

Державний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Програмна платформа кампусу гуртожитків ЗВО»

Студента 2 курсу, 2мз групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

підпис студента

Удовиці Олега
Володимировича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Рзаєва Світлана
Леонідівна

Гарант освітньої програми
кандидат педагогічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Котенко Наталія
Олексіївна

КИЇВ – 2023

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«13» грудня 2023 р.

Завдання

на випускню кваліфікаційну роботу студентіві

Удовиця Олега Володимировича

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «на тему: «Програмна платформа кампусу гуртожитків ЗВО»

Затверджена наказом ректора від «9» грудня 2022 р. № 3339

2. Строк здачі студентом закінченої роботи 1 грудня 2023 р.

3. Цільова установка та вихідні дані до роботи

Мета роботи: дослідження існуючих інформаційно-аналітичних систем обліку жителів кампусу та розробка програмного продукту, котрий буде зберігати інформацію про жителів студмістечка, дозволить, завдяки аналітичним інструментам, обирати кімнату для поселення студентів, спростить час та завдання комендантам гуртожитків.

Об'єкт дослідження: облік студентів, котрі проживають в кампусі університету.

Предмет дослідження: інформаційно-аналітична система обліку студентів-жителів кампусу.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОСНОВНІ ФУНКЦІОНАЛЬНІ ХАРАКТЕРИСТИКИ АНАЛІТИЧНОЇ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

1.1. Загальні відомості про системи обліку жителів

1.2. Застосування баз даних обліку жителів кампусу

1.3. Технічне завдання

1.4. Висновки до розділу 1

РОЗДІЛ 2. ПРЕДМЕТНА ОБЛАСТЬ РОЗРОБКИ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

2.1. Відбір інформації для інформаційно-аналітичної системи обліку жителів кампусу

2.2. Постановка задачі

2.3. Аналіз предмету розробки інформаційно-аналітичної системи обліку жителів кампусу

2.4. Опис алгоритму вибору кімнати для нового мешканця кампусу

2.5. Створення інформаційної бази в субд sqlite

2.6. Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

3.1. Створення інформаційно-аналітичної системи у середовищі visual studio .net

3.2. Опис функціональних характеристик інформаційно-аналітичної системи обліку жителів кампусу

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

АНОТАЦІЯ

Дипломна робота на тему «Програмна платформа кампусу гуртожитків ЗВО» містить 51 сторінку, 18 рисунків та 5 додатків. Перелік посилань налічує 29 найменувань.

Мета дослідження: роботи є дослідження існуючих інформаційно-аналітичних систем обліку жителів кампусу та розробка програмного продукту, котрий буде зберігати інформацію про жителів студмістечка, дозволить, завдяки аналітичним інструментам, обирати кімнату для поселення студентів, спростить час та завдання комендантам гуртожитків.

Об'єкт дослідження: є облік студентів, котрі проживають на кампусі університету.

Предметом дослідження є інформаційно-аналітична система обліку студентів-жителів кампусу.

У першому розділі проаналізовано існуючі у світі інформаційно-аналітичні системи обліку жителів кампусу, складено технічне завдання, де визначено мету, призначення та передумови створення інформаційно-аналітичної системи обліку жителів кампусу, описано вимоги до розробки програми та обраної платформи й засоби для її реалізації.

У другому розділі проаналізовано інформацію про студентів, створено анкету нового жителя кампусу та наведено її зразок, описано процес алгоритм вибору кімнати для нового жителя системою та створено і наповнено базу даних для подальшого її використання у інформаційно-аналітичній системі обліку жителів кампусу.

У третьому розділі описано процес розробки інформаційно-аналітичної системи обліку жителів кампусу. Тут подано інформацію про інтерфейс програми, описано основні моменти розробки системи та наведено фрагменти коду, що використовувався під час роботи.

ANNOTATION

The thesis on the topic "Development of analytical accounting system for campus residents" Program platform of the ZVO dormitory campus contains 51 pages, 18 drawings, and 5 appendix. The list of links includes 29 items.

The purpose of the study research of existing information-analytical accounting systems for campus residents and development of a software product that will store information about campus residents and will allow, thanks to AI tools, to choose a room for students, simplify tasks for dormitory commanders.

Research subject: an accounting of campus residents.

The study's subject information-analytical accounting system for campus residents.

