

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана
Огієнка Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота бакалавра
з теми: **«Розробка інформаційної системи обліку
оплати за проживання в гуртожитках»**

Виконав: здобувач вищої освіти групи KN1-B20

спеціальності 122 Комп'ютерні науки

Гавришко Володимир Євгенович

(прізвище, ім'я та по батькові здобувача вищої освіти)

Керівник: Іванюк Віталій Анатолійович, доктор
технічних наук, доцент

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

Консультант: Андруховський Андрій Борисович,
Individual Contributor at Aspose (Aspose.PDF
Group)

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

Рецензент: Кушнір Оксана Климівна, кандидат
економічних наук, доцент

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

м. Кам'янець-Подільський – 2024 р.

АНОТАЦІЯ

Тема: розробка інформаційної системи обліку оплати за проживання в гуртожитках.

Студент: Гавришко Володимир Євгенович

Керівник: доктор технічних наук, доцент, Іванюк Віталій Анатолійович

Консультант: Андруховський Андрій Борисович, Individual Contributor at Aspose (Aspose.PDF Group)

Дана робота являє собою розробку інформаційної системи для ефективного обліку оплати за проживання в гуртожитках. У першому розділі присвячений аналізу контексту використання систем обліку платежів для гуртожитків. Роль облікових систем у фінансовому управлінні гуртожитками, їх ключові компоненти та значення.

У другому розділі досліджуються методи реалізації програмної системи. Розглянуто визначення функціональних і технічних вимог, вибір відповідних технологій та опис структури бази даних. Цей розділ деталізує підхід до вибору інструментів та технологій, які забезпечують надійну та ефективну роботу системи, представлена архітектурна схема програмної системи, детально описана функціональність системи.

У третьому розділі розглядається представлення інтерфейсу користувача. Особливу увагу приділено вкладкам, які є ключовими компонентами системи. робота користувача в системі. Описані процеси інсталяції та системні вимоги, а також надана детальна інструкція з використання програмного продукту. Розглянуто пункти меню, що дозволяють ефективно управляти даними та здійснювати необхідні операції.

Ключові слова: інформаційна система, облік оплати, проживання в гуртожитках, фінансове управління, функціональні вимоги, технічні вимоги, користувацький інтерфейс, технології.

ANNOTATION

Topic: Development of an information system for accounting for payment for accommodation in dormitories.

Student: Gavryshko Volodymyr Yevhenovych

Academic Supervisor: Doctor of Technical Sciences, Associate Professor, Ivaniuk Vitalii Anatoliiovych

Consultant: Andrukhovskyi Andrii Borysovych, Individual Contributor at Aspose (Aspose.PDF Group)

This work is the development of an information system for efficient accounting of payments for accommodation in dormitories. The first section analyses the context of using payment accounting systems for dormitories. The role of accounting systems in the financial management of dormitories, their key components and importance.

The second section examines the methods of implementing a software system. It includes the definition of functional and technical requirements, the selection of appropriate technologies and a description of the database structure. This section details the approach to the selection of tools and technologies that ensure reliable and efficient operation of the system, presents the architectural diagram of the software system, and describes the system's functionality in detail.

The third section discusses the presentation of the user interface. Particular attention is paid to tabs, which are key components of the system. user work in the system. The installation process and system requirements are described, and detailed instructions on how to use the software product are provided. The menu items that allow you to effectively manage data and perform the necessary operations are considered.

Keywords: information system, payment accounting, dormitory accommodation, financial management, functional requirements, technical requirements, user interface, technology.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. Контекст використання систем обліку платежів для гуртожитків .	7
1.1 Характеристика обліково-аналітичного забезпечення систем управління гуртожитками	7
1.2 Особливості надання послуг в гуртожитках та їх вплив на побудову обліку	8
1.3 Автоматизовані системи управління гуртожитками	10
Висновок до розділу 1.....	11
РОЗДІЛ 2. Методи реалізації програмної системи.....	13
2.1 Визначення функціональних і технічних вимог	13
2.2 Вибір технологій (мова програмування, платформа).....	14
2.3 Архітектурна схема програмної системи.....	15
2.4 Опис структури бази даних	16
2.5 Опис функціональності системи	18
Висновок до розділу 2.....	19
РОЗДІЛ 3. Інтерфейс застосунку та робота користувача в системі	20
3.1 Концепція інтерфейсу користувача.....	20
3.2 Інструкція з використання програмного продукту.....	21
3.3 Пункт меню “Довідники”	22
3.4 Пункт меню “Студенти”	24
3.5 Пункт меню “Перегляд довідника”	27
3.6 Пункт меню “Картка студента”	28
3.7 Пункт меню “Операції”	28
Висновки до розділу 3	29
ВИСНОВКИ	30
ДОДАТКИ	32
ДОДАТОК А.	32
Реалізація вікна Гуртожитки	32
ДОДАТОК Б.	35
Реалізація додавання студентів в базу даних	35
ДОДАТОК В.....	43
Реалізація формування звіту.....	43

ВСТУП

Актуальність. У сучасний період впровадження інформаційних технологій у різних організаціях стає все більш поширеним і актуальним явищем. Використання цих технологій дозволяє підвищити рівень інформаційного забезпечення управлінських процесів та їх ефективність.

Оскільки бухгалтерські операції мають високу складність, автоматизація бухгалтерської діяльності залишається актуальною. Багато аспектів цієї діяльності залишаються неавтоматизованими, що робить впровадження автоматизації бухгалтерського обліку та інформаційних систем найважливішою задачею будь-якої установи. Перед появою комп'ютерних програм для бухгалтерського обліку потрібно було вручну вираховувати дані, складати звітність та баланси, що вимагало значних зусиль і часу. Крім того, часто виникали арифметичні помилки, які ускладнювали об'єктивну оцінку фінансового стану установи.

Об'єктом дослідження є процеси обліку оплати за проживання в студентських гуртожитках, а також існуючі інформаційні системи, які використовуються для цих цілей. Дослідження охоплює всі аспекти обліку, включаючи реєстрацію платежів, моніторинг заборгованостей та автоматизацію звітності.

Предметом дослідження є розробка інформаційної системи, яка забезпечить ефективний облік оплати за проживання в студентських гуртожитках. Особлива увага приділяється функціональним вимогам до системи, вибору технологій для її реалізації.

Метою дослідження є розробка інформаційної системи обліку оплати за проживання в гуртожитках, включаючи реєстрацію платежів, моніторинг заборгованостей та генерування звітів.

Завдання дослідження:

1. Розгляд контексту використання систем обліку платежів для гуртожитків
2. Організування методів реалізації програмної системи

3. Розробка інформаційної системи обліку

4. Інструкція з використання програмного продукту

Методи дослідження. Для втілення цієї мети використовувалися методи об'єктно-орієнтованого програмування (ООП), а також сучасні технології розробки програмних додатків. Для побудови графічного інтерфейсу використовувалися C# та .NET WinForms, а для створення та управління базою даних застосовувався XSD, що забезпечує надійну та ефективну роботу з даними.

Практичне значення отриманих результатів. Розроблена інформаційна система обліку оплати за проживання в гуртожитках забезпечує значні покращення в управлінні цими процесами. Вона надає адміністрації гуртожитків інструменти для точного та швидкого обліку платежів, моніторингу заборгованостей та генерації звітів. Впровадження цієї системи сприятиме підвищенню ефективності роботи адміністрації гуртожитків.

РОЗДІЛ 1. Контекст використання систем обліку платежів для гуртожитків

1.1 Характеристика обліково-аналітичного забезпечення систем управління гуртожитками

Обліково-аналітична підтримка системи управління гуртожитком є важливим елементом для ефективного функціонування та прийняття управлінських рішень. Вона включає в себе збір, обробку та аналіз даних, необхідних для управління фінансовими ресурсами, обліку доходів і витрат, а також забезпечення прозорості і точності бухгалтерського процесу.

Ключові компоненти обліково-аналітичної підтримки

Інформаційна система:

Автоматизація облікових процесів: спеціальні програмні засоби дозволяють автоматизувати бухгалтерський процес, скоротити обсяг ручної роботи, підвищити точність і оперативність фінансових операцій інформаційні системи для обліку платежів за проживання в гуртожитку повинні забезпечувати автоматичну генерацію платежів, формування звітів та інтеграцію з іншими системами обліку. Система повинна мати можливість інтегруватися з іншими системами бухгалтерського обліку університету, забезпечуючи централізоване управління даними та зручний обмін між різними підрозділами.

Організаційна структура та процедури управління:

Для точного відображення фінансових результатів необхідно вести як синтетичний, так і аналітичний облік. Встановлення правил і політик для обліку різних видів послуг, обробки платежів, управління заборгованістю та моніторингу виконання фінансових зобов'язань мешканцями гуртожитку.

Технологія:

Включає використання сучасного апаратного та програмного забезпечення для збору, зберігання та обробки даних. Це забезпечує швидкість і точність процесу обліку і зручний доступ до інформації для прийняття управлінських рішень. Забезпечення конфіденційності та збереження персональних даних мешканців є важливим аспектом бухгалтерської та аналітичної підтримки. Всі дані повинні бути захищені від несанкціонованого доступу і втрати.

