

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д. е. н., проф. П. М. Грицюк

« ____ » _____ 2021 р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ
«БАКАЛАВР»**

**Інформаційна система підтримки функціонування студентських
гуртожитків НУВГП**

Виконав:

здобувачка вищої освіти за ОПП
«Інформаційні системи та технології»
спеціальності 126 «Інформаційні системи
та технології», групи ІСТ-41

Наконечна Юлія Анатоліївна

Керівник:

к.е.н., доцент Волошин В.С.

Рецензент:

к.т.н., доцент Василів В.Б.

Рівне – 2021

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 55с., 63 рис., додатки на 59 стор., 12 літературних джерел.

Актуальність теми даної бакалаврської роботи полягає в тому, що в даний час, в більшості гуртожитків коменданти, адміністратори керують цією інформацією в паперовому варіанті, але такий спосіб вже не підходить для сучасного суспільства, так як людина фізично не може устежити за всіма аспектами своєї роботи, а помилки можуть привести до незворотних наслідків, тому швидка і зручна робота автоматизованих інформаційних систем безпосередньо впливає на ефективність діяльності будь-якої сучасної організації або підприємства.

Об'єкт дослідження бакалаврської роботи – студентські гуртожитки НУВГП.

Предметною областю роботи є автоматизація менеджменту студентських гуртожитків.

Метою бакалаврської роботи є розробка автоматизованої інформаційної системи керування гуртожитками, за допомогою якої буде здійснюватися взаємодія керуючого з різними інформаційними аспектами його діяльності.

У бакалаврській роботі було: здійснено аналіз предметної області, визначений набір інструментів, що сприяють проектування і створення інформаційної системи, розроблено проект інформаційної системи, який включає в себе зображення функціональної структури; реалізовано проект завдяки побудові бази даних інформаційної системи, розробці інтерфейсу, формуванню вихідної інформації.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, DESKTOP-ЗАСТОСУНОК, БАЗА ДАНИХ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ ГУРТОЖИТКІВ.....	6
1.1. Аналіз предметної області.....	6
1.2. Інформаційна система «БІТ.Гуртожиток» як засіб автоматизації обліку проживання.....	11
1.3. Сучасні тренди популярності мов програмування	14
РОЗДІЛ 2. ІНФОРМАЦІЙНА СИСТЕМА ЯК ЗАСІБ ПІДТРИМКИ ФУНКЦІОНУВАННЯ СТУДЕНТСЬКИХ ГУРТОЖИТКІВ	22
2.1. Логічна модель даних системи.....	22
2.2. Функціональні можливості	31
2.3. Аналіз вихідної інформації та інтерфейс користувача	47
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	57

ВСТУП

У сучасному світі, в будь-якій організації особливе значення має людина, яка керує певним видом діяльності. Гуртожитки не є винятком.

Керуючий гуртожитком – професія, пов'язана з роботою і обробкою великих обсягів інформації. В даний час, в більшості гуртожитків коменданти, адміністратори керують цією інформацією за допомогою старого перевіреного методу - в паперовому варіанті. Але такий спосіб вже не підходить для сучасного комп'ютеризованого суспільства, так як людина фізично не може устежити за всіма аспектами своєї роботи, а помилки можуть привести до незворотних наслідків. Навіть якщо в гуртожитку є комп'ютерна підтримка і спеціалізоване програмне забезпечення, то це програмне забезпечення не завжди відповідає вимогам самих працівників і не містить в собі повного спектру функцій. Безпомилково швидка і зручна робота автоматизованих інформаційних систем безпосередньо впливає на ефективність діяльності будь-якої сучасної організації або підприємства.

Метою роботи є створення desktop-застосунку для студентських гуртожитків НУВГП.

Об'єктом роботи є desktop-застосунок.

Предметом роботи є використання сучасної мови програмування C#, методів програмного забезпечення та платформ для розробки desktop-застосунку для студентських гуртожитків НУВГП.

Завданням бакалаврської роботи є:

- проаналізувати функціонування гуртожитків;
- дослідити інформаційну систему «БІТ.Гуртожиток»;
- описати сучасні тренди популярності мов програмування;
- проектування та розробка інформаційної системи «Гуртожитки НУВГП»;
- створення бази даних ІС «Гуртожитки НУВГП»;

- розробка інтерфейсу ІС «Гуртожитки НУВГП»;

Програмні продукти, що використовувались під час написання бакалаврської роботи: Navicat for MySQL, MySQL-Front, Microsoft Visual Studio, Microsoft Word, MS Excel, MySQL, C#.

РОЗДІЛ 1. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ ГУРТОЖИТКІВ

1.1. Аналіз предметної області

Гуртожитки НУВГП функціонують відповідно до положення Національного університету водного господарства та природокористування від вересня 2009 року «Про студентський гуртожиток» [6]. Гуртожитки НУВГП призначені для проживання на період навчання студентів з інших міст, слухачів, студентів з числа дітей сиріт та дітей, позбавлених батьківського піклування, за умови відсутності в них іншого місця проживання, аспірантів та докторантів. Також є можливість проживання для іноземних студентів, які навчаються у НУВГП.

Види гуртожитків зображено на рисунку 1.1.



Рис. 1.1. Види гуртожитків

Іногородні студенти, які навчаються на заочній формі навчання на період проведення екзаменаційної сесії та іногородні абітурієнти на період проведення вступних випробувань, також можуть бути поселені у гуртожитки.

Як виняток, працівники НУВГП можуть проживати в студентських гуртожитках до закінчення навчального року лише за рішенням ректора університету і за погодженням з первинною профспілковою організацією студентів та органами студентського самоврядування.

Студенти можуть бути переселені до іншої кімнати з поважних причин або за рішення дирекції студентського містечка. Обов'язки керівництва університету відповідно до [6] зображено на рисунку 1.2.

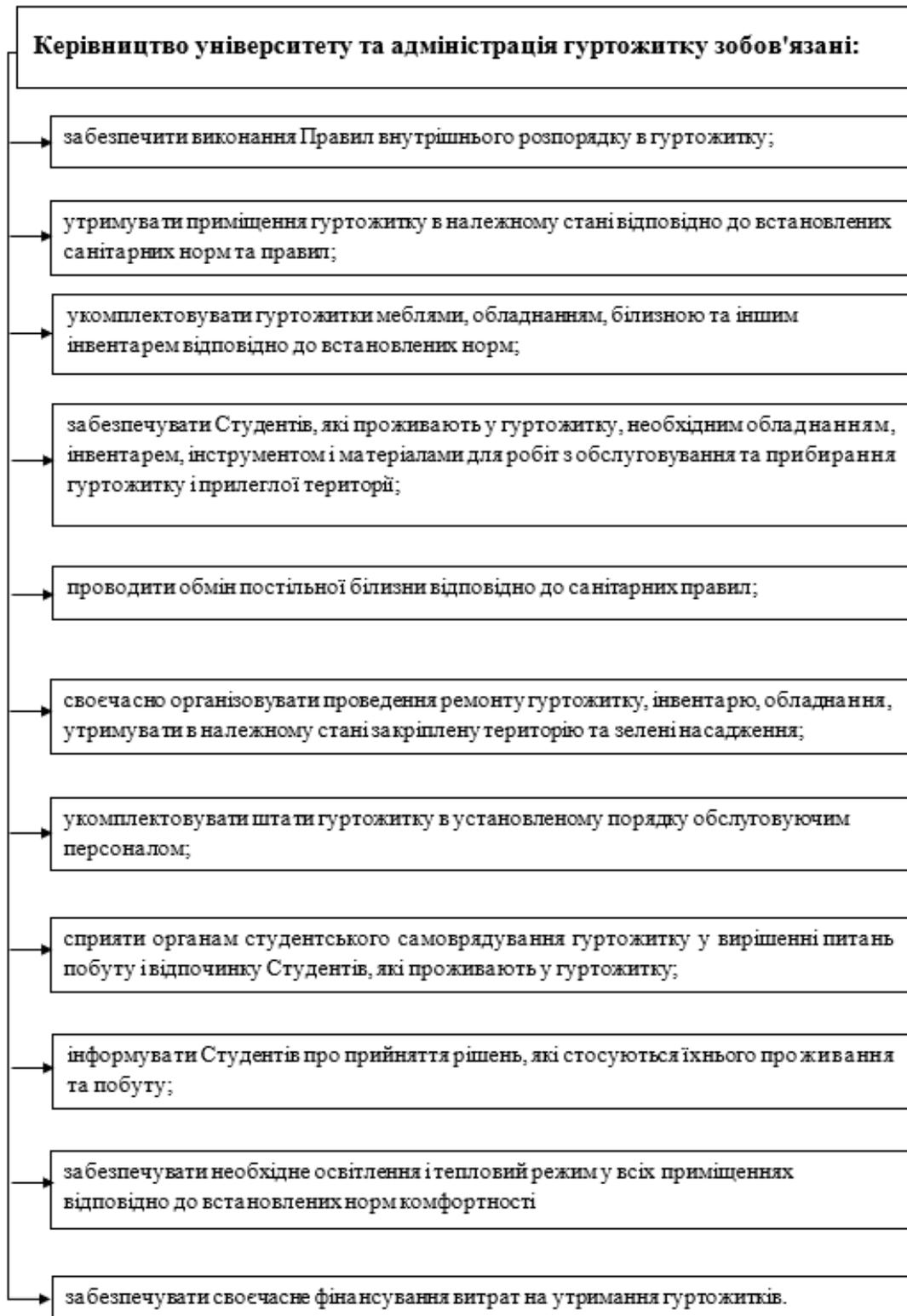


Рис. 1.2. Обов'язки керівництва університету

Відповідно до розпорядження Кабінету Міністрів України від 2020 року «Концепція державної програми відновлення та розбудови мережі гуртожитків для проживання студентів закладів вищої освіти на 2021-2025 роки» [2] за даними Державної служби статистики України, станом на початок 2019/20 навчального року, в країні функціонує 281 заклад вищої освіти, в яких здобуває вищу освіту на різних рівнях понад 1266,1 тис. студентів. Щорічно до 50% студентів потребують місця в гуртожитку. У великих університетських центрах потреба в гуртожитках для студентів найбільша. Чимало студентів не отримують можливості поселитися у гуртожитку і змушені винаймати приватне житло за високими цінами.

На балансі закладів вищої освіти, що перебувають у підпорядкуванні Міністерства освіти і науки, знаходиться 1246 гуртожитків загальною площею 5248,1 тис. м². До 90% цієї категорії гуртожитків потребують капітального або поточного ремонту, близько 10% – знаходяться в аварійному стані. Більшість із гуртожитків побудовані у 1960-70-х роках і не відповідають сучасним стандартам якості житла. Частина гуртожитків знаходиться далеко від навчальних корпусів. Актуальним залишається питання облаштування доступності гуртожитків для осіб з обмеженими фізичними можливостями.

Станом на сьогодні сформувалося три типи гуртожитків, у яких проживають студенти. Найпоширенішими є ті, що можна охарактеризувати як студентське житло «економ класу». Це гуртожитки з коридорною системою планування, душовою кімнатою у підвальному приміщенні (або однією на декілька поверхів), кухнею та вбиральною на поверсі. Студенти в такому гуртожитку проживають здебільшого у переповнених кімнатах (від 4 до 8 осіб). Проживання у такому гуртожитку є стресовим для студентів і не сприяє їх продуктивному навчанню.

Проблеми, які потребують розв'язку відповідно до [2] зображено на рисунку 1.3.

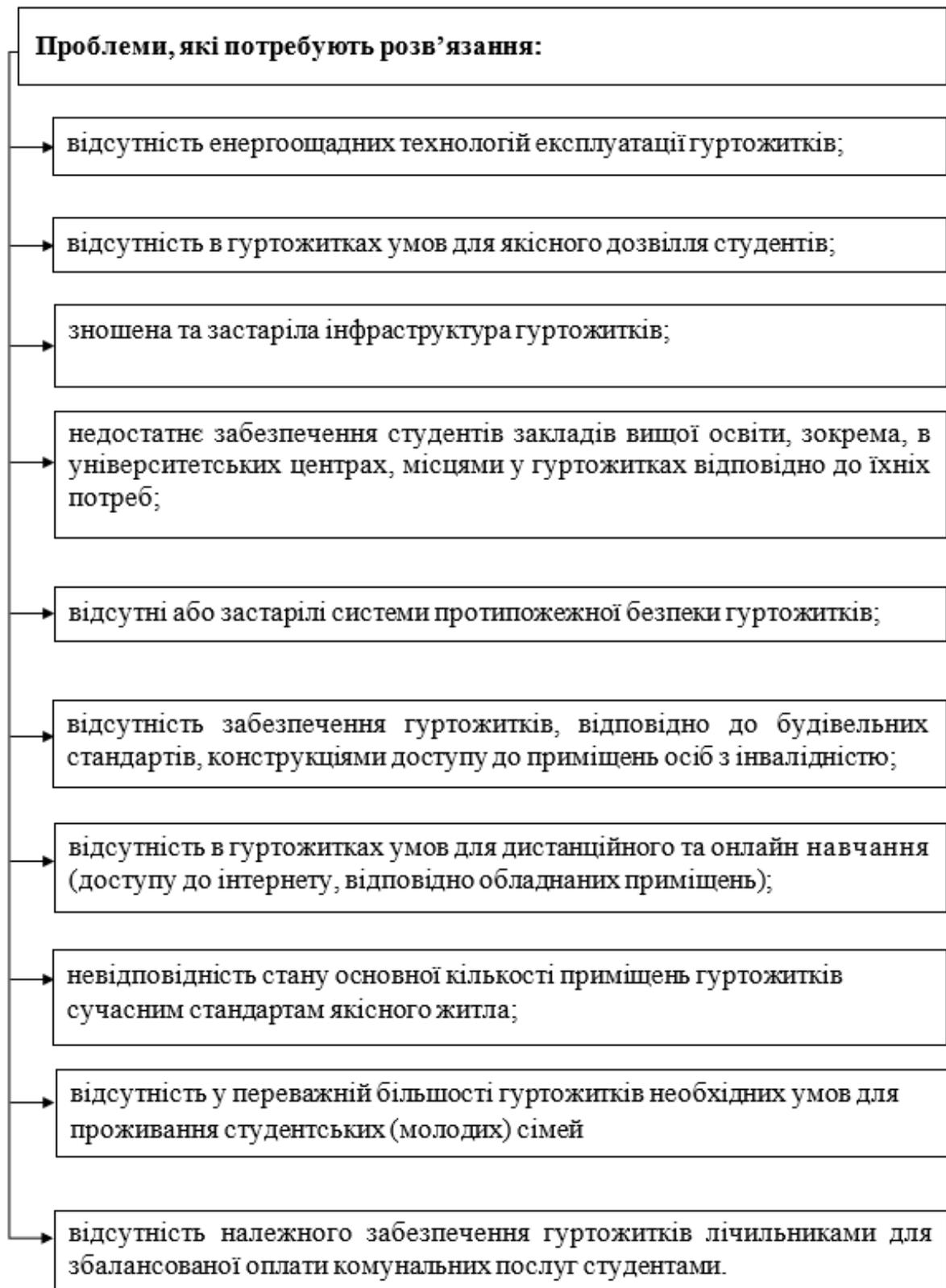


Рис.1.3. Проблеми, які потребують розв'язку

Метою Програми є забезпечення можливостей та належних умов проживання, навчання та дозвілля в гуртожитках для студентів закладів вищої освіти.

Можемо розглядати два основних варіанта розв'язання проблеми.

Перший варіант полягає у збереженні наявних способів управління та підтримки гуртожитків закладами вищої освіти. Позитивним наслідком такого варіанту може бути те, що заклади зберігатимуть певну автономію та поступово напрацюють ефективні способи управління системою власних гуртожитків. Проте, реальною загрозою залишається очікувана в таких умовах стагнація і консервація проблеми, оскільки ремонт, реконструювання та будівництво гуртожитків повністю залежатиме лише від коштів спеціального фонду закладу вищої освіти.

Другий варіант полягає у зміні концептуального підходу до вирішення проблеми студентських гуртожитків. В основі нової концепції має лежати розуміння необхідності забезпечити для студентів нову якість проживання та побуту, яка б підтримувала б у них відчуття самоповаги, відповідальності та безпеки. Виходячи з цього, проблема повинна вирішуватися через масштабні інвестиційні проєкти, які б включали капітальний ремонт та реконструкцію гуртожитків відповідно до сучасних стандартів якості житла. Необхідним уявляється внесення цих проєктів до переліку об'єктів Національного проєкту «Велике будівництво».

Другий варіант розв'язання програми є оптимальним, зокрема, з точки зору його комплексності та системності. Його реалізація передбачає залучення коштів як з державного бюджету, так і місцевих бюджетів, бюджетів закладів вищої освіти, благодійних внесків від приватних осіб та організацій.

В результаті аналізу предметної області був визначений набір інструментів, що сприяють проектуванню і створенню інформаційної системи. Засобом розробки була вибрана мова C#[5]. Вибір ґрунтується на тому, що вона є найбільш підходящою для програмованого додатка. Так само ця мова дозволить найбільш швидко створити потрібну інформаційну систему.

Програмне забезпечення Visual Studio було обрано в якості середовища розробки, так як його інструментарій та різні доповнення дуже спрощують

процес розробки інтерфейсу і написання коду[10]. NET Framework обраний в якості платформи розробки через можливість сприйняття мови C #, великої бібліотеки класів і середовища виконання Windows Forms[3]. Так само були розглянуті різні види баз даних, способи їх створення і варіанти доступу до них. Windows Forms - інтерфейс програмування додатків, відповідальний за графічний інтерфейс користувача.

Програмними продуктами для формування і керування базами даних, які будуть використані додатком, є MySQL-Front та Navicat for MySQL.

1.2. Інформаційна система «БІТ.Гуртожиток» як засіб автоматизації обліку проживання

БІТ.Гуртожиток повнофункціональне рішення для автоматизації обліку в гуртожитку. Програма призначена для ведення обліку руху грошових коштів при наданні послуг проживання гуртожитків університетами.

Конфігурація є самостійною програмою. Для роботи конфігурації необхідна наявність платформи 1С версії не нижче 8.3.6.2041.

Програма вирішує такі задачі:

- гнучке планування індивідуальної структури гуртожитку;
- облік проживаючих в гуртожитку (заїзд, переміщення, виїзд);
- формування договорів з мешканцями;
- облік осіб, додатково які проживають спільно з особою, яка уклала договір з гуртожитком;
- облік наявності ліжко-місць і їх характеристик;
- розрахунок і контроль здійснення оплати за проживання, надання додаткових послуг;
- аналіз динаміки руху грошових коштів при наданні послуг;
- звітність по чисельності проживаючих і додатково проживаючих, взаєморозрахунках і заборгованості проживаючих, наявності вільних ліжко-місць, матеріальних засобів.

ІС «БІТ.Гуртожиток» призначена для керівника гуртожитку та для бухгалтера. Функціональні можливості бухгалтера зображено на рисунку 1.4., функціональні можливості для керівника гуртожитку зображено на рисунку 1.5.

