

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інформаційної системи супроводу клієнтів готельно-ресторанного комплексу «Garden Plaza»

Виконав: студент IV курсу, групи СНС-41

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Ціпіньо М.М.

(прізвище та ініціали)

Керівник

(підпис)

Козбур Г.В

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ціпіню Михайлу Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи супроводу клієнтів готельно-ресторанного комплексу «Garden Plaza»

Керівник роботи кандидат технічних наук, доцент кафедри КН Козбур Г.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 2023р.

3. Вихідні дані до роботи Опис підприємства готельного бізнесу Garden Plaza.
Описи сучасних технологій для супроводу клієнтів, зокрема у вигляді Інтернет-сайтів компаній

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз сучасного стану та технологій для супроводу клієнтів у готельному бізнесі. Аналіз предметної галузі та існуючих рішень. Постановка задачі роботи.

2. Проектування та реалізація АІС для готелю Garden Plaza. Розробка архітектури, бази даних, інтерфейсу ПЗ. Реалізація в програмних кодах та тестування.

3. Безпека життєдіяльності, основи хорони праці. Висновки. Перелік джерел.

Додатки. Тексти розробленого програмного забезпечення

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний слайд. 2. Актуальність та характеристики роботи.

3. Вимоги до програмного продукту. 4. Модель бази даних.

5. Інструментальні засоби для реалізації системи. 6. Архітектура програмного продукту.

7. Зовнішній вигляд програмного продукту. 8. Зовнішній вигляд програмного продукту.

9. Висновки. 10. Подяка.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання 7 лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		<i>Виконано</i>
2.	Підбір джерел про методи та засоби вирішення задачі супроводу у туристичній сфері		<i>Виконано</i>
3.	Переклад та опрацювання джерел про технології створення програмних продуктів для задач супроводу		<i>Виконано</i>
4.	Виконання дослідження щодо архітектури системи її бази даних та основних алгоритмів роботи		<i>Виконано</i>
5.	Розроблення автоматизованої інформаційної системи Супроводу клієнтів для готелю Garden Plaza		
5.	Оформлення розділу «Аналіз сучасного стану ведення послуг у готельному бізнесі»		<i>Виконано</i>
6.	Оформлення розділу «Проектування та реалізація АІС ведення обліку для готелю Garden Plaza»		<i>Виконано</i>
7.	Виконання підрозділу «Безпека життєдіяльності»		<i>Виконано</i>
8.	Виконання завдання до підрозділу «Основи хорони праці»		<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи		<i>Виконано</i>
10.	Нормоконтроль		<i>Виконано</i>
11.	Перевірка на плагіат		<i>Виконано</i>
12.	Попередній захист кваліфікаційної роботи		<i>Виконано</i>
13.	Захист кваліфікаційної роботи		

Студент

_____ (підпис)

Ціпіню М.М.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Козбур Г.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційної системи супроводу клієнтів готельно-ресторанного комплексу «Garden Plaza» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Ціпіньо Михайло Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // С. 67, рис. – 26, табл. – 0, додат. – 2, бібліогр. – 26.

Ключові слова: інформаційна система, туристична галузь, супровід у готельній сфері, десктопний додаток, цифровізація готельного бізнесу.

Кваліфікаційна робота присвячена актуальній темі розвитку туристичного бізнесу та необхідності використання розумних технологій для трансформації готельного продукту.

Мета роботи: ефективне керування клієнтською базою, контроль за заселенням, управління заселенням та виселенням клієнтів, а також авторизація співробітників, яка можлива за допомогою розробки та реалізації АІС для супроводу клієнтської БД.

В першому розділі кваліфікаційної роботи досліджено літературу про сферу туризму та її автоматизацію, яка в даний час має великі перспективи для розвитку в зв'язку Індустрією 4.0. Описано існуючі додатки для автоматизації цієї сфери, а також обгрунтовано створення власної системи супроводу клієнтів в готельній сфері. Було проаналізовано методи та інструменти для, власне, розробки системи.

В другому розділі кваліфікаційної роботи описано проектування та тестування АІС для супроводу клієнтської БД. Приведено архітектуру додатку, її БД, методи та функції роботи. Середовище Visual Studio, мова

програмування C# та бібліотека Windows Forms стали основними при проектуванні програмного продукту. Тестування показало його продуктивність та перспективу у подальшому використанні даного додатку.

ANNOTATION

Information System Development for Clients Support in the Hotel and Restaurant Complex "Garden Plaza" // Qualification work of the educational level "Bachelor" // Mykhailo Mykhailovych Tsipino // Ternopil Ivan Puluj National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-41 // Ternopil, 2023 // P. 67, fig. – 26, tabl. – 0, annexes – 2, references – 26.

Keywords: information system, tourism industry, support in the hotel sector, desktop application, digitization of the hotel business.

The qualification work is dedicated to the current topic of tourism business development and the need to use smart technologies to transform the hotel product.

The goal of the work: effective management of the client base, control over check-in, management of check-in and check-out of clients, as well as authorization of employees, which is possible through the development and implementation of AIS to support the client database.

The first section of the qualification paper considered the literature on the field of tourism and its automation, which currently has great prospects for development in connection with Industry 4.0, is investigated. Already available applications for automation of this area are described, as well as advantages for creating your own customer support system in the hotel sector. The methods and tools for actually developing this system were analyzed.

In the second section of the qualification work, it is considered the design and testing of AIS to support the clients' database. The architecture of the application, its database, work methods and functions are presented. The Visual Studio environment, the C# programming language, and the Windows Forms library have

become essential when designing a software product. Testing showed its performance and perspective in further use of this application.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CRM – Customer Relationship Management

HTML – HyperText Markup Language

IDE – Integrated Development Environment

IT – Information Technology

MS – Microsoft

ОС – Операційна система

SQL – Structured query language

VS – Visual Studio

WF – Windows Forms

WPF – Windows Presentation Foundation

AIC – Автоматизована інформаційна система

БД – База даних

ВВП – Валовий внутрішній продукт

ДСТ – Державний стандарт

ІД – Ідентифікатор

ІКТ – Інформаційно-комунікаційні технології

ІС – Інформаційна система

МСБ – Малий та середній бізнес

ПЗ – Програмне забезпечення

ПК – Персональний комп'ютер

ПМД – Перша медична допомога

РБД – Реляційна база даних

СКБД – Система керування базами даних

ШІ – Штучний інтелект

ЗМІСТ

ВСТУП	10
1 АНАЛІЗ СУЧАСНОГО СТАНУ ВЕДЕННЯ КЛІЄНТІВ У ГОТЕЛЬНОМУ БІЗНЕСІ	13
1.1 Аналіз предметної області.....	13
1.2 Аналіз існуючих додатків для ведення супроводу клієнтів в готельному бізнесі	16
1.3 Постановка задачі.....	18
1.4 Вибір та обґрунтування засобів реалізації АІС готелю	19
1.5 Висновки по розділу	21
2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ СУПРОВОДУ КЛІЄНТІВ ДЛЯ ГОТЕЛЮ GARDEN PLAZA	22
2.1 Функціональне проєктування	22
2.2 Проєктування та реалізація бази даних	28
2.3 Програмна реалізація проєкту	33
2.3.1 Реалізація інтерфейсу	35
2.4 Тестування розробленого продукту	40
2.5 Висновки по розділу	44
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	45
3.1 Охорона праці і техніка безпеки в готелі "Garden Plaza".....	46
3.2 Долікарська допомога при ураженні електричним струмом в готелі "Garden Plaza"	47
Висновки	50
Перелік використаних джерел	52
Додаток. Тексти розробленого програмного забезпечення.....	55

ВСТУП

Актуальність теми. Туристичний бізнес можна вважати одним з найрозвиненіших економічних галузей багатьох країн, вони становлять вагому частку їх ВВП. Початок Індустрії 4.0 та її цифрова ера обов'язково вимагає трансформації готельного продукту за допомогою розумних технологій, які змінюють його матеріальні елементи. Беручи до уваги стан економічних показників закордонних країн, на які вплинула пандемія та економіку нашої країни, в якій ведеться вторгнення, що ускладнює негативні наслідки пандемії, туристична сфера в обох випадках зазнала різкого зменшення показників. Проте, беручи до уваги історичний досвід попередніх економічних та епідемологічних ситуацій, можна очікувати, що туризм повернеться на свої позиції, а отже, проведення аналізу та розробка ПЗ зараз є актуальними.

З огляду на стрімкий розвиток діджиталізації суспільства та розширення доступу глобальної мережі Інтернет, важливу роль набувають цифровізація бізнесу та його автоматизація. Що стосується готельної сфери, тут важливо почати з ведення БД клієнтів та контролю їх заселення. Як підсумок, для готельного бізнесу було б добре мати в користуванні власне ПЗ, в якому надається зрозумілий та повнофункціональний інтерфейс для перегляду, керування та планування бронюваннями. Такі задачі супроводу можуть застосовуватися до наступних важливих питань:

- Супровід клієнтської бази.
- Завжди актуальний список заселених номерів.
- Управління заселеннями та виселеннями клієнтів.
- Авторизація співробітників.

Готельні підприємства є різні за розмірами. Варто зазначити, що розробка власного ПЗ навіть для невеликого готельного бізнесу дозволить підвищити його конкурентоспроможність та заощадить чималий відсоток робочого часу працівників готелю, що призведе до використання їх вільного

часу для підвищення рівня їх кваліфікації в сфері обслуговування, а готельний бізнес – до підвищення рівня конкурентоспроможності, збільшення кількості клієнтів та прибутку. Тому створення і впровадження автоматизованої системи для супроводу готелю є актуальним питанням і в даній роботі буде підлягати вирішенню.

Мета і задачі дослідження. Мета кваліфікаційної роботи є розробити автоматизовану систему управління готельним підприємством з метою зменшення затрат часу персоналу, яке він витратить на супровід БД клієнтів та їх заселення. Ці функції є одним з головних критеріїв нормального функціонування готельного бізнесу, чого можна досягти, спроектувавши і реалізувавши систему для супроводу клієнтів готелю. Слід зазначити, що працівниками готелю є люди, які не є фахівцями в галузі інформаційних технологій, а їхній рівень володіння ПК є зазвичай середнім або низьким. Тому розроблення ПЗ має містити достатньо зрозумілий інтерфейс користування та мінімальну кількість функцій з можливістю їх доопрацьовувати в майбутньому. Для отримання результатів мети роботи необхідно виконати такі задачі:

- Провести аналіз предметної області, шляхів супроводу клієнтів в готельній справі, а також здійснити огляд існуючих додатків.
- Скласти алгоритми для вирішення завдання даної роботи, розробити структурну схему, проєкт системи та інші її майбутні особливості.
- Виконати огляд та аналіз комплексу програмних засобів та здійснити вибір оптимальної конфігурації для реалізації проєктованої системи.
- Виконати програмну реалізацію АІС супроводу для готелю.
- Провести тестування розробки, оцінити її ефективність, зробити висновки щодо розробки.

Практичне значення одержаних результатів полягає в розробці робочого програмного забезпечення для АІС супроводу клієнтів, що може

бути застосована для готельного бізнесу «Garden Plaza». Розробка може бути впроваджена в роботі будь-якого іншого туристичного підприємства, що надає послуги оренди номерів.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ВЕДЕННЯ ОБЛІКУ ПОСЛУГ У ГОТЕЛЬНОМУ БІЗНЕСІ

1.1 Аналіз предметної області

У сфері народного господарства багатьох невеликих країн світу важливою часткою є туризм, проте навіть у великих державах внутрішній туризм також є одним з важливих питань, включаючи не тільки економічні показники, але й іміджеві та патріотичні. Враховуючи важливість цих показників послуги в туристичній та готельній справі мають відповідати загальноприйнятим нормам, включаючи державні стандарти. [1-4].

Робота в сфері обслуговування, а саме туристична та готельна, є багатокomпонентною, проте можна виділити деякі особливості галузі:

– Бізнеси та підприємства, які надають послуги з розміщення гостей, повинні забезпечувати їх засобами особистої гігієни, засобами чистоти та побутової хімії, а також якісним обслуговуванням.

– За бажанням клієнтів або попередньою домовленістю готелі мають забезпечувати їх послугами якісного харчування, причому це стосується раціону гостей з різних географічних точок розміщення.

– Бізнесам та підприємствам необхідно внести в перелік своїх послуг супутні товари чи послуги, які допоможуть їм повноцінно працювати або ж комфортно відпочивати.

Слід зауважити, що забезпечити туристичну сферу якістю послуг означає як комфорт гостя, так і персоналу.

Зазвичай туристичні компанії мають в своєму підпорядкуванні або мережу готелів або невеликий за масштабом готель або хостел (наприклад, надає послуги з розміщення в єдиному готелі на 10 осіб).

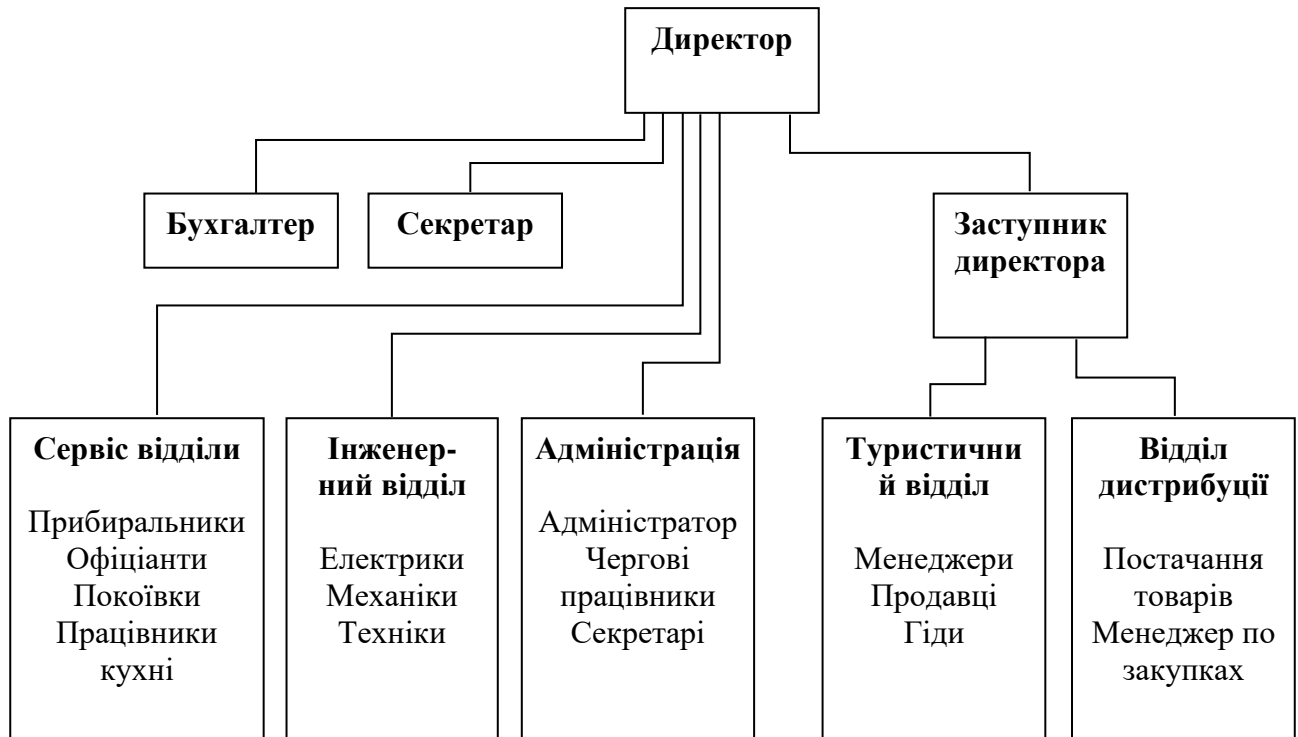
Великі розвинуті мережі зазвичай вже використовують ПЗ, яке може оптимізувати діяльність підприємства, проте в малих за масштабом готельних

зкладах така можливість буває рідко. Враховуючи показники підприємств готельного чи ресторанного бізнесу, які надають послуги обслуговування, можна стверджувати, що більшість з МСБ використовують або паперові відомості, або мають в користуванні Excel. Тому, саме в малих за масштабами бізнесів виникає потреба у створенні та автоматизації ведення супроводу клієнтською БД та процедури заселення-виселення, що призведе до зменшення часу працівників, коректного обліку та виведення реальних показників продуктивності готельної справи.