Ефективна система управління гуртожитками складний процес, який передбачає використання сучасних інформаційних технологій, чітко визначених організаційних структур і процедур, а також надійних аналітичних і бухгалтерських методів.

1.2 Особливості надання послуг в гуртожитках та їх вплив на побудову обліку

Надання послуг в гуртожитку характеризується певними особливостями, які роблять вплив на побудову системи обліку. До основних особливостей належать різноманітність послуг, що надаються, терміновість обслуговування, специфіка житлових умов та наявність різних категорій пільг для жителів.

Однією з особливостей надання послуги в гуртожитку є терміновість її виконання. Наприклад, надання місця для проживання має оформлятися відразу після запиту студента або співробітника. Це вимагає оперативного ведення бухгалтерського обліку і своєчасного відображення всіх операцій в бухгалтерських документах. Така терміновість впливає на організацію бухгалтерського обліку та аналізу фінансових результатів, оскільки необхідно відобразити дохід від проживання в системі бухгалтерського обліку на наступний день після заселення проживаючого.

Гуртожитки мають певні житлові умови, які впливають на облік витрат і доходів. Наприклад, оплата за проживання може залежати від типу кімнати, кількості проживаючих, наявності додаткових зручностей і т.д. це вимагає

розробки спеціальних облікових форм і реєстрів для роздільного відображення витрат і доходів по кожному типу житлових умов.

Кам'янець-Подільський національний університет імені Івана Огієнка пропонує 6 гуртожитків, у яких одночасно можуть проживати 2472 особи, також виділено кімнати для проживання сімей студентів. [2]. Це вимагає ретельного обліку для забезпечення ефективного управління та розрахунку вартості проживання.

В процесі надання послуг гуртожитку стикаються з наступними особливостями:

Різні послуги: крім проживання, надаються додаткові послуги (ремонт, прибирання).

Сезонність: кількість проживаючих залежить від навчального року.

Тип номера: є номери, які підходять як для одномісного проживання, так і для сімейного проживання.

Пільги: надання пільг певним категоріям студентів.

Дані аспекти впливають на побудову бухгалтерського обліку, що вимагає розробки інформаційних систем, що враховують усі змінні та забезпечують прозорість та ефективність управління фінансами. Програма повинна автоматизувати процес розрахунку платежів і формувати звіти для бухгалтерії. Це підвищить ефективність роботи бухгалтерії, скоротить кількість помилок і забезпечить прозорість фінансових операцій.

1.3 Автоматизовані системи управління гуртожитками

За допомогою АС "Студмістечко" автоматизується діяльність Генеральної дирекції навчальних закладів, спрощується та вдосконалюється документообіг, підвищується ефективність, узгодженість та надійність даних. Система автоматизує всі процеси, від розподілу місць, моніторингу та звітності до управління системами доступу.

При роботі з програмою створюється інформаційна інфраструктура студмістечка для моніторингу:

- Поселення студентів з АС «Деканат»;
- Автоматизація діяльності дирекції студентського містечка навчального закладу;
- Спрощення та вдосконалення документообігу;
- Підвищення оперативності, узгодженості та достовірності даних;
- Автоматизація всіх процесів, від розподілу місць, контролю та формування звітів до управління пропускною системою;

Згідно з введеними даними, друкуються поточні, офіційні, статистичні та звітні документи.

На підставі інформації, введеної в АС «Деканат», в програмі відображається інформація про студентів, яким потрібен гуртожиток, і персональні дані автоматично передаються в систему. Потім адміністратор студмістечка розселив студентів по їхніх гуртожитках.

Під час заселення студентам в студмістечку присвоюються кімнати та інвентар, і відповідним чином створюється замовлення на вартість проживання.

За допомогою модуля «Відвідувачі», кожен з чергових по гуртожитку веде реєстрацію осіб, які проходили в гуртожиток, із зазначенням особи та кімнати, до якої пройшов відвідувач і встановленням точного часу входу до гуртожитку та виходу з нього.

Існує можливість створювати мапи мешканців безпосередньо в системі «Студмістечко» для поселення осіб, які перебувають у відрядженні, студентів із інших навчальних закладів, аспірантів, докторантів та інших мешканців.

Інформація про стан абітурієнтів та студентів в навчальному закладі постійно поновлюється і адміністрація має точні дані про те, навчається студент в навчальному закладі чи відрахований.

Переваги впровадження:

1. Підвищення ефективності: Автоматизація рутинних процесів зменшує навантаження на персонал і підвищує швидкість виконання завдань.
2. Прозорість обліку: Чіткий та прозорий облік фінансових операцій і стану мешканців.
3. Поліпшення комунікації: Система забезпечує ефективну комунікацію між адміністрацією та мешканцями.
4. Аналіз даних: Можливість проведення детального аналізу діяльності гуртожитків для прийняття обґрунтованих управлінських рішень.

Автоматизовані системи управління гуртожитками, такі як "АС Студмістечко", сприяють оптимізації процесу управління, забезпечують високу точність бухгалтерського обліку та значно спрощують управління студентськими гуртожитками. Це покращить якість обслуговування мешканців та підвищить ефективність використання ресурсів університету.

Висновок до розділу 1

У першому розділі розглядається важливість систем обліку платежів для ефективного управління гуртожитками. Аналізуються ключові компоненти бухгалтерської та аналітичної підтримки, включаючи автоматизацію облікових процесів, що скорочує ручну працю і підвищує точність фінансових операцій. Підкреслюється важливість інтеграції цих систем з університетськими бухгалтерськими системами для забезпечення централізованого управління даними. Використання автоматизованої системи

дозволяє створювати платежі, звіти та інтегруватися з іншими системами, що значно покращує процес ведення обліку.

РОЗДІЛ 2. Методи реалізації програмної системи

2.1 Визначення функціональних і технічних вимог

Даний підрозділ присвячений визначенню функціональних та технічних вимог до розроблюваної системи обліку платежів для гуртожитків. Функціональні вимоги описують функціональні можливості програми, включаючи додавання, редагування та видалення даних про студентів, гуртожитки та факультети, пошук та фільтрацію даних, а також звітність. Технічні вимоги, з іншого боку, визначають необхідне апаратне та програмне забезпечення для нормальної роботи програми.

Функціональні вимоги:

Додавання, редагування та видалення даних про студентів, гуртожитки та факультети: Система повинна надавати можливість додавати, змінювати та видаляти дані про студентів, гуртожитки та факультети зручним та інтуїтивно зрозумілим способом.

Пошук та фільтрація даних: Потрібно забезпечити функціонал для ефективного пошуку та фільтрації даних за різними критеріями, щоб користувачі могли швидко знаходити необхідну інформацію.

Звітність: Система має забезпечувати можливість генерувати різноманітні звіти про облік платежів, діяльність гуртожитка та інші важливі показники, які можуть бути корисними для адміністрації.

Технічні вимоги:

Операційна система: Система повинна бути сумісна з популярною операційною системою, такою як Windows.

Процесор та об'єм оперативної пам'яті: Мінімальні вимоги щодо процесора та об'єму оперативної пам'яті будуть визначені під час розробки, з огляду на обсяг робочих процесів та швидкість обробки даних.

Місце на диску: Необхідно врахувати потреби в місці на диску для зберігання програмного забезпечення та баз даних, а також можливості зберігання додаткової інформації.

База даних: Обрана база даних повинна відповідати вимогам щодо ефективності, надійності та безпеки зберігання даних.

Ці вимоги будуть використовуватися як основа для розробки та тестування програмної системи обліку платежів для гуртожитків, забезпечуючи ефективну та надійну роботу програми.

2.2 Вибір технологій (мова програмування, платформа)

При виборі технологій для розробки системи обліку платежів для гуртожитків, було враховано кілька ключових чинників. Обґрунтування вибору мови програмування C# та платформи WinForms полягає в наступному:

Сучасність та об'єктно-орієнтований підхід: C# є мовою програмування, що активно розвивається та підтримує об'єктно-орієнтований підхід, що відповідає потребам сучасних програмних рішень.

Підтримка платформи .NET: C# є основною мовою програмування для платформи .NET, що забезпечує широкі можливості розробки та високу сумісність зі звичайними Windows-додатками.

Широка база розробників та багатофункціональність: C# має велику спільноту розробників, що сприяє доступності ресурсів для вирішення будь-яких технічних питань, а також велику кількість бібліотек та інструментів, що полегшують розробку та підтримку програм.

Обґрунтування вибору платформи WinForms:

Простота використання та навчання: WinForms надає простий та інтуїтивно зрозумілий інтерфейс для розробки Windows-додатків, що спрощує процес навчання та швидкої розробки.

Широка спільнота розробників та ресурси: WinForms має велику спільноту розробників, що сприяє наявності багато навчальних матеріалів, блогів, форумів та інших ресурсів для вирішення будь-яких питань щодо розробки.

Сумісність з платформою .NET: WinForms є одним із фреймворків для розробки Windows-додатків на платформі .NET, що забезпечує високу сумісність та інтеграцію з іншими .NET-технологіями.