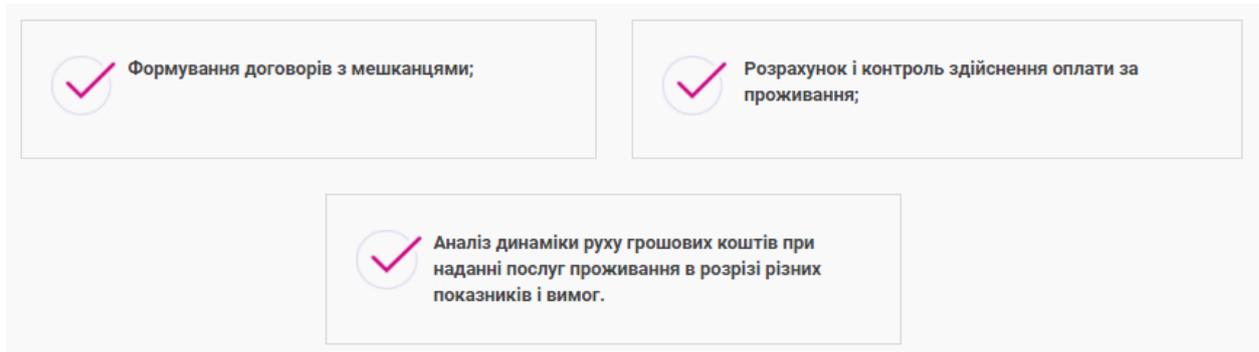


Рис.1.4. Функціональні можливості бухгалтера

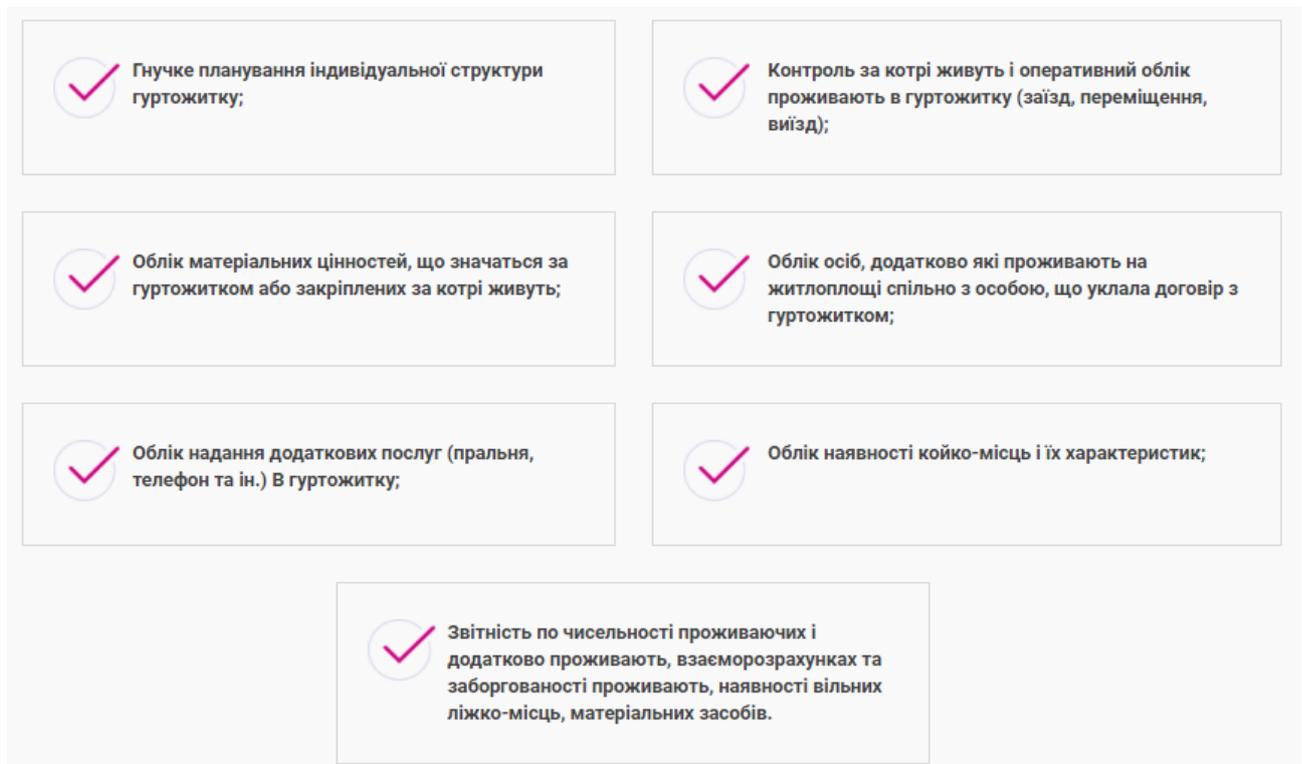


Рис.1.5. Функціональні можливості для керівника гуртожитку

Системні вимоги до комп'ютерів:

Комп'ютер сервера 1С: Підприємство 8:

- операційна система MS Windows 2000/XP/Server 2003 і вище;

- процесор Intel Pentium III 866 МГц і вище (рекомендується Intel Pentium IV/Xeon 2,4 ГГц, для кращої масштабованості рекомендується два і більше процесорів);
- оперативна пам'ять 512 Мбайт і вище (рекомендується 1024 Мбайт);
- USB-порт;
- пристрій читання компакт дисків;
- Сервер баз даних:
- Microsoft SQL Server 2000 + Service Pack 2 (рекомендується Service Pack 3a)

або:

- Microsoft SQL Server 2005

Комп'ютер сервера баз даних:

- операційна система відповідно до вимог Microsoft SQL Server;
- технічні характеристики комп'ютера повинні відповідати вимогам використовуваної версії сервера баз даних Microsoft SQL Server

Комп'ютер кінцевого використання:

- операційна система MS Windows 98/Me, MS Windows NT 4.0/2000/XP/Server 2003 (рекомендується MS Windows 2000/XP/Server 2003);
- процесор Intel Pentium II 400 МГц і вище (рекомендується Intel Pentium III 866 МГц);
- оперативна пам'ять 128 Мбайт і вище (рекомендується 256 Мбайт);
- жорсткий диск (при установці використовується близько 120 Мбайт);
- пристрій читання компакт дисків;
- USB-порт;
- SVGA дисплей;

Комп'ютер, який використовується для розробки конфігурацій:

- операційна система MS Windows 2000/XP/Server 2003;
- процесор Intel Pentium III 866 МГц і вище (рекомендується Intel Pentium IV/Celeron 1800 МГц);

- оперативна пам'ять 256 Мбайт і вище (рекомендується 512 Мбайт);
- жорсткий диск (при установці використовується близько 120 Мбайт);
- пристрій читання компакт дисків;
- USB-порт;
- SVGA дисплей;

Вартість програми:

- БІТ.Гуртожиток. Установка на 5 робочих місць – 545\$;
- БІТ.Клієнтська ліцензія на 1 робоче місце - 68\$;
- БІТ.Клієнтська ліцензія на 5 робочих місць - 235\$;
- БІТ.Клієнтська ліцензія на 10 робочих місць - 450\$;
- БІТ.Клієнтська ліцензія на 20 робочих місць - 851\$;
- БІТ.Клієнтська ліцензія на 50 робочих місць - 2045\$;
- БІТ.Клієнтська ліцензія на 100 робочих місць - 3925\$;

Отже, ІС «БІТ.Гуртожиток» має як переваги, так і недоліки. Серед переваг: наявність обліку студентів, які проживають у гуртожитку, облік місць для проживання та їх характеристика, інформація про оплату за проживання, звітність по чисельність проживаючих та заборгованості. Голові недоліки – вартість ІС та неможливість робити без платформи ІС.

1.3. Сучасні тренди популярності мов програмування

Популярність мов програмування змінюється з кожним роком все швидше. DOU.UA - український ресурс провів чергове опитування про мови програмування, метою якого було дізнатися найпопулярніші у даний час мови програмування та сфери їх використання, тому було зібрано 7211 анкет, 92% людей, які пройшли анкетування знаходяться в Україні[9].

Результати опитування, якою мовою пишете для роботи зараз зображено на рисунку 1.6.

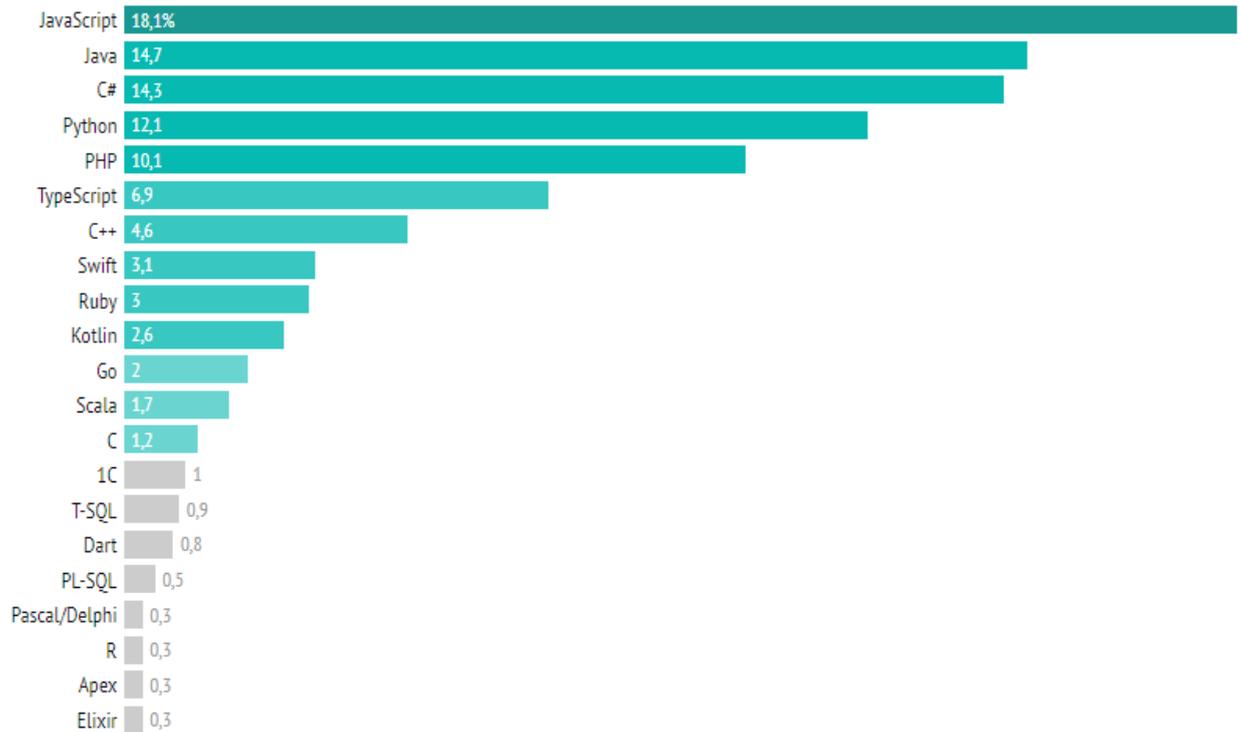


Рис.1.6. Відсоткове відношення використання мов для роботи

За результатами топ-5 мов програмування: JavaScript, Java, C#, Python, PHP при цьому частка використання Java та C# майже однакова.

Причинами популярності JavaScript є те, що ця мова призначена для початківців, середніх і просунутих розробників, вона не вимагає налаштувань середовища.

JavaScript забезпечує інтерактивність сайтів і використовується в популярних сучасних браузерах. Ці завдання можна вирішувати і за допомогою альтернативних технологій, наприклад, WebAssembly, але це не питання короткострокової перспективи. У найближчому майбутньому з великою часткою ймовірності JS залишиться основною мовою програмування фронтенд-додатків. Для розробки прогресивних веб-додатків (PWA, progressive web apps) застосовується JavaScript[8]. Наприклад, сайти, створені за допомогою генератора Gatsby.js, підтримують функціональність прогресивного веб-додатки з коробки. Це ще один плюс на користь затребуваності JS в найближчій перспективі.

Java об'єктно-орієнтована мова зі строгою типізацією. Може застосовуватися на будь-яких платформах за допомогою віртуальної Java-машини. Досить універсальний. З його допомогою можна писати десктопні програми, ігри, мобільні додатки. Зараз Java користується попитом серед android-розробників, для створення бізнес-додатків.

Популярність C# пов'язані з розширенням функціоналу платформи і переходом від Windows орієнтованої .NET Framework до крос-платформної .NET.Core. Тепер мова цікава не тільки Windows-розробникам, а й фахівцям з Linux, Mac OS, Android, iOS[11].

C# вважається порівняно легкою у вивченні, зрозумілою і відповідною для новачків. За синтаксису він схожий на Java і C ++. Попит на програмістів C# стабільно високий, мова застосовується все ширше.

Python має дуже широку сферу застосування: програми для роботів, безпілотників, додатки для ПК і мобільних пристроїв, веб-розробка та аналітика[7]. Python - лідер багатьох рейтингів популярності (в тому числі PYPL, IEEE Spectrum).

Python вже використовується деякими з найбільших імен в технології, поряд з деякими менш імовірними, але однаково вражаючими користувачами. Uber, PayPal, Google, Facebook, Instagram, Netflix, Dropbox і Reddit використовують Python при розробці та тестуванні. Крім того, Python також широко використовується в робототехніці і вбудованих системах (його навіть можна використовувати для управління Arduinos).

Мова набирає обертів завдяки своїй продуктивності, простоті освоєння, доступності на різних ОС і великих обсягів наявних напрацювань. Синтаксис досить незвичайний, але дуже раціональний, що не ускладнений громіздкими конструкціями. Популярність мови сьогодні і в майбутньому сумнівів не викликає.

Далі розглянемо , які мови використовують залежно від сфери програмування.



Рис.1.7. Відсоткове відношення використання мов для Mobile

Swift, Kotlin – найпоширеніші мови для Mobile розробки, частка використання мов сімейства C значно менша.

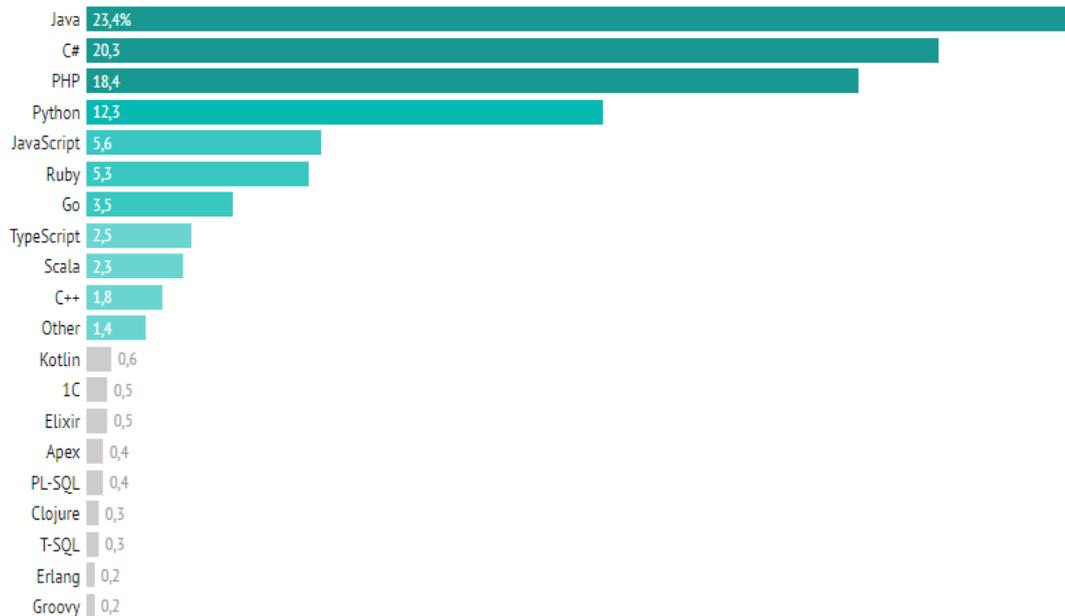


Рис.1.8. Відсоткове відношення використання мов для Back-end

Для Back-end розробки основні мови – Java, C#, PHP, меншу частку мають – Python, JavaScript, Ruby.

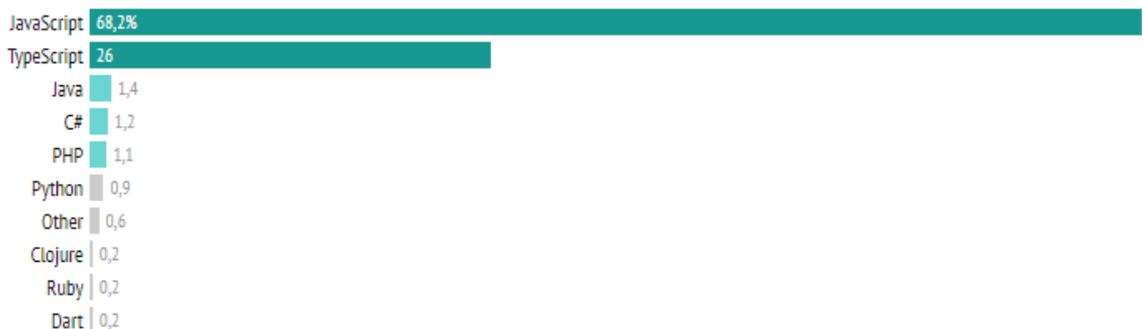


Рис.1.9. Відсоткове відношення використання мов для Front-end

Практично весь Frond-end пишеться на JavaScript та Type JavaScript.



Рис.1.10. Відсоткове відношення використання мов для Data processing

Для Data processing лідером є Python, частка T-SQL, Scala також є значною.

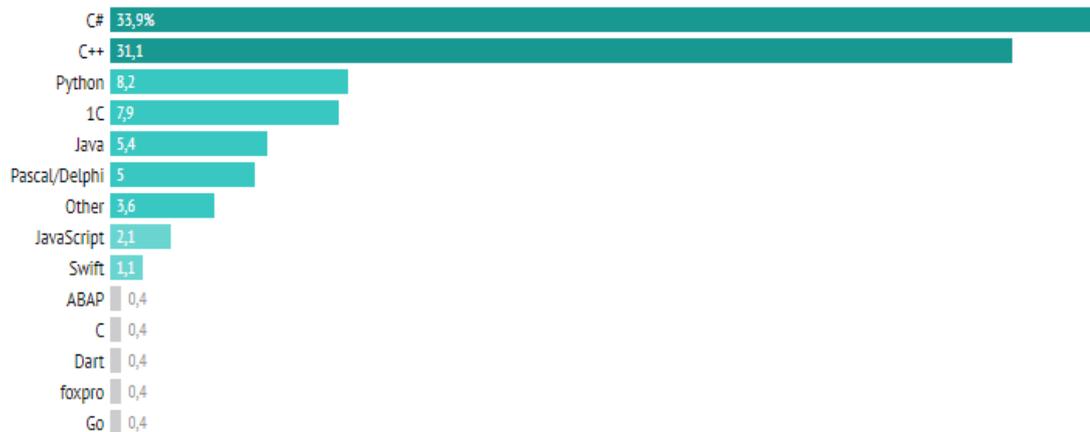


Рис.1.11. Відсоткове відношення використання мов для Desktop

У Desktop розробці основні мови – C# та C. Також часто використовуваними є Python та 1C.

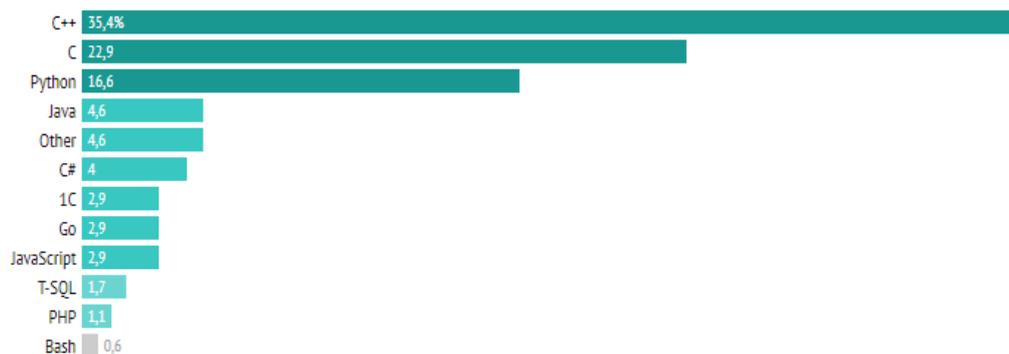


Рис.1.12. Відсоткове відношення використання мов для System

Для System розробки основні мови – C++, C, Python.

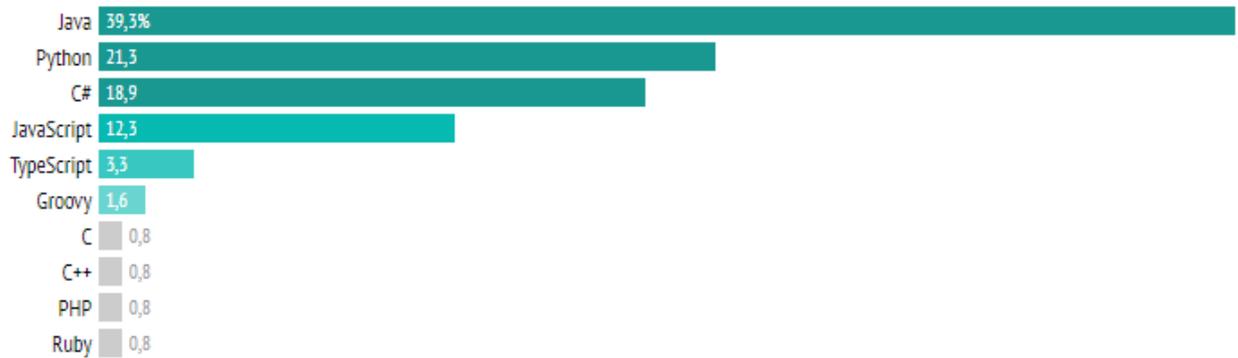


Рис.1.13. Відсоткове відношення використання мов для QA automation

Найпоширеніші мови для QA automation – Java, Python, C#. JavaScript та TypeScript також використовуються, проте частка JavaScript значно більша.

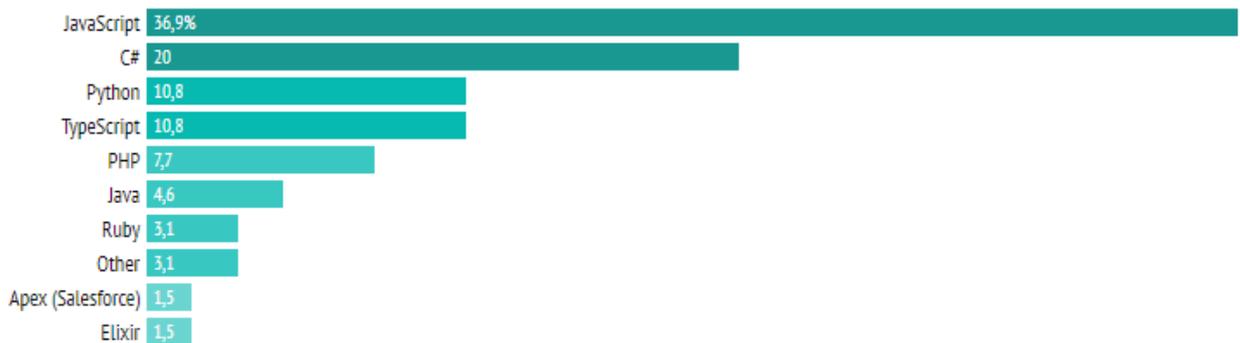


Рис.1.14. Відсоткове відношення використання мов для Full Stack

JavaScript, C#, Python, TypeScript – основні мови для Full Stack розробки, при цьому частка використання Python, TypeScript однакова.