Для реалізації автоматизованої системи потрібно зрозуміти та описати структуру та масштаби готельного бізнесу, для якого буде вестись дана розробка. Готель «Garden Plaza» – це невеликий готельно-ресторанний комплекс, розрахований на 30 номерів місткістю від 2 до 5 осіб. Готель має також маленьку садову зону та територію для парковки.

Кадрова структура готелю показана на рисунку 1.1.

Рисунок 1.1 – Структура кадрового забезпечення готелю «Garden Plaza»



Дослідження конструкції кадрів готельно-ресторанного комплексу, а саме блок адміністрації, свідчить про те, що дане підприємство може мати досить великі навантаження (відділ, який займається питаннями бронювання та поселення-виселення людей), оскільки комплекс розташований в місцевості, де зустрічається сезонність (туристичні маршрути, відрядження).

Тому для розроблення і впровадження системи автоматизації супроводу клієнтів можна піти двома шляхами:

- Перший – покупка та встановлення вже готових додатків, які реалізують супровід готельного бізнесу, або ж пошук безкоштовних ПЗ. Проте покупка ПЗ не завжди релевантна, оскільки на її реалізацію, підтримку, оновлення та навчання працівників йде велика частина бюджету підприємства.

- Другий – це розробка власної системи автоматизації супроводу, яка буде підлаштована під кількісні характеристики наявного технічного і програмного комп'ютерного забезпечення на підприємстві, із врахуванням

рівня працівників в користуванні ПК та можливості майбутніх удосконалень для повноцінної роботи готелю і всіх його відділів.

Аналіз існуючих додатків, їх переваг та недоліків дозволить прийняти найкращі рішення для власної розробки.

1.2 Аналіз систем для ведення супроводу в готельному бізнесі

Розвиток ІКТ дає можливість компаніям застосовувати та впроваджувати системи та засоби для оцифрування, автоматизації та організації супроводу роботи відповідного підприємства.

Одна з таких систем – система управління взаємовідносинами з клієнтами (CRM) – це поєднання практик, стратегій і технологій, які компанії використовують для керування та аналізу взаємодії з клієнтами та даних протягом життєвого циклу клієнта. Управління взаємовідносинами з клієнтами (CRM) в готельній індустрії – це процес побудови та управління вигідними відносинами з клієнтами. Це стратегія, яка охоплює всі відділи організації та всі рівні працівників. Мета полягає в тому, щоб покращити обслуговування клієнтів і підвищити їхню задоволеність [6], відповідно допомогти в утриманні клієнтів і стимулювати зростання продажів. Системи CRM збирають дані про клієнтів через різні канали або точки контакту між клієнтом і компанією, які можуть включати веб-сайт компанії, телефон, чат, пряму поштову розсилку, маркетингові матеріали та соціальні мережі. CRM-системи також можуть надавати співробітникам, які працюють із клієнтами, детальну інформацію про особисті дані клієнтів, історію покупок, купівельні переваги та проблеми.

Дослідження інформації про компанії, які активно користуються CRM показав, що, наприклад, в США з 2017 року показник компаній виріс з 56% до 74%, цей показник свідчить про те, що 3 з 4 компаній використовують системи для автоматизації роботи з клієнтами та їх супроводу. Аналогічні показники в

Україні є меншими, тобто в 2018 році рівень використання CRM-систем становив близько 10%. З цих відсотків більше половини мали в користуванні ці системи, решта або були на стадії впровадження або розглядали таку можливість [5].

Переваги CRM у готельній індустрії:

- Потужний маркетинговий інструмент: це чудовий маркетинговий інструмент, який допомагає готельній індустрії орієнтуватися на своїх потенційних клієнтів і розвивати свій бізнес. Його також можна використовувати для аналізу поведінки та вподобань клієнтів.
- Краще розуміння потреб клієнтів: це дає змогу краще зрозуміти потреби, бажання та очікування клієнтів.
- Збільшення прибутку: це допомагає збільшити прибуток завдяки кращому обслуговуванню клієнтів, збільшенню конверсій і зниженню витрат.
- Утримання клієнтів: це допомагає утримувати клієнтів для майбутнього бізнесу.

Недоліком існуючих CRM є те, що, з одного боку, користувачами даних систем є працівники підприємства, в даному випадку – це адміністратори, покоївки, офіціанти. Зазвичай такі працівники мають низький або середній рівень користування ПК, а дані системи мають просунуті можливості для роботи компаній, і зазвичай, вони є надлишковими для початкового впровадження і перевантажують систему та працівників, які не завжди хочуть витрачати багато часу для вивчення нових функцій.

Тому необхідністю є розробка власної системи супроводу, яка має містити максимально спрощений інтерфейс і не містити зайвих елементів, і в той же часе забезпечити функціональність супроводу клієнтів і легку адаптацію для працівників. Вагомою перевагою також є фінансове питання, оскільки вже наявні системи для керування бізнесом є платними, а власна розробка – це низька вартість з готовим функціоналом та можливістю доопрацювання.

1.3 Постановка задачі

Постановка для розробленої програми містить такі дані.

Назва продукту: Розробка інформаційної системи супроводу клієнтів готельно-ресторанного комплексу «Garden Plaza».

Тип продукту: інформаційна система для супроводу клієнтів готельного бізнесу.

Архітектура продукту: десктопний додаток.

Призначення продукту: зменшення часу працівників для ведення бази клієнтів та керування бронюваннями та поселеннями, необхідне для правильного функціонування готельного бізнесу.

Розроблена десктопна програма має повноцінно функціонувати на ПК, де встановлена ОС Windows версії 7 і вище, також ПК мають мати наступні системні вимоги:

- Процесор 2 ГГц.
- Оперативна пам'ять 4 Гб.
- 1 Гб на диску.
- Екран шириною від 1280 пікселів.
- Функціональність:
- Можливість авторизації в програмі.
- Перегляд списку номерів (цінова категорія, опис, тип), список клієнтів (які проживають, які забронювали, які виїхали), список співробітників (їхні паролі для авторизації, можливість внесення нових співробітників), реєстрація клієнтів (поселення).
- Можливість зміни записаної інформації, зміна авторизованих працівників.

Програмний продукт повинен мати зручний сучасний інтерфейс, стандартний для додатків такого типу.

Посилаючись на вимоги до програми, можна зробити обґрунтування вибору засобів та методів для його розробки, реалізації та проведення тестування.

1.4 Вибір та обґрунтування засобів реалізації системи супроводу клієнтів в готельній сфері

Серед існуючих платформ для розробки даного додатку розглянуто MS Visual Studio, NetBeans, Eclipse, IntelliJ IDEA.

Visual Studio – це інтегроване середовище розробки (IDE), яке використовується для розробки консолі, додатків графічного інтерфейсу користувача, додатків Windows Forms або WPF, веб-додатків, веб-сайтів і веб-сервісів тощо.

Visual Studio – найкращий інструмент, адаптований до роботи з продуктами Microsoft, саме його було обрано. Microsoft Visual Studio має небагато конкурентів, як-от IntelliJ IDEA, Eclipse, NetBeans, PhpStorm і Xcode тощо, тому загалом цей інструмент допомагає працювати з різними пов'язаними проектами MS.

Переваги Visual Studio (VS):

- VS допомагає розробити програми Webapps і Xamarin для потреб клієнтів.
- Зручна функція навігації.
- VS допомагає краще шукати, фільтрувати та переглядати код, з яким працює програміст.
- Має кращі налаштування та може також містити сторонні розширення.
- VS є найкращим IDE у випадку, якщо мова проєкту – C#.
- Якщо для інтерфейсу користувача використовується Angular або Angular 4, тоді VS найкраще підтримує сценарій типу та сценарій Java.

- VS допомагає створювати нативні програми для Android, iOS і Windows в одному рішенні.
- Функція Xamarin Live Player допомагає провести тестування в реальному часі на пристроях Android і Apple.
- Експерт може створювати програми за допомогою HTML, JavaScript, C#, .NET (Xamarin) і C++ тощо.

Мінуси Visual Studio:

- Встановлення VS і оновлення можуть зайняти деякий час.
- Іноді VS потребує обширних досліджень, щоб вирішити розбіжності з виконанням проектів.
- Усе попереднє програмне забезпечення може бути несумісним із новою IDE після переходу зі старої VS на новішу версію.
- VS може бути дорогим для деяких IDE порівняно з іншими на ринку. [7]

При виборі інструменту для реалізації інтерфейсу було розглянуто кілька бібліотек. Серед декількох (.NET, AWT, Swing, JavaFX, GTK+, Windows Forms) було обрано одну – Windows Forms [8,9].

Windows Forms (WF) – це набір керованих бібліотек в .NET Framework. Ці бібліотеки призначені для розробки багатофункціональних клієнтських додатків, графічний API для відображення даних і керування взаємодією користувача з більш простим розгортанням і підвищеною безпекою в клієнтських додатках.

WF пропонує обширну клієнтську бібліотеку, яка надає інтерфейс для доступу до власних елементів графічного інтерфейсу Windows і графіки з керованого коду. Побудована на керованій подією архітектурі, аналогічній клієнтам Windows, і, відповідно, його додатки будуть вводити дані від користувача для його виконання [8].

Отже, остаточна конфігурація розробки – середовище Visual Studio за допомогою мови C# та бібліотеки Windows Forms.

1.5 Висновки по розділу

Отже, в першому розділі було досліджено літературу про сферу туризму та її автоматизацію, яка в даний час має великі перспективи для розвитку в епоху Індустрії 4.0. Описано наявні вже додатки для автоматизації цієї сфери, а також переваги для створення власної системи супроводу клієнтів в готельній сфері, де важливим критерієм має бути простий та зрозумілий інтерфейс для майбутніх користувачів, які не мають високого рівня володіння ПК та додатками.

Ще однією з поставлених вимог є те, що програма має бути універсальною з реалізацією на платформі Windows, Android, та майбутніми модифікаціями стосовно функціоналу програми. Також система має мати захист, тобто авторизацію для співробітників через пароль та логін. Тому, перед початком програмування (розділ 2) було проаналізовано методи та інструменти для, власне, розробки даної системи. Це середовище Visual Studio, мова програмування C# та бібліотека Windows Forms.

2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ СУПРОВОДУ КЛІЄНТІВ ДЛЯ ГОТЕЛЮ «GARDEN PLAZA»

Комп'ютерна система управління розроблена для обробки всієї первинної інформації, необхідної для розрахунку щомісячних звітів. Окрема база даних підтримується для обробки всіх деталей, необхідних для правильного обчислення та генерації оператора.

Цей проєкт має на меті запровадити більше зручності для користувачів у різних видах діяльності, таких як оновлення записів, обслуговування та пошук. Опишемо в пункті 2.1 основні функціональні можливості.

2.1 Функціональне проєктування

Пошук запису – досить простий, оскільки всі дані про клієнта можна отримати, просто ввівши ідентифікацію цього клієнта. Аналогічно, ведення та оновлення записів також можна здійснити за допомогою ідентифікації клієнта з автоматичною генерацією всіх деталей. Ці деталі також швидко автоматично оновлюються в головному файлі, таким чином зберігаючи запис абсолютно актуальним. Вся інформація зберігається в базі даних або файлах, які доступні тільки користувачу, який авторизувався.

Визначимо основні вимоги до функціональності проєктованого ПЗ, які опишемо та унаочнимо на рисунках.

Основною метою всієї діяльності є автоматизація процесу повсякденної діяльності готелю, а саме:

- підключення до системи;
- відсутність дублювання даних;
- зменшення або повна відсутність паперової роботи;

- ефективність за часом. Кожен раз, коли новий користувач подає свої дані, база даних повинна оновлюватися автоматично. Цей запис миттєво може використовуватись іншим користувачем;
- автоматична перевірка даних. Повний контроль над проектом знаходиться в руках уповноваженої особи, яка має пароль для доступу до цього проекту. Весь контроль виконується адміністратором, а інші учасники мають право просто переглядати записи, щоб не змінювати транзакції чи записи;
- зручний інтерфейс;
- безпека та надійність даних. Безпека є основним критерієм для запропонованої системи, оскільки незаконний доступ може пошкодити базу даних;
- швидке введення та отримання даних.

Діаграму прецедентів, яка буде описувати відповідні сервіси, показано на рис. 2.1.

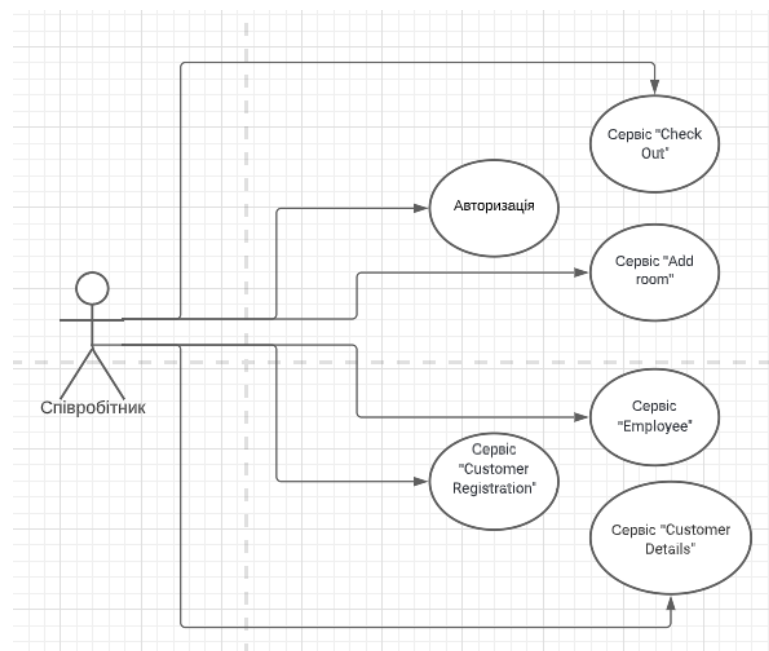


Рисунок 2.1 – Діаграма прецедентів (варіантів використання, Use Case) для проєктованої системи автоматизації готелю, опис сервісів; дійова особа – співробітник готелю

Слід зазначити, що перший варіант використання системи (Сервіс «Customer Registration») дозволяє заносити дані клієнта під час реєстрації та при поселенні, а саме – дані номера кімнати, тип ліжка, ціна. Діаграма діяльності (активності) для варіанту використання «Customer Registration» показана на рис. 2.2.