Обраний набір технологій забезпечить швидку розробку, надійність та ефективність системи обліку платежів для гуртожитків, а також забезпечить зручний та інтуїтивно зрозумілий інтерфейс для користувачів. [4, 8]

2.3 Архітектурна схема програмної системи

Система обліку оплати за проживання в гуртожитках буде складатися з кількох модулів, кожен з яких буде розбиватися на різну кількість функціональних підблоків. Головним модулем буде вкладка Довідники, яка одночасно буде елементом компонування системи. Така структура, дозволить легко та інтуїтивно користуватися системою за рахунок того, що модулі розташовуються у порядку, який зазвичай використовують для вирішення схожих задач.

Наведена схема структури системи, на якій розташовані програмні модулі.



Рис. 2.1 Схема структури системи

У систему заносяться вхідні дані — інформація про студента, що обробляються, після чого система виконує перевірку введених даних і зберігає результат.

2.4 Опис структури бази даних

База даних - це набір пов'язаних даних, які зберігаються в системі за допомогою таблиць. Дозволяє створювати, редагувати та ділитися базами даних, які називаються системами управління базами даних (СУБД). Будь-яка таблиця являє собою послідовність однорідних частин, які називаються записами. Запис також може містити один або більше іменованих полів. Імена та номери полів вказуються при створенні таблиці. Кожне поле також має певний тип даних.

Представлено структуру бази даних, яка використовується для обліку оплати за проживання в студентських гуртожитках. База даних складається з кількох основних таблиць, які взаємодіють між собою для забезпечення збереження і доступу до необхідної інформації.

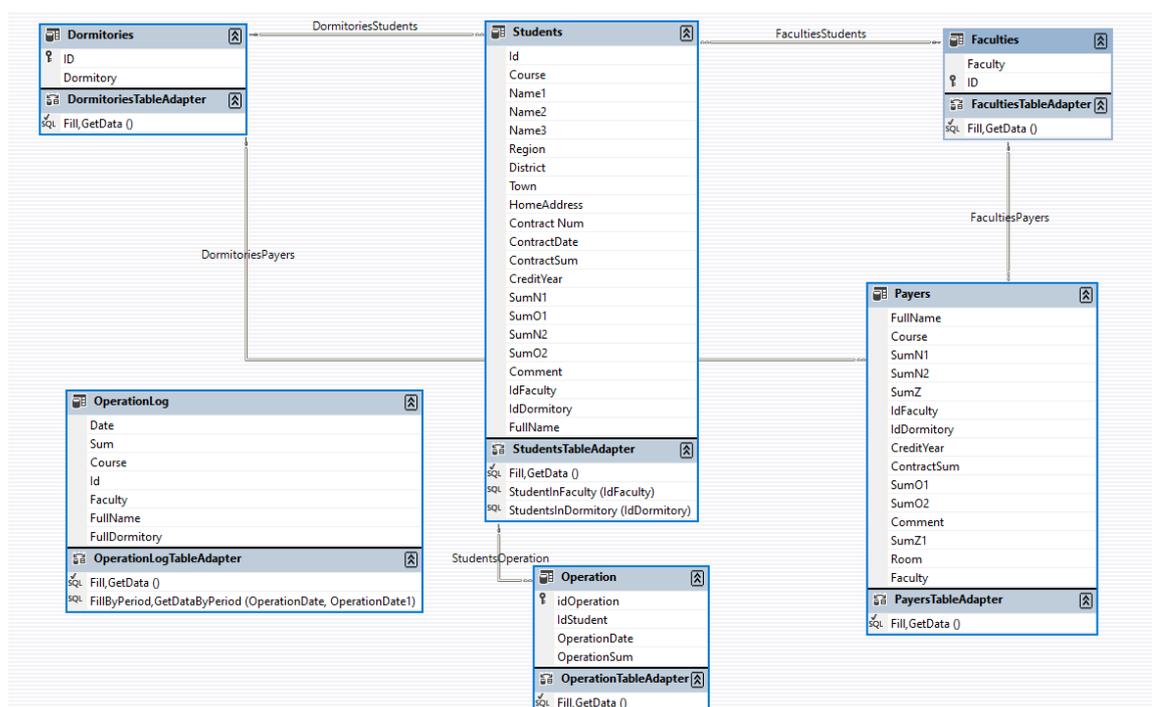


Рис. 2.2 Модель бази даних

База даних складається з наступних таблиць: “Студенти”, “Гуртожитки”, “Операції”, “Факультети”.

Структура таблиці “Студенти” представлена наступним чином:

Ім'я поля	Тип і розмір поля	Опис поля
Id	Int32	Первинний ключ

Course	byte	Курс
Surname	string	Прізвище
Name	string	Ім'я
Patronymic	string	По батькові
ContractSum	decimal	Загальна сума оплати
SumN1	decimal	Оплата за перший семестр
SumN2	decimal	Оплата за другий семестр
CreditYear	decimal	Сума за попередній рік
Comment	string	Замітки щодо студента

Таблиця 2.1 Структура таблиці “Студенти”

Таблиця складається з таких полів як Ідентифікатор, Прізвище, Ім'я, По батькові, Загальна сума оплати, Оплата за перший та другий семестр, Сума за попередній рік, Замітки про студента.

Ім'я поля	Тип і розмір поля	Опис поля
ID	Int32	Первинний ключ
Dormitory	string	Гуртожиток

Таблиця 2.2 Структура таблиці “Гуртожитки”.

Таблиця “Гуртожитки” містить зв'язок з таблицею “Студенти” по ідентифікатору, так як виражає дані, що стосуються кожного студента аналогічно із таблицею “Факультети”.

Ім'я поля	Тип і розмір поля	Опис поля
ID	Int32	Первинний ключ
Faculty	string	Назва факультету

Таблиця 2.3 Структура таблиці “Факультети”.

Таблиця “Журнал операцій” містить наступні поля.

Ім'я поля	Тип і розмір поля	Опис поля
Date	DateTime	Дата

Sum	decimal	Сумма
Course	byte	Курс
Id	Int32	Первинний ключ
Faculty	string	Назва факультету
FullName	string	Повне ім'я
FullDormitory	string	Гуртожиток

Таблиця 2.4 Структура таблиці “Журнал операцій”

2.5 Опис функціональності системи

Система містить у собі два актора, які взаємодіють із системою а також один з одним. Перший актор – це користувач який взаємодіє із системою. Другий автор – це адміністратор який може доповнювати додаток новими функціями та можливостями, і надавати необхідну технічну підтримку користувачу.

Представлена діаграма прецедентів, яка описує функції та дії акторів у системі обліку оплати за проживання в гуртожитках.

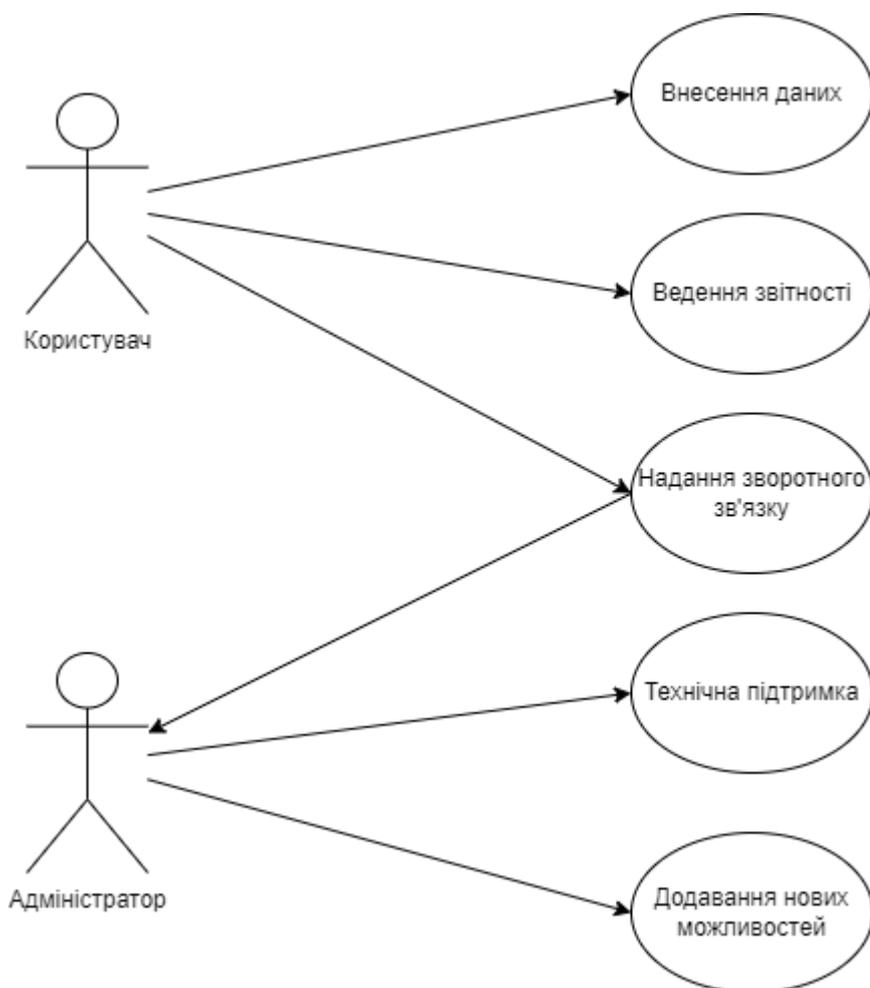


Рис. 2.3 Діаграма прецедентів системи

Висновок до розділу 2

Другий розділ присвячений тому, як впровадити програмну систему обліку платежів в гуртожитках. Описує визначення функціональних та технічних вимог до системи, вибір відповідних технологій, таких як мова програмування та платформа, а також структура бази даних. Представлено схему архітектури програмного комплексу і детально описано його функціональність. Підкреслюється важливість вибору надійних і ефективних інструментів і технологій для забезпечення стабільної роботи системи

РОЗДІЛ 3. Інтерфейс застосунку та робота користувача в системі

Система розроблена для використання на персональних комп'ютерах під управлінням операційної системи Windows з використанням технології .NET від фірми Microsoft.

