

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ
«КОМП'ЮТЕРНІ НАУКИ»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня фаховий молодший бакалавр
за спеціальністю 122 Комп'ютерні науки

на тему:

**«Інформаційна система обліку проживання
студентів у гуртожитку»**

Виконав здобувач освіти групи П-43стн
Бригалеви́ч Руслан Микола́йович

Керівник роботи:
Студзі́нський Володи́мир Ві́кторович

Рецензент:

Зміст

Вступ	6
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	8
1.1 Основні задачі	8
1.2 Огляд існуючих рішень	9
1.3 Функціональні вимоги до програми	10
1.4 Вибір технології	11
1.5 Постановка задачі	12
Висновки до розділу 1	13
Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ	15
2.1 Модель бази даних	15
2.2 Список основних форм та їх призначення	21
2.3 Архітектура програмного засобу	27
2.4 Програмна реалізація	28
2.5 Тестування програми	30
Висновки до розділу 2	31
Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ	33
3.1 Мінімальні вимоги до системи	33
3.2 Інтерфейс користувача	34
3.3 Інструкція для роботи з програмою	36
Висновки до розділу 3	38
ВИСНОВКИ	39
Перелік літературних джерел	41
Додаток А. Код для роботи з базою даних	44
Додаток Б. Програмний код модулів	46

					КРФМБ.122.043стн.017.2026.ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система обліку проживання студентів у гуртожитку			Літ.	Арк.	Аркуші		
Розробив		Бригалевиц Р М								3	51	
Перевірів		Студзінський В В										
Рецензент		.						ЖАТФК				
Н. Контр.												
Затвердив.		Гнатюк О.Ф.										

РЕФЕРАТ

Записка: 51 стор., 10 рис., 7 таблиць, 2 додатки, 20 джерел.

Об'єкт дослідження — процеси адміністративного та фінансового обліку в закладах вищої освіти. Предмет дослідження — автоматизація реєстрації мешканців, контролю заповненості житлового фонду та обліку оплат за проживання в гуртожитку.

Мета роботи — розробка кросплатформної інформаційної системи для оптимізації роботи персоналу гуртожитку та підвищення точності фінансового контролю.

В ході виконання роботи проведено аналіз предметної галузі та обґрунтовано вибір технологічного стека: мови C#, платформи .NET 9, графічної системи WPF та СУБД SQLite. Спроековано реляційну базу даних та архітектуру додатка на основі патерну MVVM, що забезпечує гнучкість та стабільність програмного продукту.

Результатом розробки є програмне забезпечення, що дозволяє вести реєстр студентів, керувати статусами кімнат, реєструвати поселення та виселення, а також автоматично розраховувати заборгованість. Реалізовано модуль генерації звітів у форматі PDF, що значно скорочує час на підготовку паперової документації. Тестування підтвердило надійність алгоритмів та відповідність інтерфейсу сучасним вимогам UX/UI.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІК ПРОЖИВАННЯ, .NET 9, WPF, SQLITE, MVVM, АВТОМАТИЗАЦІЯ, ГУРТОЖИТОК, PDF-ЗВІТНІСТЬ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД — база даних;

ЗВО — заклад вищої освіти;

ІС — інформаційна система;

ПЗ — програмне забезпечення;

СУБД — система управління базами даних;

API (Application Programming Interface) — прикладний програмний інтерфейс;

CRUD (Create, Read, Update, Delete) — чотири базові операції для роботи з даними: створення, читання, оновлення та видалення;

DI (Dependency Injection) — ін'єкція залежностей, патерн проектування для забезпечення слабкої пов'язаності компонентів;

ER-діаграма (Entity-Relationship Diagram) — діаграма «сутність-зв'язок», що використовується для проектування моделей даних;

MVVM (Model-View-ViewModel) — архітектурний патерн, що розділяє обробку даних, інтерфейс користувача та логіку представлення;

PDF (Portable Document Format) — міжплатформний формат електронних документів;

POCO (Plain Old CLR Object) — простий об'єктний клас, що не залежить від специфічних фреймворків;

UI (User Interface) — інтерфейс користувача;

WPF (Windows Presentation Foundation) — система для побудови графічних інтерфейсів користувача в Windows-додатках;

XAML (Extensible Application Markup Language) — декларативна мова розмітки, що використовується в WPF для опису інтерфейсу користувача.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ВСТУП

Актуальність теми дослідження зумовлена необхідністю цифровізації адміністративних процесів у закладах вищої освіти, зокрема в частині управління житловим фондом гуртожитків. Традиційні методи паперового обліку мешканців є малоефективними, оскільки призводять до дублювання даних та виникнення помилок при розрахунках. Створення спеціалізованої інформаційної системи дозволяє автоматизувати рутинні операції комендантів та бухгалтерів, забезпечуючи високу швидкість обробки запитів.

Об'єктом дослідження є процеси адміністрування поселення, виселення та фінансового обліку проживання студентів у гуртожитку. Предметом дослідження є розробка програмного забезпечення для автоматизації цих процесів з використанням сучасних засобів об'єктно-орієнтованого програмування. Метою роботи є проектування та реалізація інформаційної системи, що забезпечує надійне збереження даних та зручний інтерфейс для персоналу.

Для досягнення поставленої мети було обрано платформу.NET 9, яка забезпечує високу продуктивність та підтримку сучасних мовних конструкцій C#. Як технологія побудови інтерфейсу обрана WPF, що дозволяє реалізувати багатий графічний функціонал та розділити логіку і представлення за допомогою патерну MVVM. Використання бази даних SQLite є раціональним рішенням для локальних систем, оскільки вона не потребує окремого сервера та гарантує цілісність даних.

Система передбачає розмежування прав доступу за ролями: адміністратор, комендант та бухгалтер. Комендант отримує інструменти для візуального моніторингу заповненості кімнат та оперативного поселення нових мешканців. Бухгалтерський модуль системи дозволяє автоматизувати нарахування плати за проживання згідно з діючими тарифами. Важливою особливістю системи є механізм перевірки заборгованості, який блокує виселення студента до повного погашення фінансових зобов'язань.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Програмний продукт реалізує можливість формування деталізованих PDF-звітів, що значно полегшує підготовку статистичної документації. Архітектура системи побудована на принципах модульності, що забезпечує легкість масштабування та внесення змін у майбутньому. Впровадження такої системи мінімізує вплив людського фактора на коректність фінансових нарахунків. Дашборд головного вікна надає користувачу миттєвий доступ до ключових показників: кількості вільних місць та загальної суми боргу.

Наукова новизна роботи полягає в оптимізації алгоритму розрахунку помісячного балансу студента з урахуванням динамічної зміни цін. Практичне значення отриманих результатів полягає у створенні готового до експлуатації інструменту, що підвищує прозорість обліку. Кваліфікаційна робота містить опис усіх етапів життєвого циклу розробки — від аналізу вимог до тестування готового продукту. Результатом реалізації проекту є підвищення ефективності управління ресурсами гуртожитку та покращення якості обслуговування студентів.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні задачі

Автоматизація обліку в студентському гуртожитку охоплює широке коло адміністративних та господарських процесів. Першочерговою задачею є ведення детального реєстру студентів із фіксацією їхніх персональних даних та поточної контактної інформації. Система має забезпечувати оперативне управління житловим фондом через облік вільних та зайнятих місць у кожній конкретній кімнаті. Важливою функцією є реєстрація поселення студентів із прив'язкою до номерів наказів та визначенням термінів проживання. Також програма повинна контролювати процес виселення та звільнення інвентарю мешканцями.

Ефективна робота закладу неможлива без автоматизації фінансових нарахувань за проживання згідно з установленими тарифами. Програмний засіб має підтримувати облік фактичних оплат від студентів та автоматично розраховувати поточний стан їхнього балансу. Особлива увага приділяється виявленню заборгованостей та формуванню списків боржників для проведення відповідної виховної роботи. Система повинна генерувати різноманітні звіти про заповненість поверхів та рух мешканців за обраний календарний період.

Важливою задачею є розмежування прав доступу персоналу для захисту персональних даних від несанкціонованого перегляду. Крім того програмний продукт має зберігати історію проживання та платежів у спеціальному архіві для проведення аналітики за минулі роки. Інформаційна система повинна спрощувати комунікацію між комендантом та бухгалтерією через створення єдиного інформаційного простору. Усі ці задачі спрямовані на підвищення прозорості управління гуртожитком та зменшення навантаження на адміністративний персонал.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1.2 Огляд існуючих рішень

На сучасному ринку програмного забезпечення представлено кілька типів систем для управління житловим фондом, які різняться за функціональністю та ціною політикою. Нижче наведено порівняльний аналіз найбільш розповсюджених рішень у контексті їхнього застосування для українських гуртожитків.

1. Парус-Студентське містечко

Вартість від 15 000 до 40 000 грн залежно від кількості робочих місць та модулів. Повна адаптація до українського законодавства та бухгалтерських стандартів, інтеграція з іншими продуктами лінійки «Парус».

Головні недоліки це висока складність інтерфейсу, що потребує тривалого навчання персоналу, а також надмірна функціональність для невеликих установ.

2. 1С:Підприємство 8. Управління гуртожитком

Вартість від 25 000 грн за базову ліцензію плюс витрати на щорічне обслуговування. Гнучкість налаштувань під індивідуальні потреби закладу та наявність потужних інструментів для автоматизації звітності. Головні недоліки це значні системні вимоги до комп'ютерного обладнання, висока вартість супроводу фахівцями та складність самостійної підтримки бази даних.

3. Bitrix24

Вартість від 1 500 грн на місяць за комерційний тарифний план. Сучасний веб-інтерфейс, можливість доступу з мобільних пристроїв та зручні вбудовані інструменти для комунікації з мешканцями. Головні недоліки це відсутність специфічних функцій для освітніх закладів, як-от облік за факультетами, та складність реалізації індивідуальних фінансових алгоритмів.