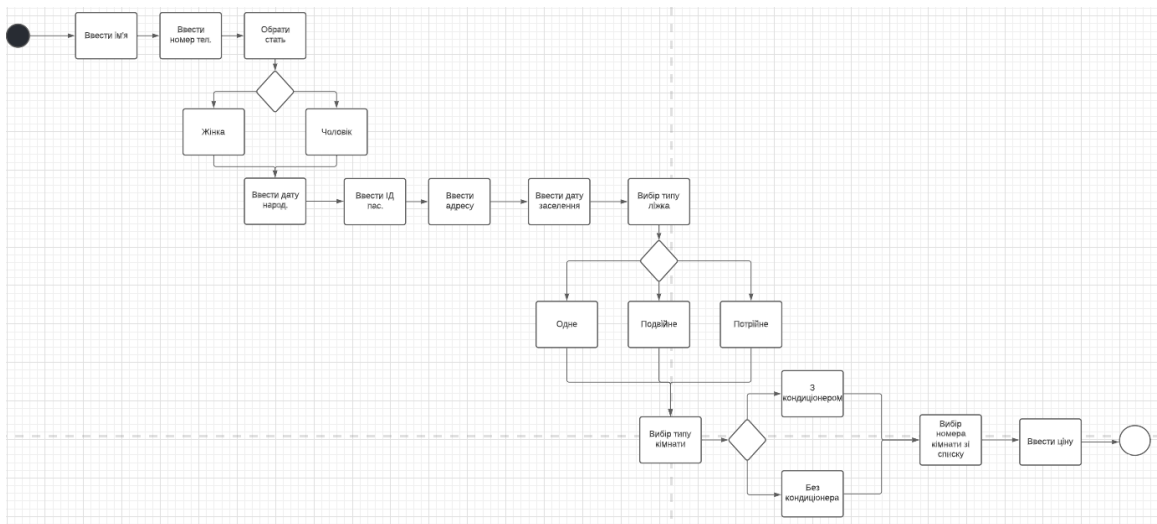


Рисунок 2.2 – Діаграма діяльності для варіанту використання «Customer Registration»

Авторизація виконується один раз, але теоретично можливий варіант неправильного введення логіна та пароля, що також слід передбачити на діаграмі діяльності (рис. 2.3).

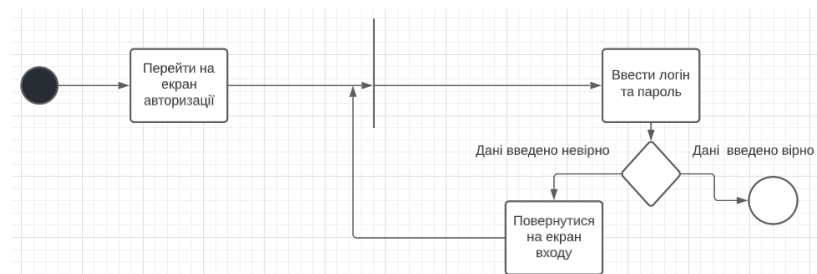


Рисунок 2.3 – Діаграма діяльності для варіанту використання «Авторизація»

Наступний варіант використання системи (Сервіс «Check Out») дозволяє знайти дані клієнта, які були введені під час реєстрації та виселити його (пошук здійснюється за допомогою введення ПІБ або за номером кімнати). Виселення проходить шляхом знаходження клієнта, потім відбувається вибір дати виселення, яке ставить даного клієнта в БД Customer Details. Діаграма діяльності (активності) для варіанту використання «Check Out» показана на рис. 2.4.

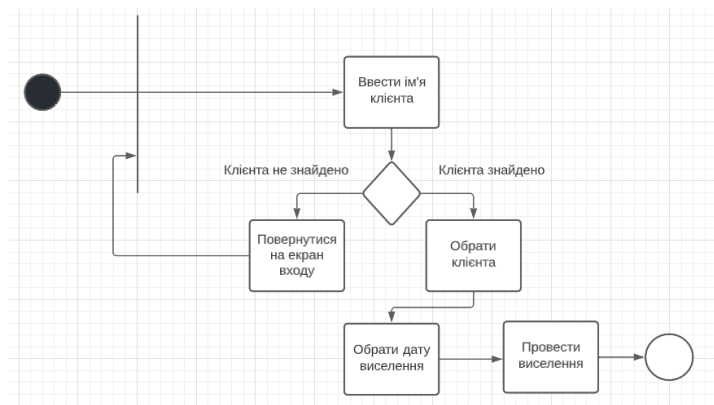


Рисунок 2.4 – Діаграма діяльності для варіанту використання «Check Out»

Варіант використання системи (Сервіс «Add room»), в якій відбувається занесення даних про кімнати (ціна, тип тощо). Діаграма діяльності (активності) для варіанту використання «Add room» показана на рис. 2.5.

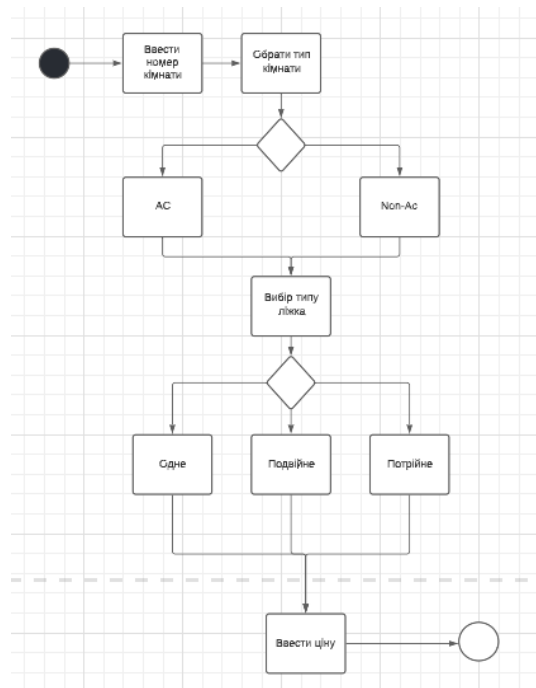


Рисунок 2.5 – Діаграма діяльності для варіанту використання «Add room»

Варіант використання для системи (Сервіс «Employee»). Тут відбувається занесення даних про користувачі програми, тобто співробітників (ПІБ, пошта, пароль тощо), дана БД допомагає переглядати користувачів та заносити нові дані, які будуть використовуватись при авторизації. Діаграма діяльності (активності) для варіанту використання «Employee» показана на рис. 2.6.

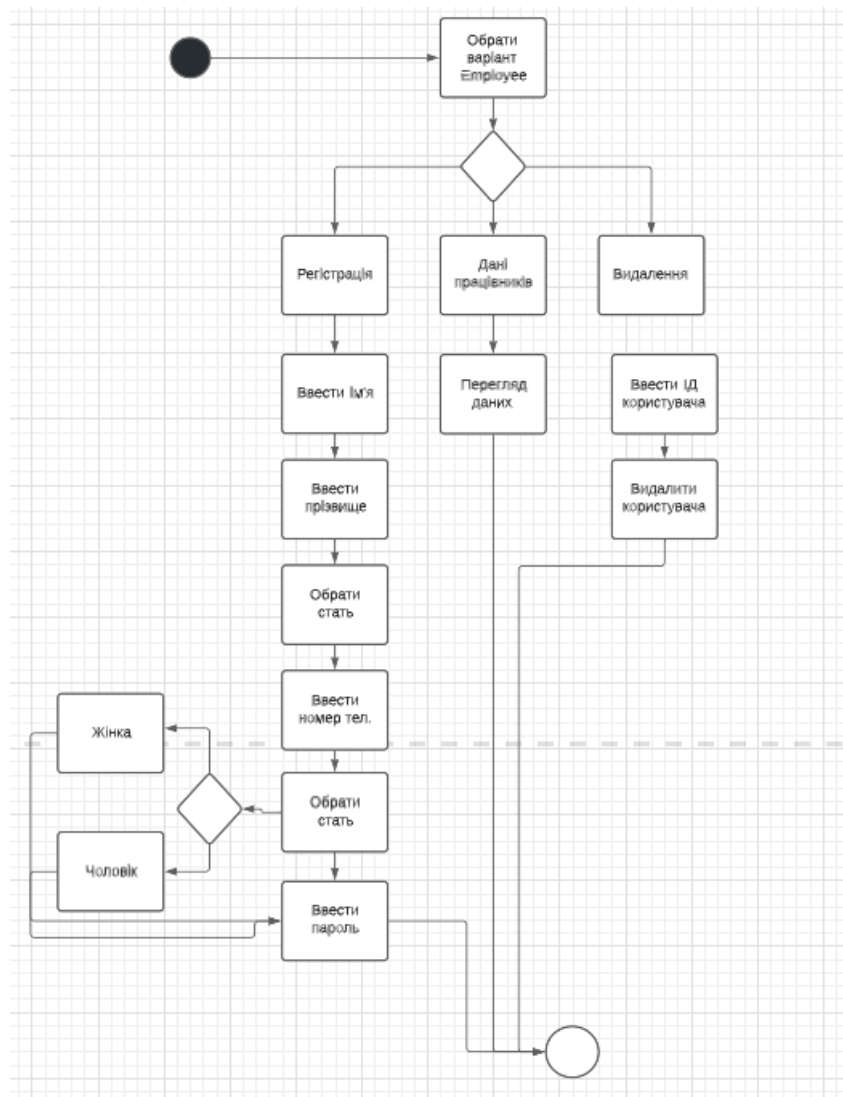


Рисунок 2.6 – Діаграма діяльності для варіанту використання «Employee»

Варіант використання для системи (Сервіс «Customer Details»). Тут відбувається огляд даних про клієнтів, дана БД допомагає переглядати клієнтів, які групуються категоріями: всі, проживаючі, виселені. Діаграма діяльності (активності) для варіанту використання «Customer Details» показана на рисунку 2.7.

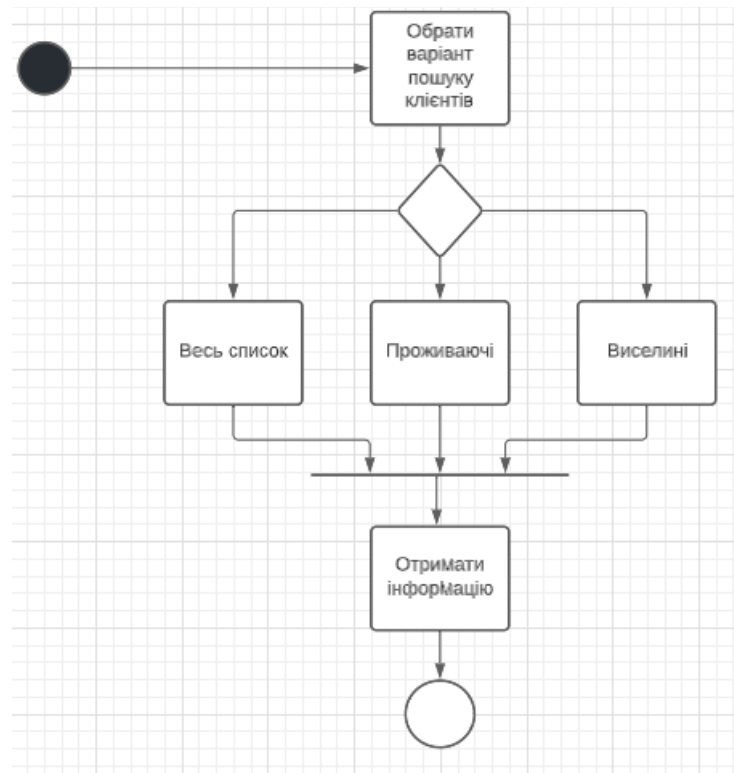


Рисунок 2.7 – Діаграма діяльності для варіанту використання «Customer Details»

Основні функціональні елементи проєктованої системи розглянуто і можна переходити до проєктування бази даних проєкту.

2.2 Проєктування та реалізація бази даних

Дані, які будуть вноситись в систему, повинні зберігатись в базі даних у вигляді таблиць, відповідно до цього є логічним використання саме РБД (реляційна база даних).

Очевидно, що в програмі будуть окремі таблиці, які призначені для зберігання даних наступних типів:

- Для списку всіх наявних номерів, що є в готелі.
- Для списку всіх клієнтів, які проживають чи проживали в готелі (що необхідно для проведення їх супроводу та статистики).
- Для списку всіх користувачів, які працюють в даній програмі.

Також доцільно ввести таблицю з реєстрацією виселення клієнта та видаленням користувачів, щоб позбавити навантаження на таблицю в БД. Планована схема даних тоді матиме вигляд як на рисунку 2.7.

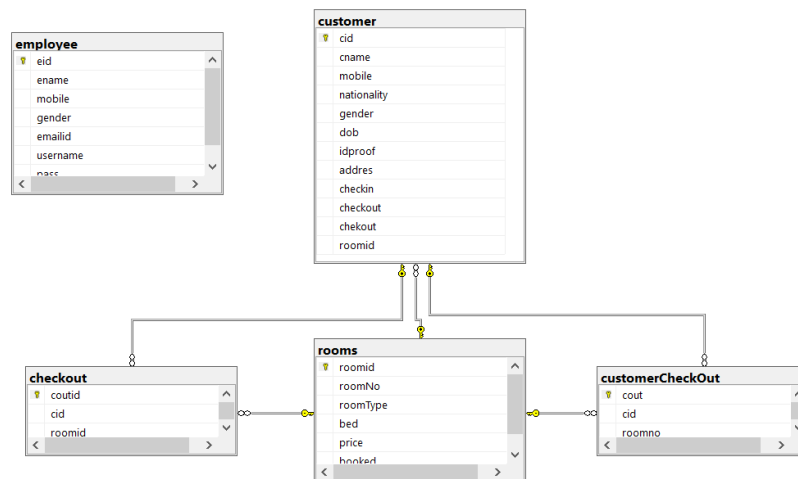


Рисунок 2.7 – Проект схеми даних системи у вигляді ERD

На рисунку 2.7 приведена діаграма «сутність-зв'язок», котра використовується для створення моделі даних. Відповідно до цього, вона забезпечує стандартний метод визначення даних і зв'язків між ними, проводиться методом задокументовування об'єктів та їхньої ініціалізації, які мають атрибути та зв'язки з другими об'єктами, важливими для даної проєктованої сфери. Тут ми можемо також провести пояснення та відтворення зберігання даних для системи, яка розробляється.

Отриману схему даних взято за фундамент в системі супроводу клієнтів готелю «Garden Plaza».

До того ж, перед реалізацією варто обрати відповідну СКБД, а саме, РБД, яка забезпечить зберігання даних в вигляді таблиць, які будуть мати між собою зв'язок. Доступ до цих даних буде здійснюватися за допомогою SQL. Вибір СКБД обумовлений тим, що дані, необхідні нам для роботи програми, будуть простими та носитимуть довідковий характер, наприклад цифрові дані

або ж текстові. Вони, зазвичай, не мають глибокою статистичної обробки засобами ШІ, тому даний вибір БД є оптимальним варіантом.

На сьогоднішній день, miniSQL, SQLite, MySQL, PostgreSQL та Oracle є популярними СКБД типу "реляційна модель". Перші дві СКБД мають обмежені можливості та використовуються досить рідко. Oracle є найпотужнішою, але пропрієтарною СКБД з високою вартістю ліцензії, що робить його неможливим для застосування у розробці. Залишаються MySQL та PostgreSQL, які також мають широкі можливості та часто використовуються в практиці.

MySQL є оптимальним варіантом для цілей цієї роботи, оскільки він має достатні можливості обробки даних і протягом багатьох років є найпоширенішою базою даних Інтернету. PostgreSQL має додаткові можливості для обробки великих обсягів корпоративних даних, але це не обов'язково є перевагою для передбачуваної мети цієї роботи. Тому в якості системи управління базами даних було обрано MySQL (а точніше MariaDB). Рекомендується додавати первинний ключ цілочисельного типу з автоінкрементом для кожної таблиці фактичної бази даних.

Реалізуємо таблиці БД в системі MySQL? результати яких наведено на рисунках 2.8-2.12.