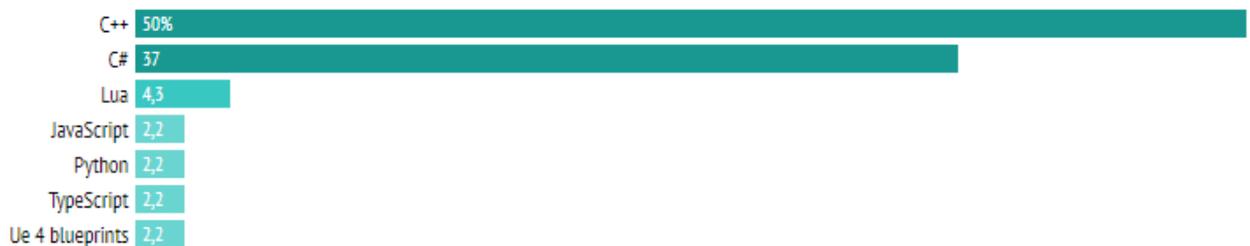


Рис.1.15. Відсоткове відношення використання мов для GameDev

Майже уся сфера GameDev пишеться на C++ та C#, значно меншу частку має Lua.

В загальному можна відмітити, що спостерігається зростання використання TypeScript, цього року він уперше обійшов за популярністю

C++. Причиною переходу користувачів з JavaScript на TypeScript є те, що JavaScript - це мова сценаріїв, а TypeScript - це об'єктно-орієнтована мова програмування. TypeScript підтримує модулі, а JavaScript - ні. TypeScript підтримує статичну типізацію, яка дозволяє вам перевіряти правильність типу під час компіляції, тоді як JavaScript не підтримує її.

Також уперше за декілька років знижується частка Python, а от частка C# вперше з 2012 року має позитивну динаміку. Основними причинами зростання популярності C# є: розширення функціоналу платформи і переходом від Windows орієнтованої .NET Framework до крос-платформної .NET.Core. Тепер мова цікава не тільки Windows-розробникам, а й фахівцям з Linux, Mac OS, Android, iOS.

Висновки по розділу 1

Проаналізувавши предметну область, був визначений набір інструментів, що сприяють проектування і створення інформаційної системи. Засобом розробки була вибрана мова C# тому, що вона найбільш підходить для розробки додатка. Дана мова дозволить найбільш швидко створити потрібну інформаційну систему. Програмне забезпечення Visual Studio було обрано в якості середовища розробки, так як його інструментарій та різні доповнення дуже спрощують процес розробки інтерфейсу і написання коду. NET Framework обраний в якості платформи розробки через можливість прийняття мови C#, великої бібліотеки класів і середовища виконання Windows Forms. Програмними продуктами для формування і керування базами даних є MySQL-Front та Navicat for MySQL.

Розглянувши ІС «БІТ.Гуртожиток», як засіб автоматизації обліку проживання, можемо зазначити основні переваги та недоліки ІС і відштовхуватись від них про проектуванні та розробці ІС «Гуртожитків НУВГП». Серед переваг: наявність обліку студентів, які проживають у гуртожитку, облік місць для проживання та їх характеристика, інформація про оплату за проживання, звітність по чисельності проживаючих та

заборгованості. Голові недоліки – вартість ІС та неможливість робити без платформи ІС.

Оглянувши сучасні тренди популярності мов програмування, можна відмітити, що спостерігається зростання використання TypeScript, цього року він уперше обійшов за популярністю C++. Також уперше за декілька років знижується частка Python, а от частка C# вперше з 2012 року має позитивну динаміку, найбільша частина desktop застосунків пишеться на C#. Саме C# була обрана для розробки інформаційної системи гуртожитків НУВГП.

РОЗДІЛ 2. ІНФОРМАЦІЙНА СИСТЕМА ЯК ЗАСІБ ПІДТРИМКИ ФУНКЦІОНУВАННЯ СТУДЕНТСЬКИХ ГУРТОЖИТКІВ

2.1. Логічна модель даних системи

Логічна модель даних, або логічна схема – це модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних), але в термінах структур даних, таких як реляційні таблиці та колонки, об'єктно-орієнтовані класи чи теги XML[4].

Метою побудови логічної моделі є отримання графічного представлення логічної структури досліджуваної предметної області.

Логічна модель предметної області ілюструє сутності, а також їх взаємовідносини між собою. Сутність описують об'єкти, які є предметом діяльності предметної області, і суб'єкти, що здійснюють діяльність в рамках предметної області. Властивості об'єктів і суб'єктів реального світу описуються за допомогою атрибутів. Взаємини між сутностями ілюструються за допомогою зв'язків. Правила та обмеження взаємин описуються за допомогою властивостей зв'язків. Зазвичай зв'язки визначають, або залежність між сутностями, або впливом однієї сутності на іншу.

Логічна модель є основою бази даних, вона повинна відображати взаємозв'язки між реляційними таблицями. Між реляційними таблицями можуть бути наступні типи зв'язків 1:1, 1:Б та Б:Б. Найбільшим поширеним зв'язком є зв'язок 1:Б.

Для створення зв'язку необхідно, щоб у головній таблиці були визначені первинні ключі. Встановлення первинного ключа для зв'язаної (підпорядкованої) таблиці не є обов'язковою умовою. Для підпорядкованої таблиці необхідно визначити поле вторинного ключа, тип даних і розміри якого повинні збігатись з полем первинного ключа головної таблиці. Імена полів первинного та вторинного ключів, між якими встановлюється зв'язок

можуть не збігатися. Вторинні ключі відрізняються від первинних тим, що для них допускається дублювання значень.

Логічну модель даних ІС «Гуртожитки НУВГП» зображено на рисунку 2.1.

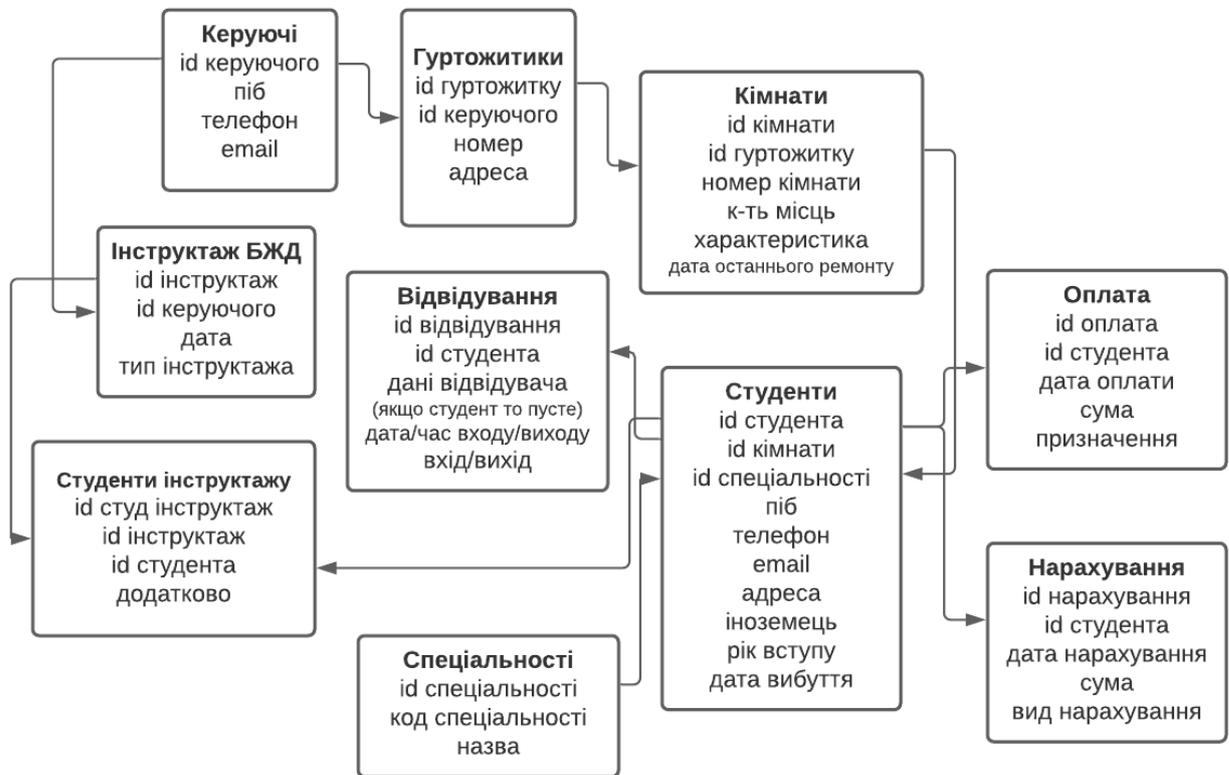


Рис.2.1. Логічна модель даних ІС «Гуртожитки НУВГП»

Для створення бази даних використовуємо додаток MySQL-Front. У БД інформаційної системи гуртожитків налічується десять сутностей, які зображені на рисунку 2.2.

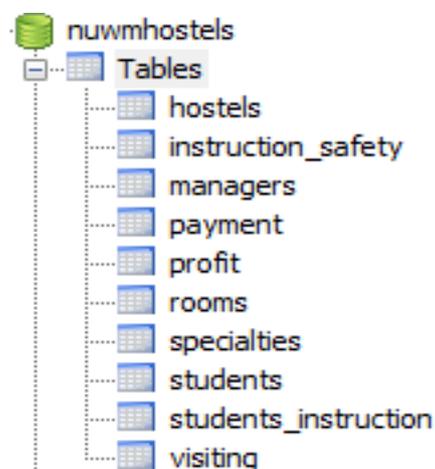


Рис.2.2. Таблиці бази даних ІС «Гуртожитки НУВГП»

Таблиця «Гуртожитки НУВГП», яка зображена на рисунку 2.3, створена за допомогою наступного SQL запиту:

```
CREATE TABLE `hostels` (
  `Id_hostel` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_manager` int(11) unsigned DEFAULT NULL,
  `number` int(2) unsigned NOT NULL DEFAULT 0,
  `adress` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Id_hostel`),
  KEY `managers` (`Id_manager`),
  CONSTRAINT `managers` FOREIGN KEY (`Id_manager`) REFERENCES
  `managers` (`Id_manager`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

Id_hostel	Id_manager	number	adress
1		1	1 вул. Чорновола, 51
2		2	2 вул. Чорновола, 55
3		3	3 вул. Мірющенко, 62
4		4	4 вул. Мірющенко, 60
5		5	5 вул. Волинська, 24
6		6	6 вул. Чорновола, 53

Рис.2.3. Таблиця «Гуртожитки НУВГП»

Наведена таблиця створена для вводу інформації про усі гуртожитки, а саме id керуючого гуртожитком, номер та адресу гуртожитку.

Таблиця «Інструктажі БЖД», яка зображена на рисунку 2.4, створена за допомогою наступного SQL запиту:

```
CREATE TABLE `instruction_safety` (
  `Id_instruction` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_manager` int(11) unsigned DEFAULT NULL,
  `date` date DEFAULT NULL,
  `type_instruction` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`Id_instruction`),
  KEY `managers` (`Id_manager`),
  CONSTRAINT `managers` FOREIGN KEY (`Id_manager`) REFERENCES
  `managers` (`Id_manager`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

Id_instruction	Id_manager	date	type_instruction
1		2020-08-30	первинний
2		2020-08-29	первинний
4		2020-08-30	первинний
5		2020-08-30	первинний
6		2020-09-01	первинний
7	1	2021-02-28	повторний
8	2	2021-02-27	повторний
9	3	2021-03-01	повторний
10	4	2021-03-05	повторний
11	5	2021-02-25	повторний

Рис.2.4. Таблиця «Інструктажі БЖД»

Наведена таблиця створена для вводу інформації про усі інструктажі БЖД, а саме id керуючого гуртожитком, який провів інструктаж, дата проведення та тип інструктажу.

Таблиця «Інструктажі БЖД», яка зображена на рисунку 2.5, створена за допомогою наступного SQL запити:

```
CREATE TABLE `managers` (
  `id_manager` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `pib` varchar(100) DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `e_mail` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_manager`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

Id_manager	pib	phone	e_mail
1	Цімох Марія Карлівна	0978866502	karlivna.manager@nuwm.edu.ua
2	Шарабура Людмила Миколаївна	0978866740	sharabura.manager@nuwm.edu.ua
3	Черній Світлана Іванівна	0738086674	chernii.manager@nuwm.edu.ua
4	Попельницька Галина Миколаївна	0678563241	popelnytska.manager@nuwm.edu.ua
5	Якимів Ярослава Степанівна	0671563248	yakymiv.manager@nuwm.edu.ua
6	Чемакіна Ольга Василівна	0978563214	chemakina.manager@nuwm.edu.ua

Рис.2.5. Таблиця «Керуючі»

Наведена таблиця створена для вводу інформації про усіх керуючих гуртожитками , а саме ПІБ керуючого, номер телефону та e-mail.

Таблиця «Оплата», яка зображена на рисунку 2.6, створена за допомогою наступного SQL запити:

```
CREATE TABLE `payment` (
  `id_payment` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_student` int(11) unsigned DEFAULT NULL,
  `date_payment` date DEFAULT NULL,
  `amount` float(6,0) DEFAULT NULL,
  `appointment` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id_payment`),
  KEY `payment` (`id_student`),
  CONSTRAINT `payment` FOREIGN KEY (`id_student`) REFERENCES `students` (`id_student`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
```

id_payment	id_student	date_payment	amount	appointment
1		2020-08-31	7080	оплата за рік
2		2020-08-30	3540	оплата за півроку
3		2020-08-30	590	оплата за вересень
4		2020-08-31	3540	оплата за півроку
5		2020-08-25	7080	оплата за рік
6		2020-08-27	590	оплата за вересень
7		2020-09-30	590	оплата за жовтень
8		2020-09-27	590	оплата за жовтень
9		2020-09-30	200	найнове відшкодування

Рис.2.6. Таблиця «Оплата»

Наведена таблиця створена для вводу інформації про здійснену оплату студентів, а саме id студента, який зробив оплату, дата, сума та призначення оплати.

Таблиця «Нарахування», яка зображена на рисунку 2.7, створена за допомогою наступного SQL запити:

```
CREATE TABLE `profit` (
  `id_profit` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_student` int(11) unsigned DEFAULT NULL,
  `date_profit` date DEFAULT NULL,
  `amount` float(6,0) DEFAULT NULL,
  `type_profit` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_profit`),
  KEY `profit` (`id_student`),
  CONSTRAINT `profit` FOREIGN KEY (`id_student`) REFERENCES `students` (`id_student`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

Id_profit	Id_student	date_profit	amount	type_profit
2	2	2020-08-30	3540	проживання
3	3	2020-08-30	590	проживання
4	4	2020-08-31	3540	проживання
5	1	2021-05-16	400	відшкодування
7	1	2020-08-30	7500	проживання

Рис.2.7. Таблиця «Нарахування»

Наведена таблиця створена для вводу інформації про нарахування студента, які він має оплатити, а саме id студента, кому було створене нарахування, дата, сума та тип нарахування.

Таблиця «Кімнати», яка зображена на рисунку 2.8, створена за допомогою наступного SQL запити:

```
CREATE TABLE `rooms` (
  `Id_room` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_hostel` int(11) unsigned DEFAULT NULL,
  `number_room` varchar(10) DEFAULT NULL,
  `number_seats` int(11) unsigned DEFAULT NULL,
  `characteristic` varchar(255) DEFAULT NULL,
  `date_last_repair` date DEFAULT NULL,
  PRIMARY KEY (`Id_room`),
  KEY `hostels` (`Id_hostel`),
  CONSTRAINT `hostels` FOREIGN KEY (`Id_hostel`) REFERENCES `hostels`
  (`Id_hostel`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=29 DEFAULT CHARSET=utf8;
```

Id_room	Id_hostel	number_room	number_seats	characteristic	date_last_repair
1		1 1101	3	3 ліжка, 2 столи, 3 стільця	2020-08-29
2		1 1102	3	3 ліжка, 1 стіл, 3 стільця	2019-08-30
3		1 1103	3	3 ліжка, 1 стіл, 3 стільця,поличка	2017-08-17
4		1 1201	3	3 ліжка, 2 столи, 3 стільця	2020-08-29
5		1 1202	3	3 ліжка, 1 стіл, 3 стільця	2019-08-27
6		1 1203	3	3 ліжка, 1 стіл, 3 стільця,поличка	2018-08-14
7		2 2101	3	3 ліжка, 1 стіл, 3 стільця	2018-09-02
8		2 2102	3	3 ліжка, 1 стіл, 3 стільця,поличка	2020-08-01
9		2 2103	3	3 ліжка, 2 столи, 3 стільця	2019-09-02
10		2 2201	3	3 ліжка, 2 столи, 3 стільця	2019-08-27
11		2 2202	3	3 ліжка, 1 стіл, 3 стільця	2020-08-17
12		2 2203	3	3 ліжка, 1 стіл, 3 стільця,поличка	2019-08-04
13		3 3101	3	3 ліжка, 1 стіл, 3 стільця	2019-08-28
14		3 3102	3	3 ліжка, 1 стіл, 3 стільця	2020-08-28
15		3 3201	3	3 ліжка, 1 стіл, 3 стільця,поличка	2019-08-01
16		3 3202	3	3 ліжка, 2 столи, 3 стільця	2018-09-03
17		4 4101	3	3 ліжка, 2 столи, 3 стільця	2017-09-01

Рис.2.8. Таблиця «Кімнати»

Наведена таблиця створена для вводу інформації про усі кімнати гуртожитку, а саме id гуртожитку, якому належить певна кімната, номер, кількість місць, характеристика та дата останнього ремонту кімнати.

Таблиця «Спеціальності», яка зображена на рисунку 2.9, створена за допомогою наступного SQL запити:

```
CREATE TABLE `specialties` (
  `Id_specialty` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `code_specialty` varchar(50) DEFAULT NULL,
  `name` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Id_specialty`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8;
```

Id_specialty	code_specialty	name
1	051	Економіка
2	061	Журналістика
3	073	Менеджмент
4	075	Маркетинг
5	081	Права
6	101	Екологія
7	113	Прикладна математика
8	121	Інженерія програмного забезпечення
9	122	Комп'ютерні науки
10	123	Комп'ютерна інженерія
11	126	Інформаційні системи та технології
12	133	Галузеве машинобудування
13	141	Електроенергетика, електротехніка та електромеханіка
14	144	Теплоенергетика
15	151	Автоматизація та комп'ютерно-інтегровані технології
16	171	Електроніка
17	184	Грицтво

Рис.2.9. Таблиця «Спеціальності»

Наведена таблиця створена для вводу інформації про спеціальності, на яких навчаються студенти, що проживають у гуртожитках, а саме код та назва спеціальності.

Таблиця «Студенти», яка зображена на рисунку 2.10, створена за допомогою наступного SQL запити:

```
CREATE TABLE `students` (
  `Id_student` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_room` int(11) unsigned DEFAULT NULL,
  `Id_specialty` int(11) unsigned DEFAULT NULL,
  `pib` varchar(255) DEFAULT NULL,
```

```

`phone` varchar(50) DEFAULT NULL,
`e_mail` varchar(100) DEFAULT NULL,
`adress` varchar(255) DEFAULT NULL,
`foreigner` tinyint(1) DEFAULT NULL,
`year_entry` varchar(4) DEFAULT NULL,
`date_leave` date DEFAULT NULL,
PRIMARY KEY (`Id_student`),
KEY `specialty` (`Id_specialty`),
KEY `rooms` (`Id_room`),
CONSTRAINT `rooms` FOREIGN KEY (`Id_room`) REFERENCES `rooms`
(`Id_room`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `specialty` FOREIGN KEY (`Id_specialty`) REFERENCES
`specialties` (`Id_specialty`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;

```

Id_student	Id_room	Id_specialty	pib	phone	e_mail	adress	foreigner	year_entry	date_leave
1	1	14	Мищенко Петро Дмитрович	0654789214	mischenko_bg17.nuwm.edu.ua	м. Нетішин вул. Соборна 31	0	2017	2021-06-30
2	1	14	Оніщук Олександр Леонідович	0974789214	onischyk_bg17.nuwm.edu.ua	м. Славути вул. Київська 2	0	2017	2021-06-30
3	1	14	Петренко Леонід Володимирович	0970789211	petrenko_bg17.nuwm.edu.ua	с. Білка вул. Спортивна 89	0	2017	2021-06-30
4	2	14	Дерун Олена Миколаївна	0678541258	deryn_bg20.nuwm.edu.ua	м. Здолбунів вул. Миру 45	0	2020	2024-06-30
5	2	14	Зній Лариса Василівна	0678541295	znyi_bg19.nuwm.edu.ua	м. Здолбунів вул. Дерев'яна 31	0	2019	2023-06-30
6	7	1	Тагаєв Тагай Тигранович	0987452143	tagaev_em18.nuwm.edu.ua	м. Стамбул	1	2018	2022-06-30
7	7	1	Саргсян Михаель Мухамедович	0963257400	sargsyan_em19.nuwm.edu.ua	м. Стамбул	1	2017	2021-06-30
8	8	3	Кривко Ірина Федорівна	0678514589	krivko_em19.nuwm.edu.ua	с. Нетреба вул. Перемоги 4	0	2019	2023-06-30

Рис.2.10. Таблиця «Студенти»

Наведена таблиця створена для вводу інформації про студентів, які проживають у гуртожитках, а саме id кімнати студента, ПІБ, номер телефону, e-mail, адреса, іноземець(1/ 0 -ні), рік вступу та дата вибуття студента.