4. Облік у хмарі

Вартість від 500 до 2 000 грн на місяць за умови передплати. Не потребує встановлення на локальні робочі станції та забезпечує автоматичне резервне копіювання інформації. Головні недоліки це повна залежність від стабільності

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

інтернет-з'єднання та значні ризики щодо конфіденційності збереження персональних даних студентів на сторонніх ресурсах.

5. Microsoft Excel

Вартість входить до стандартного пакету офісних програм. Відсутність додаткових витрат на придбання та загальна обізнаність персоналу з основними принципами роботи в таблицях. Головні недоліки це неможливість одночасної роботи багатьох користувачів, високий ризик випадкової втрати даних та повна відсутність автоматичного контролю заборгованостей.

Аналіз показав, що жодне з наведених рішень не є оптимальним для малого або середнього гуртожитку України. Професійні системи є занадто дорогими та складними в експлуатації для звичайного персоналу, тоді як універсальні хмарні сервіси не враховують специфіку внутрішніх наказів закладу. Саме тому виникає потреба у створенні власної локальної системи, яка поєднає простоту інтерфейсу, відсутність абонентської плати та надійне збереження даних на комп'ютері установи.

1.3 Функціональні вимоги до програми

На основі аналізу бізнес-процесів сформовано наступний перелік функціональних вимог до розроблюваної системи:

1. Авторизація користувачів із розмежуванням доступу для ролей «Адміністратор», «Комендант» та «Бухгалтер».
2. Ведення реєстру студентів із можливістю редагування персональних даних, контактів та статі.
3. Швидкий пошук та фільтрація списку студентів за ПІБ, групою або факультетом.
4. Облік кімнат із зазначенням поверху, типу, загальної місткості та автоматичним підрахунком вільних місць.
5. Візуальна індикація стану кімнат (вільна, частково зайнята, заповнена) за допомогою колірних маркерів.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

6. Реєстрація поселення студентів із перевіркою статі мешканця відповідно до типу обраної кімнати.
7. Проведення виселення з обов'язковою автоматичною перевіркою відсутності фінансової заборгованості.
8. Облік фінансових платежів із можливістю вказання типу оплати та додавання приміток.
9. Автоматичний розрахунок балансу студента (нарахування мінус оплати) на основі діючих тарифів.
10. Ведення історії тарифів (цін) із прив'язкою до дати початку їхньої дії.
11. Формування та експорт у формат PDF звітів про боржників, заповненість гуртожитку та рух студентів.
12. Перегляд повної архівної історії проживання та фінансової активності кожного студента.

1.4 Вибір технології

Для розробки інформаційної системи обліку проживання студентів було обрано сучасний технологічний стек, який забезпечує баланс між швидкістю розробки, продуктивністю та зручністю підтримки.

Технологія WPF (Windows Presentation Foundation) обрана як основний засіб побудови графічного інтерфейсу користувача. Вона дозволяє реалізувати декларативний підхід до створення UI за допомогою мови XAML, що чітко відокремлює зовнішній вигляд від бізнес-логіки. Завдяки потужній системі зв'язування даних (Data Binding), WPF ідеально підходить для реалізації патерну MVVM, який використано в архітектурі системи. Це забезпечує легке тестування коду та можливість гнучкого налаштування візуальних елементів, таких як картки кімнат із кольоровою індикацією або динамічні панелі редагування.

Мова програмування C# у поєднанні з платформою .NET 9 є оптимальним вибором для створення надійних десктопних додатків. Вона надає широкі можливості для асинхронного програмування, що критично

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

важливо для збереження відгуку інтерфейсу під час виконання запитів до бази даних або генерації PDF-звітів..NET 9 пропонує покращену продуктивність та доступ до сучасних бібліотек, що дозволяє швидко реалізувати складні алгоритми розрахунку фінансових балансів. Суворі типізація мови мінімізує кількість помилок на етапі розробки та полегшує подальший супровід програмного продукту.

Як систему керування базами даних обрано SQLite, що є безкоштовним рішенням з відкритим кодом. Основною перевагою SQLite для даного проекту є її портативність, оскільки вся база даних зберігається в одному локальному файлі без необхідності встановлення окремого сервера. Це спрощує розгортання системи в умовах гуртожитку та гарантує високу швидкість доступу до даних при помірних обсягах інформації. SQLite повністю підтримує транзакційність, що забезпечує цілісність фінансових записів та персональних даних студентів.

1.5 Постановка задачі

Метою даної кваліфікаційної роботи є розробка локальної інформаційної системи для автоматизації обліку мешканців студентського гуртожитку. Система повинна забезпечувати надійне збереження даних про студентів, кімнати, поселення та платежі в єдиній базі даних. Програмний продукт має бути реалізований як Windows-додаток із сучасним графічним інтерфейсом, що відповідає вимогам ергономіки та зручності використання.

Основними завданнями розробки є:

- Проектування архітектури бази даних для збереження взаємопов'язаної інформації про житловий фонд та фінансові операції.
- Реалізація модулів реєстрації поселення та виселення з автоматичним контролем заповненості місць та наявності боргів.
- Розробка механізму автоматичного нарахування оплати за проживання на основі гнучких тарифів.
- Створення інструментів для генерації та експорту статистичної звітності у форматі PDF для адміністрації закладу.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Забезпечення безпеки даних через систему авторизації та розмежування прав доступу між персоналом.

Кінцевий продукт повинен стати ефективною заміною паперовому обліку та застарілим табличним методам, підвищуючи загальну швидкість та якість адміністрування гуртожитку.

Висновки до розділу 1

У першому розділі було проведено комплексний аналіз предметної галузі та обґрунтовано необхідність автоматизації процесів обліку в студентському гуртожитку. На основі проведеного дослідження можна зробити наступні висновки:

1. Визначено, що основними задачами управління гуртожитком є ведення реєстру мешканців, контроль заповненості житлового фонду, автоматизація фінансових нарахувань та формування звітності для адміністрації.
2. Аналіз існуючих програмних рішень, таких як «Парус», «1С» та хмарні CRM-системи, показав, що вони або мають занадто високу вартість і складність для впровадження в окремих установах, або не враховують специфіку українських освітніх стандартів.
3. Сформовано перелік із 12 ключових функціональних вимог, серед яких критично важливими є розмежування прав доступу за ролями, візуальний моніторинг стану кімнат та блокування виселення студентів при наявності фінансової заборгованості.
4. Обґрунтовано вибір технологічного стека, що базується на використанні платформи.NET 9 та мови C# для забезпечення високої продуктивності, технології WPF для створення сучасного інтерфейсу та СУБД SQLite для забезпечення локального і надійного збереження даних без необхідності серверної інфраструктури.
5. Поставлено задачу на розробку інформаційної системи, яка має стати ефективною заміною паперовому обліку, забезпечити цілісність даних

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

та підвищити швидкість обслуговування студентів персоналом гуртожитку.

Результати аналізу, отримані в цьому розділі, слугуватимуть фундаментом для подальшого проектування архітектури бази даних та програмних модулів системи у наступних етапах розробки.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

2.1 Модель бази даних

2.1.1 Опис предметної галузі та основних сутностей

Предметна галузь охоплює процеси адміністративного та фінансового управління житловим фондом студентського гуртожитку. Основними об'єктами обліку є мешканці (студенти), приміщення (кімнати), факти перебування (поселення) та грошові транзакції (платежі). База даних повинна забезпечувати цілісне зберігання інформації про кожного студента, його поточне місце проживання, історію переміщень та стан взаєморозрахунків з установою.

Для реалізації функціональних вимог системи було виокремлено такі ключові сутності:

- **Студент (Student):** містить персональні та контактні дані мешканців.
- **Кімната (Room):** описує характеристики житлової площі та її місткість.
- **Поселення (Settlement):** фіксує юридичний факт проживання конкретної особи в певній кімнаті протягом визначеного часу.
- **Платіж (Payment):** відображає фінансові надходження від студентів за послуги проживання.
- **Налаштування ціни (PriceSetting):** зберігає історію змін тарифів для автоматизації нарахувань.
- **Користувач (User):** забезпечує авторизацію та розмежування прав доступу персоналу.

2.1.2 ER-діаграма

Структура розробленої ER-діаграми (**Entity-Relationship Diagram**) відображає логічні зв'язки між фізичними та абстрактними об'єктами системи. Модель побудована таким чином, щоб мінімізувати дублювання інформації та забезпечити автоматичне оновлення пов'язаних даних.

У моделі визначено шість основних сутностей: Student, Room, Settlement, Payment, PriceSetting та User.

Типи зв'язків між сутностями:

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

1. **Student — Settlement (1:N)**: один студент може мати кілька записів про поселення протягом усього терміну навчання (наприклад, при переїзді в іншу кімнату), проте кожен запис належить лише одному студенту.
2. **Room — Settlement (1:N)**: одна кімната може бути пов'язана з багатьма записами про поселення (різні студенти в різний час), але конкретне поселення прив'язане до однієї кімнати.
3. **Student — Payment (1:N)**: один студент здійснює багато платежів, кожен з яких ідентифікується як такий, що належить саме цій особі.