Имя столбца	Тип данных	Разрешить знач...
roomid	int	<input type="checkbox"/>
roomNo	varchar(250)	<input type="checkbox"/>
roomType	varchar(250)	<input type="checkbox"/>
bed	varchar(250)	<input type="checkbox"/>
price	bigint	<input type="checkbox"/>
booked	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рисунок 2.8 – Зміст таблиці Rooms

Таблиця Rooms містить дані про номер кімнати, тип кімнати, тип ліжка, ціну та стан бронювання. Ключове поле ІД кімнати. Ця таблиця має залежність, залежна від таблиць Customer, Checkout, Customercheckout.

Имя столбца	Тип данных	Разрешить знач..
eid	int	<input type="checkbox"/>
ename	varchar(250)	<input type="checkbox"/>
mobile	bigint	<input type="checkbox"/>
gender	varchar(50)	<input type="checkbox"/>
emailid	varchar(120)	<input type="checkbox"/>
username	varchar(150)	<input type="checkbox"/>
pass	varchar(150)	<input type="checkbox"/>

Рисунок 2.9 – Зміст таблиці Employee

Таблиця Employee незалежна від решти таблиць та містить дані про співробітників, а саме: ІД співробітника, тобто користувача програми, ім'я, мобільні дані, стать, поштову скриньку, прізвище, пароль, який буде використовуватися для входу в програму. Ключове поле даної таблиці – ІД користувача.

Имя столбца	Тип данных	Разрешить знач...
cout	int	<input type="checkbox"/>
cid	varchar(250)	<input type="checkbox"/>
roomno	varchar(250)	<input type="checkbox"/>

Рисунок 2.10 – Зміст таблиці CustomerCheckOut

Таблиця CustomerCheckOut залежна від таблиць Rooms, Customer. Вона буде містити дані про те, в якому номері проживає даний клієнт. Ця таблиця буде використовуватися під час процедури виселення клієнтів за його ІД, яке

буде мати внутрішній характер, а також тут знаходиться номер кімнати даного клієнта.

Имя столбца	Тип данных	Разрешить знач...
cid	int	<input type="checkbox"/>
cname	varchar(250)	<input type="checkbox"/>
mobile	bigint	<input type="checkbox"/>
nationality	varchar(250)	<input type="checkbox"/>
gender	varchar(50)	<input type="checkbox"/>
dob	varchar(50)	<input type="checkbox"/>
idproof	varchar(250)	<input type="checkbox"/>
addres	varchar(350)	<input type="checkbox"/>
checkin	varchar(250)	<input type="checkbox"/>
checkout	varchar(250)	<input checked="" type="checkbox"/>
chekout	varchar(250)	<input type="checkbox"/>
roomid	int	<input type="checkbox"/>

Рисунок 2.11 – Зміст таблиці Customer

Залежна таблиця Customer, яка буде містити дані клієнтів, а саме: ПІБ, номер телефону, національність, стать, дані паспорту, адресу, дату заселення, дату виселення, номер кімнати. Ключове поле ІД клієнта. До цієї таблиці прив'язані всі інші таблиці, окрім Employee.

Имя столбца	Тип данных	Разрешить знач...
coutid	int	<input type="checkbox"/>
cid	int	<input type="checkbox"/>
roomid	int	<input type="checkbox"/>

Рисунок 2.12 – Зміст таблиці Checkout

Таблиця Checkout буде містити дані про ІД кімнати та ІД клієнта, для забезпечення пошуку під час вибору пункту виселені клієнти. Залежить від таблиць Customer, Rooms.

Отже, основні таблиці були створені, реалізована БД та її схема зв'язку, а це означає, що можна переходити до другого етапу – розробка інтерфейсу.

2.3 Програмна реалізація проєкту

На рисунку 2.13 представлена структура файлів даного продукту, опис їхньої роботи буде описано нижче.

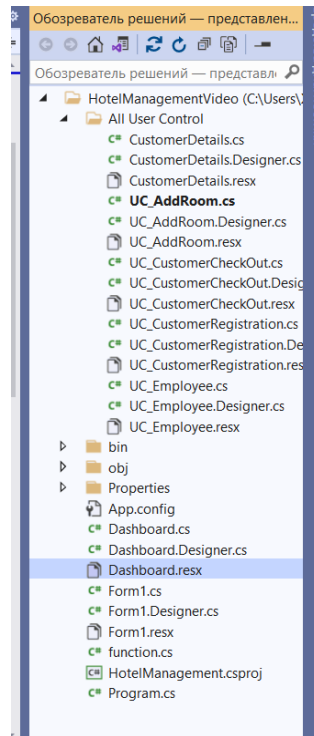


Рисунок 2.13 – Будова проєкту

Файли в папці All User Control будуть використовувати методи щодо опису відповідних сервісів, які були приведені під час створення діаграм діяльності. Опишемо коротко файли чотири важливі файли.

Структура та діяльність форми Form1.cs досить проста і насичена перевітками. По суті, тут проводиться перевірка на авторизацію користувача, відбувається запит на знаходження в БД відповідного логіна та пароля, а також тут приймається ініціалізація на відповідні запити до БД, відповідно до вибраного пошуку, занесення та видалення даних. Принцип авторизації приведено в лістингу 2.1.

Лістинг 2.1

```
private void btnLogin_Click(object sender, EventArgs e)
{
    query = "select username,pass from employee where
username          ='"+txtUsername.Text+"'          and          pass
='"+txtPassword.Text+"'";
    DataSet ds = fn.getData(query);
    if (ds.Tables[0].Rows.Count != 0)
    {
        labelError.Visible = false;
        Dashboard dash = new Dashboard();
        this.Hide();
        dash.Show();
    }
    else
    {
        labelError.Visible = true;
        txtPassword.Clear();
    }
}
```

Файл Program.cs – це основна точка входу для програми. Тут наявні лише такі методи:

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

Файл function.cs отримує доступ до БД, а також проводить оновлення видалення вставок. Доступ до БД приведено в лістингу 2.2.

Лістинг 2.2

```
public DataSet getData(String query) // Get data from database
{
    SqlConnection con = getConnection();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;
    cmd.CommandText = query;
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    da.Fill(ds);
    return ds;
}
```

Файл `Dashboard.Designer.cs` описує всі можливі методи щодо маніпуляцію з кнопками програми. Основні методи приведено в лістингу 2.3.

Лістинг 2.3

```
{
    private System.Windows.Forms.Panel panel1;
    private System.Windows.Forms.Panel panel2;
    private Guna.UI2.WinForms.Guna2CircleButton btnExit;
    private Guna.UI2.WinForms.Guna2CircleButton btnMinimize;
    private Guna.UI2.WinForms.Guna2Button btnEmployee;
    private Guna.UI2.WinForms.Guna2Button btnCustomerDetails;
    private Guna.UI2.WinForms.Guna2Button btnCheckout;
    private Guna.UI2.WinForms.Guna2Button
btnCustomerRegistration;
    private Guna.UI2.WinForms.Guna2Button btnAddRoom;
    private System.Windows.Forms.Panel MovingPanel;
    private Guna.UI2.WinForms.Guna2Ellipse guna2Ellipse1;
    private All_User_Control.UC_AddRoom uC_AddRoom1;
    private Guna.UI2.WinForms.Guna2Ellipse guna2Ellipse2;
    private All_User_Control.UC_CustomerRegistration
uC_CustomerRegistration1;
    private Guna.UI2.WinForms.Guna2Ellipse guna2Ellipse3;
    private All_User_Control.UC_CustomerCheckOut
uC_CustomerCheckOut1;
    private Guna.UI2.WinForms.Guna2Ellipse guna2Ellipse4;
    private All_User_Control.CustomerDetails customerDetails1;
    private Guna.UI2.WinForms.Guna2Ellipse guna2Ellipse5;
    private All_User_Control.UC_Employee uC_Employee1;}
}
```

Для більш детального ознайомлення можна скористатись додатком А, в якому приведений повний лістинг програми.

2.3.1 Реалізація інтерфейсу

Система для супроводу клієнтів готельно-ресторанного комплексу реалізується у вигляді десктопного додатку. У будь-якого додатку повинна бути система авторизації, титульна сторінка. Тут містяться поля для заповнення облікових даних користувача (логіна, пароля). Головне вікно ресурсу, що розробляється, показано на рисунку 2.14.



Рисунок 2.14 – Вікно авторизації

Під час авторизації користувач вводить пароль і програмне забезпечення перевіряє його дійсність. Якщо пароль дійсний, користувач може використовувати програмне забезпечення та виконувати операції, інакше, якщо пароль недійсний, то відобразиться повідомлення «Користувач/Пароль». Є можливість відновлення пароля, виходу, входу, входу нових користувачів. Також адміністратор може оновлювати зміни в програмі після входу. Після введення авторизаційної інформації співробітник компанії потрапляє в особистий кабінет – рисунок 2.15.

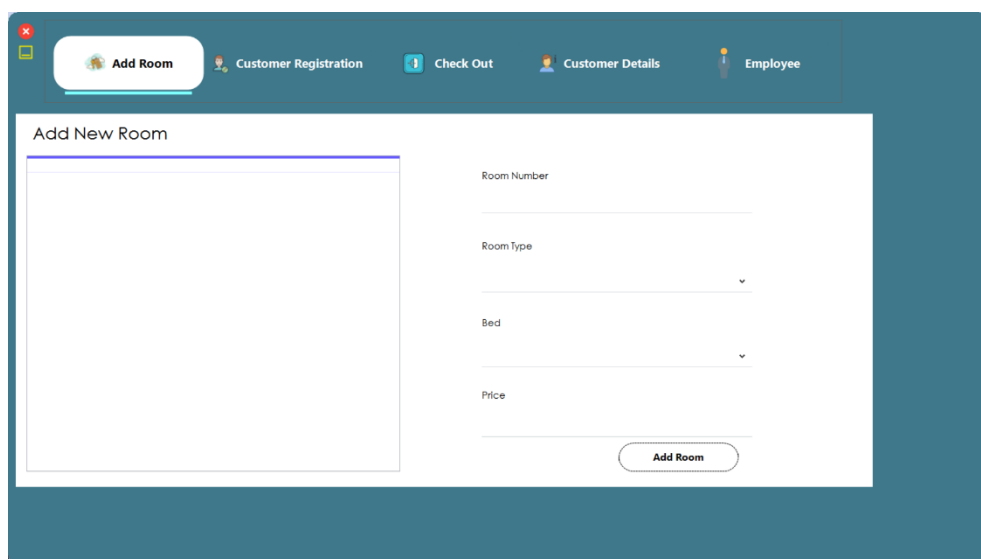


Рисунок 2.15 – Відображення вікна після авторизації

У кабінеті відображається головна сторінка з першим сервісом «Add Room», а також панель з вибором відповідного сервісу, в яких даний співробітник (який авторизувався за допомогою свого особистого логіна та пароля) може маніпулювати даними відповідно до можливостей кожного сервісу. Створення нового об'єкта (готель, кімната, клієнти, користувачі тощо). Це використовується для додавання даних про нового співробітника, видалення інформації про сутність і перегляду деталей. На цьому екрані створюється автоматичний елемент. У цій функції щоразу, коли потрібно додати нову сутність, відкриваються відповідні форми і маніпулюють базою даних, щоб перевірити, чи існують дані вже чи ні. Якщо він уже існує, він підказує, що «Запис уже існує», а якщо ні, то дані вводяться з різними перевітками.

Відзначимо, що система спроектована для того, щоб незалежно від кількості користувачів додатку, кожен із співробітників міг бачити паралельно відповідні зміни, які буде вводити чи редагувати інший користувач. При звільненні працівника його дані та пароль/логін можна буде видалити та занести дані вже нового користувача. Тут, перебуваючи в особистому кабінеті, працівник готелю може переглядати, редагувати та вводити дані відповідно до сервісів.

Функція `NEW_ROOM()`. Ця функція використовується для відкриття нового приміщення для клієнта, щоб користувач міг призначити окрему кімнату. На цьому екрані створюється автоматичний номер кімнати. Після відкриття нового приміщення для клієнта, нарешті, кімната призначається клієнту, а записи кімнат додаються до файлу даних, рисунок 2.15.

Функція `CHECKIN_CUSTOMER()`. Ця функція використовується для допуску клієнта до нашого готелю після введення всіх його персональних даних, таких як ім'я, адреса, телефон, стать, а потім йому призначається номер із функції `NEW_ROOM()`, рисунок 2.16.

The screenshot shows a web application interface for 'Customer Registration'. The top navigation bar includes 'Add Room', 'Customer Registration' (active), 'Check Out', 'Customer Details', and 'Employee'. The main form is titled 'Customer Registration' and contains the following fields:

- Name: Enter Full Name
- ID Proof: Enter ID
- Sex: dropdown menu
- Mobile No: Enter Contact No
- Address: Complete Address
- Room type: dropdown menu
- Nationality: dropdown menu
- Check In: 25.02.2023
- Room No: dropdown menu
- Gender: dropdown menu
- Price: dropdown menu
- Date of Birth: 25.02.2021

A 'Book Room' button is located at the bottom right of the form.

Рисунок 2.16 – Реалізація інтерфейсу для функції CHECKIN_CUSTOMER()

Функція CHECKOUT_CUSTOMER(). Ця функція використовується для отримання інформації про клієнта з бази даних. Коли користувач введе номер своєї кімнати, той самий номер кімнати буде перевірено в базі даних, якщо номер кімнати збігається в базі даних, тоді клієнта буде виписано з таблиці проживаючих в готелі і перенесено запис виписки в іншу таблицю бази даних, щоб менеджмент готелю мав облік клієнтів, які виїхали, рисунок 2.17.

The screenshot shows a web application interface for 'Customer Check Out'. The top navigation bar includes 'Add Room', 'Customer Registration', 'Check Out' (active), 'Customer Details', and 'Employee'. The main form is titled 'Customer Check Out' and contains the following fields:

- Name: Enter Full Name
- A large empty text area for notes or details.
- Name: Enter Full Name
- Room No: Room No
- Check Out: 27.02.2023

A 'Check Out' button is located at the bottom right of the form.

Рисунок 2.17 – Реалізація інтерфейсу для функції CHECKOUT_CUSTOMER()

Функція `BILL()`. Під час виїзду будь-якого клієнта його розрахунковий рахунок заповнюється в полі в даті виїзду мінус дата заїзду та помножується на щоденну плату за номер плюс інші витрати, а рахунок має бути збережений у таблиці в бази даних.

Функція `DISPLAY_RECORD()`. Ця функція використовується для відображення на екрані всієї транзакції, включаючи ім'я клієнта, адресу, телефон, номер ліжка тощо. Це звіт для відображення всіх записів транзакцій на екрані, рисунок 2.18.