Таблиця «Студенти інструктажу», яка зображена на рисунку 2.11, створена за допомогою наступного SQL запити:

```

CREATE TABLE `students_instruction` (
  `Id_st_instruction` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_instruction` int(11) unsigned DEFAULT NULL,
  `Id_student` int(11) unsigned DEFAULT NULL,
  `additionally` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`Id_st_instruction`),
  KEY `instruction` (`Id_instruction`),
  KEY `sruents` (`Id_student`),
  CONSTRAINT `instruction` FOREIGN KEY (`Id_instruction`) REFERENCES
`instruction_safety` (`Id_instruction`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `sruents` FOREIGN KEY (`Id_student`) REFERENCES
`students` (`Id_student`) ON DELETE CASCADE ON UPDATE CASCADE

```

) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8;

	Id_st_instruction	Id_instruction	Id_student	additionally
▶	1	1	1	1 проінструктований
	2	1	1	2 проінструктований
	3	1	1	3 проінструктований
	4	1	1	4 проінструктований
	5	1	1	5 проінструктований
	6	2	2	6 проінструктований
	7	2	2	7 проінструктований
	9	14	2	2 проінструктований
	10	14	9	9 проінструктований
	11	14	5	5 проінструктований

Рис.2.11. Таблиця «Студенти інструктажу»

Наведена таблиця створена для вводу інформації про студентів, які пройшли інструктаж, а саме, id інструктажу, id студента та поле додатково.

Таблиця «Відвідування», яка зображена на рисунку 2.12, створена за допомогою наступного SQL запити:

```
CREATE TABLE `visiting` (
  `Id_visiting` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `Id_student` int(11) unsigned DEFAULT NULL,
  `data_visitor` varchar(100) DEFAULT NULL,
  `time_e` datetime DEFAULT NULL,
  `exit_entrance` tinyint(1) DEFAULT 0,
  PRIMARY KEY (`Id_visiting`),
  KEY `students` (`Id_student`),
  CONSTRAINT `students` FOREIGN KEY (`Id_student`) REFERENCES
  `students` (`Id_student`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8;
```

	Id_visiting	Id_student	data_visitor	time_e	exit_entrance
▶	1		1 (Null)	2021-02-01 10:20:56	0
	2		2 (Null)	2021-02-01 09:00:56	0
	3		(Null) Електрик Олександр	2021-02-01 08:00:00	0
	4		3 (Null)	2021-02-01 16:00:00	0
	5		4 (Null)	2021-03-04 13:24:13	0
	6		(Null) Сантехнік Петро	2021-04-08 18:26:15	0
	7		1 (Null)	2021-04-14 13:26:50	0
	8		7 (Null)	2021-04-15 13:57:29	0
	10		1 (Null)	2021-02-02 04:20:56	1
	11		2 (Null)	2021-02-02 04:20:56	1
	12		4 (Null)	2021-02-02 01:20:56	0
	13		3	2021-05-19 11:53:00	1
	20		3	2021-05-19 12:04:00	1

Рис.2.12. Таблиця «Відвідування»

Наведена таблиця створена для вводу інформації про відвідування студентів та сторонніх осіб, а саме id студента, дані відвідувача, якщо не студент, дата/час входу/виходу та поле вхід(1)/вихід(0).

Логічну модель даних та схему даних розробленої ІС зображено на рисунку 2.13.

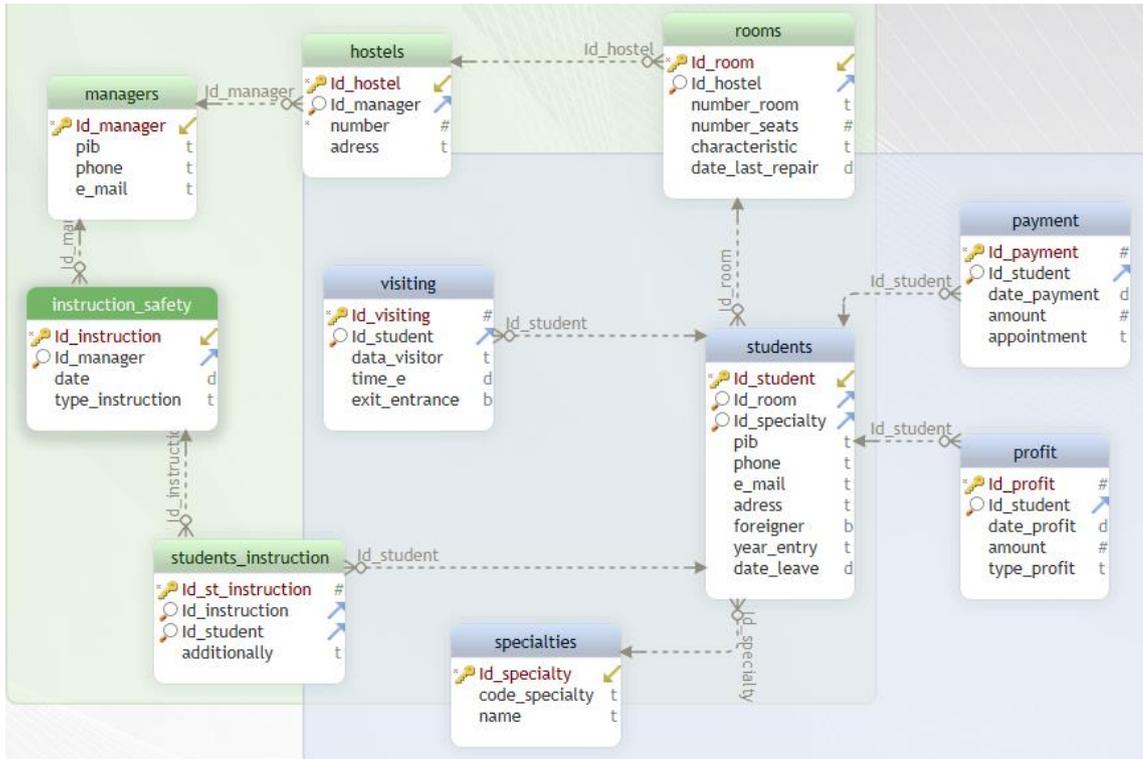


Рис.2.13. Схема даних ІС «Гуртожитки НУВГП»

Отже, під час моделювання бази даних інформаційної системи «Гуртожитків НУВГП» було створено десять сутностей, зокрема: Студенти, Відвідування, Спеціальності, Оплата, Нарахування, Гуртожитки, Кімнати, Керуючі, Інструктажі БЖД, Студенти Інструктажів, з'єднаних між собою зовнішніми ключами.

2.2. Функціональні можливості

Розроблена інформаційна система «Гуртожитки НУВГП», складається з пунктів головного меню, зображених на рисунку 2.14.

Файл Довідники Відвідування Інструктажі Звіти Довідка

Рис.2.14. Пункти головного меню ІС «Гуртожитки НУВГП»

Пункт головного меню *Довідники* складається з підпунктів зображених на рисунку 2.15.

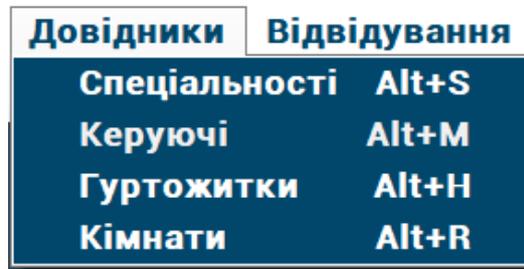


Рис.2.15. Підпункти пункту *Довідники*

Наведений пункт забезпечує доступ до інформації про спеціальності студентів, керуючих гуртожитками, гуртожитки та кімнати гуртожитків.

Натискаючи на підпункт *Спеціальності*, відкривається форма зображена на рисунку 2.16.



Код	Код спеціальності	Назва спеціальності
1	051	Економіка
2	061	Журналістика
3	073	Менеджмент
4	075	Маркетинг
5	081	Права
6	101	Екологія
7	113	Прикладна математика
8	121	Інженерія програмного забезпечення
9	122	Комп'ютерні науки
10	123	Комп'ютерна інженерія
11	126	Інформаційні системи та технології
12	133	Галузеве машинобудування

Код:

Код спеціальності:

Назва спеціальності:

Зберегти **Відміна**

Рис.2.16. Довідник «Спеціальності»

Наведена форма містить інформацію про спеціальності студентів, які проживають у гуртожитках, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля.

Також передбачена можливість редагування та видалення даних за допомогою контекстного меню.

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видалити дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.17.

Код	Код спеціальності	Назва спеціальності
1	051	Економіка
2	061	Журналістика
3	073	Менеджмент
4	075	Маркетинг
5	081	Права
6	101	Екологія
7	113	Прикладна математика
8	121	Інженерія програмного забезпечення
9	122	Комп'ютерні науки
10	123	Комп'ютерна інженерія
11	126	Інформаційні системи та технології
12	133	Галузеве машинобудування

Код:

Код спеціальності:

Назва спеціальності:

Рис.2.17. Режим редагування Довідника «Спеціальності»

Як зображено за рисунку 2.17. поля запису вставляються у відповідні поля форми, після чого керуючий гуртожитком може їх змінювати, а натиснувши на кнопку *Зберегти*, змінені дані зберігаються.

Натискаючи на підпункт головного меню *Керуючі*, відкривається форма зображена на рисунку 2.18.

Код керуючого	ПІБ керуючого	Номер телефону	Електронна пошта
1	Цімох Марія Карлівна	0978866502	karlivna.manager@nuwm.edu.ua
2	Шарабура Людмила Миколаївна	0978866740	sharabura.manager@nuwm.edu.ua
3	Черній Світлана Іванівна	0738086674	chernii.manager@nuwm.edu.ua
4	Попельницька Галина Миколаївна	0678563241	popelnytska.manager@nuwm.edu.ua
5	Якимів Ярослава Степанівна	0671563248	yakymiv.manager@nuwm.edu.ua
6	Чемакіна Ольга Василівна	0978563214	chemakina.manager@nuwm.edu.ua

Код керуючого:

ПІБ керуючого:

Номер телефону:

Електронна пошта:

Зберегти **Відміна**

Рис.2.18. Довідник «Керуючі»

Наведена форма містить інформацію про керуючих гуртожитками, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля.

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видаляти дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.19.

Керуючі

Код керуючого	ПІБ керуючого	Номер телефону	Електронна пошта
1	Цімох Марія Карлівна	0978866502	karlivna.manager@nuwm.edu.ua
2	Шарабура Людмила Миколаївна	0978866740	sharabura.manager@nuwm.edu.ua
3	Черній Світлана Іванівна	0738086674	chernii.manager@nuwm.edu.ua
4	Попельницька Галина Миколаївна	0678563241	popelnytska.manager@nuwm.edu.ua
5	Якимів Ярослава Степанівна	0671563248	yakymiv.manager@nuwm.edu.ua
6	Чемакіна Ольга Василівна	0978563214	chemakina.manager@nuwm.edu.ua

Код керуючого:

ПІБ керуючого:

Номер телефону:

Електронна пошта:

Рис.2.19. Режим редагування Довідника «Керуючі»

Як зображено за рисунку 2.19. поля запису вставляються у відповідні поля форми, після чого керуючий гуртожитком може їх змінювати, а натиснувши на кнопку *Зберегти*, змінені дані зберігаються.

Натискаючи на підпункт головного меню *Гуртожитки*, відкривається форма зображена на рисунку 2.20.

Гуртожитки

Код	ПІБ керуючого	Номер гуртожитку	Адреса
1	Цімох Марія Карлівна	1	вул. Чорновола, 51
2	Шарабура Людмила Миколаївна	2	вул. Чорновола, 55
3	Черній Світлана Іванівна	3	вул. Мірющенка, 62
4	Попельницька Галина Миколаївна	4	вул. Мірющенка, 60
5	Якимів Ярослава Степанівна	5	вул. Волинська, 24
6	Чемакіна Ольга Василівна	6	вул. Чорновола, 53

Код гуртожитку:

ПІБ керуючого:

Номер гуртожитку:

Адреса:

Рис.2.20. Довідник «Гуртожитки»

Наведена форма містить інформацію про гуртожитки, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля.

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видаляти дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.21.

Код	ПІБ керуючого	Номер гуртожитку	Адреса
1	Цімох Марія Карлівна	1	вул. Чорновола, 51
2	Шарабура Людмила Миколаївна	2	вул. Чорновола, 55
3	Черній Світлана Іванівна	3	вул. Мірющенко, 62
4	Попельницька Галина Миколаївна	4	вул. Мірющенко, 60
5	Якимів Ярослава Степанівна	5	вул. Волинська, 24
6	Чемакіна Ольга Василівна	6	вул. Чорновола, 53

Код гуртожитку:

ПІБ керуючого:

Номер гуртожитку:

Адреса:

Рис.2.21. Режим редагування Довідника «Гуртожитки»

Як зображено за рисунку 2.21. поля запису вставляються у відповідні поля форми, після чого керуючий гуртожитком може їх змінювати, а натиснувши на кнопку *Зберегти*, змінені дані зберігаються.

Пункт контекстного меню *Показати кімнати* забезпечує відкриття форми переліку усіх кімнат вибраного гуртожитку. Перелік кімнат гуртожитку номер 1 зображено на рисунку 2.22.

Кімнати гуртожитку номер : '1'

	Код кімнати	Номер гуртожитку	Номер кімнати	Кількість місць	Характеристика	Дата останнього ремонту
▶	1	1	1101	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	2	1	1102	3	3 ліжка, 1 стіл, 3 стільця	30.08.2019
	3	1	1103	3	3 ліжка, 1 стіл, 3 стільця, полиц...	17.08.2017
	4	1	1201	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	5	1	1202	3	3 ліжка, 1 стіл, 3 стільця	27.08.2019
	6	1	1203	3	3 ліжка, 1 стіл, 3 стільця, полиц...	14.08.2018

Гуртожиток:

Номер кімнати:

Кількість місць:

Характеристика:

Дата останнього ремонту:

Рис.2.22. Форма переліку Кімнат 1 гуртожитку

Наведена форма містить інформацію про усі кімнати вибраного гуртожитку, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля.

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видаляти дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.23.

Кімнати гуртожитку номер : '1'

	Код кімнати	Номер гуртожитку	Номер кімнати	Кількість місць	Характеристика	Дата останнього ремонту
▶	1	1	1101	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	2	1	1102	3	3 ліжка, 1 стіл, 3 стільця	30.08.2019
	3	1	1103	3	3 ліжка, 1 стіл, 3 стільця, полиц...	17.08.2017
	4	1	1201	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	5	1	1202	3	3 ліжка, 1 стіл, 3 стільця	27.08.2019
	6	1	1203	3	3 ліжка, 1 стіл, 3 стільця, полиц...	14.08.2018

Гуртожиток:

Номер кімнати:

Кількість місць:

Характеристика:

Дата останнього ремонту:

Рис.2.23. Режим редагування Форми Кімнат

Як зображено за рисунку 2.23. поля запису вставляються у відповідні поля форми, після чого керуючий гуртожитком може їх змінювати, а натиснувши на кнопку *Зберегти*, змінені дані зберігаються.

Пункт контекстного меню *Показати студентів* забезпечує відкриття форми переліку студентів, які проживають у вибраній кімнаті. Перелік студентів, які проживають у кімнаті номер 1101 зображено на рисунку 2.24.

Студенти, які проживають у кімнаті номер : '1101'

Пошук по ПІБ студента:

Код студента	ПІБ студента	Номер кімнати	спеціальність	Мобільний номер	Електронна пошта	Адреса	Іноземець?	Рік вступу	Дата вибуття	
▶	1	Міщенко Петро Дмитрович	1101	Теплоенергет...	0654789214	mischenko_bg1...	м. Нетішин ву...	<input type="checkbox"/>	2017	30.06.20...
	2	Ончук Олександр Леонідович	1101	Теплоенергет...	0974789214	onischuk_bg1...	м. Славути ву...	<input type="checkbox"/>	2017	30.06.20...
	3	Петренко Леонід Володимиро...	1101	Теплоенергет...	0970789211	petrenko_bg1...	с. Білка вул. С...	<input type="checkbox"/>	2017	30.06.20...

ПІБ студента:

Номер кімнати:

Спеціальність:

Мобільний номер:

Електронна пошта:

Адреса:

- Іноземець

Рік вступу:

Дата вибуття:

Рис.2.24. Форма переліку Студентів, які проживають у 1101 кімнаті

Наведена форма містить інформацію про студентів, які проживають у вибраній кімнаті, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*, тобто заселяти студентів. Кнопка *Відміна* очищує дані, які були записані у поля.

З метою забезпечення цілісності даних, при заселенні студента відбувається перевірка наявності вільних місць у кімнаті, якщо кімната повністю заповнена, то заселення не відбувається, повідомлення про помилку зображено на рисунку 2.25.

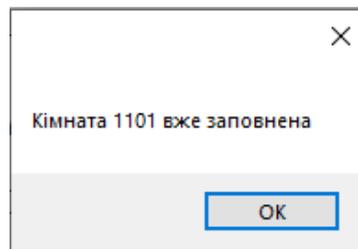


Рис.2.25. Повідомлення про помилку при заселенні студента

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видаляти дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.26.

Студенти, які проживають у кімнаті номер : '1101'

Пошук по ПІБ студента:

Код студента	ПІБ студента	Номер кімнати	спеціальність	Мобільний номер	Електронна пошта	Адреса	Іноземець?	Рік вступу	Дата вибуття
1	Мищенко Петро Дмитрович	1101	Теплоенергетика	0654789214	mischenko_bg17.nuwm.edu	м. Нетішин вул. Соборна 3	<input type="checkbox"/>	2017	30.06.2021
2	Онїщук Олександр Леонідович	1101	Теплоенергетика	0974789214	onischuk_bg17.nuwm.edu	м. Славути вул. С. Білка вул. С. Білка	<input type="checkbox"/>	2017	30.06.2021
3	Петренко Леонід Володимирович	1101	Теплоенергетика	0970789211	petrenko_bg17.nuwm.edu	с. Білка вул. С. Білка	<input type="checkbox"/>	2017	30.06.2021

ПІБ студента: Адреса:
 Номер кімнати: - Іноземець
 Спеціальність: Рік вступу:
 Мобільний номер: Дата вибуття: 30 червня 2021 р.
 Електронна пошта:

Рис.2.26. Режим редагування Форми Студенти

У наведеній формі є можливість пошуку студента по ПІБ, вводимо відповідні дані у поле та натискаємо кнопку *Пошук*, процес пошуку зображений на рисунку 2.27.

Код студента	ПІБ студента	Номер кімнати	спеціальність	Мобільний номер	Електронна пошта	Адреса	Іноземець?	Рік вступу	Дата вибуття
9	Рівна Діана Тимофіївна	2102	Маркетинг	0731102258	rivna_em17.n...	м. Львів вул. ...	<input type="checkbox"/>	2017	30.06.2017

Рис.2.27. Пошук студента по ПІБ

На формі Студенти розміщена кнопка *Нарахування*, вибираємо певного студента та, натискаючи на кнопку, відкривається форма зображено на рисунку 2.28.

Код	Дата нарахування	Сума	Тип нарахування
5	16.05.2021	400	відшкодування
7	30.08.2020	7500	проживання

ПІБ студента:

Дата нарахування: 9 червня 2021 р.

Сума нарахування:

Тип нарахування:

Рис.2.28. Форма «Нарахування Студента»

Наведена форма містить інформацію про нарахування вибраного студента, які студент повинен оплатити.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля. Також керуючий може видаляти записи, за допомогою контекстного меню.

На формі Студенти розміщена кнопка *Оплата*, вибираємо певного студента та, натискаючи на кнопку, відкривається форма зображено на рисунку 2.29.

Оплата студента : 'Мищенко Петро Дмитрович'

	Код	Дата оплати	Сума	Призначення
▶	1	31.08.2020	7080	оплата за рік
	11	16.05.2021	400	відшкодування

ПІБ студента:

Дата оплати:

Сума оплати:

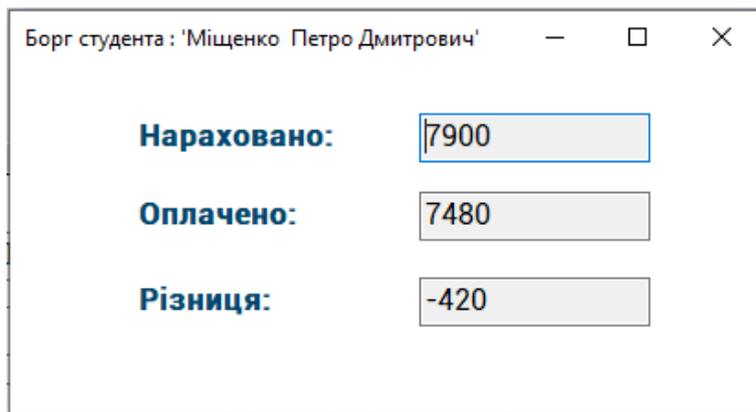
Призначення:

Рис.2.29. Форма «Оплата Студента»

Наведена форма містить інформацію про оплату вибраного студента.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля. Також керуючий може видаляти записи, за допомогою контекстного меню.