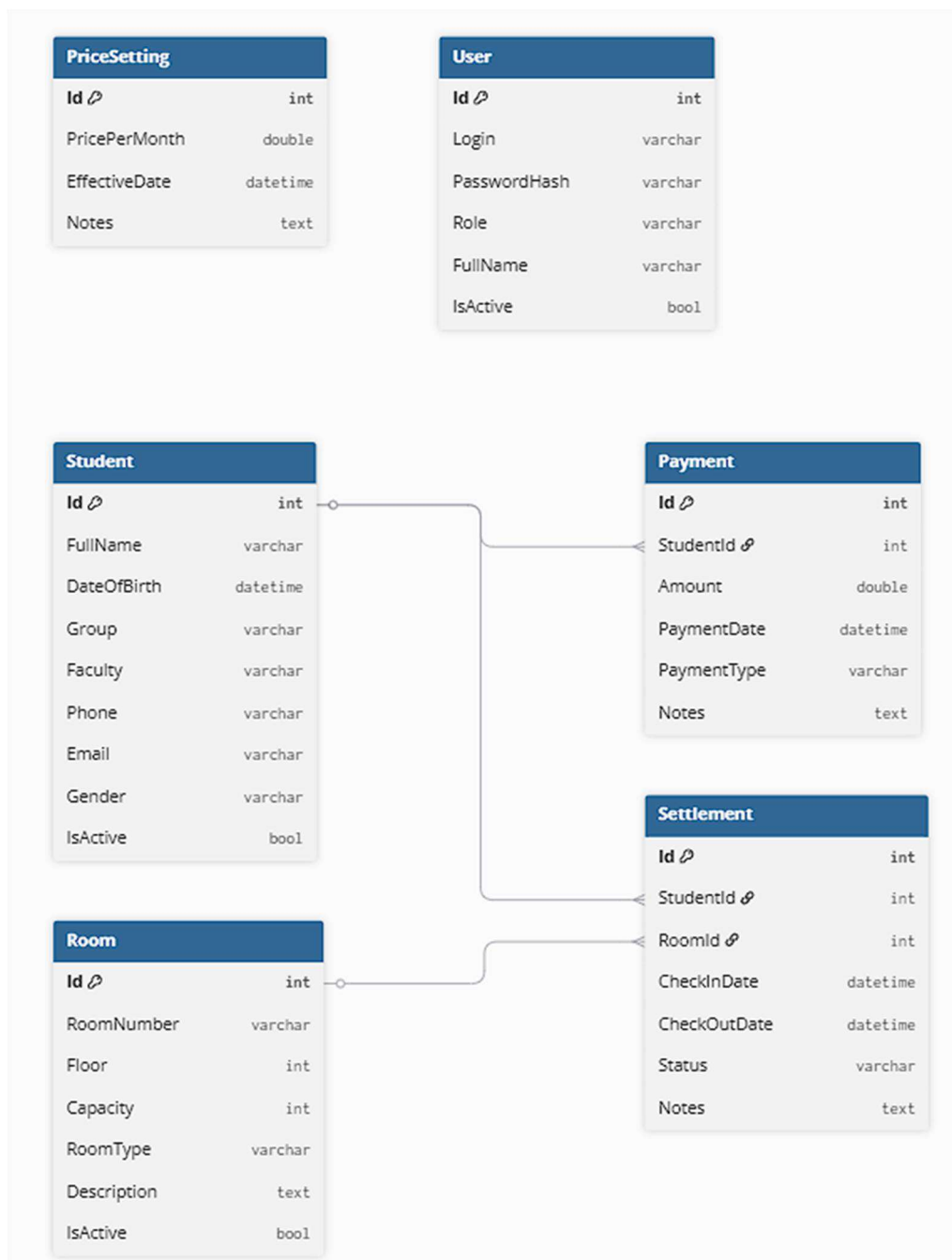


Рисунок 2.1.1. – ER-діаграма

Нижче наведено детальну класифікацію компонентів моделі даних:

Сутності та атрибути:

1. Студент (Student):

Атрибути: Id (первинний ключ), FullName, DateOfBirth, Group, Faculty, Phone, Email, Gender, IsActive.

Класифікація: атрибути FullName та Phone є простими, однозначними; атрибут IsActive — логічний.

2. Кімната (Room):

Атрибути: Id (первинний ключ), RoomNumber, Floor, Capacity, RoomType, Description, IsActive.

Класифікація: атрибути RoomNumber та Floor є простими; RoomType — класифікаційний атрибут.

3. Поселення (Settlement):

Атрибути: Id (первинний ключ), StudentId (зовнішній ключ), RoomId (зовнішній ключ), CheckInDate, CheckOutDate, Status, Notes.

Класифікація: дати є складними атрибутами; Status — перелічуваний.

4. Платіж (Payment):

Атрибути: Id (первинний ключ), StudentId (зовнішній ключ), Amount, PaymentDate, PaymentType, Notes.

Класифікація: Amount — числовий атрибут (подвійної точності).

Зв'язки:

«Проживання» (Student — Settlement): тип 1:N. Студент може заселятися декілька разів за історію обліку.

«Розміщення» (Room — Settlement): тип 1:N. Кімната містить історію всіх поселених у неї осіб.

«Здійснення оплати» (Student — Payment): тип 1:N. Студент генерує множину фінансових транзакцій.

Ця модель забезпечує цілісність даних через використання механізмів зовнішніх ключів (Foreign Keys), що унеможливорює видалення студента або кімнати, якщо до них прив'язані активні поселення або платежі. Уникнення

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

надмірності (нормованість) досягається за рахунок того, що персональні дані студента зберігаються лише в одній таблиці, а в інших таблицях використовуються посилання на його ідентифікатор (StudentId). Наприклад, при зміні прізвища студента, інформація автоматично оновиться у всіх звітах, оскільки вони посилаються на первинний ключ таблиці Student.

2.1.3 Фізична модель бази даних у SQLite

Фізична модель бази даних визначає конкретну реалізацію структури зберігання даних у середовищі СУБД SQLite. Враховуючи особливості обраної технології, типи даних, такі як DateTime та bool, трансформуються у TEXT та INTEGER відповідно. Нижче наведено детальний опис кожної таблиці системи.

Таблиця 2.1.3.1. Структура таблиці Students (Студенти)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор студента	PK (Autoinc)
FullName	TEXT	Прізвище, ім'я та по батькові	NOT NULL
DateOfBirth	TEXT	Дата народження мешканця	NOT NULL
Group	TEXT	Академічна група студента	NOT NULL
Faculty	TEXT	Факультет навчання	NOT NULL
Phone	TEXT	Контактний номер телефону	NOT NULL
Email	TEXT	Адреса електронної пошти	NOT NULL
Gender	TEXT	Стать (Чоловіча / Жіноча)	NOT NULL
IsActive	INTEGER	Статус (1 — активний, 0 — видалений)	NOT NULL

Таблиця 2.1.3.2. Структура таблиці Rooms (Кімнати)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор кімнати	PK (Autoinc)
RoomNumber	TEXT	Номер кімнати	NOT NULL
Floor	INTEGER	Поверх розміщення	NOT NULL
Capacity	INTEGER	Загальна кількість місць у кімнаті	NOT NULL
RoomType	TEXT	Тип кімнати (Чоловіча / Жіноча / Змішана)	NOT NULL
Description	TEXT	Додатковий опис приміщення	NOT NULL
IsActive	INTEGER	Статус активності кімнати	NOT NULL

Таблиця 2.1.3.3. Структура таблиці Settlements (Поселення)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор поселення	PK (Autoinc)
StudentId	INTEGER	Ідентифікатор студента	FK (Students.Id)
RoomId	INTEGER	Ідентифікатор кімнати	FK (Rooms.Id)
CheckInDate	TEXT	Дата заселення	NOT NULL
CheckOutDate	TEXT	Дата фактичного виселення	NULL
Status	TEXT	Статус (Активне / Завершено)	NOT NULL
Notes	TEXT	Примітки до поселення	NOT NULL

Таблиця 2.1.3.4. Структура таблиці Payments (Платежі)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор платежу	PK (Autoinc)
StudentId	INTEGER	Ідентифікатор студента, що платить	FK (Students.Id)
Amount	REAL	Сума грошового переказу	NOT NULL
PaymentDate	TEXT	Дата здійснення оплати	NOT NULL
PaymentType	TEXT	Тип платежу (готівка, термінал тощо)	NOT NULL
Notes	TEXT	Додаткові примітки	NOT NULL

Таблиця 2.1.3.5. Структура таблиці PriceSettings (Налаштування ціни)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор запису	PK (Autoinc)
PricePerMonth	REAL	Вартість проживання за місяць	NOT NULL
EffectiveDate	TEXT	Дата, з якої діє вказана ціна	NOT NULL
Notes	TEXT	Опис або обґрунтування ціни	NOT NULL

Таблиця 2.1.3.6. Структура таблиці Users (Користувачі системи)

Назва поля	Тип даних SQLite	Опис поля	Ключ
Id	INTEGER	Унікальний ідентифікатор користувача	PK (Autoinc)
Login	TEXT	Логін для входу в систему	NOT NULL
PasswordHash	TEXT	Хешований пароль доступу	NOT NULL
Role	TEXT	Роль (Адмін / Комендант / Бухгалтер)	NOT NULL
FullName	TEXT	Повне ім'я співробітника	NOT NULL
IsActive	INTEGER	Стан облікового запису	NOT NULL