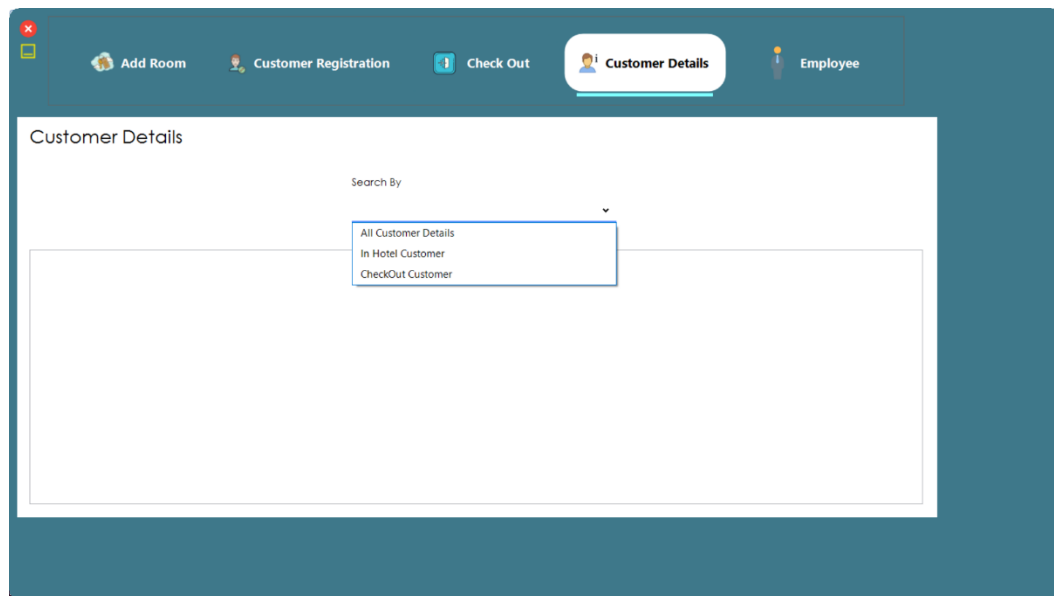


Рисунок 2.18 – Реалізація інтерфейсу для функції `DISPLAY_RECORD()`

Також реалізована функція `Employee()`. Ця функція використовується для відображення на екрані всієї інформації про працівника, який являється користувачем програми, включаючи ім'я працівника, телефон, електронна пошта тощо. Це звіт для відображення всіх записів транзакцій на екрані, рисунок 2.19.

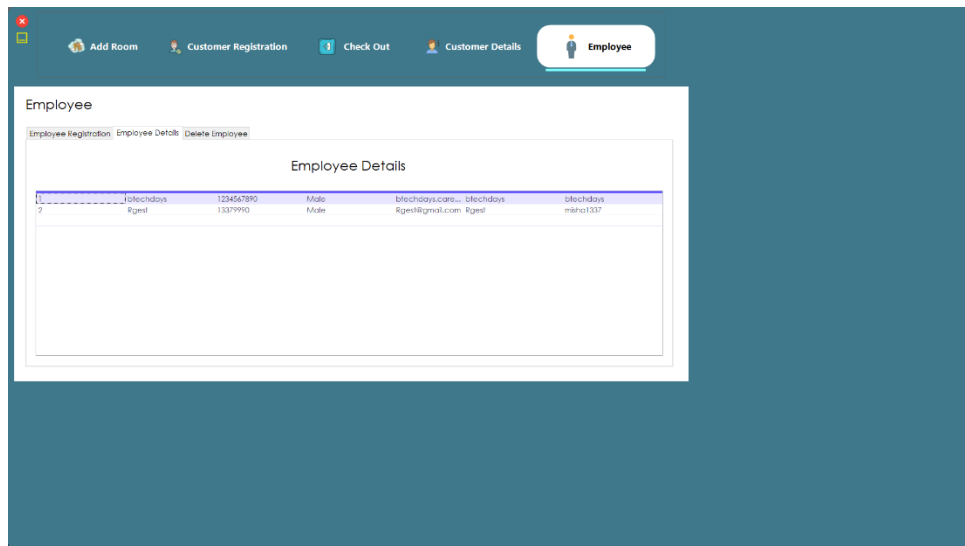


Рисунок 2.19 – Реалізація інтерфейсу для функції Employee()

Також тут можна буде проводити не лише реєстрацію та перегляд працівників, а й їхнє видалення з БД даної програми.

2.4 Тестування розробленого продукту

Будь-який програмний продукт має етапи проектування, розробки та тестування. Перші два етапи ми вже описали вище, тому в цьому підрозділі буде описано тестування. Отже, програмний продукт був розроблений для супроводу клієнтського бази у вигляді «десктопного додатку», нижче приведені результати тестування.

- Під час запуску додатку, спостерігалось стабільне завантаження без помилок та аварійних завершень. Інтерфейс програми також коректно видавав зображення.

- При тестуванні програмного продукту, заповнення БД, видалення та інших маніпуляцій додаток працює на достатньому рівні: без зависання, зупинок, пауз.

Висновок щодо тестування програми полягає в наступному, розроблений продукт виконує поставлене перед ним завдання. Він адекватно працює з БД, виконує перевірку авторизації, при навантаженні БД не видає

помилки або аварійних завершень. Також розроблений додаток протестований на зручність та інтуїтивність, інформація виводиться в зрозумілих таблицях та розміщення в різних секціях, а отже працювати з ним можна, навіть якщо в працівника низький рівень володіння ПК.

Кожне тестування приведено з детальним описом, нижче, відповідно до розроблених діаграм діяльності.

а) «Авторизація».

Тут є можливість трьох варіантів дій:

– Користувач ввів неправильний логін/пароль, система ж видає помилку та просить ввести повторно – рисунок 2.20.

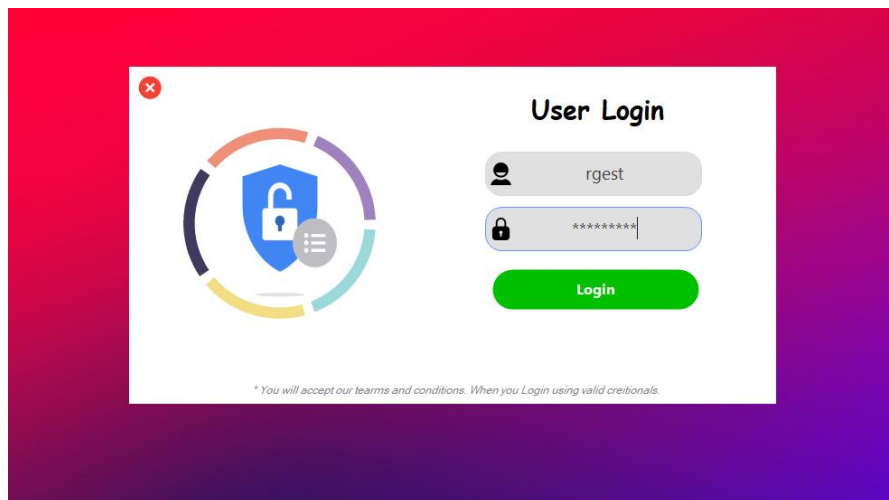


Рисунок 2.20 – Перевірка авторизації

– Користувач вводить правильний логін/пароль та при перевірці системою відбувається перехід на головне меню.

б) «Перегляд даних».

Тут ми знайдемо інформацію детальну про клієнтів, яка відображається у вкладці відомості про клієнтів, примітка спочатку вона порожня – рисунок 2.21, інформація буде накопичуватися при заповненні БД.

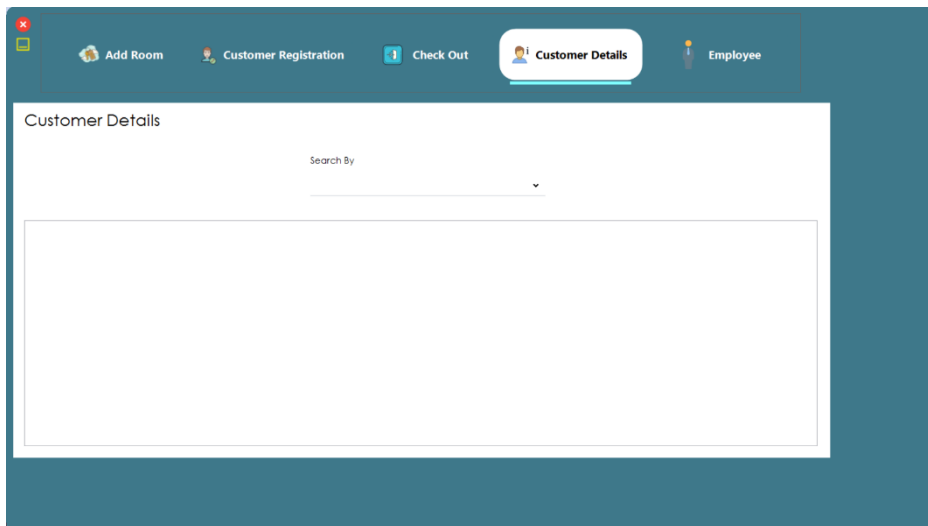


Рисунок 2.21 – Тестування вкладки «Customer Details»

Для отримання інформації про клієнтів в нас є вибір пошуку по таким критеріям як: клієнти, які проживають в готелі, клієнти, які виселились і перегляд всіх клієнтів. Для цього ми обираємо один з варіантів і отримуємо дані про клієнтів, рисунок 2.22.

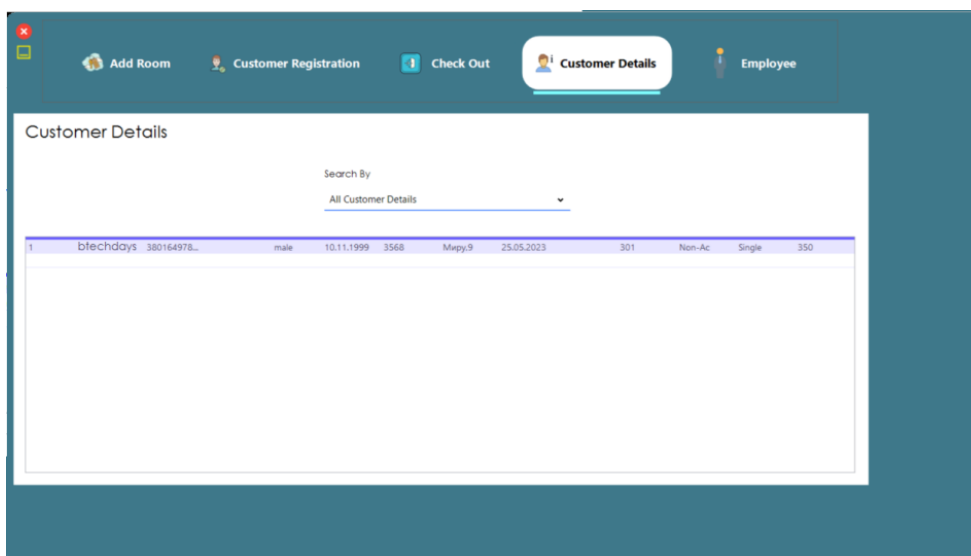


Рисунок 2.22 – Тестування вкладки «Customer Details»

в) «Видалення даних у системі».

Для видалення даних, наприклад користувача, необхідно обрати вкладку Employee, а в ній – Delete Employee, рисунок 2.23.

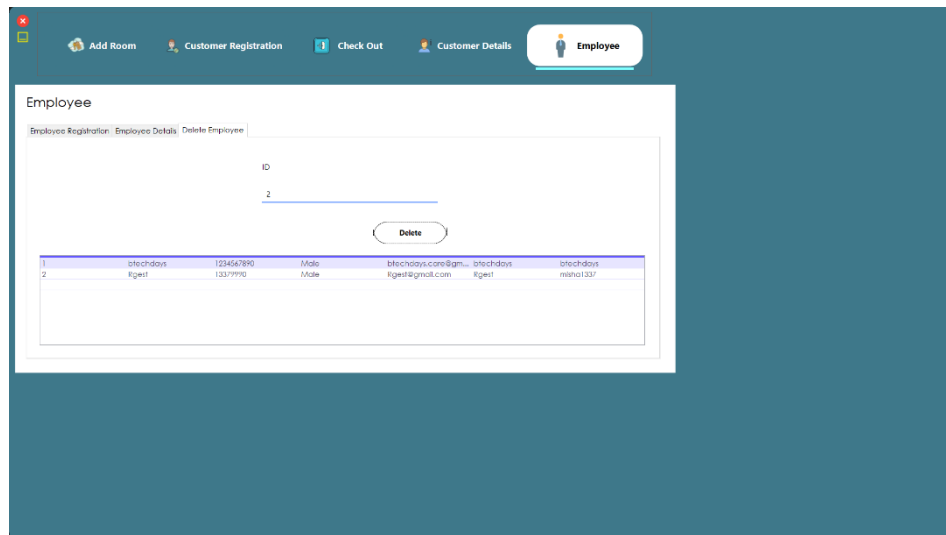


Рисунок 2.23 – Тестування вкладки «Employee»

Далі ми прописуємо ІД користувача, яке створюється під час реєстрації, та натискаємо на видалення (рисунок 2.24).

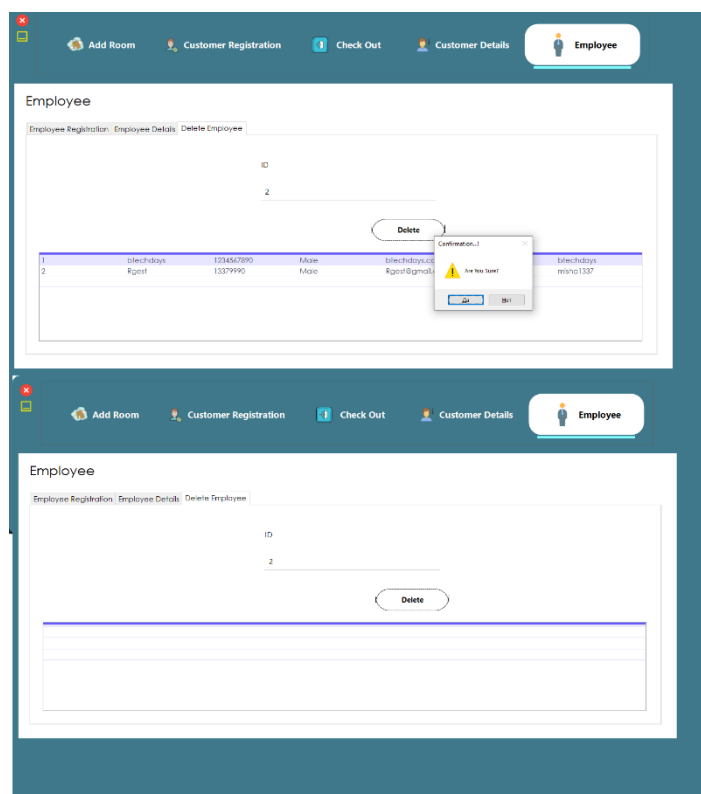


Рисунок 2.24 – Тестування вкладки «Employee»

Після підтвердження про видалення, яке з'являється після натиснення кнопки видалення, користувача буде видалено з програми (рис. 2.24).

2.5 Висновки по розділу

У розділі описано функціональні можливості програмного продукту:

- деталі кімнати;
- відкриття нової кімнати;
- зміна номеру, призначеного для деталі реєстрації заїзду та виїзду;
- прийом нового клієнта;
- виписка клієнта;
- призначення кімнат відповідно до потреб клієнта;
- відомості про клієнта;
- реєстрація клієнта;
- Виїзд клієнта;
- Інформація про номер;
- Загальна кількість клієнтів, яка відображається у звіті.

Також описано результати тестування додатку, вони показали придатність розробки до впровадження.

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Охорона праці і техніка безпеки в готелі “Garden Plaza”

Згідно з завданням на дипломний проєкт необхідно виконати розробку автоматизованої інформаційної системи обліку для готелю «Garden Plaza». Створення (розробником) та експлуатація (працівниками готелю) такої системи обумовлює використання комп'ютерної техніки, що пов'язане з наявністю потенційно-небезпечних та шкідливих факторів у приміщенні де встановлено комп'ютери. Робота людей у такому приміщенні повинна регламентуватися вимогами охорони праці щодо цього.

В ст. 43 Конституції України записано: «Кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає, або на яку вільно погоджується», «Кожен має право на належні, безпечні і здорові умови праці, на заробітну плату, не нижчу від визначеної законом».

Забезпечення вимог охорони праці в Україні базується на нормах, записаних у Конституції України, Законах України «Про охорону праці», «Про охорону здоров'я», «Про забезпечення санітарного та епідеміологічного благополуччя населення», Кодексом цивільного захисту України, а також КЗпП.