Застосунок встановлюється на пристрій користувача шляхом встановлення файлу та його автоматичного налаштування. Оскільки додаток розроблений на платформі .NET, пристрій користувача повинен містити потрібні файли SDK цієї технології для коректної роботи. Для роботи додатку необхідно мати версію операційної системи не нижче Windows 7.

3.1 Концепція інтерфейсу користувача

Графічний інтерфейс користувача відповідає наступним вимогам: ясність, відповідність, виразність, послідовність. Для забезпечення цих вимог був розроблений інтерфейс програмного продукту. Він включає в себе декілька модулів. Це довідники, операції, звіти, та налаштування. Коли користувач відкриває додаток перед ним з'являється вікно в якому він може обрати потрібну йому вкладку. Всі вкладки розташовані по порядку. Після того як користувач обирає потрібну йому, він переходить на вікно, в якому відображається потрібна для роботи інформація. Вкладка довідники містить операції із списком студентів, гуртожитків, факультетів. Вкладка операції надає користувачу можливість переглянути інформацію про всі зміни в таблицях, або окремо про кожного студента.

Вкладка довідники. Слугує для виведення на екран випадаючого меню і надання користувачу можливості вибору одного із них. Меню складається з декількох елементів, а саме:

- гуртожитки — можливість додати, редагувати, та видалити;
- студенти, яке в свою чергу має можливість додати нового студента та переглянути довідник;
- факультети — аналогічно до гуртожитків;

- операції — історія всіх змін в базі даних.

Вкладка операції. Призначена для виведення інформації про зміни в базі даних. Складається з такого списку:

- всі операції;
- операції за 3-6 місяців;
- додати операцію;
- журнал по студента.

3.2 Інструкція з використання програмного продукту

При вході в систему обліку оплати за проживання перед користувачем з'являється головне вікно, в якому користувачу необхідно обрати одну із запропонованих вкладок. Вкладки розташовані одна за одною.

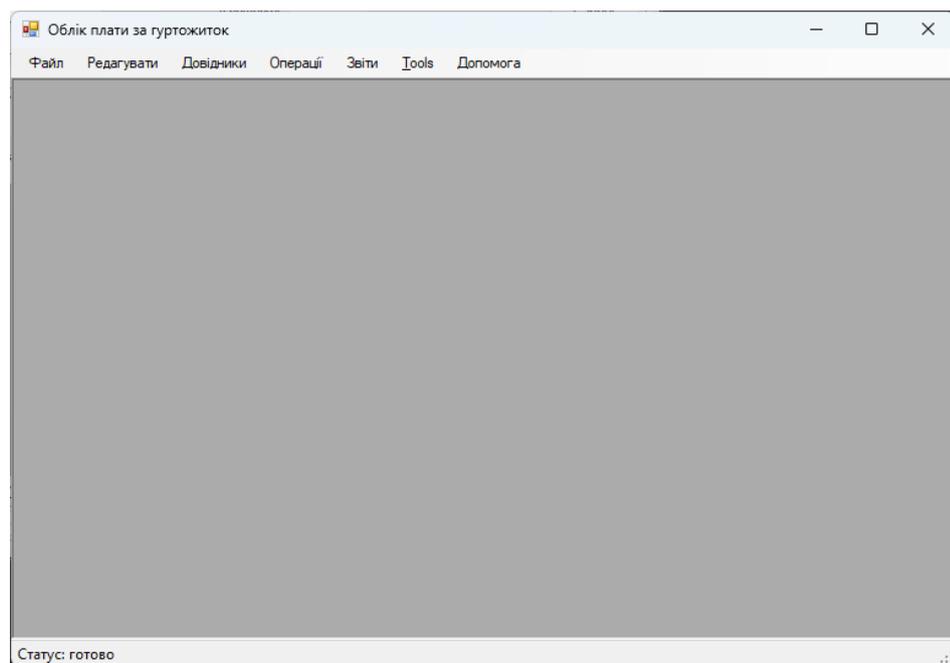


Рис. 3.1 Головне вікно

Після того як користувач обрав вкладку, він бачить перед собою список елементів керування. За допомогою цього списку користувач має доступ до функцій системи.

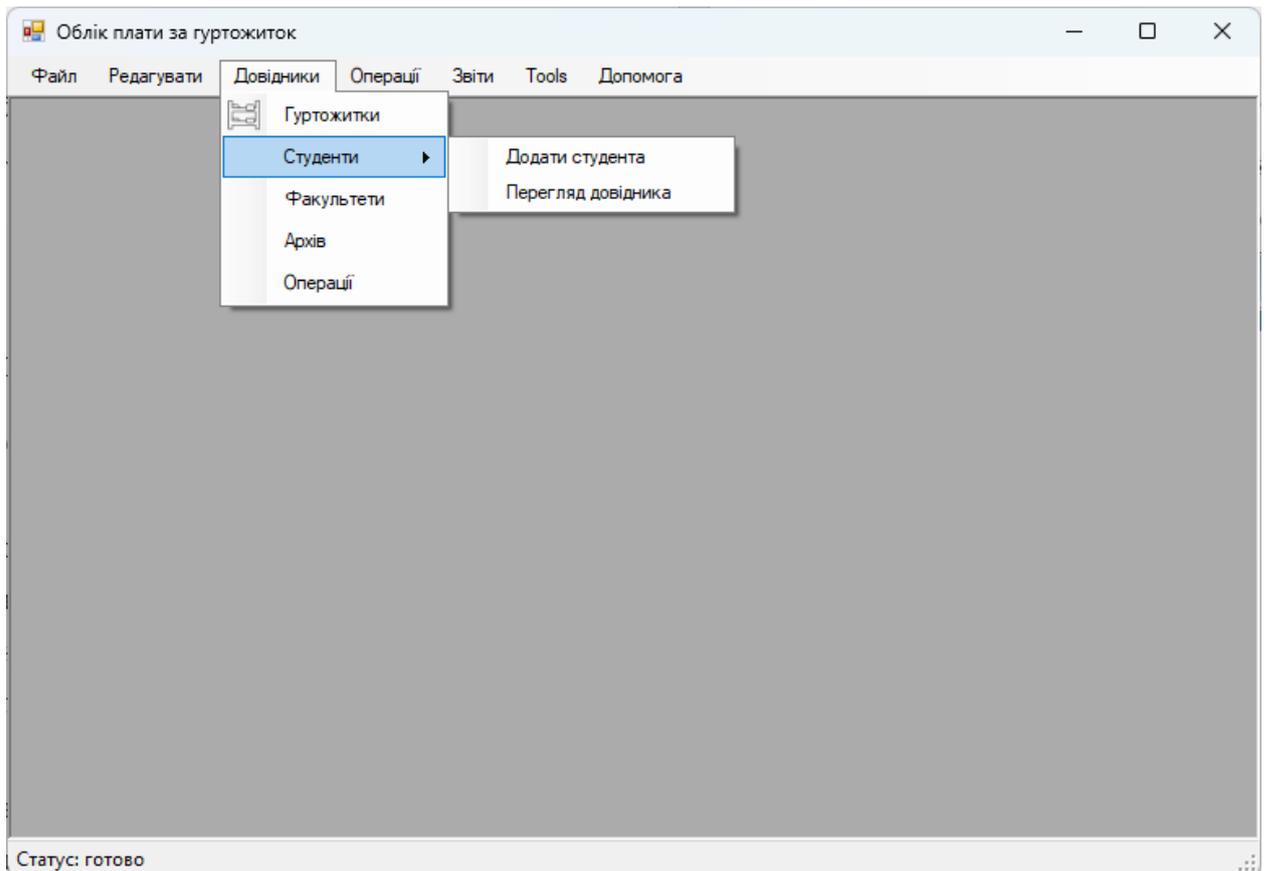


Рис. 3.2 Список елементів керування

Розроблено такі пункти роботи з системою: “Довідники”, “Студенти”, “Операції”.

За допомогою пункту меню “Довідники” можна переглянути список гуртожитків, факультетів, а також історію всіх операцій в базі даних.

За допомогою пункту меню “Студенти” можна відкрити нове вікно із формою для додавання нового студента, та переглянути довідник із повним списком студентів.

За допомогою пункту меню “Операції” користувач може переглянути звіт за операціями які він проводив за весь час, або вибірково за 3-6 місяців, і картка студента в якій можна переглянути інформацію по студенту.