ІС може розраховувати борг студента. Натискаючи на кнопку *Борг по оплаті*, система автоматично розраховує борг студента, відкриваючи форму зображену на рисунку 2.30.

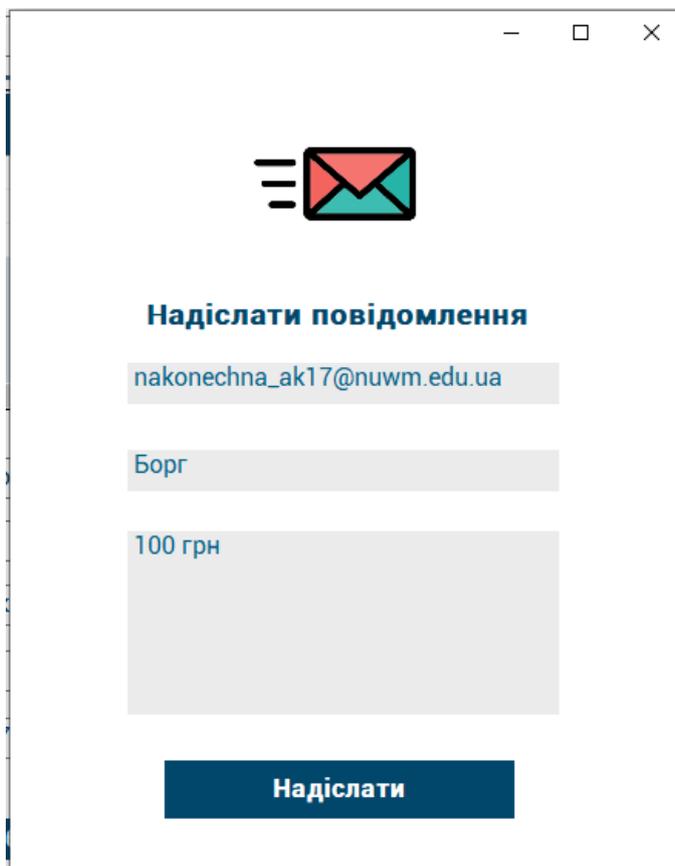


Борг студента: 'Мищенко Петро Дмитрович'

Нараховано:	<input type="text" value="7900"/>
Оплачено:	<input type="text" value="7480"/>
Різниця:	<input type="text" value="-420"/>

Рис.2.30. Форма «Борг Студента»

З метою автоматизації процесів, було передбачено можливість написання листів на електронну пошту студента. Тобто, при наявності боргу, керуючий гуртожитком може надіслати лист студенту, натискаючи на кнопку *Написати студенту* відкривається форма, вводимо e-mail студента, предмет повідомлення та суму боргу, приклад повідомлення зображений на рисунку 2.31.



Надіслати повідомлення

Рис.2.31. Приклад повідомлення

Далі натискаємо кнопку *Надіслати* та бачимо сповіщення зображено на рисунку 2.32.

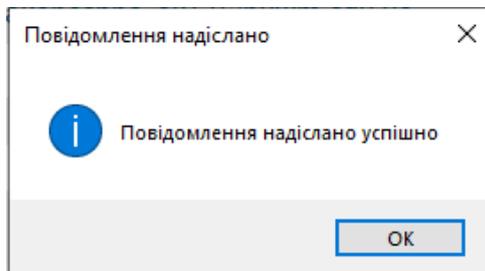


Рис.2.32. Сповіщення про успішно надіслане повідомлення

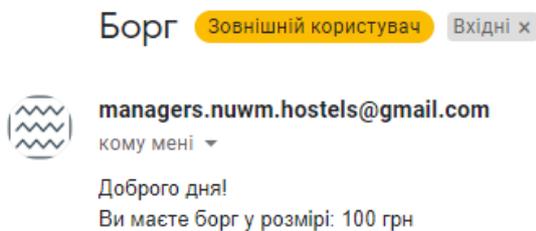


Рис.2.33. Повідомлення, яке прийшло на пошту

Натискаючи на підпункт головного меню *Кімнати*, відкривається форма зображена на рисунку 2.34.

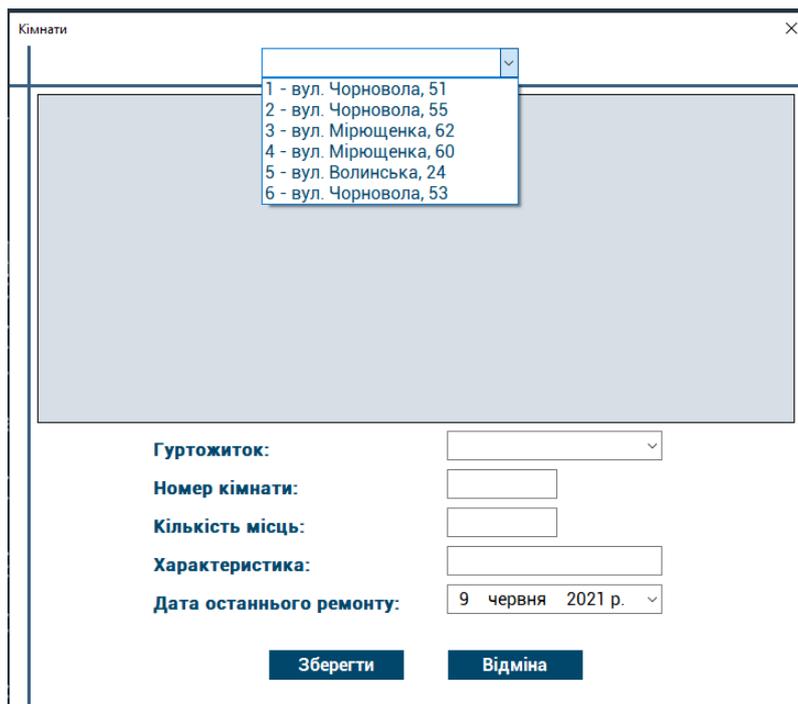


Рис.2.34. Довідник «Кімнати»

Вибираємо номер гуртожитку та бачимо інформацію про кімнати вибраного гуртожитку, зображену на рисунку 2.35.

Кімнати

1 - вул. Черновола, 51

	Код кімнати	Номер гуртожитку	Номер кімнати	Кількість місць	Характеристика	Дата останнього ремонту
▶	1	1	1101	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	2	1	1102	3	3 ліжка, 1 стіл, 3 стільця	30.08.2019
	3	1	1103	3	3 ліжка, 1 стіл, 3 стільця, полиц...	17.08.2017
	4	1	1201	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
	5	1	1202	3	3 ліжка, 1 стіл, 3 стільця	27.08.2019
	6	1	1203	3	3 ліжка, 1 стіл, 3 стільця, полиц...	14.08.2018

Гуртожиток:

Номер кімнати:

Кількість місць:

Характеристика:

Дата останнього ремонту: 9 червня 2021 р.

Зберегти **Відміна**

Рис.2.35. Кімнати 1 Гуртожитку

Наведена форма містить інформацію про усі кімнати вибраного гуртожитку, для звичайних користувачів доступний тільки перегляд інформації.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля.

Вибираючи запис та натискаючи на контекстне меню, можна редагувати дані, натиснувши на пункт *Редагувати*, та видаляти дані, пункт *Видалити*. Також перейти у режим редагування можна за допомогою подвійного кліку на запис, режим редагування форми зображений на рисунку 2.36.

Код кімнати	Номер гуртожитку	Номер кімнати	Кількість місць	Характеристика	Дата останнього ремонту
1	1	1101	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
2	1	1102	3	3 ліжка, 1 стіл, 3 стільця	30.08.2019
3	1	1103	3	3 ліжка, 1 стіл, 3 стільця, полиц...	17.08.2017
4	1	1201	3	3 ліжка, 2 столи, 3 стільця	29.08.2020
5	1	1202	3	3 ліжка, 1 стіл, 3 стільця	27.08.2019
6	1	1203	3	3 ліжка, 1 стіл, 3 стільця, полиц...	14.08.2018

Гуртожиток:

Номер кімнати:

Кількість місць:

Характеристика:

Дата останнього ремонту:

Рис.2.36. Режим редагування Довідника «Кімнати»

Як зображено за рисунку 2.36, поля запису вставляються у відповідні поля форми, після чого керуючий гуртожитком може їх змінювати, а натиснувши на кнопку *Зберегти*, змінені дані зберігаються.

Пункт контекстного меню *Показати студентів* забезпечує відкриття форми переліку студентів, які проживають у вибраній кімнаті. Перелік студентів, які проживають у кімнаті номер 1101 зображено на рисунку 2.26.

Натискаючи на підпункт головного меню *Відвідування*, відкривається форма зображена на рисунку 2.37.

Код відвідування	Код студента	Дані відвідувача	Час	Вхід/Вихід	Піб студента
23	3		19.05.2021 12:09	<input checked="" type="checkbox"/>	Петренко Леонід Володимирович
22	3	Столяр	19.05.2021 12:09	<input checked="" type="checkbox"/>	Петренко Леонід Володимирович
20	3		19.05.2021 12:04	<input checked="" type="checkbox"/>	Петренко Леонід Володимирович
13	3		19.05.2021 11:53	<input checked="" type="checkbox"/>	Петренко Леонід Володимирович
8	7		15.04.2021 13:57	<input type="checkbox"/>	Сарсян Михаель Мухамедович
7	1		14.04.2021 13:26	<input type="checkbox"/>	Мищенко Петро Дмитрович
6		Сантехнік Петро	08.04.2021 18:26	<input type="checkbox"/>	
5	4		04.03.2021 13:24	<input type="checkbox"/>	Лелен Ольга Миколаївна

Код студента:

Дані відвідувача:

Вхід/Вихід
 Вхід
 Вихід

Рис.2.37. Форма «Відвідування»

Наведена форма призначена для фіксування входу/виходу студентів та сторонніх осіб. Також передбачена можливість перегляду даних за вказаний період.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля. Пункт контекстного меню *Видалити* видаляє вибраний запис.

Натискаючи на підпункт головного меню *Інструктажі*, відкривається форма зображена на рисунку 2.38.

Інструктажі
×

Код інструктажу	ПІБ керуючого	Дата інструктажу	Тип інструктажу
9	Черній Світлана Іванівна	01.03.2021	повторний
8	Шарабура Людмила Миколаїв...	27.02.2021	повторний
7	Цімох Марія Карлівна	28.02.2021	повторний
6	Чемакіна Ольга Василівна	01.09.2020	первинний
5	Якимів Ярослава Степанівна	30.08.2020	первинний
4	Попельницька Галина Микола...	30.08.2020	первинний
2	Шарабура Людмила Миколаїв...	29.08.2020	первинний
1	Цімох Марія Карлівна	30.08.2020	первинний

Відповідальний за інструктаж:

Дата проведення інструктажу:

Тип інструктажу:

Зберегти
Відміна

Код інструктованого студента	Код інструктажу	ПІБ студента	Додатково
1	1	Міщенко Петро Дмитрович	проінструктований
2	1	Оніщук Олександр Леонідович	проінструктований
3	1	Петренко Леонід Володимиро...	проінструктований
4	1	Дерун Олена Миколаївна	проінструктований
5	1	Змій Лариса Василівна	проінструктований

Код інструктажу:

ПІБ студента:

Додатково:

Зберегти
Відміна

Рис.2.38. Форма «Інструктажі»

Наведена форма призначена для внесення даних про інструктажі. Натискаючи на певний інструктаж, у нижній таблиці відображаються студенти, які були проінструктовані.

Керуючий гуртожитком має можливість додавати нові записи у форму, заповнюючи відповідні поля та натиснувши кнопку *Зберегти*. Кнопка *Відміна* очищує дані, які були записані у поля. Пункт контекстного меню *Видалити* видаляє вибраний запис, для редагування запису – пункт *Редагувати*.

На завершення необхідно відмітити, що інтерфейс інформаційної системи оформлено відповідно до брендбуку НУВГП[1].

Отже, для ІС «Гуртожитки НУВГП» створено чотири довідники: *Спеціальності*, *Керуючі*, *Гуртожитки*, *Кімнати* для перегляду інформації, додавання, редагування та видалення. Забезпечена можливість перегляду кімнат вибраного гуртожитку та студентів, які проживають у відповідній кімнаті. Забезпечена цілісність даних, виконується перевірка наявності вільних місць у кімнаті, якщо місць немає, то виводиться відповідне повідомлення. Також передбачено можливість розрахунку боргу студента. Для автоматизації процесів забезпечена можливість оповіщення студента про наявний борг на електронну пошту. ІС містить форму *Відвідування* для фіксації входу/виходу студентів та інших осіб, а також форму *Інструктажі* для внесення даних про інструктажі і студентів, які були проінструктовані.

2.3. Аналіз вихідної інформації та інтерфейс користувача

За допомогою інформаційної системи можна сформувати ордер на поселення для кожного студента. Для цього потрібно у головному меню ІС вибрати пункт *Довідники – Кімнати – Показати студентів* обрати студента та натиснути на кнопку *Сформувати ордер на поселення*. Дана кнопка відкриває документ Word, в якому знаходить автоматично сформований ордер, зображений на рисунку 2.39

Корінець ордера № _____ Підпис _____	Форма №1
	Національний університет водного господарства та природокористування
	Ордер № _____ від _____ 20__ р. на ліжко-місце
	Направляється студент
	Міщенко Петро Дмитрович Прізвище
	Теплоенергетика Спеціальність
	Для поселення до гуртожитку № <u> 1 </u> на ліжко-місце в кімнаті № <u> 1101 </u> Відповідальна посадова особа НУВГП <hr/> Підпис Ордер та квитанція про сплату зберігати до закінчення Терміну проживання студента! Ордер дійсний протягом 10 днів.
	<hr/> Залишається у адміністратора

Рис.2.39. Приклад автоматично сформованого ордеру на поселення

На головному меню інформаційної системи розміщений пункт *Звіти*, натиснувши на нього, відкриваються звіти, зображені на рисунку 2.40, які ІС може автоматично формувати.

Звіти	Довідка
Іноземці	Alt+Z
Студенти, що запізнюються	Alt+X
Кімнати, що потребують ремонту	Alt+C
Боржники по оплаті	Alt+A

Рис.2.40. Звіти, які формуються автоматично

Звіт *Іноземці* забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, зображена на рисунку 2.41, в якій відображаються усі іноземні студенти, що мешкають у гуртожитках.

№ з/п	ПІБ студента	Номер кімнати	Номер телефону	Адреса
1	Тагаєв Тагай Тигранович	2101	987452143	м. Стамбул
2	Саргсян Михаель Мухамедович	2101	963257400	м. Стамбул
3	Аріель	1102	48586686	Кіпр

Рис.2.41. Звіт «Іноземці»

Звіт *Студенти, що запізнюються* забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, зображена на рисунку 2.42, в якій відображаються студенти, які запізнюються, тобто приходять з 24 до 6 години.

№ з/п	ПІБ студента	Номер кімнати	Час входу	Номер телефону
1	Оніщук Олександр Леонідович	1101	02.02.2021 4:20:56	974789214
2	Міщенко Петро Дмитрович	1101	02.02.2021 4:20:56	654789214
3	Дерун Олена Миколаївна	1102	02.02.2021 1:20:56	678541258

Рис.2.42. Звіт «Студенти, що запізнюються»

Звіт *Кімнати, що потребують ремонту* забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, зображена на рисунку 2.43, в якій відображаються десять кімнати, в яких давно не робили ремонт.

№ з/п	Номер гуртожитку	Номер кімнати	Дата останнього ремонту
1	4	4202	26.07.2015 0:00:00
2	6	6202	02.08.2015 0:00:00
3	6	6201	01.09.2016 0:00:00
4	1	1103	17.08.2017 0:00:00
5	4	4101	01.09.2017 0:00:00
6	6	6102	29.07.2018 0:00:00
7	6	6101	14.08.2018 0:00:00
8	1	1203	14.08.2018 0:00:00
9	2	2101	02.09.2018 0:00:00
10	3	3202	03.09.2018 0:00:00

Рис.2.43. Звіт «Кімнати, що потребують ремонту»

Звіт *Боржники по оплаті* забезпечує відкриття файлу Excel, в якому міститься автоматично згенерована таблиця, зображена на рисунку 2.44, в якій відображаються студенти, які мають борг, та сума боргу.

№ з/п	Гуртожиток №	Кімната	Студент	Нараховано, грн	Оплачено, грн	Борг, грн
1	1	1101	Мищенко Петро Дмитрович	7900	7480	420

Рис.2.44. Звіт «Боржники по оплаті»

Для роботи з інформаційною системою запускаємо додаток, відкривається головна форма ІС, зображена на рисунку 2.45.

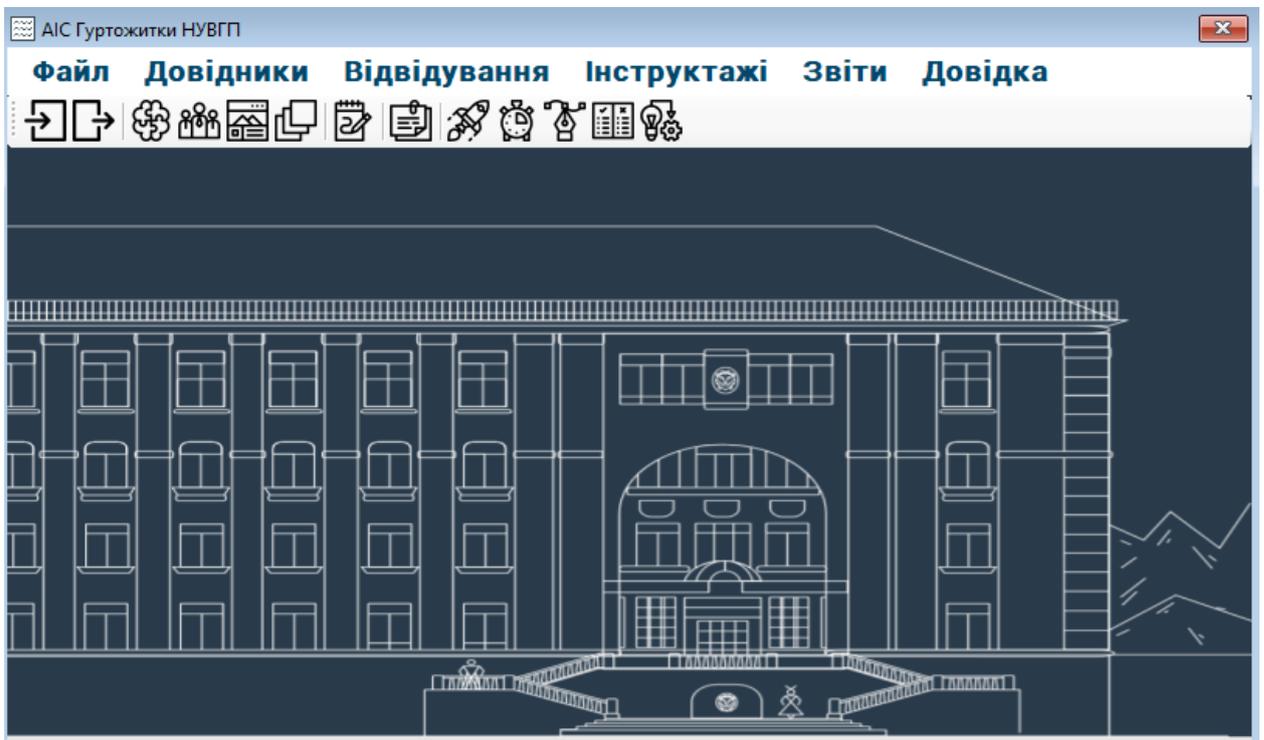


Рис.2.45. Головна форма ІС «Гуртожитки НУВГП»

Для входу у систему натискаємо *Файл – Вхід у систему*, відкривається вікно авторизації, зображене на рисунку 2.46.

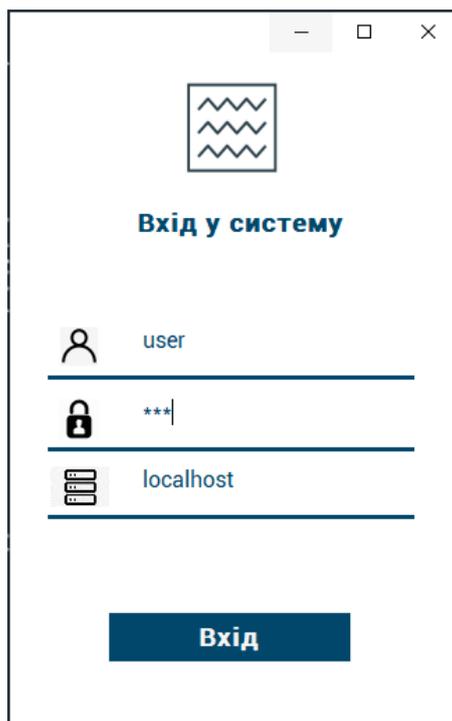


Рис. 2.46. Вхід для звичайного користувача

Оскільки керуючий гуртожитком може редагувати, додавати та видаляти записи , він заходить під іншим логіном, який зображений на рисунку 2.47.