The first section analyzes the existing information and analytical systems of campus residents, the technical task, which defines the purpose and prerequisites for creating an information-analytical system of campus residents, describes the requirements for program development and selected platform and tools for its implementation.

The second section analyzes information about students, creates a questionnaire for a new campus resident and provides a sample, describes the process of algorithm for choosing a room for a new resident by the system and created and filled a database for further use in the information-analytical system of campus residents.

The third section describes the process of developing an information-analytical accounting system for campus residents. There is the information about the program interface, which describes the main points of system development and fragments of code used during the development of program.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ОСНОВНІ ФУНКЦІОНАЛЬНІ ХАРАКТЕРИСТИКИ ПРОГРАМНОЇ ПЛАТФОРМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ	7
<i>1.1. Загальні відомості про системи обліку жителів</i>	<i>7</i>
<i>1.2. Застосування баз даних обліку жителів кампусу</i>	<i>9</i>
<i>1.3. Технічне завдання</i>	<i>11</i>
<i>1.4. Висновки до розділу 1</i>	<i>18</i>
РОЗДІЛ 2. ПРЕДМЕТНА ОБЛАСТЬ РОЗРОБКИ ПРОГРАМНОЇ ПЛАТФОРМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ	19
<i>2.1. Вибір інформації для програмної платформи кампусу гуртожитків ЗВО</i>	<i>19</i>
<i>2.2. Постановка задачі</i>	<i>22</i>
<i>2.3. Аналіз предмету розробки програмної платформи кампусу гуртожитків ЗВО</i>	<i>23</i>
<i>2.4. Опис алгоритму вибору кімнати для нового мешканця кампусу</i>	<i>23</i>
<i>2.5. Створення інформаційної бази в СУБД SQLite</i>	<i>25</i>
<i>2.6. Висновки до розділу 2</i>	<i>32</i>
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ ПЛАТФОРМИ КАМПУСУ ГУРТОЖИТКІВ ЗВО	33
<i>3.1. Створення програмної платформи кампусу гуртожитків ЗВО</i>	<i>33</i>
<i>3.2. Опис функціональних характеристик програмної платформи кампусу гуртожитків ЗВО</i>	<i>44</i>

					<i>ДТЕУ 121 02мз-17.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка аналітичної системи обліку жителів кампусу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>01.11.23</i>		<i>Зміст</i>	<i>2</i>	<i>51</i>
<i>Керівник</i>		<i>Рзаєєв С.Л</i>		<i>01.11.23</i>		<i>Факультет інформаційних технологій, 2 курс, 2мз група</i>		
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>01.11.23</i>				
<i>Розроб.</i>		<i>Удовеця О.В.</i>		<i>01.11.23</i>				
					<i>Зміст</i>			

3.3. Висновки до розділу 3.....	45
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТКИ.....	52



						Аржуш
Зм.	Аржуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
						3

ВСТУП

В сучасному світі створення програмної платформи кампусу гуртожитків ЗВО (заклад вищої освіти) є важливим завданням для будь-якого університету, незалежно від його місця розташування. Розробка відповідного програмного забезпечення надасть зручність та ефективність у керуванні інформацією про студентів, які проживають на території університетського кампусу гуртожитків.

Кампус гуртожитків є важливою складовою територією університету. Відповідальність за цей простір лежить на багатьох людях, а також вимагає обробки великої кількості документації. З метою спрощення управління цією інформацією запропоновано розробити програмний продукт, який надасть зручність та швидкість обробки даних, а також здатність аналізувати отриману інформацію.

У сучасному світі, ведення та зберігання інформації на паперових носіях або у табличних редакторах вважається незручним та неефективним підходом. З кожним днем обсяг даних зростає, і керувати ними стає все складніше. Досвід показує, що такий підхід вимагає значних затрат часу та ресурсів, і не є продуктивним. Тому пропонується використовувати систему управління базами даних (СУБД) для поліпшення процесу обліку жителів кампусу.