3.3 Пункт меню “Довідники”

Нажимаючи елемент керування Гуртожитки, появляється нове вікно із можливостями редагування.

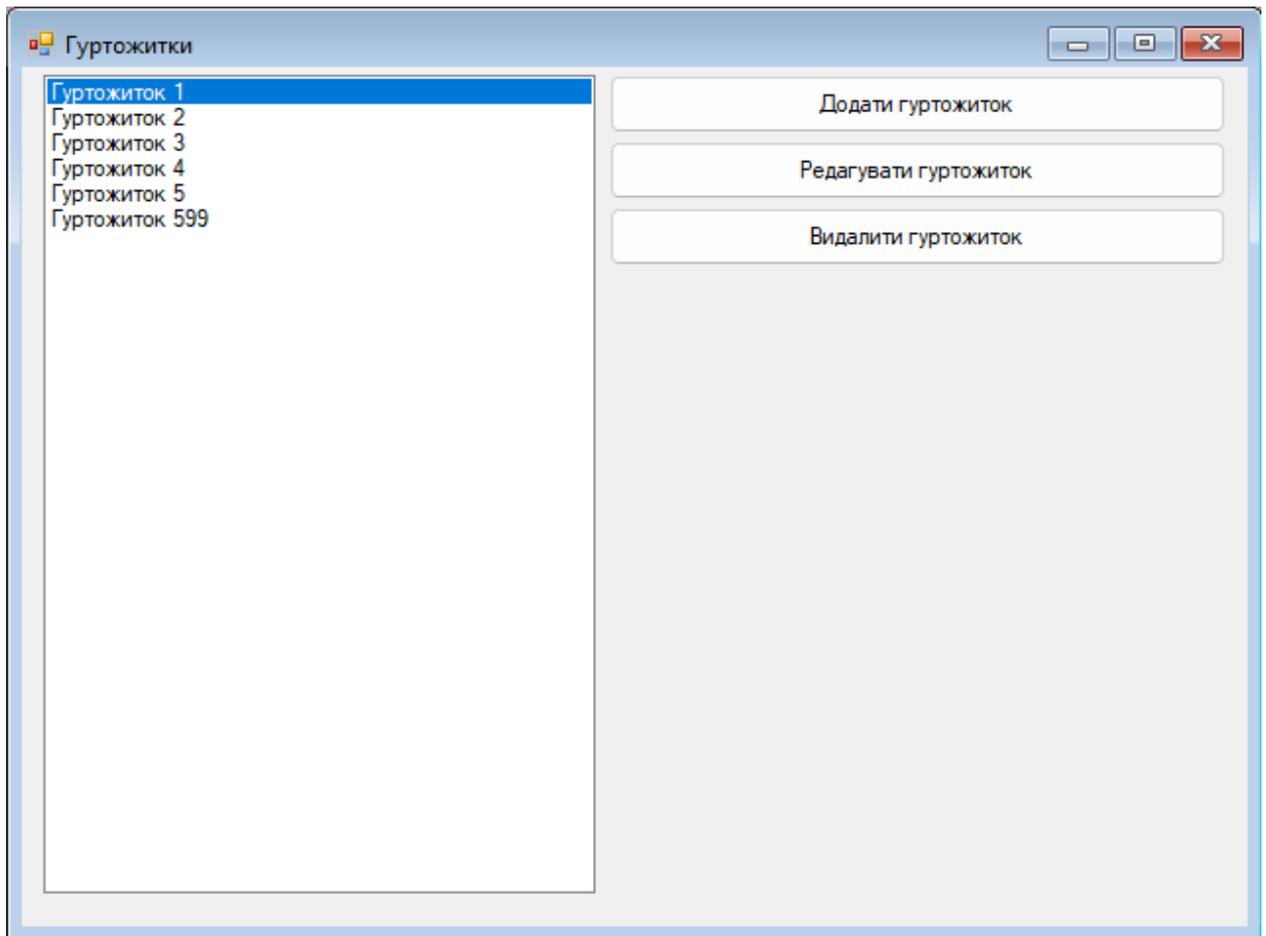


Рис. 3.3 Вікно гуртожитки

Елемент керування Операції, виглядає наступним чином, вікно із історією змін.

The screenshot shows a window titled 'FormOperations' with a toolbar containing navigation and action icons. Below the toolbar are two data tables.

	Id	Course	Name1	Name2	Name3	Regi
▶	57158	4	Авраменко	Світлана	Орестівна	
	57159	5	Адаменко	Тетяна	Степанівна	
	58393	3	Кондратюк	Світлана	Олегівна	
	58394	3	Коновал	Наталія	Володимирівна	
	58396	3	Коноваленко	Христина-Вікторія	Юрівна	
	58397	3	Коновалов	Наталія	Сергіївна	
	58398	3	Коновалова	Соломія	Іванівна	
	58399	3	Кононенко	Ірина	Анатоліївна	
	58650	4	Міняйло	Тетяна	Олександрівна	
	57165	4	Адамчук	Роман	Сергійович	

	idOperation	IdStudent	OperationDate	OperationSum
▶	90589	57158	16.12.2022 19:33	1657
	90590	57158	04.04.2023 16:42	1255,4
	90591	57158	12.02.2023 8:16	1880,9

Рис. 3.4 Вікно операції

3.4 Пункт меню “Студенти”

Для того щоб відкрити вікно для заповнення даних про студента потрібно на елементі списку натиснути кнопку “Додати студента”.

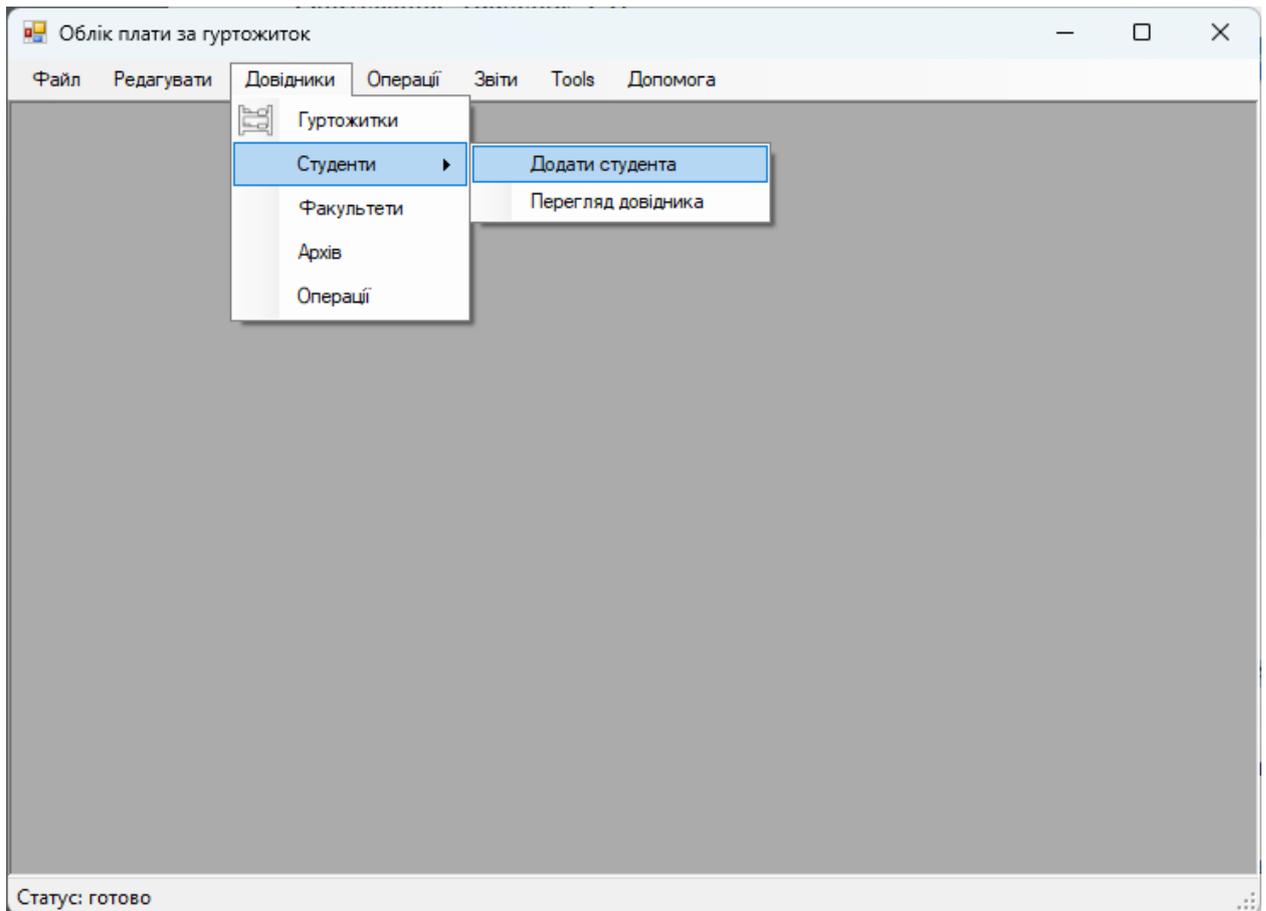


Рис. 3.5 Пункт меню "Додати студента"

Вигляд вікна для введення даних студента такий: всі поля мають вбудовану валідацію для уникнення введення некоректної інформації.