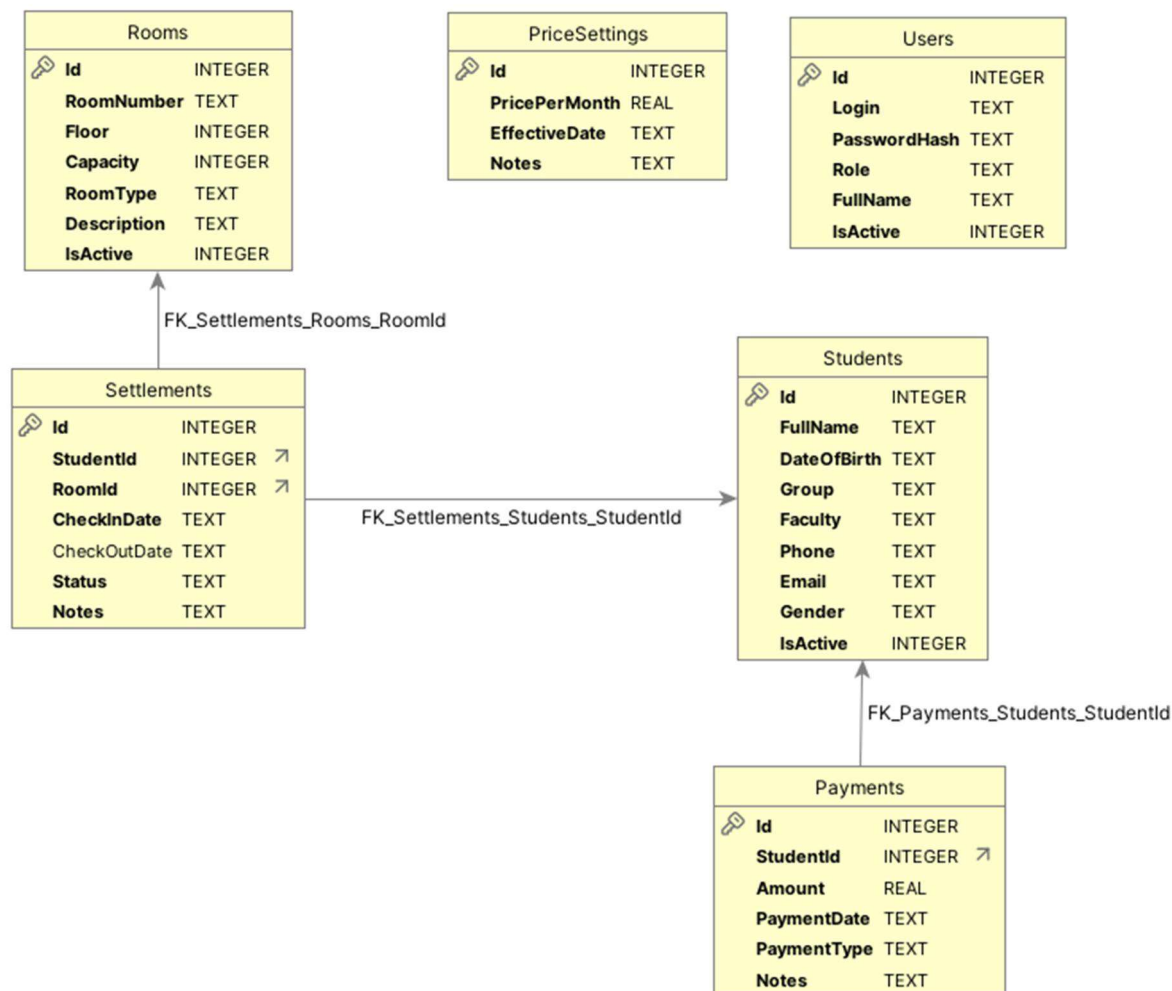


Рисунок 2.1.2. – Схема бази даних

Таблиця 2.1.3.7. Структура таблиць бази даних

Таблиця	Опис призначення	Зв'язки
Students	Зберігання персональних даних мешканців	One-to-Many до Settlements, Payments
Rooms	Дані про житловий фонд гуртожитку	One-to-Many до Settlements
Settlements	Журнал поселень та виселень	Пов'язує Students та Rooms
Payments	Облік фінансових надходжень	Many-to-One до Students
PriceSettings	Історія тарифних планів	Незалежна (використовується сервісами)
Users	Адміністрування доступу співробітників	Внутрішня таблиця авторизації

2.2 Список основних форм та їх призначення

Графічний інтерфейс інформаційної системи побудований за принципом модульності, де кожна форма відповідає за конкретний аспект життєвого циклу студента в гуртожитку. Форма авторизації є першим етапом взаємодії користувача з програмою, забезпечуючи захищений вхід до системи. Вона вимагає введення текстового логіна та пароля, які перевіряються на відповідність у базі даних, після чого встановлюється сесія з певним рівнем доступу.

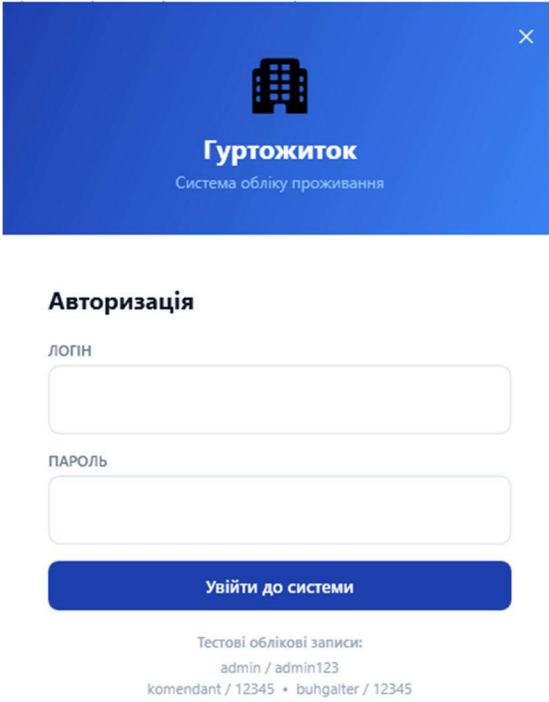


Рисунок 2.2.1. – Форма авторизації

Головне вікно системи виконує роль центрального вузла управління, де за допомогою навігаційної панелі користувач перемикається між функціональними розділами, а дашборд відображає зведену статистику про кількість вільних місць та загальний стан фінансової заборгованості у реальному часі.

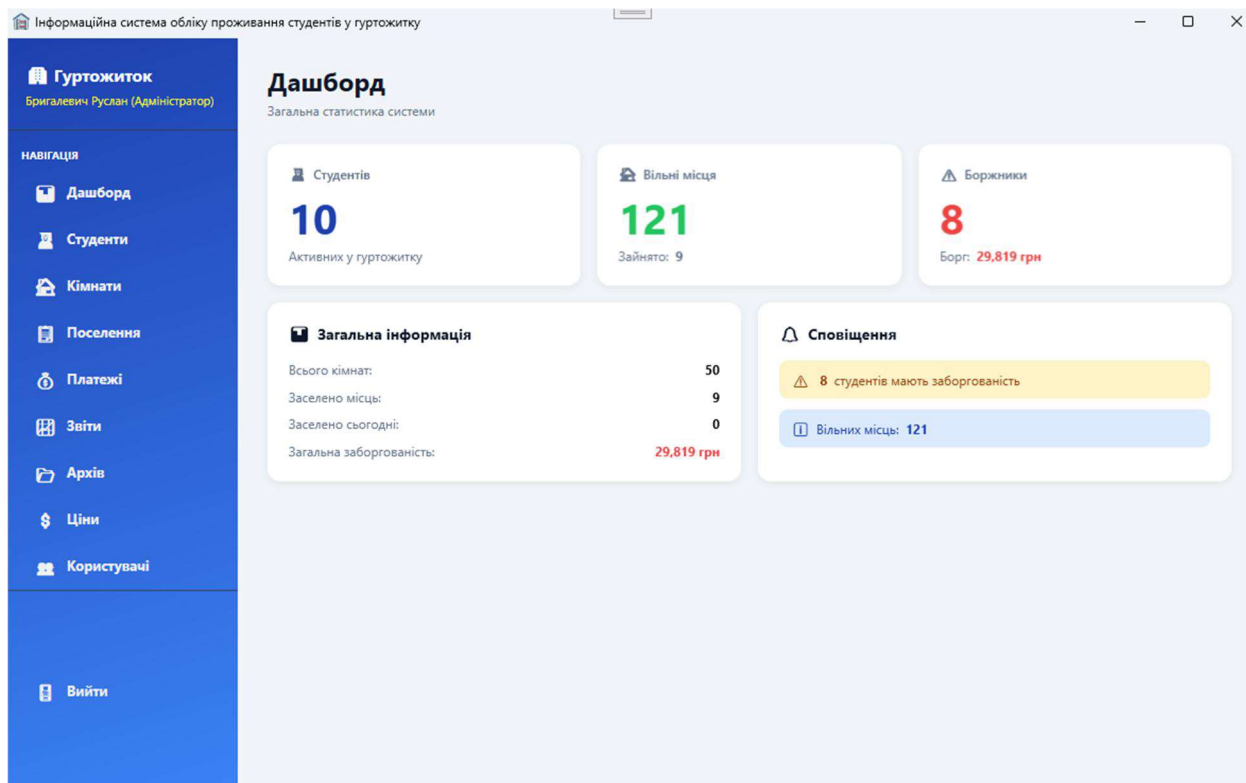


Рисунок 2.2.2. – Головне вікно

Розділ управління студентами призначений для ведення детального електронного реєстру мешканців. Форма введення даних студента передбачає заповнення текстових полів для повного імені, назви академічної групи та факультету, а також вибір статі зі спадного списку. Для забезпечення контактного зв'язку вводяться номер телефону у числовому форматі та адреса електронної пошти, а вибір дати народження здійснюється через графічний календарний інтерфейс. Кожна картка студента підтримує статус активності, що дозволяє зберігати інформацію про виселених осіб без їх остаточного видалення з бази даних.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Інформаційна система обліку проживання студентів у гуртожитку

Гуртожиток

Бригалевиц Руслан (Адміністратор)

НАВІГАЦІЯ

Дашборд

Студенти

Кімнати

Поселення

Платежі

Звіти

Архів

Ціни

Користувачі

Вийти

Студенти

Облік студентів гуртожитку

+ Додати студента

ПІБ	Група	Факультет	Телефон	Стать	Дії
Іваненко Олексій Миколі	КН-21	Факультет ін	0501234567	Чоловіча	 
Бондаренко Олена Сергії	ФЛ-21	Філологічні	0671234571	Жіноча	 
Гриценко Ірина Василівн	КН-21	Факультет ін	0671234574	Жіноча	 
Коваленко Марія Іванівн	ФЛ-21	Філологічні	0501234570	Жіноча	 
Кравченко Максим Юрій	МА-22	Математичн	0631234575	Чоловіча	 
Лисенко Василь Петрови	ЕК-22	Економічні	0501234573	Чоловіча	 
Мельник Тетяна Олексан	БІ-22	Біологічний	0631234572	Жіноча	 
Петренко Дмитро Олексі	КН-21	Факультет ін	0671234568	Чоловіча	 
Сидоренко Андрій Васил	ЕК-22	Економічні	0631234569	Чоловіча	 
Шевченко Наталія Олексі	МА-22	Математичн	0501234576	Жіноча	 

Новий студент

ПІБ *

ДАТА НАРОДЖЕННЯ

28.03.2008

СТАТЬ

Чоловіча

ГРУПА

ФАКУЛЬТЕТ

ТЕЛЕФОН

EMAIL

Зберегти

Скасувати

Рисунок 2.2.3. – Управління студентами

Модуль управління кімнатами реалізований у вигляді сітки візуальних карток, кожна з яких відображає номер приміщення, поверх та тип заселення. Форма редагування кімнати потребує введення цілочисельного значення поверху та максимальної місткості, що є критично важливим для контролю щільності поселення. Користувач також вказує гендерне призначення кімнати (чоловіча, жіноча або змішана), що дозволяє системі автоматично фільтрувати доступні варіанти під час реєстрації нового мешканця. Візуальна індикація стану кожної кімнати змінюється динамічно залежно від співвідношення фактично зайнятих місць до загальної місткості.