Один з популярних типів СУБД - це засоби обробки табличної інформації. Вони дозволяють ефективно обробляти великі обсяги даних завдяки своїм інструментальним можливостям. Автоматизація процесу ведення інформації про студентів допоможе краще управляти різними аспектами, зменшить ризик помилок, зробить систему зручною для використання та зекономить час. За

					<i>ДТЕУ 121 02мз-17.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Програмна платформа кампусу гуртожитків ЗВО</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>24.02.23</i>		<i>В</i>	<i>4</i>	<i>51</i>
<i>Керівник</i>		<i>Рзаєєв С.Л.</i>		<i>24.02.23</i>		<i>Факультет інформаційних технологій, 2 курс, 2мз група</i>		
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>24.02.23</i>				
<i>Розроб.</i>		<i>Удовця О.В.</i>		<i>24.02.23</i>				
					<i>Вступ</i>			

допомогою такої системи обліку жителів кампусу процес стане більш гнучким, сучасним та розвиненим. об'єктами, зменшить ризик виникнення помилок, буде більш зручною для використання та дозволить зекономити час. Завдяки такій системі ведення обліку жителів кампусу стане більш гнучким, модернізованим та розвиненішим.

Зрозуміти, як краще організувати поселення студентів, як і де зручніше зберігати інформацію про них – є важливим завданням не тільки для коменданта гуртожитку, а і для дирекції студентського містечка. Також володіти інформацією про своїх студентів воліють деканати усіх факультетів. Облік жителів студмістечка – є одним із процесів, котрий потребує певної автоматизації та спрощення завдань для керівництва. На вирішення досить легких питань вручну витрачається забагато часу та зусиль. Саме тому, для того щоб покращити цей процес і розробляється програмна платформа кампусу гуртожитків ЗВО.

Розуміння того, як краще організувати поселення студентів і зберігати інформацію про них, є важливим завданням для комендантів гуртожитків, дирекції студентського містечка та деканатів університету. Облік жителів студентського містечка вимагає автоматизації та спрощення, оскільки ручне виконання навіть найпростіших завдань забирає багато часу та зусиль. Саме тому розробляється інформаційно-аналітична система обліку жителів кампусу.

Під час поселення у гуртожитки, студенти заповнюють анкети та підписують контракти, оплачують проживання та стоять у чергах. Ці завдання можна спростити шляхом заповнення анкет та отримання платіжних реквізитів онлайн, надсилання документів електронною поштою для внесення їх комендантом до системи, яка зберігатиме всю необхідну інформацію. Це дозволить економити час як комендантам, так і студентам, а також сприятиме

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

зменшенню використання паперу, сприяючи більш екологічній діяльності університету.

Метою дослідження випускної кваліфікаційної роботи є дослідження існуючих програмних платформ кампусу гуртожитків та розробка програмного продукту, забезпечуватиме зберігання інформації про мешканців студентського містечка. Цей програмний продукт, завдяки своїм аналітичним інструментам, дозволить вибирати оптимальні кімнати для поселення студентів і спростить роботу комендантів гуртожитків, зберігаючи при цьому час.

Об'єктом дослідження є облік студентів, котрі проживають на кампусі університету.

Предметом дослідження є програмна платформа кампусу гуртожитків ЗВО.

Завдання дослідження:

- визначити основні поняття та терміни, котрі будуть використовуватися у роботі;
- розробка СУБД для програмної платформи кампусу гуртожитків ЗВО;
- проектування запитів для бази даних;
- створення програмного продукту, що надасть можливість зручно управляти даними про студентів-жителів кампусу;
- розробка програмного продукту «Campus».

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

ДТЕУ 121 02мз-17.МР

РОЗДІЛ 1

ОСНОВНІ ФУНКЦІОНАЛЬНІ ХАРАКТЕРИСТИКИ ПРОГРАМНОЇ ПЛАТФОРМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

1.1. Загальні відомості про системи обліку жителів

Університети по всьому світу потребують сучасну систему обліку жителів студентських містечок. Під час аналізу управління інформацією про студентів, які проживають у гуртожитках Державного торговельно-економічного університету (ДТЕУ), було виявлено потребу у більш сучасному підході до ведення аналітичної інформації про мешканців.

Результати дослідження показали, що в гуртожитку ДТЕУ для обліку жителів досі використовується табличний редактор та багато паперових документів. По-перше, цей підхід не дозволяє одночасний доступ до даних, як для коменданта, так і для дирекції студентського містечка. Він також є ненадійним і не дуже зручним у використанні. По-друге, у ХХІ столітті більшість підприємств та організацій переходять на безпаперовий спосіб діяльності. Тому доцільно модернізувати систему обліку жителів кампусу з метою зменшення використання паперу та забезпечення ефективного рішення, що заощаджує час усіх зацікавлених сторін.