Поля для введення:

- "Гуртожиток": тут можна вибрати гуртожиток, у якому проживає студент.
- "Вартість": тут можна вказати вартість проживання в гуртожитку.
- "Курс": тут можна вказати курс навчання студента.
- "Семестр": тут можна вибрати семестр, за який вноситься інформація.
- "Прізвище": тут можна ввести прізвище студента.
- "Ім'я": тут можна ввести ім'я студента.
- "По батькові": тут можна ввести по батькові студента.
- "Примітка": тут можна ввести примітку про студента.

Кнопки:

- "Додати": ця кнопка використовується для додавання нового рядка даних про студента.
- "Видалити усі": ця кнопка використовується для видалення всіх рядків даних про студентів.
- "Видалити один рядок": ця кнопка використовується для видалення виділеного рядка даних про студента.
- "Ok": ця кнопка використовується для збереження введених даних.
- "Скасувати": ця кнопка використовується для скасування введених даних.

Введення списку студентів

Гуртожиток: Гуртожиток 1

Курс: 4

Прізвище: !

Ім'я: Це поле обов'язкове для заповнення!

По батькові:

Примітка:

Вартість:

I семестр:

II семестр:

Додати

Ok

Скасувати

	Курс	Прізвище	Ім'я	По батькові	Примітка	Вартість
*						

Видалити усі

Видалити один рядок

Рис. 3.6 Вікно занесення студента в базу даних

3.5 Пункт меню “Перегляд довідника”

Пункт "Перегляд списку студентів" використовується для перегляду інформації про студентів. Вікно має наступний вигляд.

Гуртожиток: Кількість студентів:

	Курс	Прізвище	Ім'я	По батькові	Примітка	Мин. рік	I семестр	II семестр
	4	Авраменко	Світлана	Орестівна	к.220 АТО 50%	0,0000	2400	2400
	5	Адаменко	Тетяна	Степанівна	к.217	0,0000	2600	2600
	3	Кондратюк	Світлана	Олегівна	к.223	0,0000	4800	4800
	3	Коновал	Наталія	Володимирівна	к.223	-2600	4800	4800
▶	3	Коноваленко	Христина-Вікторія	Юрївна	к.222а	0,0000	4800	4800
	3	Коновалов	Наталія	Сергіївна	к.222а	0,0000	4800	4800
	3	Коновалова	Соломія	Іванівна	к.210	0,0000	4800	4800
	3	Кононенко	Ірина	Анатоліївна	к.523	0,0000	4800	4800
	4	Адамчук	Роман	Сергійович	к.327	0,0000	4800	4800
	4	Александрова	Діана	Русланівна	к.527	0,0000	2600	2600
	4	Алексеева	Марина	Олександрівна	к.114 АТО 50%	0,0000	2400	2400
	4	Алієв	Мирослава	Юрївна	к.114 АТО 50%	0,0000	2400	2400
	5	Андреева	Маріна	Володимирівна	к.217	0,0000	2600	2600
	5	Андреев	Анна	Романівна	к.415	0,0000	2600	2600
	3	Конончук	Дарія	Костянтинівна	к.140 АТО 50%	0,0000	2400	2400
	3	Корж	Тетяна	Мирславівна	к.407	0,0000	4800	4800
	3	Корнієнко	Володимир	Євгенович	к.128	-3266	4800	4800

Рис. 3.7 Вікно списку студентів

3.6 Пункт меню “Картка студента”

Дане вікно призначене для перегляду інформації про мешканців використовуючи фільтрацію по грутожиткам та факультетам.

The screenshot shows a window titled "Картка студента" with the following elements:

- Filters:**
 - Гуртожиток 3
 - фіз.мат
 - Берега
 - Дмитро
 - Валентинович
- Student List:**

Студенти
Берега Дмитро Валентинович
Берега Юрій Олегович
Берега Олена Володимирівна
Берега Денис Олександрович
Білан Арсен
Кривонос Ярослав Васильович
- Summary:**
 - Вартість: 5 200,00
 - Минулий рік: 0,00
- Transaction Table:**

	Дата операції	Сума операції
▶	20.09.2023 17:32	1 551,70
	28.11.2021 16:00	1 509,40
	23.05.2022 4:52	1 667,20
	19.05.2021 22:54	1 186,80
	08.02.2022 1:14	1 261,10
- Account Details:**

	Нараховано	Сплачено	Залишок
1 сем	2 600,00	0,00	2 600,00
2 сем	2 600,00	0,00	2 600,00
с.147к.77			
- Buttons:** Recalc

Рис. 3.8 Вікно картки студента

3.7 Пункт меню “Операції”

Для перегляду інформацію по проведенню платежів потрібно скористатися одним із представлених підпунктів.

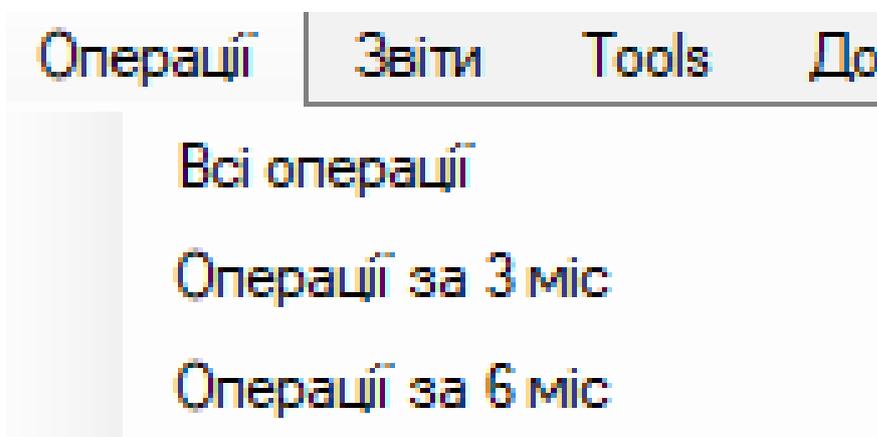


Рис. 3.9 Вибір підпункту

Також реалізована можливість введення дати вручну або вибору мишкою через піктограму календаря.

Дата	Загальна кількість	Прізвище та ім'я	Гуртожиток	Факультет
12.05.2023	1 780,60	Стеценко Олена Михайлівна	Гуртожиток 5	empty
12.05.2023	1 450,30	Макаренко Галина Володимирівна	Гуртожиток 5 315/2	пед
12.05.2023	1 801,70	Гура Дмитро Олександрович	Гуртожиток 4 511/2	фіз.вих
12.05.2023	1 224,80	Кушніренко Володимир Дмитрович	Гуртожиток 4 113/2	іноз
12.05.2023	1 352,30	Величко Олександр Іванович	Гуртожиток 3 с.335к.53	empty
13.05.2023	1 740,60	Герасименко Юрій Юрійович	Гуртожиток 4 406/2 до 31....	іст
13.05.2023	1 589,80	Степанов Олеся Дмитрівна	Гуртожиток 3 с.221к.13 п...	empty
13.05.2023	1 304,00	Костюченко Андріана Романівна	Гуртожиток 1 к.225	укр
14.05.2023	1 587,20	Вороніна Андрій Васильович	Гуртожиток 4 603/3	іст
14.05.2023	1 328,40	Івченко Наталія Андріївна	Гуртожиток 5 908/2 до 31....	СОПСР
14.05.2023	1 471,50	Сирота Вікторія Віталіївна	Гуртожиток 5 403/3	СОПСР
14.05.2023	1 552,20	Єфіменко Ірина Костянтинівна	Гуртожиток 5 518/2	пед
14.05.2023	1 278,90	Суріменко Юлія Володимирівна	Гуртожиток 3 с.233к.24	пед

Рис. 3.10 Вікно журналу операцій

Висновки до розділу 3

Третій розділ розглядає питання інтерфейсу користувача та його роботи з системою. Особливу увагу приділено ключовим вкладкам системи, процесам інсталяції та системним вимогам. Надано детальну інструкцію з використання програмного продукту, включаючи опис пунктів меню, що дозволяють ефективно управляти даними та здійснювати необхідні операції. Висвітлено концепцію інтерфейсу користувача, яка забезпечує зручність та інтуїтивність використання застосунку.