Рис. 2.47. Вхід для керуючого гуртожитком

Щоб побачити інформацію про розробника системи, натискаємо на пункт головного меню *Довідка*, відкривається форма зображена на рисунку 2.48.

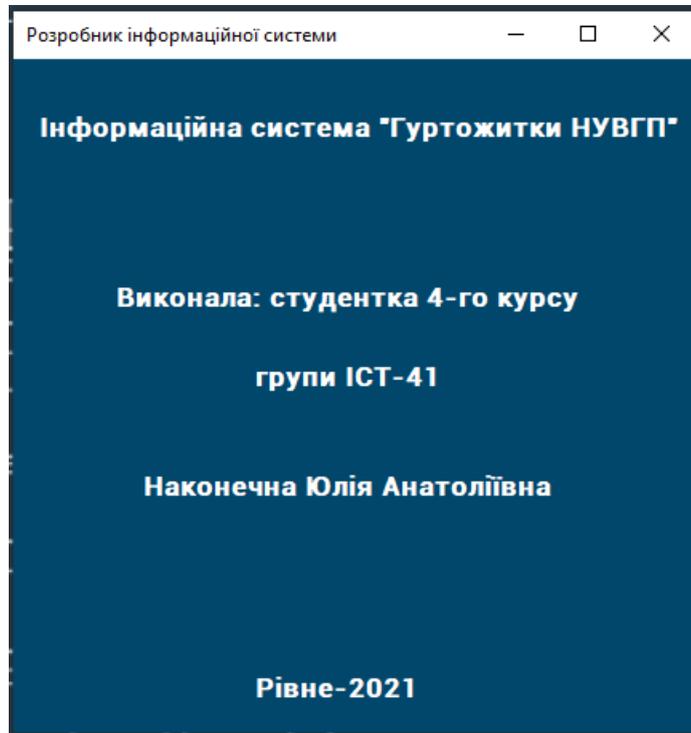


Рис. 2.48. Інформація про розробника ІС

Також вхід та вихід, відкриття інших форм, можна здійснити за допомогою панелі інструментів, розміщених під головним меню.

Отже, за допомогою інформаційної системи можна сформувати ордер на поселення, сформувати звіти, а саме: іноземці, які проживають у гуртожитку, студенти, які запізнюються, кімнати, які потребують ремонту, студенти, які мають борги. Під час розробки інтерфейсу ІС «Гуртожитки НУВГП» було створено головне меню, за допомогою якого можна швидко отримати доступ до форм із даними та забезпечена можливість входу для звичайного користувача та керуючого гуртожитком.

Висновки по розділу 2

Під час моделювання бази даних інформаційної системи «Гуртожитків НУВГП» було створено десять сутностей, зокрема: Студенти, Відвідування,

Спеціальності, Оплата, Нарахування, Гуртожитки, Кімнати, Керуючі, Інструктажі БЖД, Студенти Інструктажів, з'єднаних між собою зовнішніми ключами.

Для функціонування ІС «Гуртожитки НУВГП» створено чотири довідники: *Спеціальності*, *Керуючі*, *Гуртожитки*, *Кімнати* для перегляду інформації, додавання, редагування та видалення. Забезпечена можливість перегляду кімнат вибраного гуртожитку та студентів, які проживають у відповідній кімнаті. Забезпечена цілісність даних, виконується перевірка наявності вільних місць у кімнаті, якщо місць немає, то виводиться відповідне повідомлення. Також передбачено можливість розрахунку боргу студента. Для автоматизації процесів забезпечена можливість оповіщення студента про наявний борг на електронну пошту. ІС містить форму *Відвідування* для фіксації входу/виходу студентів та інших осіб, а також форму *Інструктажі* для внесення даних про інструктажі і студентів, які були проінструктовані.

Під час розробки інтерфейсу ІС «Гуртожитки НУВГП» було створено головне меню, за допомогою якого можна швидко отримати доступ до форм із даними та забезпечена можливість входу для звичайного користувача та керуючого гуртожитком. Також за допомогою інформаційної системи можна сформувати ордер на поселення, сформувати звіти, а саме: іноземці, які проживають у гуртожитку, студенти, які запізнюються, кімнати, які потребують ремонту, студенти, які мають борги.

ВИСНОВКИ

Результатом виконаної роботи є desktop-додаток, призначений для управління гуртожитками НУВГП.

Проаналізувавши предметну область, було визначено найбільш оптимальний набір інструментів, що сприяють проектуванню, створенню і функціонуванню інформаційної системи, а саме: мова програмування C#, середовище розробки Visual Studio 2019, платформа NET Framework, бази даних MySQL з компонентами MySQL-Front та Navicat for MySQL, Microsoft Word та Excel як засоби експорту даних.

Розглянувши ІС «БІТ.Гуртожиток», як засіб автоматизації обліку проживання, виявлено основні особливості даної системи, а також головний її недолік – неможливість відокремленого функціонування без платформи ІС: Підприємство 8.

Під час моделювання бази даних інформаційної системи «Гуртожитків НУВГП» було створено десять сутностей, зокрема: Студенти, Відвідування, Спеціальності, Оплата, Нарахування, Гуртожитки, Кімнати, Керуючі, Інструктажі БЖД, Студенти Інструктажів, з'єднаних між собою зовнішніми ключами.

Для функціонування ІС «Гуртожитки НУВГП» створено необхідні довідники. Забезпечено можливість перегляду кімнат вибраного гуртожитку та студентів, які проживають у відповідній кімнаті. Підтримується цілісність даних, зокрема виконується перевірка наявності вільних місць у кімнаті. Передбачено можливість розрахунку боргу студента. Для автоматизації процесів забезпечена можливість оповіщення студента про наявний борг на електронну пошту. ІС містить форму для фіксації входу/виходу студентів та інших осіб, а також форму для внесення даних про інструктажі. В інформаційній системі «Гуртожитки НУВГП» передбачена можливість автоматичного формування вихідних даних, а саме: ордер на поселення, звіт про іноземців, студентів, що запізнюються або мають борги, а також кімнати, які потребують ремонту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Правила використання символіки НУВГП (Брендбук). URL: <https://nuwm.edu.ua/publiczna-informacija/brendbuk> (дата звернення 29.05.2021).
2. Гольцман В. MYSQL 5.0. БИБЛИОТЕКА ПРОГРАММММТА. Питер, 2010.
3. Концепція державної програми відновлення та розбудови мережі гуртожитків для проживання студентів закладів вищої освіти на 2021-2025 роки. URL: https://mon.gov.ua/storage/app/media/gromadske-obgovorennya/2020/08/17/Proekt_rozporiadzhennia_Konceptia_20.08.17.doc (дата звернення 29.05.2021).
4. Лабор В. В. Си Шарп: Создание приложений для Windows. Минск, 2003. -384с.
5. Логічна модель даних. URL: https://uk.wikipedia.org/wiki/Логічна_модель_даних (дата звернення 02.06.2021).
6. Осипов Н.А. Разработка Windows приложений на С# - СПб: НИУ ИТМО, 2012. – 74с.
7. Положення про студентський гуртожиток Національного університету водного господарства та природокористування. URL: <https://start.nuwm.edu.ua/images/Docs/campus/Положення%20про%20студентський%20гуртожиток.pdf> (дата звернення 29.05.2021).
8. 6 причин, чому Python - це мова програмування майбутнього. URL: <http://helpexe.ru/programmirovanie/6-prichin-pochemu-python-jazyk-programmirovaniya> (дата звернення 29.05.2021).
9. 4 причини, чому JavaScript такий популярний. URL: <https://ichi.pro/ru/4-priciny-pocemu-javascript-tak-popularen-111955255305031> (дата звернення 29.05.2021).

10. Рейтинг мов програмування 2021: частка Python зменшується, а TypeScript обійшов C++. URL: <https://dou.ua/lenta/articles/language-rating-jan-2021/> (дата звернення 30.05.2021).
11. Середовище розробки Microsoft Visual Studio. URL: https://informatics.in.ua/programming_csharp/part_01.php (дата звернення 30.05.2021).
12. Яку мову програмування вчити прямо зараз: 9 самих затребуваних. URL: <https://trends.rbc.ru/trends/education/601c1a6b9a79472c4806230a><https://trends.rbc.ru/trends/education/601c1a6b9a79472c4806230a> (дата звернення 30.05.2021).

ДОДАТКИ

Код головної форми ІС «Гуртожитки НУВГП»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
using Excel = Microsoft.Office.Interop.Excel;

namespace nuwmhostels
{
    public partial class Form1 : Form
    {
        static public MySqlConnection con = new MySqlConnection();
        MySqlCommand cmd;

        public Form1()
        {
            InitializeComponent();
        }

        private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DialogResult dialogResult = MessageBox.Show
                ("Ви впевнені, що хочете завершити роботу?", "Вихід", MessageBoxButtons.YesNo,
                MessageBoxIcon.Information);

            if (dialogResult == DialogResult.Yes)
            {
                Close();
            }
        }
        private void спеціальностіToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form2 newForm = new Form2();
            newForm.ShowDialog();
        }
        private void керуючіToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form3 newForm = new Form3();
            newForm.ShowDialog();
        }
        private void гуртожиткиToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form4 newForm = new Form4();
            newForm.ShowDialog();
        }
        private void кімнатиToolStripMenuItem_Click(object sender, EventArgs e)
        {

```

```

        Form5 newForm = new Form5();
        newForm.ShowDialog();
    }
    private void відвідуванняToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        Form11 newForm = new Form11();
        newForm.ShowDialog();
    }
    private void інструктажіToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        Form12 newForm = new Form12();
        newForm.ShowDialog();
    }
    private void кімнатиЩоПотребуютьРемонтToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        string fileName = "rooms_repair.xlsx";
        string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
        string targetPath = Environment.CurrentDirectory + @"\rep";
        string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
        string destFile = System.IO.Path.Combine(targetPath, fileName);
        System.IO.File.Copy(sourceFile, destFile, true);

        Excel.Application app = new Excel.Application();
        app.Visible = false;
        string filename = Environment.CurrentDirectory + @"\rep\rooms_repair.xlsx";
        Excel.Workbook wb = app.Workbooks.Open(filename,
        Type.Missing, Type.Missing, Type.Missing, Type.Missing,
        Type.Missing, Type.Missing, Type.Missing, Type.Missing,
        Type.Missing, Type.Missing, Type.Missing, Type.Missing,
        Type.Missing, Type.Missing);
        Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

        string query2 = @"SELECT
    hostels.number,
    rooms.number_room,
    rooms.date_last_repair
    FROM
    rooms
    Inner Join hostels ON rooms.Id_hostel = hostels.Id_hostel
    ORDER BY
    rooms.date_last_repair ASC
    LIMIT 10

";
        int i = 8;
        using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query2, Form1.con))
        {
            DataTable table = new DataTable();
            table.Locale = System.Globalization.CultureInfo.InvariantCulture;
            da_sp.Fill(table);
            foreach (DataRow row in table.Rows)
            {
                ws.Cells[i, 3] = row["number"].ToString();
                ws.Cells[i, 4] = row["number_room"].ToString();
                ws.Cells[i, 5] = row["date_last_repair"].ToString();
            }
        }
    }
}

```

```

        i = i + 1;
    }
}

app.Visible = true;
}

private void іноземціToolStripMenuItem_Click(object sender, EventArgs e)
{
    string fileName = "foreigners.xlsx";
    string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName);
    System.IO.File.Copy(sourceFile, destFile, true);

    Excel.Application app = new Excel.Application();
    app.Visible = false;
    string filename = Environment.CurrentDirectory + @"\rep\foreigners.xlsx";
    Excel.Workbook wb = app.Workbooks.Open(filename,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing);
    Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

    string query2 = @"SELECT
students.pib,
rooms.number_room,
students.phone,
students.adress
FROM
students
Inner Join rooms ON students.Id_room = rooms.Id_room
WHERE
students.foreigner = '1'
";

    int i = 8;
    using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query2, Form1.con))
    {
        DataTable table = new DataTable();
        table.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_sp.Fill(table);
        foreach (DataRow row in table.Rows)
        {
            ws.Cells[i, 3] = row["pib"].ToString();
            ws.Cells[i, 4] = row["number_room"].ToString();
            ws.Cells[i, 5] = row["phone"].ToString();
            ws.Cells[i, 6] = row["adress"].ToString();

            i = i + 1;
        }
    }
}

```

```

    app.Visible = true;
}

private void студентиЩоЗапізнюютьсяToolStripMenuItem_Click(object sender, EventArgs
e)
{
    string fileName = "students_late.xlsx";
    string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName);
    System.IO.File.Copy(sourceFile, destFile, true);

    Excel.Application app = new Excel.Application();
    app.Visible = false;
    string filename = Environment.CurrentDirectory + @"\rep\students_late.xlsx";
    Excel.Workbook wb = app.Workbooks.Open(filename,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing);
    Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

    string query2 = @"
SELECT
students.pib,
rooms.number_room,
TIME(visiting.time_e),
visiting.time_e,
students.phone
FROM
visiting
Inner Join students ON visiting.Id_student = students.Id_student
Inner Join rooms ON students.Id_room = rooms.Id_room
WHERE
TIME(visiting.time_e) BETWEEN '00:00:00' AND '06:00:00'
ORDER BY
visiting.time_e DESC

";

    int i = 8;
    using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query2, Form1.con))
    {
        DataTable table = new DataTable();
        table.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_sp.Fill(table);
        foreach (DataRow row in table.Rows)
        {
            ws.Cells[i, 3] = row["pib"].ToString();
            ws.Cells[i, 4] = row["number_room"].ToString();
            ws.Cells[i, 5] = row["time_e"].ToString();
            ws.Cells[i, 6] = row["phone"].ToString();

            i = i + 1;
        }
    }
}

```

```

    app.Visible = true;
}

private void вхідУСистемуToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form14 newForm = new Form14();
    newForm.ShowDialog();
}

private void боржникиПоОплатіToolStripMenuItem_Click(object sender, EventArgs e)
{
    //копіювання файлів
    string fileName = "borg.xls";
    string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName);
    System.IO.File.Copy(sourceFile, destFile, true);
    //завершення копіювання
    Excel.Application app = new Excel.Application();
    app.Visible = false;
    string filename = Environment.CurrentDirectory + @"\rep\borg.xls";
    Excel.Workbook wb = app.Workbooks.Open(filename,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing);
    Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];
    //формуємо таблицю з даними про всіх студентів
    string query = @"SELECT
hostels.number,
rooms.number_room,
students.pib,
students.Id_student
FROM
students
Inner Join rooms ON students.Id_room = rooms.Id_room
Inner Join hostels ON rooms.Id_hostel = hostels.Id_hostel
ORDER BY
hostels.number ASC,
rooms.number_room ASC";

    int i = 8;
    int j = 0;
    int nomer_zp = 0;
    int borg = 0;
    int nar = 0;
    int opl = 0;
    using (MySqlDataAdapter da_st = new MySqlDataAdapter(query, Form1.con))
    {
        DataTable table_st = new DataTable();
        table_st.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_st.Fill(table_st);
        foreach (DataRow row in table_st.Rows)
        {

```

```

//борги
string Id_student = row["Id_student"].ToString();
string query2 = @"SELECT
SUM(0-profit.amount) AS amount
FROM
profit
WHERE
profit.Id_student = " + Id_student + @"
UNION
SELECT
SUM(payment.amount) AS amount
FROM
payment
WHERE
payment.Id_student = " + Id_student;

using (MySqlDataAdapter da_borg = new MySqlDataAdapter(query2, Form1.con))
{
    DataTable table_borg = new DataTable();
    table_borg.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_borg.Fill(table_borg);
    DataRow[] rows = table_borg.Select();
    j = table_borg.Rows.Count;
    if (j > 1)
    {
        if (DBNull.Value.Equals(rows[0]["amount"]))
        {
            nar = 0;
        }
        else
        {
            nar = Math.Abs(Convert.ToInt16(rows[0]["amount"]));
        }
        if (DBNull.Value.Equals(rows[1]["amount"]))
        {
            opl = 0;
        }
        else
        {
            opl = Convert.ToInt16(rows[1]["amount"]);
        }
        borg = nar - opl;
        if (borg > 0)
        {
            nomer_zp++;
            ws.Cells[i, 2] = nomer_zp.ToString();
            ws.Cells[i, 3] = row["number"].ToString();
            ws.Cells[i, 4] = row["number_room"].ToString();
            ws.Cells[i, 5] = row["pib"].ToString();
            ws.Cells[i, 6] = nar.ToString();
            ws.Cells[i, 7] = opl.ToString();
            ws.Cells[i, 8] = borg.ToString();
            i++;
        }
    }
}
}

```

```

        } //кінець циклу
    }

    app.Visible = true;
} //кінець процедури

private void розробникиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Form15 newForm = new Form15();
    newForm.ShowDialog();
}
}
}

```

Код форми «Спеціальності»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form2 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        public Form2()
        {
            //specialties.Id_specialty,
            InitializeComponent();
            query = @"SELECT
specialties.Id_specialty,
specialties.code_specialty,
specialties.name
FROM
specialties
ORDER BY
specialties.Id_specialty ASC";
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "specialties");
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "specialties";
            dataGridView1.Columns[0].HeaderText = "Код ";
            dataGridView1.Columns[1].HeaderText = "Код спеціальності";
            dataGridView1.Columns[2].HeaderText = "Назва спеціальності";
            dataGridView1.Columns[1].Width = 150;
        }
    }
}

```

```

dataGridView1.Columns[2].Width = 325;
dataGridView1.Refresh();
}

private void Form2_Shown(object sender, EventArgs e)
{
    insert_mode = true;
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `specialties` (`code_specialty`, `name`) VALUES (";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ")"; // convert.toint16(text)
    }
    else
    {
        query2 = "UPDATE `specialties` SET";
        query2 = query2 + "`code_specialty` = " + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "`name` = " + Convert.ToString(textBox3.Text) + " ";
        query2 = query2 + " WHERE `Id_specialty` = " +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();

    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "specialties");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();

    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }
    }

    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

```

```

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `specialties` WHERE `Id_specialty`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "specialties");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            insert_mode = true;
        }

        catch

        {
            MessageBox.Show("Видалення неможливо. Керуючий має пов`язані записи",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
}

```

```

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        insert_mode = true;
    }

}
}

```

Код форми «Керуючі»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form3 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        public Form3()
        {
            InitializeComponent();
            query = @"SELECT
            managers.Id_manager,
            managers.pib,
            managers.phone,
            managers.e_mail
            FROM
            managers
            ORDER BY
            managers.Id_manager ASC";
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "managers");
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "managers";
            dataGridView1.Columns[0].HeaderText = "Код керуючого";
            dataGridView1.Columns[1].HeaderText = "ПІБ керуючого";
            dataGridView1.Columns[2].HeaderText = "Номер телефону";
            dataGridView1.Columns[3].HeaderText = "Електронна пошта";
            dataGridView1.Columns[0].Width = 90;
            dataGridView1.Columns[1].Width = 210;
            dataGridView1.Columns[2].Width = 100;
            dataGridView1.Columns[3].Width = 210;
            dataGridView1.Refresh();
        }
    }
}

```

```

private void Form3_Shown(object sender, EventArgs e)
{
    insert_mode = true;
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `managers` (`pib`,`phone`,`e_mail`) VALUES (";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox4.Text) + ")";
    }
    else
    {
        query2 = "UPDATE `managers` SET";
        query2 = query2 + "`pib`=" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "`phone`=" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "`e_mail`=" + Convert.ToString(textBox4.Text) + " ";
        query2 = query2 + " WHERE `Id_manager`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "managers");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();

    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }
    }

    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox4.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `managers` WHERE `Id_manager`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "managers");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
            insert_mode = true;
        }

        catch

        {
            MessageBox.Show("Видалення неможливо. Керуючий має пов'язані записи",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
}

```

```

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        insert_mode = true;
    }
}
}

```

Код формы «Гуртожитки»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form4 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;

        public Form4()
        {
            InitializeComponent();
            query = @"SELECT
            hostels.Id_hostel,
            managers.pib,
            hostels.number,
            hostels.adress,
            managers.Id_manager
            FROM
            managers
            Inner Join hostels ON managers.Id_manager = hostels.Id_manager";
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "hostels");
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "hostels";
            dataGridView1.Columns[0].HeaderText = "Код";
            dataGridView1.Columns[1].HeaderText = "ПІБ керуючого";
            dataGridView1.Columns[2].HeaderText = "Номер гуртожитку";
            dataGridView1.Columns[3].HeaderText = "Адреса";
            dataGridView1.Columns[4].Visible = false;
            dataGridView1.Columns[0].Width = 90;
            dataGridView1.Columns[3].Width = 150;
            dataGridView1.Columns[2].Width = 100;
            dataGridView1.Columns[1].Width = 230;
        }
    }
}