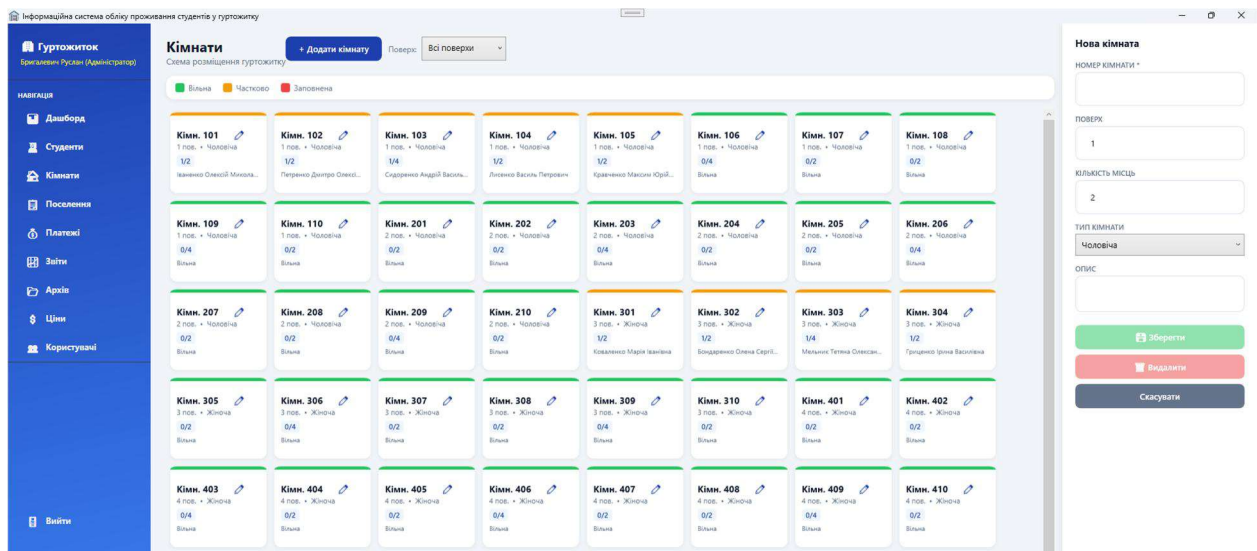


Рисунок 2.2.4. – Управління кімнатами

Форма поселення та виселення є основним інструментом коменданта для реєстрації руху студентів. При створенні нового запису користувач обирає студента та кімнату із запропонованих списків, які автоматично відфільтровані за критеріями активності та статі. Обов'язковим для заповнення є поле дати заселення, тоді як дата виселення залишається порожньою до моменту фактичного завершення проживання. Система блокує функцію виселення через спеціальне діалогове вікно, якщо за обраним студентом зафіксовано фінансову заборгованість, що змушує персонал спершу врегулювати питання оплати.

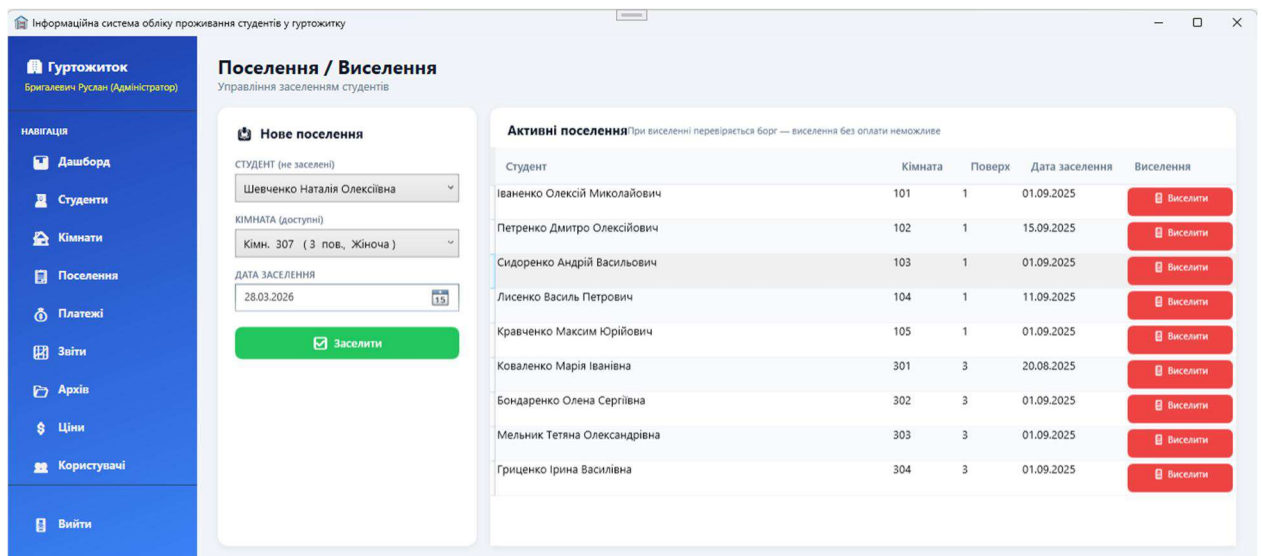


Рисунок 2.2.5. – Форма поселення та виселення

Фінансовий блок системи представлений формами обліку платежів та налаштування тарифів. Вікно додавання платежу вимагає введення суми у форматі дійсного числа, вибору дати транзакції та зазначення типу операції. Програма підтримує помісячну деталізацію нарахувань, де для кожного студента відображається накопичувальний баланс, що розраховується на основі діючих цін. Форма управління тарифами дозволяє встановлювати вартість проживання за місяць із фіксацією дати початку дії нової ціни, що забезпечує коректність ретроспективних розрахунків при зміні економічних умов.

Платежі
Облік оплати · нарахувань · балансів

Студент
Бондаренко Олена Сергіївна

Нараховано: 6,127.42 грн
Оплачено: 6,390.98 грн
Баланс: +263.56 грн (переплата)

Тарифи
950.00 грн/міс з 01.01.2026
850.00 грн/міс з 01.08.2025
Змінити → розділ «Ціни»

Баланс студентів

Студент	Нараховано	Оплачено	Баланс	Кімната
Бондаренко Олена Сергіївна	6,127.42 грн	6,390.98 грн	+263.56 грн (переплата)	302
Гриценко Ірина Василівна	6,127.42 грн	2,550.00 грн	-3,577.42 грн (борг)	304
Іваненко Олексій Миколайович	6,127.42 грн	1,700.00 грн	-4,427.42 грн (борг)	101

Помісячні нарахування: Бондаренко Олена Сергіївна

Місяць	Нараховано	Надійшло оплат	Залишок (накопич.)
вересень 2025	850.00 грн	1,700.00 грн	+850.00 грн (переплата)
жовтень 2025	850.00 грн	0.00 грн	+0.00 грн (переплата)
листопад 2025	850.00 грн	999.99 грн	+149.99 грн (переплата)
грудень 2025	850.00 грн	0.00 грн	-700.01 грн (борг)
січень 2026	950.00 грн	2,840.99 грн	+1,190.98 грн (переплата)
лютий 2026	950.00 грн	0.00 грн	+240.98 грн (переплата)
березень 2026	950.00 грн	850.00 грн	+140.98 грн (переплата)

Всі проплати: Бондаренко Олена Сергіївна

Дата	Сума	Тип	Примітки	Дія
01.03.2026	850.00 грн	Оплата за проживання		
14.01.2026	2,840.99 грн	Оплата за проживання		
14.11.2025	999.99 грн	Оплата за проживання		
05.09.2025	1,700.00 грн	Оплата за проживання	Початковий платіж	

Рисунок 2.2.6. – Форма обліку платежів та налаштування тарифів

Для адміністративного контролю та підготовки паперової документації використовується форма звітів та архівний модуль.

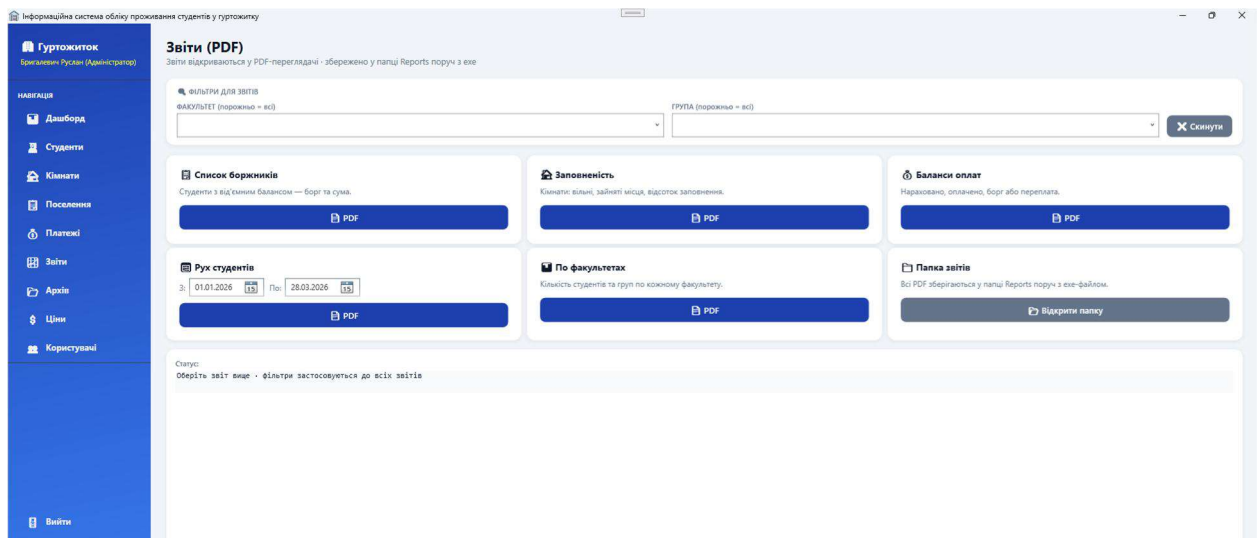


Рисунок 2.2.7. – Форма звітів

Розділ звітів дозволяє користувачу обрати один із типів статистичних вибірок та налаштувати фільтри за факультетом або часовим періодом. Результатом роботи цієї форми є генерація стилізованого PDF-документа, що містить табличні дані та підсумкові показники.