ВИСНОВКИ

В рамках даної дипломної роботи було проведено дослідження і розробку інформаційної системи для обліку оплати витрат на проживання в студентських гуртожитках. Проаналізована сучасна інформаційна система, яка використовується для оплати проживання в гуртожитках. Виявлені сильні сторони, які послужили основою для формулювання вимог до нової системи. Визначено основні функціональні вимоги до системи: введення платежів, формування списків, автоматизація звітності. Що в свою чергу дозволило створити технічні завдання для розробки програмних продуктів. Розроблено схему архітектури програмного комплексу, вибір технології мови програмування, платформи був зроблений на основі аналізу вимог до продуктивності і надійності системи. Створено додаток, що відповідає певним вимогам. Система була протестована і підтверджена на відповідність функціональним і технічним вимогам. Розроблено докладні інструкції по використанню програмного продукту, які спрощують користувачам навчання роботі з системою. З впровадженням цієї інформаційної системи процес обліку оплати проживання в студентських гуртожитках буде більш автоматизованим, точним і займе менше часу. Це може значно підвищити ефективність управління гуртожитками, знизити ризик помилок в обліку платежів і поліпшити якість обслуговування студентів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. АС «Студмістечко» - АСУ «ВНЗ» URL: <https://vuz.osvita.net/as-studmistechko/>
2. Гуртожитки К-ПНУ. URL: <https://profkom.kpnu.edu.ua/poselennia-v-hurtozhytky-2/>
3. Гавришко Володимир, Розробка інформаційної системи обліку оплати за проживання в гуртожитках. Збірник матеріалів наукової конференції за підсумками науково-дослідної роботи здобувачів вищої освіти фізико-математичного факультету у 2023-2024 н. р. Кам'янець-Подільський, 2024. с.18-21
4. Desktop Guide (Windows Forms .NET): веб-сайт. URL: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview>
5. Створення додатку Windows Forms у Visual Studio з використанням C#: веб-сайт. URL: <https://learn.microsoft.com/uk-ua/visualstudio/ide/create-csharp-winform-visual-studio>
6. Ян Гріффітс, Метью Адамс, Джессі Ліберті Programming C# 8.0: Build Windows, Web, and Desktop Applications 2020. 797 с. URL: <https://www.amazon.com/Programming-8-0-Windows-Desktop-Applications/dp/1492056812>
7. Ендрю Троелсен Pro C# 7: With .NET and .NET Core 2017. 1437 с. URL: <https://www.amazon.com/Pro-NET-Core-Andrew-Troelsen/dp/1484230175>
8. Кріс Селлс, Майкл Вайнхардт Windows Forms Programming in C# 2006. 1032 с. URL: <https://www.amazon.com/Windows-Forms-Programming-Microsoft-Development/dp/0321267966>
9. Онлайн посібник по C# від Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>

ДОДАТКИ

ДОДАТОК А.

Реалізація вікна Гуртожитки

```

using DormitoryFeeTrackerNet.Properties;
using System;
using System.Data.OleDb;
using System.Windows.Forms;

namespace DormitoryFeeTrackerNet
{
    public partial class FormDormitories : Form
    {
        public FormDormitories()
        {
            InitializeComponent();
        }

        private void FormDormitories_Load(object sender, EventArgs e)
        {
            studentsTableAdapter.Fill(kpnuPayDataSet.Students);
            dormitoriesTableAdapter.Fill(kpnuPayDataSet.Dormitories);
            listBox1.SelectedIndex = 0;
        }

        private void SaveDatabase()
        {
            Validate();
            dormitoriesBindingSource.EndEdit();
            tableAdapterManager.UpdateAll(kpnuPayDataSet);
        }

        private void ButtonAdd_Click(object sender, EventArgs e)
        {
            buttonEdit.Enabled = false;
            buttonRemove.Enabled = false;
            label1.Visible = true;
            label1.Text = Resources.messageDormitoryAddNew;
            textBoxDormitory.Visible = true;
            buttonSave.Visible = true;
            buttonCancel.Visible = true;
        }

        private void ButtonEdit_Click(object sender, EventArgs e)
        {
            buttonAdd.Enabled = false;
            buttonRemove.Enabled = false;
            label1.Visible = true;
            label1.Text = Resources.messageDormitoryEdit;
            var row = GetRowById();
            textBoxDormitory.Text = row.Dormitory;
            textBoxDormitory.Visible = true;
            buttonSave.Visible = true;
            buttonCancel.Visible = true;
        }

        private KpnuPayDataSet.DormitoriesRow GetRowById()
        {
            var id = int.Parse(listBox1.SelectedValue.ToString());
            return kpnuPayDataSet.Dormitories.FindByID(id);
        }

        private void ButtonRemove_Click(object sender, EventArgs e)
        {

```

```

var dialogResult = MessageBox.Show(
    Resources.AreYouSure,
    Resources.Question,
    MessageBoxButtons.YesNo,
    MessageBoxIcon.Question);
if (dialogResult == DialogResult.No)
{
    return;
}
var id = int.Parse(listBox1.SelectedValue.ToString());
var students = studentsTableAdapter.StudentsInDormitory(id);
if (students > 0)
{
    var message =
        string.Format(Resources.messageDormitoryHaveStudents,
students);
    MessageBox.Show(message, Resources.Warning,
        MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    return;
}
dormitoriesBindingSource.RemoveCurrent();
SaveDatabase();
}

private void ButtonSave_Click(object sender, EventArgs e)
{
    if (buttonAdd.Enabled)
    {
        kpnuPayDataSet.Dormitories.AddDormitoriesRow(textBoxDormitory.Text);
    }
    if (buttonEdit.Enabled)
    {
        var row = GetRowById();
        row.Dormitory = textBoxDormitory.Text;
    }
    try
    {
        SaveDatabase();
    }
    catch (OleDbException ex)
    {
        MessageBox.Show(ex.Message, "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        dormitoriesTableAdapter.Fill(kpnuPayDataSet.Dormitories);
        ClearControls();
    }
}

private void ButtonCancel_Click(object sender, EventArgs e)
{
    ClearControls();
}

private void ClearControls()
{
    labell.Visible = false;
    textBoxDormitory.Visible = false;
    textBoxDormitory.Clear();
    buttonAdd.Enabled = true;
    buttonEdit.Enabled = true;
    buttonRemove.Enabled = true;
    buttonSave.Visible = false;
}

```

```
        buttonCancel.Visible = false;
    }
}
```

ДОДАТОК Б.

Реалізація додавання студентів в базу даних

```

using DormitoryFeeTrackerNet.Properties;
using System;
using System.ComponentModel;
using System.Data;
using System.Windows.Forms;

namespace DormitoryFeeTrackerNet
{
    public partial class FormStudentAdd : Form
    {
        private bool isClosing = false;
        public FormStudentAdd()
        {
            InitializeComponent();
        }

        private void FormStudentAdd_Load(object sender, EventArgs e)
        {
            this.dormitoriesTableAdapter.Fill(this.kpnuPayDataSet.Dormitories);
        }

        private void FormStudentAdd_FormClosing(object sender,
        FormClosingEventArgs e)
        {
            isClosing = true;
        }

        private void okButton_Click(object sender, EventArgs e)
        {
            dormitoriesStudentsBindingSource.EndEdit();
            tableAdapterManager.UpdateAll(kpnuPayDataSet);
        }

        private void courseTextBox_KeyPress(object sender, KeyPressEventArgs e)
        {
            TextBox textBox = (TextBox)sender;

            if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }

            if (e.KeyChar != (char)Keys.Back && (textBox.Text + e.KeyChar).Length
> 1)
            {
                e.Handled = true;
            }
        }

        private void courseTextBox_Validating(object sender, CancelEventArgs e)
        {
            TextBox textBox = (TextBox)sender;
            int number;

            if (!isClosing)
            {
                if (string.IsNullOrEmpty(textBox.Text))
                {
                    e.Cancel = true;
                    errorProvider.SetError(textBox, Resources.messageRequired);
                }
            }
        }
    }
}

```

```

        else if (!int.TryParse(textBox.Text, out number) || number < 1 ||
number > 4)
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Введіть лише одну цифру в
межах від 1 до 4!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void lastNameTextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= 'A' &&
e.KeyChar <= 'Z') || char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
    }
}

private void lastNameTextBox_Validating(object sender, CancelEventArgs e)
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void lastNameTextBox_Validated(object sender, EventArgs e)
{
    TextBox textBox = (TextBox)sender;

    if (!string.IsNullOrEmpty(textBox.Text))
    {
        string input = textBox.Text.Trim();
        textBox.Text = char.ToUpper(input[0]) +
input.Substring(1).ToLower();
    }
}

private void firstNameTextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= 'A' &&
e.KeyChar <= 'Z') || char.IsDigit(e.KeyChar))

```

```

        {
            e.Handled = true;
        }
    }

private void firstNameTextBox_Validating(object sender, CancelEventArgs e)
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void firstNameTextBox_Validated(object sender, EventArgs e)
{
    TextBox textBox = (TextBox)sender;

    if (!string.IsNullOrEmpty(textBox.Text))
    {
        string input = textBox.Text.Trim();
        textBox.Text = char.ToUpper(input[0]) +
input.Substring(1).ToLower();
    }
}

private void patronymicTextBox_KeyPress(object sender, KeyPressEventArgs
e)
{
    if ((e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= 'A' &&
e.KeyChar <= 'Z') || char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
    }
}

private void patronymicTextBox_Validated(object sender, EventArgs e)
{
    TextBox textBox = (TextBox)sender;

    if (!string.IsNullOrEmpty(textBox.Text))
    {
        string input = textBox.Text.Trim();
        textBox.Text = char.ToUpper(input[0]) +
input.Substring(1).ToLower();
    }
}

private void contractSumTextBox_KeyPress(object sender, KeyPressEventArgs
e)
{
    TextBox textBox = (TextBox)sender;