```

```

        dataGridView1.Refresh();
    }
}

private void Form4_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    string query3 = @"SELECT
managers.Id_manager,
managers.pib
FROM
managers";

    using (MySqlDataAdapter da_hos = new MySqlDataAdapter(query3, Form1.con))
    {
        DataTable table = new DataTable();
        table.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_hos.Fill(table);
        comboBox1.DataSource = table;
        comboBox1.DisplayMember = table.Columns["pib"].ColumnName;
        comboBox1.ValueMember = table.Columns["Id_manager"].ColumnName;
        comboBox1.Text = "";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `hostels` (`Id_manager`,`number`,`adress`) VALUES (";
        query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ")";
    }
    else
    {
        query2 = "UPDATE `hostels` SET";
        query2 = query2 + "`Id_manager`=" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "`number`=" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "`adress`=" + Convert.ToString(textBox3.Text) + " ";
        query2 = query2 + " WHERE `Id_hostel`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }

    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "hostels");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
}

```

```

try
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[2].Value.ToString().Contains(textBox2.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }
    }
}
catch
{
    dataGridView1.Rows[0].Selected = true;
}

dataGridView1.Enabled = true;
comboBox1.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
insert_mode = true;
}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        comboBox1.SelectedValue = dataGridView1.CurrentRow.Cells[4].Value;
        comboBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    comboBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {

```



```

using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form5 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;

        public Form5()
        {
            InitializeComponent();
        }

        private void Form5_Shown(object sender, EventArgs e)
        {
            insert_mode = true;
            string query2 = @"SELECT
            hostels.Id_hostel,
            concat (hostels.number,' - ',hostels.adress ) as hos
            FROM
            hostels
            ";

            using (MySqlDataAdapter da_hos = new MySqlDataAdapter(query2, Form1.con))
            {
                DataTable table = new DataTable();
                table.Locale = System.Globalization.CultureInfo.InvariantCulture;
                da_hos.Fill(table);
                comboBox1.DataSource = table;
                comboBox1.DisplayMember = table.Columns["hos"].ColumnName;
                comboBox1.ValueMember = table.Columns["Id_hostel"].ColumnName;
                comboBox1.Text = "";
                //DataTable table2 = new DataTable();
                //table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
                //da_hos.Fill(table2);
                comboBox2.DataSource = table;
                comboBox2.DisplayMember = table.Columns["hos"].ColumnName;
                comboBox2.ValueMember = table.Columns["Id_hostel"].ColumnName;
                comboBox2.Text = "";
            }
        }

        private void comboBox1_SelectionChangeCommitted(object sender, EventArgs e)
        {
            query = @"SELECT
            rooms.Id_room,
            hostels.number,
            rooms.number_room,
            rooms.number_seats,
            rooms.characteristic,
            rooms.date_last_repair,

```

```

hostels.Id_hostel
FROM
rooms
Inner Join hostels ON rooms.Id_hostel = hostels.Id_hostel
WHERE
hostels.Id_hostel = " + Convert.ToString(comboBox1.SelectedValue) ;
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "rooms");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "rooms";
dataGridView1.Columns[0].HeaderText = "Код кімнати";
dataGridView1.Columns[1].HeaderText = "Номер гуртожитку";
//dataGridView1.Columns[2].HeaderText = "Адреса гуртожитку";
dataGridView1.Columns[2].HeaderText = "Номер кімнати";
dataGridView1.Columns[3].HeaderText = "Кількість місць";
dataGridView1.Columns[4].HeaderText = "Характеристика";
dataGridView1.Columns[5].HeaderText = "Дата останнього ремонту";
dataGridView1.Columns[6].Visible = false;
dataGridView1.Columns[0].Width = 80;
dataGridView1.Columns[1].Width = 93;
dataGridView1.Columns[2].Width = 80;
dataGridView1.Columns[3].Width = 80;
dataGridView1.Columns[4].Width = 200;
//dataGridView1.Columns[2].Width = 100;
dataGridView1.Refresh();
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `rooms`
(^Id_hostel`, `number_room`, `number_seats`, `characteristic`, `date_last_repair`) VALUES (";
        query2 = query2 + "" + Convert.ToString(comboBox2.SelectedValue) + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + """;
    }
    else

    {
        query2 = "UPDATE `rooms` SET";
        query2 = query2 + "`Id_hostel`=" + Convert.ToString(comboBox2.SelectedValue) + ",";
        query2 = query2 + "`number_room`=" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "`number_seats`=" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "`characteristic`=" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "`date_last_repair`=" + dateTimePicker1.Value.ToString("yyyy-MM-dd")
+ """;
        query2 = query2 + " WHERE `Id_room`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;

```

```

cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "rooms");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
try
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[2].Value.ToString().Contains(textBox1.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }
    }
}
catch
{
    dataGridView1.Rows[0].Selected = true;
}
dataGridView1.Enabled = true;
comboBox1.Enabled = true;
comboBox2.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
insert_mode = true;
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    comboBox1.Enabled = true;
    comboBox2.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        comboBox2.SelectedValue = dataGridView1.CurrentRow.Cells[6].Value;
        comboBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[4].Value);
        dateTimePicker1.Value = Convert.ToDateTime(dataGridView1.CurrentRow.Cells[5].Value);
    }
}

```

```

        dataGridView1.Enabled = false;
        insert_mode = false;
        comboBox1.Enabled = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `rooms` WHERE `Id_room`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "rooms");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            comboBox1.Enabled = true;
            comboBox2.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            insert_mode = true;
        }
        catch

        {
            MessageBox.Show("Видалення неможливо. Кімната має пов`язані записи", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void студентиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if(dataGridView1.RowCount==0)
    {
        MessageBox.Show("Виберіть кімнату або введіть нову", "Увага", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }

    Form6 newForm = new Form6();
    Form6.Id_room = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
}

```

```

        Form6.number_room = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        Form6.hostel_number = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        newForm.ShowDialog();
    }
}
}

```

Код формы «Студенти»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
using Word = Microsoft.Office.Interop.Word;
using Excel = Microsoft.Office.Interop.Excel;

namespace nuwmhostels
{
    public partial class Form6 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        static public string Id_room;
        static public string number_room;
        static public string pib;
        static public string hostel_number;
        static public string specialty;

        public Form6()
        {
            InitializeComponent();
        }
    }
}

```

```
private void Form6_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    Text = "Студенти, які проживають у кімнаті номер : " + number_room + "";
    query = @"SELECT
students.Id_student,
students.pib,
rooms.number_room,
specialties.name,
students.phone,
students.e_mail,
students.adress,
students.foreigner,
students.year_entry,
students.date_leave,
specialties.Id_specialty,
rooms.Id_room
FROM
rooms
Inner Join students ON students.Id_room = rooms.Id_room
Inner Join specialties ON students.Id_specialty = specialties.Id_specialty
WHERE
rooms.number_room = " + number_room;
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "students");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "students";
dataGridView1.Columns[0].HeaderText = "Код студента";
dataGridView1.Columns[1].HeaderText = "ПІБ студента";
dataGridView1.Columns[2].HeaderText = "Номер кімнати";
dataGridView1.Columns[3].HeaderText = "Спеціальність";
dataGridView1.Columns[4].HeaderText = "Мобільний номер";
dataGridView1.Columns[5].HeaderText = "Електронна пошта";
dataGridView1.Columns[6].HeaderText = "Адреса";
dataGridView1.Columns[7].HeaderText = "Іноземець?";
dataGridView1.Columns[8].HeaderText = "Рік вступу";
```

```

dataGridView1.Columns[9].HeaderText = "Дата вибуття";
dataGridView1.Columns[10].HeaderText = "Id_specialty";
dataGridView1.Columns[11].HeaderText = "Id_room";
dataGridView1.Columns[10].Visible = false;
dataGridView1.Columns[11].Visible = false;
//dataGridView1.Columns[2].Visible = false;
dataGridView1.Columns[1].Width = 200;

```

```

string query3 = @"SELECT
specialties.Id_specialty,
specialties.name
FROM
specialties";

```

```

using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query3, Form1.con))

```

```

{
    DataTable table = new DataTable();
    table.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_sp.Fill(table);
    comboBox1.DataSource = table;
    comboBox1.DisplayMember = table.Columns["name"].ColumnName;
    comboBox1.ValueMember = table.Columns["Id_specialty"].ColumnName;
    comboBox1.Text = "";
}

```

```

string query4 = @"SELECT
rooms.Id_room,
rooms.number_room
FROM
rooms";

```

```

using (MySqlDataAdapter da_rm = new MySqlDataAdapter(query4, Form1.con))

```

```

{
    DataTable table2 = new DataTable();
    table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_rm.Fill(table2);
    comboBox2.DataSource = table2;
    comboBox2.DisplayMember = table2.Columns["number_room"].ColumnName;
    comboBox2.ValueMember = table2.Columns["Id_room"].ColumnName;
}

```

```

        comboBox2.Text = "";
    }
}
private void button1_Click(object sender, EventArgs e)
{
    if (Convert.ToInt64(comboBox2.SelectedValue)<0)
    {
        MessageBox.Show("Виберіть кімнату");
        return;
    }
    //перевірка на кількість кімнат
    string query3 = @"SELECT
Count(students.Id_student) AS kilc,
rooms.number_seats
FROM
students
Inner Join rooms ON rooms.Id_room = students.Id_room
WHERE
students.Id_room = " + Convert.ToString(comboBox2.SelectedValue) + " ";

    using (MySqlDataAdapter da_hos = new MySqlDataAdapter(query3, Form1.con))
    {
        DataTable table = new DataTable();
        table.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_hos.Fill(table);
        DataRow[] rows = table.Select();
        int kilc = Convert.ToInt16(rows[0]["kilc"]);
        int number_seats = Convert.ToInt16(rows[0]["number_seats"]);

        if (insert_mode == false && Convert.ToString(comboBox2.SelectedValue)==
Convert.ToString(dataGridView1.CurrentRow.Cells[11].Value))
        {
            kilc = kilc - 1;
        }
        if (number_seats <= kilc)
        {
            MessageBox.Show("Кімната "+ comboBox2.Text+ " вже заповнена");
            return;
        }
    }
}

```

```

    }
}
//додавання запису
//перевірка на пусу дату вибуття
string data_v;
if (dateTimePicker1.Checked)
{
    data_v = " " + dateTimePicker1.Value.ToString("yyyy-MM-dd") + " ";
}
else
{
    data_v = " Null ";
}
string query2;
if (insert_mode == true)
{
    query2 = "INSERT INTO `students`
(`pib`,`Id_room`,`Id_specialty`,`phone`,`e_mail`,`adress`,`foreigner`,`year_entry`,`date_leave`)
VALUES (";
    query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
    query2 = query2 + "" + Convert.ToString(comboBox2.SelectedValue) + ",";
    query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
    query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox3.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox4.Text) + ",";
    query2 = query2 + "" + Convert.ToString(checkBox1.Checked) + ",";
    query2 = query2 + "" + Convert.ToString(textBox5.Text) + ",";
    query2 = query2 + data_v + ")";
}
else
{
    query2 = "UPDATE `students` SET";
    query2 = query2 + "`pib`=" + Convert.ToString(textBox1.Text) + ",";
    query2 = query2 + "`Id_room`=" + Convert.ToString(comboBox2.SelectedValue) + ",";
    query2 = query2 + "`Id_specialty`=" + Convert.ToString(comboBox1.SelectedValue) +
";";
    query2 = query2 + "`phone`=" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "`e_mail`=" + Convert.ToString(textBox3.Text) + ",";

```

```

query2 = query2 + "`adress`=" + Convert.ToString(textBox4.Text) + ",";
query2 = query2 + "`foreigner`=" + Convert.ToString(checkBox1.Checked) + ",";
query2 = query2 + "`year_entry`=" + Convert.ToString(textBox5.Text) + ",";
query2 = query2 + "`date_leave`=" + data_v + " ";
query2 = query2 + " WHERE `Id_student`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "students");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
try
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox1.Text))
        {
            dataGridView1.Rows[i].Selected = true;
            break;
        }
    }
}
catch
{
    dataGridView1.Rows[0].Selected = true;
}
dataGridView1.Enabled = true;
comboBox1.Text = "";
comboBox2.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";

```

```

        textBox3.Text = "";
        textBox4.Text = "";
        textBox5.Text = "";
        insert_mode = true;
    }
private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    comboBox1.Text = "";
    comboBox2.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    dateTimePicker1.Text = "";
    insert_mode = true;
}
private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        comboBox2.SelectedValue = dataGridView1.CurrentRow.Cells[11].Value;
        comboBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        comboBox1.SelectedValue = dataGridView1.CurrentRow.Cells[10].Value;
        comboBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[4].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[5].Value);
        textBox4.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[6].Value);
        checkBox1.Checked = Convert.ToBoolean(dataGridView1.CurrentRow.Cells[7].Value);
        textBox5.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[8].Value);
        if (Convert.IsDBNull(dataGridView1.CurrentRow.Cells[9].Value)!=true)
        {
            dateTimePicker1.Checked = true;
            dateTimePicker1.Value =
Convert.ToDateTime(dataGridView1.CurrentRow.Cells[9].Value);
        }
    }
}

```

```

else
{
    dateTimePicker1.Checked = false;
}
insert_mode = false;
}
}
private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити
запис?", "Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `students` WHERE `Id_student`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "students");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            comboBox1.Text = "";
            comboBox2.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
            textBox5.Text = "";

```

```

        dateTimePicker1.Text = "";
        insert_mode = true;
    }
    catch
    {
        MessageBox.Show("Видалення неможливо. Студент має пов`язані записи",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    Form8 newForm = new Form8();
    Form8.Id_student = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    Form8.pib = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
    newForm.ShowDialog();
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    Form9 newForm = new Form9();
    Form9.Id_student = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    Form9.pib = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
    newForm.ShowDialog();
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    Form10 newForm = new Form10();
    Form10.Id_student = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    Form10.pib = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
    newForm.ShowDialog();
}

```

```

private void button6_Click(object sender, EventArgs e)
{
    insert_mode = true;
    Text = "Студенти, які проживають у кімнаті номер : " + number_room + "";
}

```

```

query = @"SELECT
students.Id_student,
students.pib,
rooms.number_room,
specialties.name,
students.phone,
students.e_mail,
students.adress,
students.foreigner,
students.year_entry,
students.date_leave,
specialties.Id_specialty,
rooms.Id_room
FROM
rooms
Inner Join students ON students.Id_room = rooms.Id_room
Inner Join specialties ON students.Id_specialty = specialties.Id_specialty
WHERE
students.pib Like '%" + textBox6.Text + "%'";
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "students");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "students";
dataGridView1.Columns[0].HeaderText = "Код студента";
dataGridView1.Columns[1].HeaderText = "ПІБ студента";
dataGridView1.Columns[2].HeaderText = "Номер кімнати";
dataGridView1.Columns[3].HeaderText = "Спеціальність";
dataGridView1.Columns[4].HeaderText = "Мобільний номер";
dataGridView1.Columns[5].HeaderText = "Електронна пошта";
dataGridView1.Columns[6].HeaderText = "Адреса";
dataGridView1.Columns[7].HeaderText = "Іноземець?";
dataGridView1.Columns[8].HeaderText = "Рік вступу";
dataGridView1.Columns[9].HeaderText = "Дата вибуття";
dataGridView1.Columns[10].HeaderText = "Id_specialty";
dataGridView1.Columns[11].HeaderText = "Id_room";
dataGridView1.Columns[10].Visible = false;
dataGridView1.Columns[11].Visible = false;

```

```

//dataGridView1.Columns[2].Visible = false;
dataGridView1.Columns[1].Width = 200;
}

private void button7_Click(object sender, EventArgs e)
{
    string fileName = "order.docx";
    string fileName2 = "order " +
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value) + ".docx";
    string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName2);
    System.IO.File.Copy(sourceFile, destFile, true);

    string filename = Environment.CurrentDirectory + @"\rep\" + fileName2;
    Microsoft.Office.Interop.Word.Application word = new
Microsoft.Office.Interop.Word.Application();
    Microsoft.Office.Interop.Word.Document doc = new
Microsoft.Office.Interop.Word.Document();
    word.Visible = false;
    doc = word.Documents.Open(filename);
    doc.Activate();
    Microsoft.Office.Interop.Word.Range rng = doc.Bookmarks["pib"].Range;
    rng.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
    Microsoft.Office.Interop.Word.Range rng2 = doc.Bookmarks["hos"].Range;
    rng2.Text = hostel_number;

    Microsoft.Office.Interop.Word.Range rng3 = doc.Bookmarks["rom"].Range;
    rng3.Text = number_room;
    Microsoft.Office.Interop.Word.Range rng4 = doc.Bookmarks["sp"].Range;
    rng4.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);

    word.Visible = true;
}

private void button8_Click(object sender, EventArgs e)
{

```

```

        Form13 newForm = new Form13();
        newForm.ShowDialog();
        Form13.Id_student = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
}
}

```

Код форми «Кімнати»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form7 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        static public string Id_hostel;
        static public string number;
        public Form7()
        {
            InitializeComponent();
        }

        private void Form7_Shown(object sender, EventArgs e)
        {
            insert_mode = true;
            Text = "Кімнати гуртожитку номер : " + number + "";
            query = @"SELECT
rooms.Id_room,
hostels.number,
rooms.number_room,
rooms.number_seats,
rooms.characteristic,
rooms.date_last_repair,
hostels.Id_hostel
FROM
rooms
Inner Join hostels ON rooms.Id_hostel = hostels.Id_hostel
WHERE
hostels.number = " + number;
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "rooms");
        }
    }
}

```

```

dataGridView1.DataSource = ds;
dataGridView1.DataMember = "rooms";
dataGridView1.Columns[0].HeaderText = "Код кімнати";
dataGridView1.Columns[1].HeaderText = "Номер гуртожитку";
//dataGridView1.Columns[2].HeaderText = "Адреса гуртожитку";
dataGridView1.Columns[2].HeaderText = "Номер кімнати";
dataGridView1.Columns[3].HeaderText = "Кількість місць";
dataGridView1.Columns[4].HeaderText = "Характеристика";
dataGridView1.Columns[5].HeaderText = "Дата останнього ремонту";
dataGridView1.Columns[6].Visible = false;
dataGridView1.Columns[0].Width = 80;
dataGridView1.Columns[1].Width = 93;
dataGridView1.Columns[2].Width = 80;
dataGridView1.Columns[3].Width = 80;
dataGridView1.Columns[4].Width = 200;
//dataGridView1.Columns[2].Width = 100;
dataGridView1.Refresh();

string query2 = @"SELECT
hostels.Id_hostel,
concat (hostels.number, ' - ',hostels.adress ) as hos
FROM
hostels
WHERE
hostels.number = " + number;

using (MySqlDataAdapter da_hos = new MySqlDataAdapter(query2, Form1.con))
{
    DataTable table = new DataTable();
    table.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_hos.Fill(table);
    comboBox1.DataSource = table;
    comboBox1.DisplayMember = table.Columns["hos"].ColumnName;
    comboBox1.ValueMember = table.Columns["Id_hostel"].ColumnName;
}
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `rooms`
(Id_hostel`,`number_room`,`number_seats`,`characteristic`,`date_last_repair`) VALUES (";
        query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + """;
    }
    else
    {
        query2 = "UPDATE `rooms` SET";
    }
}

```

```

        query2 = query2 + "`Id_hostel`=" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "`number_room`=" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "`number_seats`=" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "`characteristic`=" + Convert.ToString(textBox3.Text) + ",";
        query2 = query2 + "`date_last_repair`=" + dateTimePicker1.Value.ToString("yyyy-MM-dd")
+ ",";
        query2 = query2 + " WHERE `Id_room`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "rooms");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
    try
    {

        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if (dataGridView1.Rows[i].Cells[2].Value.ToString().Contains(textBox1.Text))
            {
                dataGridView1.Rows[i].Selected = true;

                break;
            }
        }
    }
    catch
    {
        dataGridView1.Rows[0].Selected = true;
    }
    dataGridView1.Enabled = true;
    comboBox1.Text = "";

    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    comboBox1.Enabled = true;

    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

```

```

}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        comboBox1.SelectedValue = dataGridView1.CurrentRow.Cells[6].Value;
        comboBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        textBox3.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[4].Value);
        dateTimePicker1.Value = Convert.ToDateTime(dataGridView1.CurrentRow.Cells[5].Value);

        dataGridView1.Enabled = false;
        insert_mode = false;
        comboBox1.Enabled = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `rooms` WHERE `Id_room`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "rooms");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            comboBox1.Enabled = true;

            textBox2.Text = "";
            textBox3.Text = "";
            insert_mode = true;

        }
        catch
        {
            MessageBox.Show("Видалення неможливо. Кімната має пов`язані записи", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

    }

    private void показатиСтудентівToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (dataGridView1.RowCount == 0)
        {
            MessageBox.Show("Виберіть кімнату або введіть нову", "Увага", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }

        Form6 newForm = new Form6();
        Form6.Id_room = Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        Form6.hostel_number = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        Form6.number_room = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        newForm.ShowDialog();
    }
}
}
}