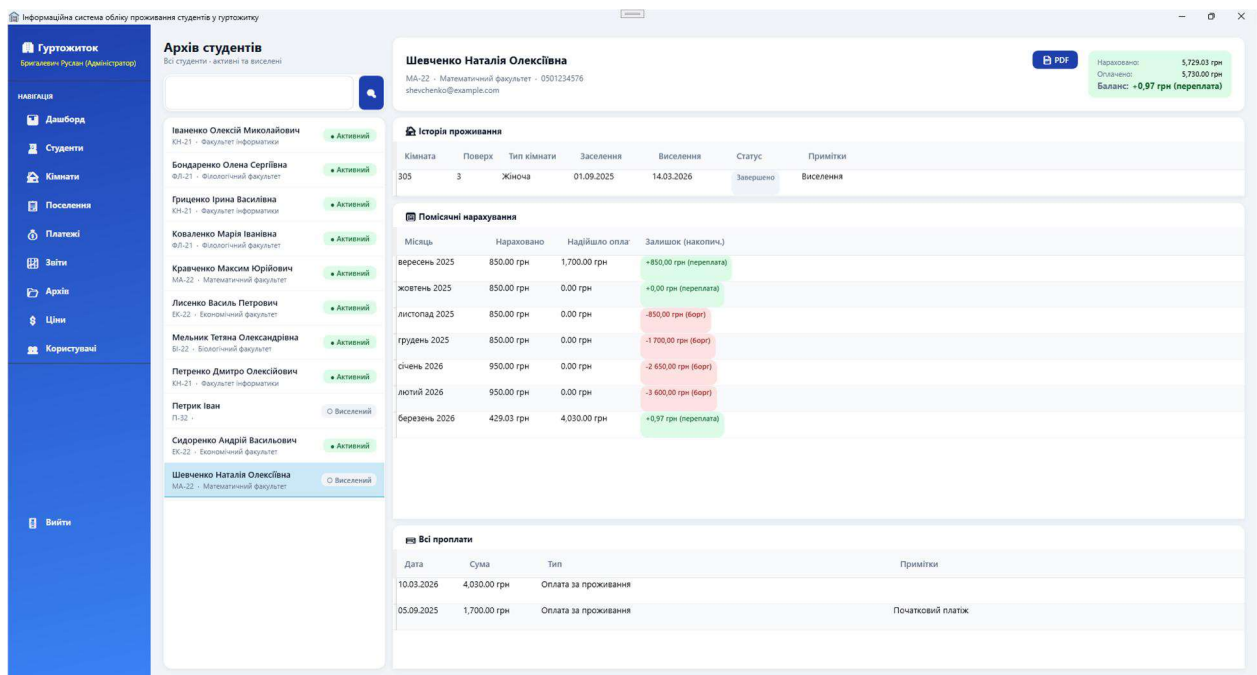


Рисунок 2.2.8. – Форма архів

Архівна форма надає можливість перегляду повної ретроспективи проживання будь-якого студента, включаючи всі його попередні кімнати,

здійснені оплати та історію нарахувань, що забезпечує прозорість обліку протягом усього терміну навчання.

2.3 Архітектура програмного засобу

Програмний комплекс побудований на базі сучасної платформи **.NET 9** із використанням архітектурного патерну **MVVM (Model-View-ViewModel)**, що є стандартом для розробки надійних та масштабованих додатків на технології **WPF**. Такий підхід забезпечує чітке розділення відповідальності між візуальним інтерфейсом, бізнес-логікою та моделлю даних, що спрощує тестування та подальший супровід системи.

Архітектура системи складається з трьох основних рівнів:

1. Рівень представлення (View) Цей шар відповідає за графічний інтерфейс користувача та взаємодію з ним. Представлення реалізовані у вигляді вікон (Window) та користувацьких елементів керування (UserControl), описаних мовою XAML. Важливою особливістю рівня представлення є використання глобальних ресурсів і конвертерів для динамічної зміни інтерфейсу, наприклад, перемикання розділів через SectionVisConverter або кольорова індикація статусів. Представлення не містять логіки обробки даних, а лише відображають стан, отриманий від ViewModel через механізм прив'язки даних (Data Binding).

2. Рівень логіки представлення (ViewModel)

ViewModel виступає посередником між інтерфейсом та бізнес-логікою. Кожна форма має відповідний клас-модель представлення, успадкований від BaseViewModel, який реалізує інтерфейс INotifyPropertyChanged для автоматичного оновлення UI при зміні властивостей. Взаємодія користувача з системою обробляється через команди (RelayCommand та AsyncRelayCommand), що дозволяє виконувати тривалі операції, такі як завантаження статистики або генерація звітів, в асинхронному режимі без блокування основного потоку.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Рівень моделі та сервісів (Model & Services)

Цей рівень включає в себе структури даних (РОСО-класи) та сервіси для роботи з ними. Моделі описують сутності предметної галузі: студентів, кімнати, платежі та поселення. Для виконання складних бізнес-процесів використовуються спеціалізовані сервіси:

- **IStudentService, IRoomService:** базові операції над реєстрами.
- **IPaymentCalculationService:** виконує розрахунок помісячних нарахувань та залишків на основі історії тарифів.
- **IAuthService:** забезпечує перевірку облікових даних та управління сесією поточного користувача.
- **IStatisticsService:** формує агреговані дані для дашборду.

Взаємодія з базою даних

Доступ до даних реалізовано через контекст бази даних (EntityFramework Core або прямі запити до SQLite), що інкапсульовано всередині сервісів. Використання **Dependency Injection (DI)** дозволяє централізовано керувати життєвим циклом сервісів та моделей представлення, що реалізовано через статичний клас App.Services. Це забезпечує легку заміну компонентів системи та можливість використання спільних станів, наприклад, через глобальний клас AppState, що зберігає інформацію про поточного авторизованого користувача та його права.

Обрана архітектура дозволяє системі ефективно обробляти складні зв'язки між сутностями, забезпечувати високу швидкість відгуку інтерфейсу та гарантувати цілісність даних при виконанні фінансових і адміністративних операцій.

2.4 Програмна реалізація

Програмна реалізація інформаційної системи базується на використанні сучасних інструментів розробки середовища **.NET 9** та мови програмування **C#**. Основним технологічним рішенням для побудови інтерфейсу є **WPF**, що

дозволило реалізувати складні графічні елементи та забезпечити високу швидкість відгуку додатку за рахунок асинхронного виконання операцій.

Взаємодія з базою даних здійснюється через сервісний шар, що дозволяє інкапсулювати логіку запитів до SQLite. Для забезпечення цілісності фінансових розрахунків реалізовано спеціалізований сервіс `PaymentCalculationService`, який на основі даних таблиці `PriceSettings` динамічно обчислює заборгованість для кожного студента. У процесі реалізації було застосовано сувору перевірку вхідних даних, зокрема при обробці грошових сум, де програмний код автоматично нормалізує роздільники (крапка/кома) та видаляє зайві пробіли для запобігання помилкам парсингу.

Для того, щоб інтерфейс залишався чуйним під час тривалих операцій (наприклад, при завантаженні великих списків або генерації PDF-файлів), у системі широко застосовано асинхронні команди `AsyncRelayCommand`. Це дозволяє виконувати фонові задачі паралельно з основним потоком UI. Наприклад, при переході до розділу «Дашборд» система асинхронно оновлює статистику про вільні місця та боржників, не блокуючи при цьому навігаційне меню.

Одним із найскладніших етапів реалізації стала підсистема генерації PDF-звітів, побудована на базі бібліотеки `PdfSharpCore`. Програмно реалізовано набір допоміжних методів для малювання таблиць, автоматичного розбиття контенту на сторінки (`EnsureSpace`) та форматування заголовків із використанням корпоративних кольорів системи. Кожен звіт генерується динамічно на основі поточних фільтрів, що встановлені користувачем у графічному інтерфейсі.

Програмна реалізація авторизації базується на перевірці хешованих паролів, що зберігаються у базі даних. На рівні коду `MainViewModel` реалізовано логіку видимості елементів інтерфейсу залежно від ролі поточного користувача (`IsAdmin`, `IsKomendant`, `IsBuhgalter`), що гарантує доступ до чутливих даних лише уповноваженим особам. Окрему увагу

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

придільено безпеці операції виселення: програмний код містить логічну перевірку, яка перериває процес, якщо баланс студента є від'ємним, виводячи відповідне попередження у UI .

У підсумку, програмна реалізація системи відповідає принципам об'єктно-орієнтованого проектування, забезпечуючи стабільну роботу додатку в локальному середовищі гуртожитку та надаючи користувачам надійні інструменти для щоденного обліку.

2.5 Тестування програми

Для підтвердження працездатності та надійності розробленої інформаційної системи було проведено комплексне тестування всіх її функціональних модулів. Процес перевірки охоплював верифікацію коректності збереження даних у базі SQLite та стабільності роботи графічного інтерфейсу на платформі WPF. Особлива увага приділялася тестуванню логіки авторизації, де перевірялася реакція системи на введення невірних облікових даних та спроби несанкціонованого доступу. У межах тестування фінансового модуля було змодельовано різні сценарії нарахування оплати, що підтвердило точність алгоритмів розрахунку балансу студента . Перевірка модуля поселення продемонструвала коректну роботу фільтрів вибору кімнат відповідно до статі студента та актуального стану житлового фонду .