На сьогоднішній день деякі університети вже успішно впроваджують автоматизовані системи обліку жителів кампусу. Наприклад, в Сполучених Штатах Америки студенти можуть заповнити всі необхідні дані онлайн та надіслати їх на спеціальну електронну адресу університету. Цей метод подання

Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02мз-17.МР</i>			
Зав. кафедри		Криворучко О.В.		10.04.23	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		10.04.23		Р1	7	51
Гарант		Котенко Н.О.		10.04.23		Факультет інформаційних технологій, 2 курс, 2мз група		
Розроб.		Удовеця О.В.		10.04.23				
					Основні функціональні характеристики інформаційної системи обліку жителів кампусу та технічне завдання			

документів зберігає час на поїздки та особисті зустрічі, а також уникне чергування для подачі документів. Такий підхід також сприяє значній економії паперу, оскільки всі необхідні форми можна заповнити онлайн.

Також, в університеті Отербайн в штаті Огайо існує програмне забезпечення, яке містить базу даних про всіх студентів, що проживають на кампусі, і мають доступ до цієї інформації уповноважені особи. Автор використала цю базу даних як приклад для розробки програмної платформи кампусу гуртожитків для Державного торговельно-економічного університету (ДТЕУ).

Така система управління інформацією є дуже зручною та доступною. Наприклад, у Німеччині під час вступу до університету, коли студент подає заяву он-лайн, він відразу має змогу подати форму на проживання в університеті та надіслати всі потрібні дані в електронному вигляді, що також в рази знижує використання паперу. В деяких університетах Німеччини такі бази даних для обліку жителів використовуються тільки управлінцями гуртожитків, а потім вже потрібні дані надходять за запитом у вигляді звітів до уповноважених осіб.

В університетах Естонії гуртожитки взагалі існують окремо від університету, хоча й підпорядковуються йому. Наприклад, в Талінському технічному університеті (ТТУ) вони мають свою автономію. Але під час вступу студент також заповнює форму на те, де він збирається проживати. Інформацію з даної форми отримує уповноважена особа і в он-лайн режимі вирішуються всі запити.

Такий розподіл та ведення інформаційної діяльності в електронному вигляді дозволяє економити час як працівникам університетів, так і студентам.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02мз-17.МР</i>				8

Також вона зберігає нашу планету чистою, оскільки зменшується в рази використання паперу.

Проаналізувавши процес ведення обліку жителів кампусу в різних університетах країн, було прийнято рішення розробити програмний продукт, який би забезпечував зберігання всієї інформації про студентів, проживаючих у гуртожитках університету, та надавав доступ до цієї інформації уповноваженим особам, таким як комендант гуртожитку, дирекція студентського містечка та деканати факультетів. Однак головною метою системи є використання аналітичних методів та штучного інтелекту для автоматичного підбору оптимальної кімнати для нового мешканця. Це спростить процес ручного розподілу місць комендантами гуртожитків, який відбувається щорічно.

Перед розробкою програмного продукту вкрай важливо повністю зрозуміти всі його аспекти, підготувати необхідні матеріали та опрацювати інформаційну базу. Тому першим кроком є розробка технічного завдання для програмного продукту. Це дозволить чітко визначити вимоги до системи, її функціональні можливості та інші важливі аспекти перед подальшою розробкою.

1.2. Застосування баз даних обліку жителів кампусу

Інформаційно-аналітична система (ІАС) – це комп’ютерна система, котра дозволяє отримувати інформацію, зберігати її, редагувати та створювати нові дані, здійснювати їх обробку та аналіз.

Завданнями ІАС є ефективно зберігання, обробка та аналіз даних. Технологічна платформа ІАС дозволяє підприємству (організації) здійснювати інтеграцію та координацію його бізнес-процесів. [13]

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02мз-17.МР</i>				9

Забезпечення для організації єдиного інформаційного простору та гарантія, що ці дані будуть доступні для всіх уповноважених осіб – є головним завданням ІАС.

Використання баз даних є сучасним та ефективним рішенням для управління інформацією. Існує багато різних видів систем управління базами даних (СУБД), які використовуються залежно від їхнього призначення, зв'язку з повним програмним забезпеченням, конфігурації з операційними системами та завантаження пам'яті на комп'ютері користувача.