Основним законодавчим документом в галузі охорони праці є Закон України «Про охорону праці» [21], дія якого поширюється на всі підприємства, установи і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють, а також залучені до праці на цих підприємствах. Стаття 4 Закону «Про охорону праці» формулює принципи державної політики у сфері охорони праці. Ці принципи в силу вказаного їх визначення, слід кваліфікувати і як принципи інституту охорони праці, які підлягають застосуванню при відсутності відповідних конкретних правових норм, і як такі, що визначають напрямки подальшого розвитку інституту

охорони праці, і як обов'язки держави, які вона взяла на себе і має додержувати при здійсненні своєї політики у майбутньому.

Охорона праці – один з центральних інститутів трудового права. Він має виключно практичне значення. Недодержання вимог охорони праці створює небезпеку для здоров'я і життя працівників. У свою чергу і ті, кого законодавець називає власником або уповноваженим ним органом, несуть сувору, у т.ч. кримінальну, відповідальність за порушення охорони праці.

Під час роботи працівників у готелях можуть виникати різноманітні небезпечні та шкідливі фактори. Небезпечні і шкідливі виробничі фактори згідно ДСТ 12.0.003-74 підрозділяються на чотири групи: фізичні, хімічні, біологічні і психофізіологічні [22].

Що стосується шкідливих факторів, то для працівників готелів можна виділити наступні найбільш суттєві із них, які є присутніми також і у готелі «Garden Plaza», що розглядається:

а) фізичні:

- Підвищені рівні електромагнітного випромінювання (від роботи техніки, розміщеної у приміщеннях готелю, зокрема, комп'ютерної та комунікаційної типу Wi-Fi, яким мають бути покриті усі сучасні готелі).
- Підвищені рівні запиленості повітря робочої зони (що є можливим при здійсненні прибирань окремих приміщень готелю, зокрема, таких що тривалий час не експлуатувалися).
- Знижена або підвищена вологість повітря робочої зони (у тому числі, під час вологих прибирань кімнат а також при очищенні поверхонь у санітарних вузлах номерів готелю).
- Підвищений рівень шуму (особливо від міжміської автомагістралі, якщо такі проходять поруч із будівлею).
- Підвищений рівень прямої та відбитої блескості (найбільш актуально для комп'ютерів, встановлених монітором до вікон).
- Недостатня яскравість світлового зображення.

- Відсутність або недостача природного світла (що часто спостерігається у коридорах готелів та службових приміщеннях, що часто проєктують без вікон).

- Недостатня освітленість робочої зони.

б) хімічні:

- Подразнюючої дії (при роботі з засобами побутової хімії, зокрема, для очищення поверхонь, прання, вологого прибирання, хімчистки і т.п.).

в) біологічні:

- Від дії патогенних мікроорганізмів (вірусів та бактерій від постояльців готелю).

г) психофізіологічні:

- Напруга зору.
- Напруга уваги.
- Інтелектуальні навантаження.
- Емоційні навантаження.
- Тривалі статичні навантаження.
- Монотонність праці.

Тільки урахування та нейтралізація усього комплексу вказаних факторів дозволяє забезпечити вимоги техніки безпеки та відсутність шкоди для здоров'я та, особливо, небезпеки для життя людей, задіяних у роботі готелю.

3.2 Долікарська допомога при ураженні електричним струмом в готелі “Garden Plaza”

Перша Медична Допомога (ПМД) – комплекс невідкладних медичних заходів, які проводяться людині, що раптово захворіла або постраждала, на місці пригоди та під час її транспортування до медичного закладу.

При проживанні в готельних комплексах зустрічаються випадки, коли працівниками готелю було допущено помилки при перевірці електроліній та

електричного обладнання. В результаті чого виникають електротравми та електричні опіки в працюючих та проживаючих, тут також слід брати до уваги і необережне поводження з електричними приборами самих проживаючих. Тому при виникненні таких ситуацій, працівники готелю мають мати відповідні знання щодо надання ПМД при ураженні електричним струмом.

Електротравма – травма, що виникає при дії електричного струму на організм.

Електричний опік – це ушкодження поверхні тіла під дією електричної дуги або великих струмів, що проходять через тіло людини.

Симптоми. Хворий миттєво втрачає свідомість, відмічається судомне скорочення м'язів, зупинка дихання, різкий розлад серцевої діяльності. Якщо ураження не призвело до миттєвої загибелі та через деякий час свідомість постраждалого відновилося, то в нього визначається головний біль, сонливість, загальна слабкість, млявість, пронос. Місцево відмічаються сліди опіку у вигляді жовтувато-бурих плям та смуг.

Електричні травми являють собою чітко виражені місцеві пошкодження тканин організму, викликані впливом електричного струму або електричної дуги. Зазвичай це ураження шкіри, рідше – інших м'яких тканин, а також зв'язок і кісток. Головне при наданні першої допомоги – якнайшвидше звільнити потерпілого від дії струму.

Для звільнення потерпілого при ураженні електричним струмом необхідно:

- вимкнути рубильник або викрутити запобіжник;
- вимкнути струмоведучі, мережу живлення;
- сухою палкою відкинути від потерпілого провід, який знаходиться

під напругою, відтягнути потерпілого від електричних проводів, від струмопровідних частин установки. При напрузі в установках до 1000 В можна взятися за сухий одяг потерпілого, не торкаючись відкритих частин тіла. Слід користуватися гумовими рукавичками або намотати на руку шарф,

прогумований плащ і т.п. Рекомендується стати на ізольований предмет (на суху дошку, на згорток сухого одягу).

- У випадку судорожного обхвату потерпілим електричного проводу, який знаходиться під напругою, розгорнути руки потерпілому, відриваючи його від проводу послідовним відгинанням окремих пальців. При цьому працівник, який надає допомогу, повинен бути у діелектричних рукавицях і знаходитись на ізольованій від землі основі.

Схема дій:

- переконатися у відсутності небезпеки;
- якщо постраждалий перебуває під дією електричного струму, при можливості припинити його дію: вимкнути джерело струму, відкинути електричний провід за допомогою сухої дерев'яної палиці чи іншого електронепровідного засобу;
- провести огляд постраждалого, визначити наявність свідомості, дихання;
- викликати бригаду екстреної (швидкої) медичної допомоги;
- якщо у постраждалого відсутнє дихання, розпочати проведення серцево-легеневої реанімації;
- якщо постраждалий без свідомості, але дихання збережене, надати постраждалому стабільного положення;
- накласти на місця опіку чисті, стерильні пов'язки;
- забезпечити постійний нагляд за постраждалим до приїзду бригади екстреної (швидкої) медичної допомоги;
- при погіршенні стану постраждалого до приїзду бригади екстреної (швидкої) медичної допомоги повторно зателефонувати диспетчеру екстреної медичної допомоги.

ВИСНОВКИ

У кваліфікаційній роботі на здобуття освітнього ступеня «бакалавр» відповідно до завдання була розроблена та протестована ІС супроводу клієнтської БД для готельного комплексу «Garden Plaza».

Початково було проаналізовано особливості діяльності компаній, що належать до зазначеного профілю готелів. Оскільки структура цього готелю є досить розгалуженою, – в даному виді готелю є велика кількість як працівників так і клієнтів, – тому для повноцінної процедури заселення/виселення необхідно витратити значний обсяг часу. З цього випливає, що розробка та впровадження АІС, що автоматизують супровід клієнтів, є обґрунтованою і ефективною з економічної точки зору.

Були встановлені вимоги до майбутнього програмного продукту та сформульовано завдання. Основна вимога полягає у максимальній простоті інтерфейсу користувача та обмеженні функцій до необхідних елементів, без зайвих опцій. З цієї причини використання готових систем не забезпечує найвищої ефективності в даному випадку, тому потрібна індивідуальна розробка програмного продукту.

Ще однією з поставлених вимог є те, що програма має бути універсальною з реалізацією на платформі Windows, Android, та майбутніми модифікаціями стосовно функціоналу програми. Також система має захист, тобто авторизацію для співробітників через пароль та логін. Проаналізовано існуючі методи та інструменти та обґрунтовано їх вибір для реалізації даної розробки. Це середовище Visual Studio, мова програмування C# та бібліотека Windows Forms.

До того ж, перед реалізацією обрано відповідну СКБД, а саме, РБД, яка забезпечить зберігання даних в вигляді таблиць, які будуть мати між собою зв'язок. Доступ до цих даних буде здійснюватися за допомогою MySQL. Вибір СКБД обумовлений тим, що дані, необхідні нам для роботи програми будуть

простими та носитимуть довідковий характер, наприклад цифрові дані або ж текстові.

Розробку програмного продукту успішно апробовано, а також описано функціональні можливості:

- деталі кімнати;
- відкриття нової кімнати;
- зміна номеру, призначеного для деталі реєстрації заїзду та виїзду;
- прийом нового клієнта;
- виписка клієнта;
- призначення кімнат відповідно до потреб клієнта;
- відомості про клієнта;
- реєстрація клієнта;
- виїзд клієнта;
- інформація про номер;
- загальна кількість клієнтів, яка відображається у звіті.

Отримані результати розробки можуть бути впроваджені для ефективного управління великими готелями та готельними комплексами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 28681.1-95. Проектування туристських послуг. Вид. офіц.
2. ДСТУ 28681.3-95. Вимоги з забезпечення безпеки туристів та екскурсантів. Вид. офіц.
3. ДСТУ 28681.4-95. Класифікація готелів. Вид. офіц.
4. Воронцов Ю.А., Козинец А.В. Стандарти сервісів для створення розподілених ІС / Ю.А.Воронцов, А.В.Козинец, // Вік якості. 2016. №3.
5. Мова програмування С# [Електронний ресурс]- Режим доступу: https://uk.wikipedia.org/wiki/C_Sharp.
6. Крістіан Нейгел та ін. С# 5.0 та платформа .NET 4.5 для професіоналів = Professional C# 5.0 and .NET 4.5., 2013. 1440 с.
7. Безкоштовні середовища програмування С ++, С # [електронний ресурс] //Режим доступу: <http://www.interestprograms.ru/articles/ide/besplatnye-sredyprogramirovaniya-c-csharp>.
8. Скит Д. С# для професіоналів: тонкості програмування, 3-є вид. Пер. з англ. – Львів «Вільямс», 2014. – 608 с.
9. Кузніцов М.В. Об'єктно-орієнтоване програмування. – БХВ-Одеса, 2017.
10. Дюбуа П. MySQL.: Пер. з англ. К.: «Вільямс», - 1056 с.
11. Хокинс С. Администрирование Web-сервера Apache и руководство по электронной коммерции.: Пер. с англ. М.: Издательский дом «Вильямс», - 336 с.: ил.
12. Герберт Шілдрт. Повне керівництво С# 4.0. Київ. 2007. 776 с.
13. Котов О.М. Мова С#. Короткий опис та введення у технології програмування. Навчальне видання, 2014 210 с.
14. Harrison G. and Feuerstein C. MySQL Stored Procedure Programming. 2006.

15. Silberschatz, Abraham; Sudarshan, S. (2011). Database system concepts (вид. 6).
16. Butek, R.: Which style of WSDL should I use?, IBM developerWorks 2003
17. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns – Elements of reusable object-oriented software. Addison-Wesley 1995, ISBN 0201633612
18. "SOA Source Book - What Is SOA?". collaboration.opengroup.org. Retrieved, 2021.
19. "Service-Oriented Architecture Standards - The Open Group". www.opengroup.org.
20. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. Відомості Верховної Ради України (ВВР), 1992, № 49, ст.668.
21. ГОСТ 12.0.003-74 Небезпечні та шкідливі виробничі фактори. Класифікація.
22. Турченко М. О. Маркетинг : навч. посіб. / М. О. Турченко, М. Д. Швець, Т. С. Карпан. – Рівне : НУВГП, 2020. – 293 с.
23. Домедична допомога (алгоритми, маніпуляції): Методичний посібник / [В.О. Крилюк, С.О. Гур'єв, А.А. Гудима та ін.] – К.: НВП "Інтерсервіс", 2014. – 84 с.
24. Екстрена та невідкладна медична допомога. Том І: допомога травмованим на догоспітальному етапі: національний підручник / В.О. Крилюк, С.О. Гур'єв, Г.В. Загорій, А.А. Гудима, Н.І. Іскра та ін. – Київ. – 2017. – 504 с
25. Вертайм К. Цифровий маркетинг: як збільшити продажі з допомогою мереж, вікі-ресурсів, мобільних телефонів та ін.. технологій: пер. з англ. / Кент Вертайм, Ян Фенвик; Пер. з англ. – К.: Юрайт. - 377 с.

26. Michael Bell (2008). "Introduction to Service-Oriented Modeling".
Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley &
Sons. p. 3. ISBN 978-0-470-14111-3.

ДОДАТКИ

ЛІСТИНГ ПРОГРАМНОГО КОДУ

Файл form.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HotelManagementVideo
{
    public partial class Form1 : Form
    {
        function fn = new function();
        String query;
        public Form1()
        {
            InitializeComponent();
        }
        private void btnExit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        private void btnLogin_Click(object sender, EventArgs e)
        {
            query = "select username,pass from employee where username
= '"+txtUsername.Text+"' and pass = '"+txtPassword.Text+"'";
            DataSet ds = fn.getData(query);
            if (ds.Tables[0].Rows.Count != 0)
            {
                labelError.Visible = false;
                Dashboard dash = new Dashboard();
                this.Hide();
                dash.Show();
            }
            else
            {
                labelError.Visible = true;
                txtPassword.Clear();
            }
        }
    }
}
```

Файл Function.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace HotelManagementVideo
```



```

{
    class function
    {
        protected SqlConnection getConnection()
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = "data source = COMPUTER;database=myHotel;integrated
security =True";
            return con;
        }
        public DataSet getData(String query) // Get data from database
        {
            SqlConnection con = getConnection();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = con;
            cmd.CommandText = query;
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds);
            return ds;
        }
        public void setData(String query, String message) // Insertion Deletion Updation
        {
            SqlConnection con = getConnection();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = con;
            con.Open();
            cmd.CommandText = query;
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("'" + message + "'", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
        public SqlDataReader getForCombo(String query)
        {
            SqlConnection con = getConnection();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = con;
            con.Open();
            cmd = new SqlCommand(query, con);
            SqlDataReader sdr = cmd.ExecuteReader();
            return sdr;
        }
    }
}

```

Файл

```

namespace HotelManagementVideo
{
    partial class Dashboard
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>