```

```

        if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
        {
            e.Handled = true;
        }
    }

private void contractSumTextBox_TextChanged(object sender, EventArgs e)
{
    int contractSum, sumN1, sumN2;

    if (int.TryParse(contractSumTextBox.Text, out contractSum))
    {
        sumN1 = contractSum / 2;
        sumN2 = contractSum - sumN1;

        sumN1TextBox.TextChanged -= sumN1TextBox_TextChanged;
        sumN2TextBox.TextChanged -= sumN2TextBox_TextChanged;

        sumN1TextBox.Text = sumN1.ToString();
        sumN2TextBox.Text = sumN2.ToString();

        sumN1TextBox.TextChanged += sumN1TextBox_TextChanged;
        sumN2TextBox.TextChanged += sumN2TextBox_TextChanged;
    }
}

e) private void contractSumTextBox_Validating(object sender, CancelEventArgs
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void sumN1TextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    TextBox textBox = (TextBox)sender;

    if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
    {
        e.Handled = true;
    }
}

private void sumN1TextBox_TextChanged(object sender, EventArgs e)
{
    int contractSum, sumN1, sumN2;

    if (int.TryParse(sumN1TextBox.Text, out sumN1) &&
int.TryParse(sumN2TextBox.Text, out sumN2))
    {
        contractSum = sumN1 + sumN2;
    }
}

```

```

        contractSumTextBox.TextChanged -= contractSumTextBox_TextChanged;
        contractSumTextBox.Text = contractSum.ToString();
        contractSumTextBox.TextChanged += contractSumTextBox_TextChanged;
    }
}

private void sumN1TextBox_Validating(object sender, CancelEventArgs e)
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void sumN2TextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    TextBox textBox = (TextBox)sender;

    if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
    {
        e.Handled = true;
    }
}

private void sumN2TextBox_TextChanged(object sender, EventArgs e)
{
    int contractSum, sumN1, sumN2;

    if (int.TryParse(sumN1TextBox.Text, out sumN1) &&
int.TryParse(sumN2TextBox.Text, out sumN2))
    {
        contractSum = sumN1 + sumN2;

        contractSumTextBox.TextChanged -= contractSumTextBox_TextChanged;
        contractSumTextBox.Text = contractSum.ToString();
        contractSumTextBox.TextChanged += contractSumTextBox_TextChanged;
    }
}

private void sumN2TextBox_Validating(object sender, CancelEventArgs e)
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;

```

```

        errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
    }
    else
    {
        e.Cancel = false;
        errorProvider.SetError(textBox, "");
    }
}
else
{
    e.Cancel = false;
}
}

e) private void creditYearTextBox_KeyPress(object sender, KeyPressEventArgs
{
    TextBox textBox = (TextBox)sender;

    if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
    {
        e.Handled = true;
    }
}

e) private void creditYearTextBox_Validating(object sender, CancelEventArgs
{
    TextBox textBox = (TextBox)sender;
    if (!isClosing)
    {
        if (string.IsNullOrEmpty(textBox.Text))
        {
            e.Cancel = true;
            errorProvider.SetError(textBox, "Це поле обов'язкове для
заповнення!");
        }
        else
        {
            e.Cancel = false;
            errorProvider.SetError(textBox, "");
        }
    }
    else
    {
        e.Cancel = false;
    }
}

private void buttonAdd_Click(object sender, EventArgs e)
{
    if (!ValidateChildren())
        return;

    string lastName = lastNameTextBox.Text;
    string firstName = firstNameTextBox.Text;
    string patronymic = patronymicTextBox.Text;
    string comment = commentTextBox.Text;

    byte course = byte.Parse(courseTextBox.Text);
    int creditYear = int.Parse(creditYearTextBox.Text);
    int sumN1 = int.Parse(sumN1TextBox.Text);
    int sumN2 = int.Parse(sumN2TextBox.Text);
    int cost = int.Parse(contractSumTextBox.Text);

    string[] row = {

```

```

course.ToString(),
lastName,
firstName,
patronymic,
comment,
creditYear.ToString(),
sumN1.ToString(),
sumN2.ToString()
};

var dataRowView = dormitoriesBindingSource.Current as DataRowView;
var dormitoriesRow = dataRowView.Row as KpnuPayDataSet.DormitoriesRow;
kpnuPayDataSet.Students.AddStudentsRow(
    course,
    lastName,
    firstName,
    patronymic,
    string.Empty,
    string.Empty,
    string.Empty,
    string.Empty,
    0,
    DateTime.Today,
    cost,
    creditYear,
    sumN1,
    0,
    sumN2,
    0,
    comment,
    null,
    dormitoriesRow);
}

private void deleteButton_Click(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0)
    {
        dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);
    }
}

private void deleteAllButton_Click(object sender, EventArgs e)
{
    kpnuPayDataSet.Students.Clear();
}

private void exitButton_Click(object sender, EventArgs e)
{
    bool hasData = false;

    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        foreach (DataGridViewCell cell in row.Cells)
        {
            if (cell.Value != null)
            {
                string cellValue = cell.Value.ToString().Trim();
                if (!string.IsNullOrEmpty(cellValue))
                {
                    hasData = true;
                    break;
                }
            }
        }
    }
    if (hasData)

```

```
        {
            break;
        }
    }

    if (hasData)
    {
        DialogResult result = MessageBox.Show("Ви впевнені? Дані можуть  
бути втрачені.", "Попередження", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

        if (result == DialogResult.Yes)
        {
            this.Close();
        }
    }
    else
    {
        this.Close();
    }

    isClosing = false;
    this.Close();
}
}
}
```

Реалізація формування звіту

```

using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Text;
using System.Windows.Forms;

namespace DormitoryFeeTrackerNet
{
    public partial class FormStudents : Form, IPrintable
    {
        public bool IsSimplePrint => throw new NotImplementedException();

        public FormStudents()
        {
            InitializeComponent();
        }

        private void FormStudents_Load(object sender, EventArgs e)
        {
            this.studentsTableAdapter.Fill(this.kpnuPayDataSet.Students);
            this.dormitoriesTableAdapter.Fill(this.kpnuPayDataSet.Dormitories);
        }

        private void dormitoriesBindingSource_CurrentChanged(object sender,
            EventArgs e)
        {
            if (sender is BindingSource bindingSource)
            {
                if (bindingSource.Current is DataRowView dataRowView)
                {
                    if (dataRowView.Row is KpnuPayDataSet.DormitoriesRow row)
                    {
                        textBoxStudentNumber.Text =
studentsTableAdapter.StudentsInDormitory(row.ID).ToString();
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            var form = MdiParent as Form1;
            form.ShowSingleChildForm<FormStudentAdd>();
        }

        public LinkedList<string> GetPrintContent()
        {
            LinkedList<string> lines = new LinkedList<string>();
            var sb = new StringBuilder();
            sb.AppendLine("Список студентів");
            sb.AppendLine("-----");
            sb.AppendLine("Курс | Прізвище, ім'я та по батькові | Примітка");
            sb.AppendLine("| За рік | I сем | II сем");
            sb.AppendLine("-----");

            var linesCount = 6;

```

```

        foreach (KpnuPayDataSet.StudentsRow row in
this.kpnuPayDataSet.Students.Rows)
        {
            var course = DBNull.Value.Equals(row["Course"]) ? string.Empty :
row["Course"].ToString();
            int totalSpaces = 4;
            int spacesToAdd = Math.Max(0, totalSpaces - course.Length);
            string padding = new string(' ', spacesToAdd - 1);
            course = padding + course.PadRight(totalSpaces - padding.Length);

            var name1 = DBNull.Value.Equals(row["Name1"]) ? string.Empty :
row["Name1"].ToString();
            var name2 = DBNull.Value.Equals(row["Name2"]) ? string.Empty :
row["Name2"].ToString();
            var name3 = DBNull.Value.Equals(row["Name3"]) ? string.Empty :
row["Name3"].ToString();
            var fullName = $"{name1} {name2} {name3}".Trim();

            var comment = DBNull.Value.Equals(row["Comment"]) ? string.Empty :
row["Comment"].ToString().Trim();
            const int maxCommentLength = 6;
            if (comment.Length > maxCommentLength)
            {
                comment = comment.Substring(0, maxCommentLength) + "...";
            }

            var creditYear = DBNull.Value.Equals(row["CreditYear"]) ?
string.Empty : row["CreditYear"].ToString();
            var sumN1 = DBNull.Value.Equals(row["SumN1"]) ? string.Empty :
row["SumN1"].ToString();
            var sumN2 = DBNull.Value.Equals(row["SumN2"]) ? string.Empty :
row["SumN2"].ToString();

            var str = $"{course} | {fullName,-37} | {comment,-8} |
{creditYear,-6} | {sumN1,-5} | {sumN2,-5}";

            sb.AppendLine(str);
            if (++linesCount == 68)
            {
                lines.AddLast(sb.ToString());
                sb.Clear();
                linesCount = 0;
            }
        }
        return lines;
    }

    public string GetPrintName()
    {
        return "Students";
    }

    public LinkedList<string> GetTextContent()
    {
        throw new NotImplementedException();
    }

    public LinkedList<Stream> GetGraphicsContent()
    {
        throw new NotImplementedException();
    }
}
}

```