```

Код форми «Нарахування»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form8 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        static public string Id_student;
        static public string pib;
        public Form8()
        {
            InitializeComponent();
        }
        private void Form8_Shown(object sender, EventArgs e)
        {
            insert_mode = true;
            Text = "Нарахування студента : " + pib + "";
            query = @"SELECT
                profit.Id_profit,
                profit.Id_student,
                profit.date_profit,
                profit.amount,
                profit.type_profit
            FROM

```

```

profit
WHERE profit.Id_student = " + Id_student;
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "profit");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "profit";
dataGridView1.Columns[0].HeaderText = "Код";
dataGridView1.Columns[1].HeaderText = "Код студента";
//dataGridView1.Columns[2].HeaderText = "ПІБ студента";
dataGridView1.Columns[2].HeaderText = "Дата нарахування";
dataGridView1.Columns[3].HeaderText = "Сума";
dataGridView1.Columns[4].HeaderText = "Тип нарахування";
dataGridView1.Columns[1].Visible = false;
dataGridView1.Columns[2].Width = 110;
dataGridView1.Columns[4].Width = 110;

string query4 = @"SELECT
students.Id_student,
students.pib
FROM
students
WHERE students.Id_student = " + Id_student;

using (MySQLDataAdapter da_rm = new MySQLDataAdapter(query4, Form1.con))
{
    DataTable table2 = new DataTable();
    table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_rm.Fill(table2);
    comboBox1.DataSource = table2;
    comboBox1.DisplayMember = table2.Columns["pib"].ColumnName;
    comboBox1.ValueMember = table2.Columns["Id_student"].ColumnName;
    comboBox1.Text = "";
}
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `profit` (`Id_student`,`date_profit`,`amount`,`type_profit`) VALUES
(";

        query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + """;

    }
    else
    {
        query2 = "UPDATE `profit` SET";
        query2 = query2 + "`Id_student` = " + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "`date_profit` = " + dateTimePicker1.Value.ToString("yyyy-MM-dd") + ",";
        query2 = query2 + "`amount` = " + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "`type_profit` = " + Convert.ToString(textBox2.Text) + """;
    }
}

```

```

        query2 = query2 + " WHERE `Id_profit`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "profit");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
    try
    {
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if (dataGridView1.Rows[i].Cells[3].Value.ToString().Contains(textBox1.Text))
            {
                dataGridView1.Rows[i].Selected = true;

                break;
            }
        }
    }
    catch
    {
        dataGridView1.Rows[0].Selected = true;
    }
    dataGridView1.Enabled = true;
    comboBox1.Enabled = true;

    textBox1.Text = "";
    textBox2.Text = "";

    insert_mode = true;
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    comboBox1.Text = "";

    textBox1.Text = "";
    textBox2.Text = "";

    insert_mode = true;
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {

```



```

}

private void Form9_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    Text = "Оплата студента : " + pib + "";
    query = @"SELECT
payment.Id_payment,
payment.Id_student,
payment.date_payment,
payment.amount,
payment.appointment
FROM
payment
WHERE payment.Id_student = " + Id_student;

    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "payment");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "payment";
    dataGridView1.Columns[0].HeaderText = "Код";
    dataGridView1.Columns[1].HeaderText = "Код студента";

    dataGridView1.Columns[2].HeaderText = "Дата оплаты";
    dataGridView1.Columns[3].HeaderText = "Сума";
    dataGridView1.Columns[4].HeaderText = "Призначення";
    dataGridView1.Columns[1].Visible = false;
    dataGridView1.Columns[2].Width = 110;
    dataGridView1.Columns[4].Width = 110;

    string query4 = @"SELECT
students.Id_student,
students.pib
FROM
students
WHERE students.Id_student = " + Id_student;

    using (MySqlDataAdapter da_rm = new MySqlDataAdapter(query4, Form1.con))
    {
        DataTable table2 = new DataTable();
        table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_rm.Fill(table2);
        comboBox1.DataSource = table2;
        comboBox1.DisplayMember = table2.Columns["pib"].ColumnName;
        comboBox1.ValueMember = table2.Columns["Id_student"].ColumnName;
        comboBox1.Text = "";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {

```

```

        query2 = "INSERT INTO `payment` (`Id_student`,`date_payment`,`amount`,`appointment`)
VALUES (";
        query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + "");
    }
    else

    {
        query2 = "UPDATE `payment` SET";
        query2 = query2 + "`Id_student`=" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "`date_payment`=" + dateTimePicker1.Value.ToString("yyyy-MM-dd") +
";";
        query2 = query2 + "`amount`=" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "`appointment`=" + Convert.ToString(textBox2.Text) + " ";
        query2 = query2 + " WHERE `Id_payment`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "payment");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
    try
    {
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if (dataGridView1.Rows[i].Cells[3].Value.ToString().Contains(textBox1.Text))
            {
                dataGridView1.Rows[i].Selected = true;

                break;
            }
        }
    }
    catch
    {
        dataGridView1.Rows[0].Selected = true;
    }
    dataGridView1.Enabled = true;
    comboBox1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    insert_mode = true;
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;

```



```

{
public partial class Form10 : Form
{
    MySqlCommand cmd;
    string query;
    static public string Id_student;
    static public string pib;
    public Form10()
    {
        InitializeComponent();
    }
    private void Form10_Shown(object sender, EventArgs e)
    {
        try
        {
            Text = "Борг студента : " + pib + "";
            query = @"SELECT
SUM(0-profit.amount) AS amount
FROM
profit
WHERE
profit.Id_student = " + Id_student + @"
UNION
SELECT
SUM(payment.amount) AS amount
FROM
payment
WHERE
payment.Id_student = " + Id_student;
            using (MySqlDataAdapter da_hos = new MySqlDataAdapter(query, Form1.con))
            {
                DataTable table = new DataTable();
                table.Locale = System.Globalization.CultureInfo.InvariantCulture;
                da_hos.Fill(table);
                DataRow[] rows = table.Select();
                int nar ;
                int opl ;

                int j = table.Rows.Count;
                if (j > 1)
                {
                    if (DBNull.Value.Equals(rows[0]["amount"]))
                    {
                        nar = 0;
                    }
                    else
                    {
                        nar = Math.Abs(Convert.ToInt16(rows[0]["amount"]));
                    }
                    if (DBNull.Value.Equals(rows[1]["amount"]))
                    {
                        opl = 0;
                    }
                    else
                    {
                        opl = Convert.ToInt16(rows[1]["amount"]);
                    }
                }
            }
        }
    }
}

```



```

visiting.time_e,
visiting.exit_entrance,
students.pib
FROM
visiting
Left Join students ON visiting.Id_student = students.Id_student
WHERE
    visiting.time_e >= "" + data_p + @" AND
    visiting.time_e <= "" + data_z + "" ORDER BY time_e DESC";
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "visiting");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "visiting";
dataGridView1.Columns[0].HeaderText = "Код відвідування";
dataGridView1.Columns[1].HeaderText = "Код студента";
dataGridView1.Columns[2].HeaderText = "Дані відвідувача";
dataGridView1.Columns[3].HeaderText = "Час";
dataGridView1.Columns[4].HeaderText = "Вхід/Вихід";
dataGridView1.Columns[5].HeaderText = "Піб студента";
dataGridView1.Columns[0].Width = 120;
dataGridView1.Columns[3].Width = 150;
dataGridView1.Columns[5].Width = 240;
dataGridView1.Columns[2].Width = 130;
dataGridView1.Refresh();
}

private void button3_Click(object sender, EventArgs e)
{
    string query2;
    string query4;
    //перевірка на кількість кімнат
    string query5 = @"SELECT
Count(students.Id_student) as kilc
FROM
students
WHERE
students.Id_student = "" + textBox1.Text + "" ";

    using (MySQLDataAdapter da_hos = new MySQLDataAdapter(query5, Form1.con))
    {
        DataTable table2 = new DataTable();
        table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_hos.Fill(table2);
        DataRow[] rows = table2.Select();
        int kilc = Convert.ToInt16(rows[0]["kilc"]);

        if (kilc == 0 && textBox2.Text == "")
        {
            MessageBox.Show("Студента з кодом "" + textBox1.Text + "" не існує");
            return;
        }

        //додавання

        string exit_entrance;

```

```

if (radioButton1.Checked)
{
    exit_entrance = "1";
}
else
{
    exit_entrance = "0";
}
DateTime data1;
data1 = DateTime.Now;
if (kilc == 0)
{
    query2 = "INSERT INTO `visiting` (`data_visitor`, `time_e`, `exit_entrance`) VALUES (";

    query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "" + data1.ToString("yyyy-MM-dd HH:mm") + ",";
    query2 = query2 + " " + exit_entrance + " )";

}
else
{
    query2 = "INSERT INTO `visiting` (`Id_student`, `data_visitor`, `time_e`, `exit_entrance`)
VALUES (";
    query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "" + data1.ToString("yyyy-MM-dd HH:mm") + ",";
    query2 = query2 + " " + exit_entrance + " )";
}
}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
//оновлюємо дані в таблиці dataGridView1
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "visiting");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
//виділення доданого запису
try
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if (dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox1.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }
    }
}
catch
{

```

```

        dataGridView1.Rows[0].Selected = true;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити запис?",
"Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `visiting` WHERE `Id_visiting`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "visiting");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            insert_mode = true;
        }
        catch
        {
            MessageBox.Show("Видалення неможливо, має пов`язані записи", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
}
}
}
}

```

Код форми «Інструктажі»

```

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;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form12 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        public Form12()
        {
            InitializeComponent();
        }
        private void Form12_Shown(object sender, EventArgs e)
        {
            insert_mode = true;
            query = @"SELECT
instruction_safety.Id_instruction,
managers.pib,
instruction_safety.date,
instruction_safety.type_instruction,
instruction_safety.Id_manager
FROM
instruction_safety
Inner Join managers ON instruction_safety.Id_manager = managers.Id_manager
ORDER BY
instruction_safety.Id_instruction DESC";

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "instruction_safety");
            dataGridView1.DataSource = ds;
            dataGridView1.DataMember = "instruction_safety";
            dataGridView1.Columns[0].HeaderText = "Код інструктажу";
```

```

dataGridView1.Columns[1].HeaderText = "ПІБ керуючого";
dataGridView1.Columns[2].HeaderText = "Дата інструктажу";
dataGridView1.Columns[3].HeaderText = "Тип інструктажу";
dataGridView1.Columns[4].Visible = false;
dataGridView1.Columns[1].Width = 200;
dataGridView1.Refresh();

string query3 = @"SELECT
managers.Id_manager,
managers.pib
FROM
managers
";
using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query3, Form1.con))
{
    DataTable table = new DataTable();
    table.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_sp.Fill(table);
    comboBox1.DataSource = table;
    comboBox1.DisplayMember = table.Columns["pib"].ColumnName;
    comboBox1.ValueMember = table.Columns["Id_manager"].ColumnName;
    comboBox1.Text = "";
}
string query5 = @"SELECT
students.Id_student,
students.pib
FROM
students
";
using (MySqlDataAdapter da_st = new MySqlDataAdapter(query5, Form1.con))
{
    DataTable table2 = new DataTable();
    table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_st.Fill(table2);
    comboBox3.DataSource = table2;
    comboBox3.DisplayMember = table2.Columns["pib"].ColumnName;
    comboBox3.ValueMember = table2.Columns["Id_student"].ColumnName;
}

```

```

        comboBox3.Text = "";
    }
}
private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `instruction_safety` (`Id_manager`,`date`,`type_instruction`)
VALUES (";

        query2 = query2 + "" + Convert.ToString(comboBox1.SelectedValue) + ",";
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + """;
    }
    else
    {
        query2 = "UPDATE `instruction_safety` SET";

        query2 = query2 + "`Id_manager`=" + Convert.ToString(comboBox1.SelectedValue) +
";";

        query2 = query2 + "`date`=" + dateTimePicker1.Value.ToString("yyyy-MM-dd") + ",";
        query2 = query2 + "`type_instruction`=" + Convert.ToString(textBox1.Text) + "" ";
        query2 = query2 + " WHERE `Id_instruction`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "instruction_safety");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
    try
    {

```

```

for (int i = 0; i < dataGridView1.RowCount; i++)
{
    dataGridView1.Rows[i].Selected = false;
    if (dataGridView1.Rows[i].Cells[3].Value.ToString().Contains(textBox1.Text))
    {
        dataGridView1.Rows[i].Selected = true;
        break;
    }
}
}
catch
{
    dataGridView1.Rows[0].Selected = true;
}
dataGridView1.Enabled = true;
comboBox1.Text = "";
textBox1.Text = "";
insert_mode = true;
}
private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    comboBox1.Text = "";
}
private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        comboBox1.SelectedValue = dataGridView1.CurrentRow.Cells[4].Value;
        comboBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        dateTimePicker1.Value =
Convert.ToDateTime(dataGridView1.CurrentRow.Cells[2].Value);
        textBox1.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

```

```

    }
}
private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити
запис?", "Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `instruction_safety` WHERE `Id_instruction`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "instruction_safety");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            comboBox1.Text = "";
            insert_mode = true;
        }
        catch
        {
            MessageBox.Show("Видалення неможливо, має пов'язані записи", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
private void dataGridView1_Click(object sender, EventArgs e)
{
    string query3;

```

```

{
    query3 = @"SELECT
students_instruction.Id_st_instruction,
students_instruction.Id_instruction,
students_instruction.Id_student,
students.pib,
students_instruction.additionally
FROM
students_instruction

Inner Join students ON students_instruction.Id_student = students.Id_student
WHERE
students_instruction.Id_instruction = " +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    MySqlDataAdapter da = new MySqlDataAdapter(query3, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "students_instruction");
    dataGridView2.DataSource = ds;
    dataGridView2.DataMember = "students_instruction";
    dataGridView2.Columns[0].HeaderText = "Код проінструктованого студента";
    dataGridView2.Columns[1].HeaderText = "Код інструктажу";
    dataGridView2.Columns[3].HeaderText = "ПІБ студента";
    dataGridView2.Columns[4].HeaderText = "Додатково";
    dataGridView2.Columns[2].Visible = false;
    dataGridView2.Columns[1].Width = 100;
    dataGridView2.Columns[3].Width = 200;
    dataGridView2.Columns[4].Width = 150;
    dataGridView2.Refresh();
}
string query4 = @"SELECT
Id_instruction
FROM
instruction_safety
WHERE Id_instruction = " + Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);

```

```

using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query4, Form1.con))
{
    DataTable table1 = new DataTable();
    table1.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_sp.Fill(table1);
    comboBox2.DataSource = table1;
    comboBox2.DisplayMember = table1.Columns["Id_instruction"].ColumnName;
    comboBox2.ValueMember = table1.Columns["Id_instruction"].ColumnName;
    comboBox2.Text = "";
}
}
private void button3_Click(object sender, EventArgs e)
{
    string query7;
    if (insert_mode == true)
    {
        query7 = "INSERT INTO `students_instruction`
(Id_instruction`,`Id_student`,`additionally`) VALUES (";

        query7 = query7 + "" + Convert.ToString(comboBox2.SelectedValue) + ",";
        query7 = query7 + "" + Convert.ToString(comboBox3.SelectedValue) + ",";
        query7 = query7 + "" + Convert.ToString(textBox2.Text) + """;
    }
    else
    {
        query7 = "UPDATE `students_instruction` SET";
        query7 = query7 + "`Id_instruction`=" + Convert.ToString(comboBox2.SelectedValue)
+ ",";
        query7 = query7 + "`Id_student`=" + Convert.ToString(comboBox3.SelectedValue) +
",";
        query7 = query7 + "`additionally`=" + Convert.ToString(textBox2.Text) + """;
        query7 = query7 + " WHERE `Id_st_instruction`=" +
Convert.ToString(dataGridView2.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query7;
    cmd.CommandType = CommandType.Text;
}

```

```

cmd.ExecuteNonQuery();
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "instruction_safety");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
dataGridView1_Click(this, e);
dataGridView1.Enabled = true;
dataGridView2.Enabled = true;
comboBox2.Text = "";
comboBox3.Text = "";
textBox2.Text = "";
insert_mode = true;
}
private void редагуватиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    if (dataGridView2.RowCount != 0)
    {
        comboBox2.SelectedValue = dataGridView2.CurrentRow.Cells[1].Value;
        comboBox2.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[1].Value);
        comboBox3.SelectedValue = dataGridView2.CurrentRow.Cells[2].Value;
        comboBox3.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[3].Value);
        textBox2.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[4].Value);
        dataGridView2.Enabled = false;
        insert_mode = false;
    }
}
private void видалитиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете видалити
запис?", "Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `students_instruction` WHERE `Id_st_instruction`=" +
Convert.ToString(dataGridView2.CurrentRow.Cells[0].Value);

```

```

cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySqlDataAdapter da = new MySqlDataAdapter(query2, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "students_instruction");
dataGridView2.DataSource = ds;
dataGridView2.Refresh();
dataGridView2.Enabled = true;
textBox2.Text = "";
comboBox2.Text = "";
comboBox3.Text = "";
insert_mode = true;
}
catch
{
    MessageBox.Show("Видалення неможливо, має пов'язані записи", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void button4_Click(object sender, EventArgs e)
{
    dataGridView2.Enabled = true;
    textBox2.Text = "";
    comboBox2.Text = "";
    comboBox3.Text = "";
}
}
}

```

Код форми «Надіслати повідомлення»

```

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;
using System.Net;
using System.Net.Mail;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace nuwmhostels
{
    public partial class Form13 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        static public string Id_student;
        public Form13()
        {
            InitializeComponent();
        }

        private void textBox1_Enter(object sender, EventArgs e)
        {
            if (textBox1.Text == "Email")
            {
                textBox1.Clear();
                textBox1.ForeColor = Color.FromArgb(0, 97, 135);
            }
        }

        private void textBox1_Leave(object sender, EventArgs e)
        {
            if (textBox1.Text == "")
            {
                textBox1.ForeColor = Color.FromArgb(200,200,200);
                textBox1.Text = "Email";
            }
        }

        private void textBox2_Enter(object sender, EventArgs e)
        {
            if (textBox2.Text == "Предмет")
            {
                textBox2.Clear();
                textBox2.ForeColor = Color.FromArgb(0, 97, 135);
            }
        }

        private void textBox2_Leave(object sender, EventArgs e)
        {
            if (textBox2.Text == "")
            {
                textBox2.ForeColor = Color.FromArgb(200, 200, 200);
                textBox2.Text = "Предмет";
            }
        }
    }
}

```

```

    }
}

private void textBox3_Enter(object sender, EventArgs e)
{
    if (textBox3.Text == "Повідомлення")
    {
        textBox3.Clear();
        textBox3.ForeColor = Color.FromArgb(0, 97, 135);
    }
}

private void textBox3_Leave(object sender, EventArgs e)
{
    if (textBox3.Text == "")
    {
        textBox3.ForeColor = Color.FromArgb(200, 200, 200);
        textBox3.Text = "Повідомлення";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    string to, from, pass, messageBody;
    MailMessage message = new MailMessage();
    to = textBox1.Text;
    from = "managers.nuwm.hostels@gmail.com";
    pass = "nuwmhostels2021";
    messageBody = textBox3.Text;
    message.To.Add(to);
    message.From = new MailAddress(from);
    message.Body = "Доброго дня! " + "<br>Ви маєте борг у розмірі: " + messageBody;
    message.Subject = textBox2.Text;
    message.IsBodyHtml = true;
    SmtplibClient smtp = new SmtplibClient("smtp.gmail.com");
    smtp.EnableSsl = true;
    smtp.Port = 587;
    smtp.DeliveryMethod = SmtplibDeliveryMethod.Network;
    smtp.Credentials = new NetworkCredential(from, pass);
    try
    {
        smtp.Send(message);
        DialogResult code = MessageBox.Show("Повідомлення надіслано успішно",
"Повідомлення надіслано", MessageBoxButtons.OK, MessageBoxIcon.Information);
        if (code == DialogResult.OK)
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

    }
    private void label2_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
}

```

Код форми «Вхід у систему»

```

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 nuwmhostels
{
    public partial class Form14 : Form
    {
        public Form14()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form1.con.Close();
            string user_name = textBox1.Text;
            string user_password = textBox2.Text;
            string user_connection_string;
            string user_server = textBox3.Text;
            if (user_password == "")
            {
                user_connection_string = "server=" + user_server + ";User Id=" + user_name +
                    ";Persist Security Info=True;database=nuwmhostels";
            }
            else
            {
                user_connection_string = "server=" + user_server + ";User Id=" + user_name +
                    ";password=" + user_password + ";Persist Security Info=True;database=nuwmhostels";
            }
            try
            {
                Form1.con.ConnectionString = user_connection_string;
                Form1.con.Open();
                MessageBox.Show("Підключення відбулось успішно", "Підключення",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
                this.Close();
            }
            catch
            {

```

```
        MessageBox.Show("Помилка при підключенні", "Увага", MessageBoxButtons.OK,  
        MessageBoxIcon.Question);  
    }  
}  
}
```

Код форми «Розробник інформаційної системи»

```
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 nuwmhostels  
{  
    public partial class Form15 : Form  
    {  
        public Form15()  
        {  
            InitializeComponent();  
        }  
    }  
}
```