```

```

protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}
#region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Dashboard));
    this.panel1 = new System.Windows.Forms.Panel();
    this.MovingPanel = new System.Windows.Forms.Panel();
    this.btnEmployee = new Guna.UI2.WinForms.Guna2Button();
    this.btnCustomerDetails = new Guna.UI2.WinForms.Guna2Button();
    this.btnCheckout = new Guna.UI2.WinForms.Guna2Button();
    this.btnCustomerRegistration = new Guna.UI2.WinForms.Guna2Button();
    this.btnAddRoom = new Guna.UI2.WinForms.Guna2Button();
    this.panel2 = new System.Windows.Forms.Panel();
    this.btnExit = new Guna.UI2.WinForms.Guna2CircleButton();
    this.btnMinimize = new Guna.UI2.WinForms.Guna2CircleButton();
    this.guna2Elipse1 = new Guna.UI2.WinForms.Guna2Elipse(this.components);
    this.guna2Elipse2 = new Guna.UI2.WinForms.Guna2Elipse(this.components);
    this.guna2Elipse3 = new Guna.UI2.WinForms.Guna2Elipse(this.components);
    this.guna2Elipse4 = new Guna.UI2.WinForms.Guna2Elipse(this.components);
    this.guna2Elipse5 = new Guna.UI2.WinForms.Guna2Elipse(this.components);
    this.customerDetails1 = new
HotelManagementVideo.All_User_Control.CustomerDetails();
    this.uC_CustomerCheckOut1 = new
HotelManagementVideo.All_User_Control.UC_CustomerCheckOut();
    this.uC_CustomerRegistration1 = new
HotelManagementVideo.All_User_Control.UC_CustomerRegistration();
    this.uC_AddRoom1 = new HotelManagementVideo.All_User_Control.UC_AddRoom();
    this.uC_Employee1 = new HotelManagementVideo.All_User_Control.UC_Employee();
    this.panel1.SuspendLayout();
    this.panel2.SuspendLayout();
    this.SuspendLayout();
    //
    // panel1
    //
    this.panel1.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(62)))),
((int)(((byte)(120)))), ((int)(((byte)(138)))));
    this.panel1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.panel1.Controls.Add(this.MovingPanel);
    this.panel1.Controls.Add(this.btnEmployee);
    this.panel1.Controls.Add(this.btnCustomerDetails);
    this.panel1.Controls.Add(this.btnCheckout);
    this.panel1.Controls.Add(this.btnCustomerRegistration);
    this.panel1.Controls.Add(this.btnAddRoom);
    this.panel1.Location = new System.Drawing.Point(57, 12);
    this.panel1.Name = "panel1";
    this.panel1.Size = new System.Drawing.Size(1249, 130);
    this.panel1.TabIndex = 0;
    //
    // MovingPanel
    //

```

```

        this.MovingPanel.BackColor =
System.Drawing.Color.FromArgb(((int)(((byte)(128)))), ((int)(((byte)(255)))),
((int)(((byte)(255))))));
        this.MovingPanel.Location = new System.Drawing.Point(31, 111);
        this.MovingPanel.Name = "MovingPanel";
        this.MovingPanel.Size = new System.Drawing.Size(199, 5);
        this.MovingPanel.TabIndex = 5;
        //
        // btnEmployee
        //
        this.btnEmployee.BorderRadius = 26;
        this.btnEmployee.ButtonMode = Guna.UI2.WinForms.Enums.ButtonMode.RadioButton;
        this.btnEmployee.CheckedState.FillColor = System.Drawing.Color.White;
        this.btnEmployee.CheckedState.ForeColor = System.Drawing.Color.Black;
        this.btnEmployee.CheckedState.Parent = this.btnEmployee;
        this.btnEmployee.CustomImages.Parent = this.btnEmployee;
        this.btnEmployee.FillColor =
System.Drawing.Color.FromArgb(((int)(((byte)(62)))), ((int)(((byte)(120)))),
((int)(((byte)(138))))));
        this.btnEmployee.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.btnEmployee.ForeColor = System.Drawing.Color.White;
        this.btnEmployee HoverState.Parent = this.btnEmployee;
        this.btnEmployee.Image =
((System.Drawing.Image)(resources.GetObject("btnEmployee.Image")));
        this.btnEmployee.ImageSize = new System.Drawing.Size(50, 50);
        this.btnEmployee.Location = new System.Drawing.Point(994, 24);
        this.btnEmployee.Name = "btnEmployee";
        this.btnEmployee.ShadowDecoration.Parent = this.btnEmployee;
        this.btnEmployee.Size = new System.Drawing.Size(236, 85);
        this.btnEmployee.TabIndex = 4;
        this.btnEmployee.Text = "Employee";
        this.btnEmployee.Click += new System.EventHandler(this.btnEmployee_Click);
        //
        // btnCustomerDetails
        //
        this.btnCustomerDetails.BorderRadius = 26;
        this.btnCustomerDetails.ButtonMode =
Guna.UI2.WinForms.Enums.ButtonMode.RadioButton;
        this.btnCustomerDetails.CheckedState.FillColor = System.Drawing.Color.White;
        this.btnCustomerDetails.CheckedState.ForeColor = System.Drawing.Color.Black;
        this.btnCustomerDetails.CheckedState.Parent = this.btnCustomerDetails;
        this.btnCustomerDetails.CustomImages.Parent = this.btnCustomerDetails;
        this.btnCustomerDetails.FillColor =
System.Drawing.Color.FromArgb(((int)(((byte)(62)))), ((int)(((byte)(120)))),
((int)(((byte)(138))))));
        this.btnCustomerDetails.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.btnCustomerDetails.ForeColor = System.Drawing.Color.White;
        this.btnCustomerDetails HoverState.Parent = this.btnCustomerDetails;
        this.btnCustomerDetails.Image =
((System.Drawing.Image)(resources.GetObject("btnCustomerDetails.Image")));
        this.btnCustomerDetails.ImageSize = new System.Drawing.Size(30, 30);
        this.btnCustomerDetails.Location = new System.Drawing.Point(752, 24);
        this.btnCustomerDetails.Name = "btnCustomerDetails";
        this.btnCustomerDetails.ShadowDecoration.Parent = this.btnCustomerDetails;
        this.btnCustomerDetails.Size = new System.Drawing.Size(236, 85);
        this.btnCustomerDetails.TabIndex = 3;
        this.btnCustomerDetails.Text = "Customer Details";
        this.btnCustomerDetails.Click += new
System.EventHandler(this.btnCustomerDetails_Click);
        //
        // btnCheckout

```

```

//
this.btnCheckout.BorderRadius = 26;
this.btnCheckout.ButtonMode = Guna.UI2.WinForms.Enums.ButtonMode.RadioButton;
this.btnCheckout.CheckedState.FillColor = System.Drawing.Color.White;
this.btnCheckout.CheckedState.ForeColor = System.Drawing.Color.Black;
this.btnCheckout.CheckedState.Parent = this.btnCheckout;
this.btnCheckout.CustomImages.Parent = this.btnCheckout;
this.btnCheckout.FillColor =
System.Drawing.Color.FromArgb(((int)(((byte)62))), ((int)(((byte)120))),
((int)(((byte)138))));
    this.btnCheckout.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
    this.btnCheckout.ForeColor = System.Drawing.Color.White;
    this.btnCheckout HoverState.Parent = this.btnCheckout;
    this.btnCheckout.Image =
((System.Drawing.Image)(resources.GetObject("btnCheckout.Image")));
    this.btnCheckout.ImageSize = new System.Drawing.Size(50, 50);
    this.btnCheckout.Location = new System.Drawing.Point(510, 24);
    this.btnCheckout.Name = "btnCheckout";
    this.btnCheckout.ShadowDecoration.Parent = this.btnCheckout;
    this.btnCheckout.Size = new System.Drawing.Size(236, 85);
    this.btnCheckout.TabIndex = 2;
    this.btnCheckout.Text = "Check Out";
    this.btnCheckout.Click += new System.EventHandler(this.btnCheckout_Click);
//
// btnCustomerRegistration
//
this.btnCustomerRegistration.BorderRadius = 26;
this.btnCustomerRegistration.ButtonMode =
Guna.UI2.WinForms.Enums.ButtonMode.RadioButton;
    this.btnCustomerRegistration.CheckedState.FillColor =
System.Drawing.Color.White;
    this.btnCustomerRegistration.CheckedState.ForeColor =
System.Drawing.Color.Black;
    this.btnCustomerRegistration.CheckedState.Parent =
this.btnCustomerRegistration;
    this.btnCustomerRegistration.CustomImages.Parent =
this.btnCustomerRegistration;
    this.btnCustomerRegistration.FillColor =
System.Drawing.Color.FromArgb(((int)(((byte)62))), ((int)(((byte)120))),
((int)(((byte)138))));
    this.btnCustomerRegistration.Font = new System.Drawing.Font("Segoe UI",
14.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
    this.btnCustomerRegistration.ForeColor = System.Drawing.Color.White;
    this.btnCustomerRegistration.HoverState.Parent =
this.btnCustomerRegistration;
    this.btnCustomerRegistration.Image =
((System.Drawing.Image)(resources.GetObject("btnCustomerRegistration.Image")));
    this.btnCustomerRegistration.ImageSize = new System.Drawing.Size(35, 35);
    this.btnCustomerRegistration.Location = new System.Drawing.Point(255, 24);
    this.btnCustomerRegistration.Name = "btnCustomerRegistration";
    this.btnCustomerRegistration.ShadowDecoration.Parent =
this.btnCustomerRegistration;
    this.btnCustomerRegistration.Size = new System.Drawing.Size(249, 85);
    this.btnCustomerRegistration.TabIndex = 1;
    this.btnCustomerRegistration.Text = "Customer Registration";
    this.btnCustomerRegistration.Click += new
System.EventHandler(this.btnCustomerRegistration_Click);
//
// btnAddRoom
//
this.btnAddRoom.BorderRadius = 26;
this.btnAddRoom.ButtonMode = Guna.UI2.WinForms.Enums.ButtonMode.RadioButton;

```

```

        this.btnAddRoom.CheckedState.FillColor = System.Drawing.Color.White;
        this.btnAddRoom.CheckedState.ForeColor = System.Drawing.Color.Black;
        this.btnAddRoom.CheckedState.Parent = this.btnAddRoom;
        this.btnAddRoom.CustomImages.Parent = this.btnAddRoom;
        this.btnAddRoom.FillColor =
System.Drawing.Color.FromArgb(((int)(((byte)(62)))), ((int)(((byte)(120)))),
((int)(((byte)(138))))));
        this.btnAddRoom.Font = new System.Drawing.Font("Segoe UI", 14.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.btnAddRoom.ForeColor = System.Drawing.Color.White;
        this.btnAddRoom HoverState.Parent = this.btnAddRoom;
        this.btnAddRoom.Image =
((System.Drawing.Image)(resources.GetObject("btnAddRoom.Image")));
        this.btnAddRoom.ImageSize = new System.Drawing.Size(40, 40);
        this.btnAddRoom.Location = new System.Drawing.Point(13, 24);
        this.btnAddRoom.Name = "btnAddRoom";
        this.btnAddRoom.ShadowDecoration.Parent = this.btnAddRoom;
        this.btnAddRoom.Size = new System.Drawing.Size(236, 85);
        this.btnAddRoom.TabIndex = 0;
        this.btnAddRoom.Text = "Add Room";
        this.btnAddRoom.Click += new System.EventHandler(this.btnAddRoom_Click);
//
// panel2
//
        this.panel2.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
        this.panel2.Controls.Add(this.uC_Employee1);
        this.panel2.Controls.Add(this.customerDetails1);
        this.panel2.Controls.Add(this.uC_CustomerCheckout1);
        this.panel2.Controls.Add(this.uC_CustomerRegistration1);
        this.panel2.Controls.Add(this.uC_AddRoom1);
        this.panel2.Location = new System.Drawing.Point(12, 159);
        this.panel2.Name = "panel2";
        this.panel2.Size = new System.Drawing.Size(1297, 505);
        this.panel2.TabIndex = 1;
//
// btnExit
//
        this.btnExit.CheckedState.Parent = this.btnExit;
        this.btnExit.CustomImages.Parent = this.btnExit;
        this.btnExit.FillColor = System.Drawing.Color.Empty;
        this.btnExit.Font = new System.Drawing.Font("Segoe UI", 9F);
        this.btnExit.ForeColor = System.Drawing.Color.White;
        this.btnExit.HoverState.Parent = this.btnExit;
        this.btnExit.Image =
((System.Drawing.Image)(resources.GetObject("btnExit.Image")));
        this.btnExit.ImageSize = new System.Drawing.Size(30, 30);
        this.btnExit.Location = new System.Drawing.Point(8, 12);
        this.btnExit.Name = "btnExit";
        this.btnExit.ShadowDecoration.Mode =
Guna.UI2.WinForms.Enums.ShadowMode.Circle;
        this.btnExit.ShadowDecoration.Parent = this.btnExit;
        this.btnExit.Size = new System.Drawing.Size(43, 36);
        this.btnExit.TabIndex = 2;
        this.btnExit.Click += new System.EventHandler(this.btnExit_Click);
//
// btnMinimize
//
        this.btnMinimize.CheckedState.Parent = this.btnMinimize;
        this.btnMinimize.CustomImages.Parent = this.btnMinimize;

```

```

        this.btnMinimize.FillColor = System.Drawing.Color.Empty;
        this.btnMinimize.Font = new System.Drawing.Font("Segoe UI", 9F);
        this.btnMinimize.ForeColor = System.Drawing.Color.White;
        this.btnMinimize.HoverState.Parent = this.btnMinimize;
        this.btnMinimize.Image =
((System.Drawing.Image)(resources.GetObject("btnMinimize.Image")));
        this.btnMinimize.ImageSize = new System.Drawing.Size(30, 30);
        this.btnMinimize.Location = new System.Drawing.Point(8, 45);
        this.btnMinimize.Name = "btnMinimize";
        this.btnMinimize.ShadowDecoration.Mode =
Guna.UI2.WinForms.Enums.ShadowMode.Circle;
        this.btnMinimize.ShadowDecoration.Parent = this.btnMinimize;
        this.btnMinimize.Size = new System.Drawing.Size(43, 36);
        this.btnMinimize.TabIndex = 3;
        this.btnMinimize.Click += new System.EventHandler(this.btnMinimize_Click);
        //
        // guna2Elipse1
        //
        this.guna2Elipse1.BorderRadius = 30;
        this.guna2Elipse1.TargetControl = this;
        //
        // guna2Elipse2
        //
        this.guna2Elipse2.BorderRadius = 30;
        this.guna2Elipse2.TargetControl = this;
        //
        // guna2Elipse3
        //
        this.guna2Elipse3.BorderRadius = 30;
        this.guna2Elipse3.TargetControl = this;
        //
        // guna2Elipse4
        //
        this.guna2Elipse4.BorderRadius = 30;
        this.guna2Elipse4.TargetControl = this;
        //
        // guna2Elipse5
        //
        this.guna2Elipse5.BorderRadius = 30;
        this.guna2Elipse5.TargetControl = this;
        //
        // customerDetails1
        //
        this.customerDetails1.BackColor = System.Drawing.Color.White;
        this.customerDetails1.Location = new System.Drawing.Point(0, 0);
        this.customerDetails1.Name = "customerDetails1";
        this.customerDetails1.Size = new System.Drawing.Size(1342, 583);
        this.customerDetails1.TabIndex = 3;
        //
        // uC_CustomerCheckOut1
        //
        this.uC_CustomerCheckOut1.AutoSize = true;
        this.uC_CustomerCheckOut1.BackColor = System.Drawing.Color.White;
        this.uC_CustomerCheckOut1.Location = new System.Drawing.Point(0, 0);
        this.uC_CustomerCheckOut1.Name = "uC_CustomerCheckOut1";
        this.uC_CustomerCheckOut1.Size = new System.Drawing.Size(1316, 582);
        this.uC_CustomerCheckOut1.TabIndex = 2;
        //
        // uC_CustomerRegistration1
        //
        this.uC_CustomerRegistration1.BackColor = System.Drawing.Color.White;
        this.uC_CustomerRegistration1.Location = new System.Drawing.Point(0, 0);
        this.uC_CustomerRegistration1.Name = "uC_CustomerRegistration1";