Важливим етапом стало стрес-тестування механізму блокування виселення, яке підтвердило неможливість завершення проживання при наявності навіть мінімальної заборгованості . Тестування інтерфейсу користувача показало високу ефективність використання асинхронних команд, оскільки програма зберігала стабільну швидкість відгуку під час генерації об'ємних PDF-звітів. Усі виявлені під час ітераційного тестування незначні помилки у відображенні даних були оперативно усунені через коригування логіки у відповідних моделях представлення . Крім того була проведена перевірка цілісності бази даних при раптовому завершенні роботи програми, що підтвердило надійність транзакційної моделі SQLite.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Результати тестування модуля управління користувачами довели ефективність системи розмежування прав доступу, де кожна роль мала доступ лише до дозволеного функціоналу. Процес валідації форм редагування підтвердив захищеність системи від введення некоректних значень або пустих обов'язкових полів. Проведене бета-тестування на реальних наборах даних дозволило зробити висновок про повну відповідність розробленого програмного продукту поставленим функціональним вимогам. Отримана висока оцінка зручності інтерфейсу свідчить про успішну реалізацію принципів ергономіки в дизайні додатку. Підсумовуючи результати перевірки можна стверджувати, що інформаційна система готова до впровадження в експлуатаційне середовище гуртожитку. Програма демонструє високий рівень відмовостійкості та гарантує точність фінансового й адміністративного обліку. Виконані тести підтвердили масштабованість архітектури системи та можливість її подальшого розширення новими модулями. Розроблений засіб забезпечує автоматизацію ключових бізнес-процесів установи на професійному рівні.

Висновки до розділу 2

У другому розділі було здійснено повне проектування інформаційної системи обліку проживання студентів, що дозволило сформувати надійний технологічний фундамент для подальшої програмної реалізації. За результатами проведеної роботи можна зробити наступні висновки:

1. Розроблено інфологічну модель предметної галузі, у межах якої виокремлено шість основних сутностей: студенти, кімнати, поселення, платежі, налаштування цін та користувачі. Побудована ER-діаграма відображає логічні зв'язки між цими об'єктами, забезпечуючи відсутність надмірності даних та дотримання принципів нормалізації.
2. Спроектовано фізичну структуру бази даних у середовищі СУБД SQLite. Для кожної з таблиць визначено оптимальні типи даних, встановлено

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

первинні та зовнішні ключі, що гарантує цілісність інформації на рівні збереження у файловій системі.

3. Визначено склад та призначення основних екранних форм системи. Графічний інтерфейс спроектовано таким чином, щоб забезпечити зручне введення персональних даних, візуальний моніторинг житлового фонду за допомогою колірної індикації та автоматизоване управління фінансовими розрахунками.
4. Обґрунтовано вибір архітектурного патерну MVVM (Model-View-ViewModel). Дана архітектура дозволяє відокремити бізнес-логіку від візуального представлення, що забезпечує гнучкість системи, спрощує її тестування та дозволяє реалізувати складні асинхронні операції без блокування інтерфейсу користувача.
5. Описано особливості програмної реалізації ключових механізмів, зокрема системи розрахунку заборгованості, модуля генерації PDF-звітів та механізмів безпеки, які базуються на розмежуванні прав доступу та валідації вхідних даних.
6. Проведене тестування підтвердило стабільність роботи спроектованих алгоритмів та їх відповідність функціональним вимогам, що були висунуті на етапі аналізу предметної галузі.

Розроблена проектна документація та архітектурні рішення є достатніми для переходу до етапу впровадження системи та її подальшої експлуатації в умовах студентського гуртожитку.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

3.1 Мінімальні вимоги до системи

Для забезпечення стабільної та коректної роботи інформаційної системи «Інформаційна система обліку проживання студентів у гуртожитку» необхідно дотримуватися певних апаратних та програмних вимог. Оскільки програмний продукт розроблено на базі сучасної платформи .NET 9, центральний процесор повинен підтримувати архітектуру x64 з тактовою частотою не менше 2.0 ГГц для швидкої обробки асинхронних запитів. Об'єм оперативної пам'яті має складати щонайменше 4 ГБ, що дозволить системі одночасно утримувати в пам'яті графічний інтерфейс WPF та виконувати операції з базою даних SQLite. Для встановлення додатку та зберігання локальної бази даних разом із генерованими PDF-звітами необхідно мати принаймні 500 МБ вільного місця на жорсткому диску або SSD-накопичувачі.

Програмна частина вимагає наявності встановленої операційної системи Windows 10 версії 1809 або новішої, оскільки саме ці версії гарантують повну сумісність із середовищем виконання .NET 9. Користувачу необхідно попередньо встановити пакет .NET Desktop Runtime 9.0, який є критично важливим для запуску десктопних WPF-додатків. Графічний адаптер комп'ютера повинен підтримувати DirectX 9 або вище для апаратного прискорення рендерингу елементів керування та плавної анімації інтерфейсу. Монітор повинен мати мінімальну роздільну здатність 1280x720 пікселів, що дозволить коректно відображати всі панелі навігації та таблиці з даними без втрати інформації.

Оскільки система передбачає експорт документації, у програмному середовищі має бути встановлений будь-який сучасний переглядач PDF-файлів для відкриття звітів безпосередньо після їх генерації. Наявність підключення до локальної мережі або мережі Інтернет не є обов'язковою для базової роботи, адже база даних SQLite зберігається локально у файлі dormitory.db. Для друку сформованих звітів про боржників або заповненість кімнат до системи має бути підключений стандартний лазерний або

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

струменевий принтер. Маніпулятор миша та клавіатура є необхідними засобами введення для заповнення текстових форм та навігації між розділами програми.

Система вимагає прав адміністратора Windows при першому запуску для створення папки «Reports» у робочій директорії додатка. Налаштування брандмауера зазвичай не потрібні, оскільки програма не використовує мережеві порти для роботи з локальною СУБД. Рекомендується використовувати 64-бітну версію операційної системи для досягнення максимальної продуктивності при виконанні складних розрахунків балансу студентів. Дотримання цих мінімальних вимог гарантує відсутність затримок при роботі з архівом та асинхронному оновленні дашборду .

Розроблений засіб може успішно експлуатуватися навіть на комп'ютерах офісного сегмента зі скромними характеристиками. Виконання зазначених умов забезпечує надійне збереження персональних даних мешканців та стабільне формування фінансової звітності гуртожитку.

3.2 Інтерфейс користувача

Інтерфейс програми розроблено на основі сучасних принципів дизайну користувацького досвіду (UX), з акцентом на чисті візуальні форми, високу контрастність та логічне групування елементів. Основним стилістичним рішенням є використання білого фону для робочих областей у поєднанні з глибоким синім кольором (PrimaryBrush) для навігаційних та акцентних компонентів. Програмний комплекс використовує єдину віконну архітектуру, де доступ до різних функціональних модулів здійснюється через бічну панель навігації.

Панель навігації та ідентифікації

Бічна панель (Sidebar) розташована у лівій частині головного вікна і забезпечує миттєвий перехід між розділами системи. У верхній частині панелі міститься блок ідентифікації, який відображає повне ім'я поточного користувача та його роль (наприклад, «Адміністратор» або «Комендант»), що

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

підкреслює персоналізацію сесії. Елементи навігації представлені кнопками з тематичними іконками (Емої) та текстовими підписами: « Дашборд», « Студенти», « Кімнати», « Поселення», « Платежі», « Звіти» та інші. Кожна кнопка має візуальний ефект при наведенні та активує відповідний розділ через механізм команд.

Візуалізація даних та картки

Для відображення інформації в системі використовуються два основних підходи: табличний та картковий. Розділ «Кімнати» реалізовано за допомогою сітки карток, кожна з яких має верхню кольорову смугу, що відображає статус заповненості: зелений (вільно), жовтий (є місця) або червоний (заповнено). Картка містить номер кімнати, поверх, тип приміщення та список поточних мешканців у скороченому вигляді. Табличне представлення (DataGrid) використовується в архівах та списках платежів, забезпечуючи чітке розділення рядків за допомогою чергування кольорів та виділення активного вибору блакитним фоном.

Форми редагування та бічні панелі

Унікальною особливістю інтерфейсу є використання контекстних бічних панелей (Edit Panels) для додавання та редагування записів. Замість відкриття множини модальних вікон, форма редагування плавно з'являється у правій частині екрана, накладаючи на основний контент м'яку тінь (DropShadowEffect). Ці панелі містять елементи керування «Modern Style»:

- **ModernTextBox:** текстові поля з чіткими межами та акцентним фокусом.
- **ModernComboBox:** випадаючі списки для вибору категорій (факультети, типи кімнат, ролі).
- **DatePicker:** графічний календар для зручного вибору дат народження або платежів.

Сигнальна система та сповіщення

Інтерфейс активно взаємодіє з користувачем через систему статусних повідомлень та бейджів. У верхній частині робочих областей розташовані

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

кольорові блоки сповіщень: зелені — для успішних операцій (« Студента додано») та червоні — для попереджень або помилок. Особливо деталізованим є блок «Борг» у модулі поселення, який з'являється при спробі виселення боржника; він має градієнтний червоний фон, іконку знаку стоп та виділену жирним шрифтом суму заборгованості, що робить помилку неможливою для ігнорування.

Завдяки використанню асинхронного завантаження, перемикання між цими елементами інтерфейсу відбувається без затримок, а інтерактивні елементи (кнопки « PDF», « Оновити», « Зберегти») мають виразні кольори, що відповідають їхньому функціональному призначенню.

3.3 Інструкція для роботи з програмою

Процес експлуатації інформаційної системи розпочинається з етапу автентифікації користувача у спеціалізованому вікні входу. Користувачеві необхідно послідовно заповнити текстові поля для введення логіна та пароля, після чого ініціювати перевірку облікових даних для отримання доступу до головного інтерфейсу системи . У разі успішної авторизації відкривається основне вікно, де на бічній панелі відображається ім'я співробітника та його роль, що визначає обсяг доступних функцій для коменданта, бухгалтера або адміністратора .