```

```

this.uC_CustomerRegistration1.Size = new System.Drawing.Size(1342, 583);
this.uC_CustomerRegistration1.TabIndex = 1;
//
// uC_AddRoom1
//
this.uC_AddRoom1.BackColor = System.Drawing.Color.White;
this.uC_AddRoom1.Location = new System.Drawing.Point(0, 0);
this.uC_AddRoom1.Name = "uC_AddRoom1";
this.uC_AddRoom1.Size = new System.Drawing.Size(1337, 583);
this.uC_AddRoom1.TabIndex = 0;
//
// uC_Employee1
//
this.uC_Employee1.BackColor = System.Drawing.Color.White;
this.uC_Employee1.Location = new System.Drawing.Point(0, -1);
this.uC_Employee1.Name = "uC_Employee1";
this.uC_Employee1.Size = new System.Drawing.Size(1342, 583);
this.uC_Employee1.TabIndex = 4;
//
// Dashboard
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)62)))),
((int)(((byte)120))), ((int)(((byte)138))));
this.ClientSize = new System.Drawing.Size(1329, 676);
this.Controls.Add(this.btnMinimize);
this.Controls.Add(this.btnExit);
this.Controls.Add(this.panel2);
this.Controls.Add(this.panel1);
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
this.Name = "Dashboard";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Dashboard";
this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
this.Load += new System.EventHandler(this.Dashboard_Load);
this.panel1.ResumeLayout(false);
this.panel2.ResumeLayout(false);
this.panel2.PerformLayout();
this.ResumeLayout(false);
}
#endregion
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.Panel panel2;
private Guna.UI2.WinForms.Guna2CircleButton btnExit;
private Guna.UI2.WinForms.Guna2CircleButton btnMinimize;
private Guna.UI2.WinForms.Guna2Button btnEmployee;
private Guna.UI2.WinForms.Guna2Button btnCustomerDetails;
private Guna.UI2.WinForms.Guna2Button btnCheckout;
private Guna.UI2.WinForms.Guna2Button btnCustomerRegistration;
private Guna.UI2.WinForms.Guna2Button btnAddRoom;
private System.Windows.Forms.Panel MovingPanel;
private Guna.UI2.WinForms.Guna2Elipse guna2Elipse1;
private All_User_Control.UC_AddRoom uC_AddRoom1;
private Guna.UI2.WinForms.Guna2Elipse guna2Elipse2;
private All_User_Control.UC_CustomerRegistration uC_CustomerRegistration1;
private Guna.UI2.WinForms.Guna2Elipse guna2Elipse3;
private All_User_Control.UC_CustomerCheckOut uC_CustomerCheckOut1;
private Guna.UI2.WinForms.Guna2Elipse guna2Elipse4;
private All_User_Control.CustomerDetails customerDetails1;
private Guna.UI2.WinForms.Guna2Elipse guna2Elipse5;
private All_User_Control.UC_Employee uC_Employee1;
}

```

```
}
```

Файл program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HotelManagementVideo
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```


ТЕСТУВАННЯ ПРОГРАМИ

The screenshot shows the 'Add New Room' interface. At the top, there is a navigation bar with five options: 'Add Room' (highlighted), 'Customer Registration', 'Check Out', 'Customer Details', and 'Employee'. Below the navigation bar, the main content area is titled 'Add New Room'. On the left, there is a table with one row of data:

1	301	Non-Ac	Single	350	YES

On the right, there are several form fields:

- Room Number: 302
- Room Type: Ac (dropdown menu)
- Bed: Double (dropdown menu)
- Price: 450

At the bottom right, there is a button labeled 'Add Room'.

Рисунок 1 – Інтерфейс програмного продукту (запис кімнат)

The screenshot shows the 'Customer Registration' interface. At the top, there is a navigation bar with five options: 'Add Room', 'Customer Registration' (highlighted), 'Check Out', 'Customer Details', and 'Employee'. Below the navigation bar, the main content area is titled 'Customer Registration'. The form contains the following fields:

- Name: Михайло
- ID Proof: 2365
- Bed: Single (dropdown menu)
- Mobile No: 068-234-15-69
- Address: Шевченка,8
- Room Type: (dropdown menu)
- Nationality: українець
- Check In: 25.05.2022 (calendar icon)
- Room No: (dropdown menu)
- Gender: Male (dropdown menu)
- Price: 450
- Date of Birth: 31.03.2002 (calendar icon)

At the bottom right, there is a button labeled 'Allote Room'.

Рисунок 2 – Інтерфейс програмного продукту (запис клієнтів)

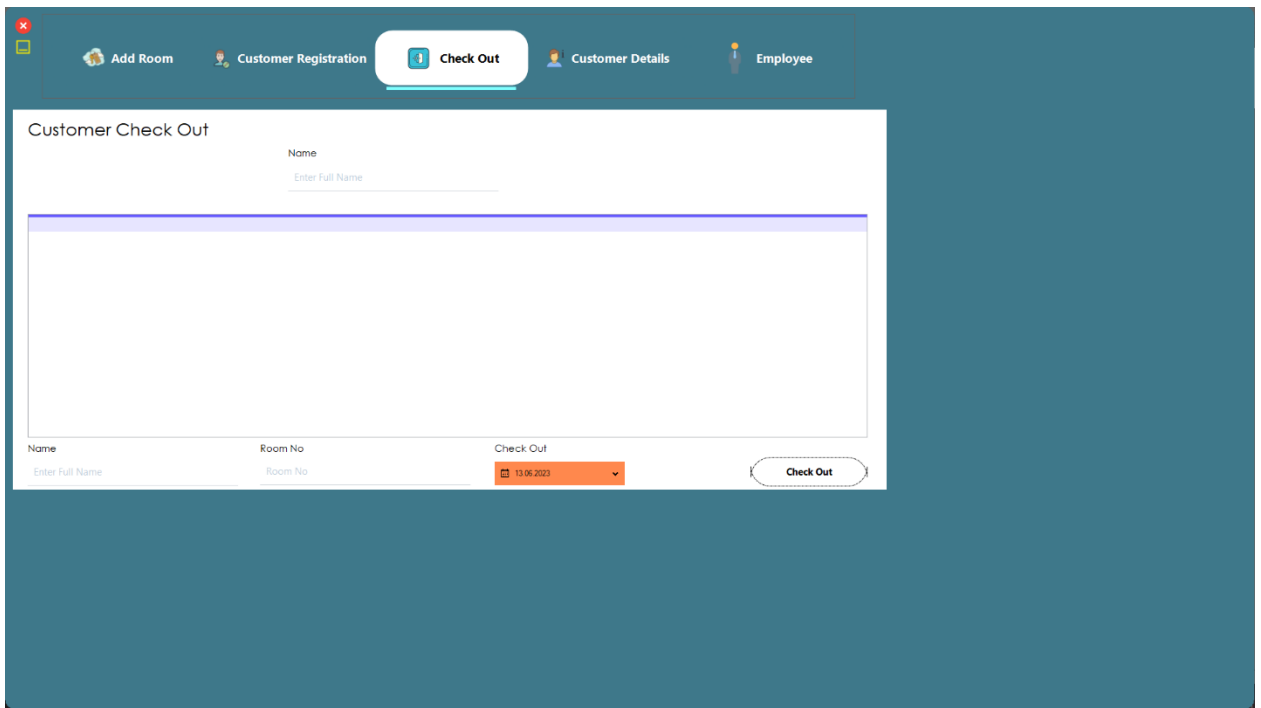


Рисунок 3 – Інтерфейс програмного продукту (виселення клієнтів)

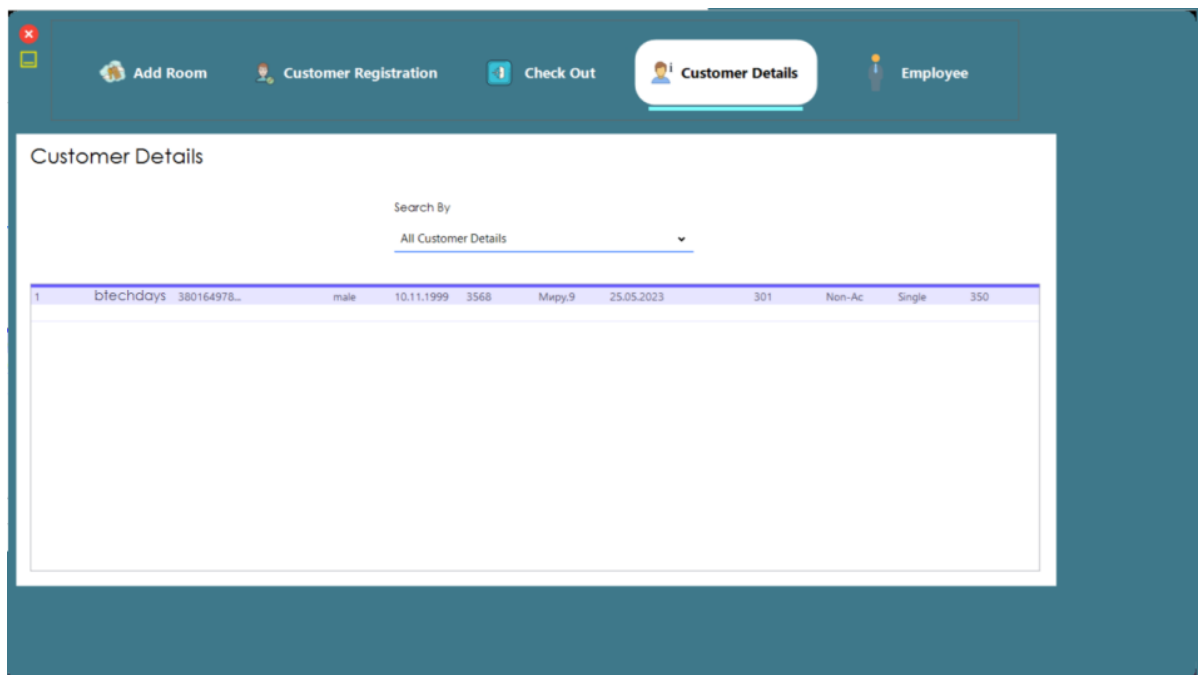


Рисунок 4 – Інтерфейс програмного продукту (перевірка даних клієнтів)

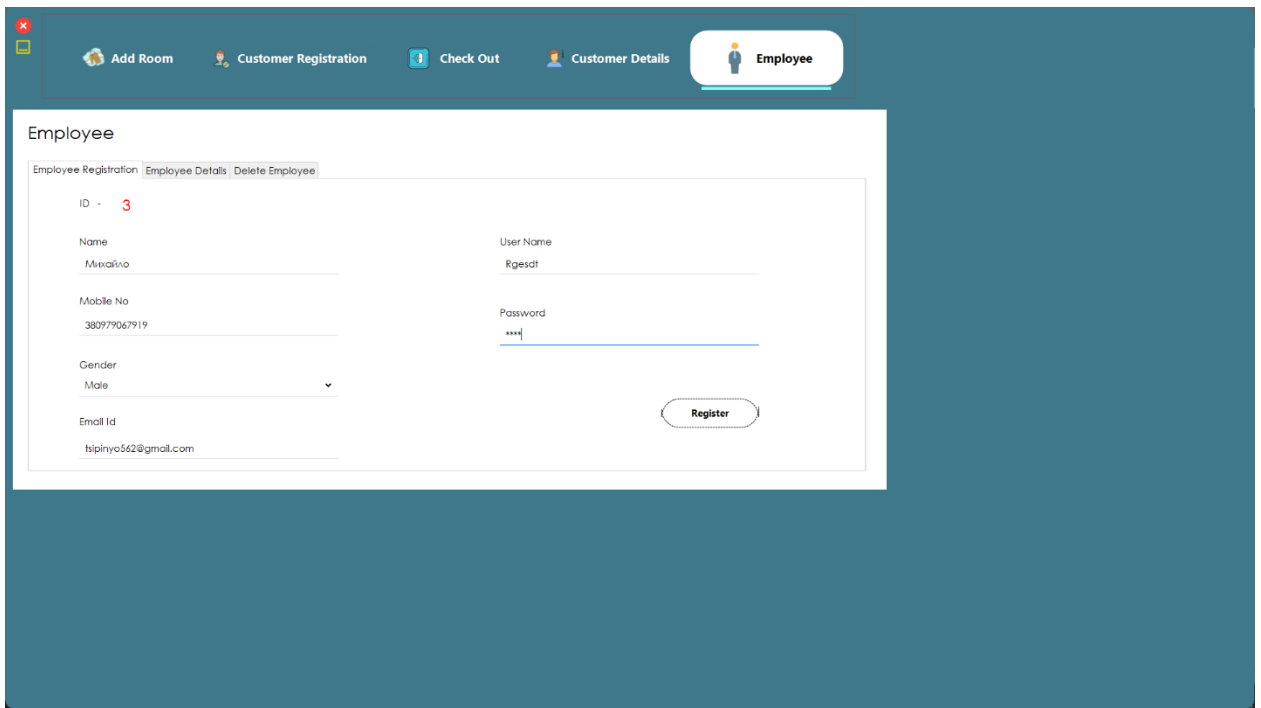


Рисунок 5 – Інтерфейс програмного продукту (реєстрація користувачів)

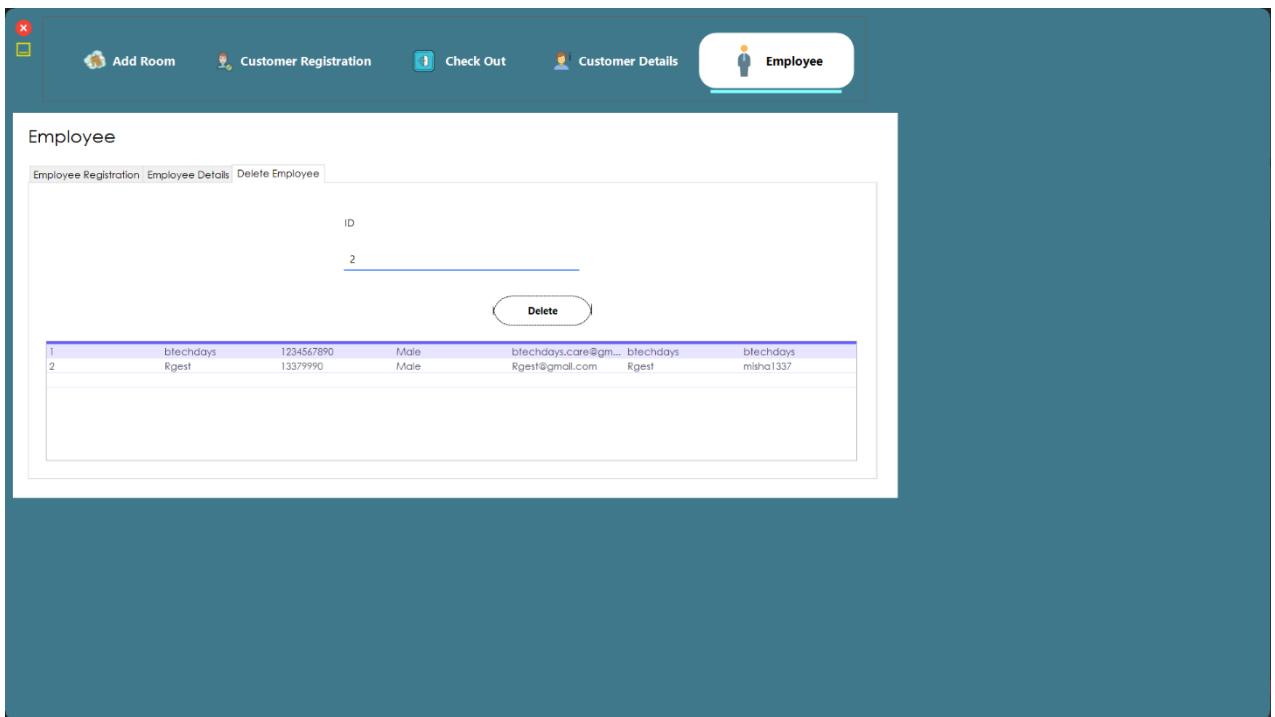


Рисунок 6 – Інтерфейс програмного продукту (перегляд користувачів)