Робота з житловим фондом зосереджена в модулі управління кімнатами, де візуалізація стану приміщень реалізована через систему кольорових карток. Для внесення змін до характеристик кімнати або додавання нового об'єкта використовується бічна панель редагування, де вказуються номер приміщення, поверх, тип заселення та загальна місткість . Система автоматично оновлює візуальну індикацію заповненості на основі співвідношення активних поселень до встановленого ліміту місць, що дозволяє персоналу миттєво оцінювати наявність вільного фонду .

Ведення бази даних студентів здійснюється в однойменному розділі, де передбачено інструменти для створення та коригування анкети мешканця. Під

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

час реєстрації нової особи необхідно внести повне ім'я, дату народження, академічну групу, факультет та актуальні контактні дані, включаючи номер телефону та електронну адресу . Пошукова система дозволяє швидко знаходити потрібний запис за ПІБ або іншими ідентифікаторами, що значно прискорює процес адміністрування .

Процедура поселення реалізується через поєднання даних студента та обраної кімнати у відповідному модулі. Користувач обирає особу зі списку незаселених студентів та вказує доступну кімнату, при цьому система автоматично перевіряє відповідність статі студента до типу приміщення . Завершення процесу проживання ініціюється натисканням кнопки виселення, проте програма виконує обов'язкову автоматичну перевірку фінансового стану студента . Якщо за мешканцем зафіксовано борг, система блокує операцію та виводить попередження з точною сумою недоплати, вимагаючи попереднього погашення заборгованості .

Фінансовий облік передбачає реєстрацію платежів та моніторинг накопичувальних балансів. Бухгалтер має можливість додавати нові записи про оплату, вказуючи суму, дату та призначення платежу, що автоматично відображається на поточному балансі студента . Для забезпечення точності нарахувань у системі ведеться історія тарифів, де фіксується вартість проживання та дата введення її в дію . Розділ звітів дозволяє формувати стилізовані PDF-документи, включаючи списки боржників та статистику заповненості по поверхах, які автоматично зберігаються в локальній папці для подальшого друку .

Повна ретроспектива взаємодії зі студентами доступна в архівному модулі, який зберігає інформацію про всіх мешканців, незалежно від їхнього поточного статусу. В архіві можна переглянути хронологію всіх поселень, помісячну розбивку нарахувань та повну історію здійснених платежів для кожної особи . Використання даного функціоналу дозволяє адміністрації здійснювати глибокий аналіз діяльності гуртожитку та надавати необхідні довідки за минулі періоди.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Висновки до розділу 3

У третьому розділі було детально розглянуто аспекти практичного впровадження та правила експлуатації інформаційної системи обліку проживання студентів. На основі опису інтерфейсу та функціональних можливостей програми можна зробити наступні висновки:

1. Встановлено мінімальні системні вимоги, які підтверджують можливість запуску додатка на комп'ютерах офісного сегмента з операційною системою Windows та встановленим середовищем .NET 9, що робить систему доступною для широкого кола навчальних закладів.
2. Описано графічний інтерфейс користувача, який побудовано за сучасними стандартами UX-дизайну. Використання бічної панелі навігації, візуальних карток кімнат із кольоровою індикацією та контекстних панелей редагування забезпечує інтуїтивно зрозуміле середовище для роботи персоналу .
3. Розроблено детальну інструкцію з виконання ключових адміністративних операцій. Процеси поселення та виселення максимально автоматизовані, а вбудовані алгоритми валідації статі студента та перевірки фінансової заборгованості мінімізують ризик виникнення помилок через людський фактор .
4. Продемонстровано ефективність роботи модуля звітності, який дозволяє миттєво генерувати юридично значущі PDF-документи (списки боржників, відомості про заповненість тощо) безпосередньо з робочого місця коменданта або бухгалтера .
5. Підтверджено важливість архівного модуля для ведення довгострокової статистики. Можливість перегляду повної ретроспективи нарахунків та проживання навіть виселених студентів гарантує прозорість обліку та спрощує підготовку історичних довідок .

Розроблене керівництво користувача доводить, що створена інформаційна система є завершеним програмним продуктом, який поєднує в собі високу функціональність із простотою щоденного використання.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено сучасну інформаційну систему обліку проживання студентів у гуртожитку, яка успішно вирішує завдання автоматизації адміністративної та фінансової діяльності установи. Під час дослідження проведено аналіз предметної галузі, що дозволило виокремити ключові бізнес-процеси: поселення, виселення, нарахування оплати та формування статистичної звітності. На основі порівняльного аналізу існуючих рішень обґрунтовано необхідність створення локального програмного продукту, який поєднує безкоштовність володіння із високою швидкістю роботи. Обраний технологічний стек у складі .NET 9, WPF та SQLite забезпечив створення продуктивного додатка з багатим графічним інтерфейсом та надійним механізмом збереження даних.

Проектування системи базувалося на архітектурному патерні MVVM, що дозволило досягти високої гнучкості коду та розділити логіку обробки даних від елементів представлення. Розроблена фізична модель бази даних гарантує цілісність інформації про студентів, житловий фонд та фінансові транзакції завдяки використанню системи зовнішніх ключів. Впроваджена система розмежування прав доступу забезпечує безпеку персональних даних мешканців, надаючи адміністраторам, комендантам та бухгалтерам лише необхідний функціонал. Одним із найважливіших досягнень проекту стала реалізація механізму фінансового контролю, який унеможливорює виселення студентів за наявності заборгованості.

Програмна реалізація асинхронних команд дозволила зберегти високу швидкість відгуку інтерфейсу навіть під час виконання складних розрахунків балансів або генерації об'ємних PDF-звітів. Модуль звітності забезпечує миттєве формування документації, що значно скорочує час на підготовку паперових звітів для адміністрації навчального закладу. Створений «Дашборд» надає користувачам можливість оперативно моніторити заповненість гуртожитку та загальний стан фінансової дисципліни мешканців у реальному часі. Візуалізація житлового фонду через кольорові картки кімнат

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

спрощує процес розселення та робить його більш інтуїтивно зрозумілим для персоналу .

Тестування системи підтвердило її стабільність, відсутність критичних помилок та повну відповідність усім висунутим функціональним вимогам. Розроблене керівництво користувача містить вичерпні інструкції, що дозволяють персоналу швидко опанувати роботу з програмою без тривалого спеціального навчання. Впровадження системи дозволить значно підвищити прозорість обліку, мінімізувати вплив людського фактора та забезпечити точність фінансових нарахунків. Перспективи подальшого розвитку проекту можуть включати інтеграцію з онлайн-банкінгом для автоматичного імпорту платежів та розробку мобільного додатка для студентів. У підсумку, мета кваліфікаційної роботи досягнута в повному обсязі, а розроблений продукт готовий до практичної експлуатації в умовах студентського гуртожитку.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Про освіту : Закон України від 05.09.2017 р. № 2145-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2145-19> (дата звернення: 12.03.2026).
2. Про вищу освіту : Закон України від 01.07.2014 р. № 1556-VII. URL: <https://zakon.rada.gov.ua/laws/show/1556-18> (дата звернення: 12.03.2026).
3. Положення про особливості користування гуртожитками закладів фахової передвищої та вищої освіти : наказом Міністерства освіти і науки України від 21.11.2019 р. № 1452.
4. Про затвердження Примірного положення про користування гуртожитками : постанова Кабінету Міністрів України від 20.06.2018 р. № 498.
5. ДСТУ ISO/IEC/IEEE 12207:2023. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення (ISO/IEC/IEEE 12207:2017, IDT). Київ : ДП «УкрНДНЦ», 2023. 114 с.
6. Троєлсен Е., Джепікс Ф. Мова програмування C# 10 та платформа .NET 6 : основні принципи та приклади застосування. 11-те вид. Київ : Спадщина, 2023. 1240 с.
7. Швець І. Т. Проектування баз даних : навчальний посібник. Львів : Новий Світ-2000, 2021. 256 с.
8. Гайдаржи В. І. Розробка програмного забезпечення на платформі .NET : навчальний посібник. Житомир : ЖДТУ, 2022. 312 с.
9. Пасічник В. В., Резніченко В. А. Організація баз даних та знань. Київ : ВГК, 2020. 448 с.
10. Кравець П. О. Об'єктно-орієнтоване програмування : навч. посібник. Львів : Видавництво Львівської політехніки, 2022. 624 с.
11. Вульф О. Г. SQLite. Повне керівництво. Стенфорд : Пріс, 2021. 410 с.
12. Ріхтер Д. CLR via C#. Програмування на платформі Microsoft .NET Framework 4.5 мовою C#. Київ : Діалектика, 2019. 896 с.
13. Стілмен Е., Грін Д. Head First C#. Вивчення C# через практику. 4-те вид. Оксфорд : O'Reilly Media, 2021. 832 с.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

14. .NET Documentation : офіційна документація платформи .NET від Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/> (дата звернення: 15.03.2026).
15. Windows Presentation Foundation (WPF) Desktop Guide : офіційна документація Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/> (дата звернення: 15.03.2026).
16. Entity Framework Core Guide : документація з об'єктно-реляційного відображення для .NET. URL: <https://learn.microsoft.com/en-us/ef/core/> (дата звернення: 18.03.2026).
17. SQLite Documentation : офіційний опис механізмів СУБД SQLite. URL: <https://www.sqlite.org/docs.html> (дата звернення: 20.03.2026).
18. Мартин Р. Чиста архітектура. Мистецтво розробки програмного забезпечення. Київ : Фабула, 2023. 368 с.
19. Фаулер М. Шаблони корпоративних програмних архітектур. Бостон : Addison-Wesley, 2020. 544 с.
20. Тестування програмного забезпечення : методи та підходи : навчальний посібник / за ред. О. В. Коваленка. Харків : ХНУРЕ, 2022. 210 с.

					КРФМБ.122.043стн.017.2026.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42