Система управління базами даних дозволяє повністю зберігати та організувати всю інформацію на сервері університету та надавати доступ до неї уповноваженим особам. Це забезпечує зручний та простий спосіб управління даними, не вимагаючи великих знань технічних деталей. Система управління базами даних включає програмне забезпечення, яке дозволяє створювати необхідні структури для роботи з базами даних та вводити потрібні елементи управління. Основна функція СУБД - керування даними у зовнішній пам'яті. Вона забезпечує збереження та обробку даних, перетворення даних відповідно до запитів для користувача.

Також в рамках даної теми важливим є вивчення проблеми поселення студентів у гуртожитки, яка виникає щорічно. Розробка програмного застосунку може зробити процес збереження та користування інформацією про студентів більш зручним і ефективним.

У гуртожитках ДТЕУ проживає близько 2,5 тисяч студентів у гуртожитках. Університет має 7 гуртожитків для студентів бакалаврського та магістерського ступеня та працівників університету. Для коменданта кожного гуртожитку важливо мати повну інформацію про кожного студента, який проживає у відповідному гуртожитку. Доступ до цієї інформації також мають

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	10

дирекція студентського містечка та деканати факультетів. З метою синхронізації даних та зменшення використання паперу, автор даної роботи вирішила розробити програмний застосунок, який допоможе зберігати всю необхідну інформацію в одному місці.

Кожен гуртожиток має студентів з різних факультетів, а в кімнатах можуть проживати студенти різних курсів і навіть магістри з різних факультетів та спеціальностей. Іноді це може призвести до конфліктів між співмешканцями. Тому важливо зберігати інформацію таким чином, щоб при поселенні нових студентів була можливість направити їх у кімнату, де вже проживають люди з їх факультету, курсу або, якщо є можливість, навіть з їх групи.

Отже, потрібно розробити програмний застосунок, який буде зберігати інформацію про кожен кімнату та проживаючих у ній студентів. Він повинен надавати зручні можливості для конфігурування та управління даними, а також автоматизувати деякі процеси обліку та спростити поселення у кімнати. Застосунок має забезпечувати ефективне співставлення студентів з подібними факультетами, курсами або групами, щоб забезпечити сприятливе середовище спільного проживання та запобігти можливим конфліктам.

1.3. Технічне завдання

1. Загальні відомості

Програмний продукт для зберігання інформації про мешканців студентського містечка та аналізу кімнат її жителів.

1.1. Найменування системи

1.1.1. Повне найменування системи

«Програмна платформа кампусу гуртожитків ЗВО «Campus»»

1.1.2. Скорочене найменування системи

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР			11

«Campus», програмний продукт, додаток, програма, система, інформаційно-аналітична система.

1.2. Головний бенефіціар та потенційні користувачі системи

Для початку «Програмна платформа кампусу гуртожитків ЗВО «Campus»» розробляється як програма для внутрішнього користування тільки працівниками Державного торговельно-економічного університету, у його подальшій роботі та покращенні, якщо він буде зручним для користування, його можна буде поширити, дозволивши організації налаштовувати його під свій кампус та використовувати у власних цілях. Адже програмний продукт є вирішенням багатьох проблем, пов'язаних з поселенням студентів у гуртожитки та веденням обліку про них.

2. Мета та призначення створення системи

2.1. Призначення системи

Головним призначенням системи є створення інформаційної бази даних про жителів студмістечку, а також аналіз кожного з жителів для вираховування кращої кімнати для нового мешканця. Це дозволить уникнути конфліктів та проблем, пов'язаних з проживанням у гуртожитку, зменшить використання паперу та взагалі всю волокнину з документами, адже всі документи можна надсилати у електронному вигляді.

2.2. Мета створення системи

На разі не існує аналогів такої програми на території України, користуватися звичайним табличним редактором є досить не зручно у 21 ХХІ столітті, тим паче, коли світ пропонує широкі рішення багатьох проблем. Саме тому основною метою створення інформаційно-аналітичної системи обліку жителів кампусу є звільнення коменданта гуртожитку від введення вручну інформації, втрати часу на аналіз документів та слідкування за усіма

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	12

мешканцями, завдяки автоматизації багатьох процесів та інструментів аналітичних функцій для кращого ведення обліку всіх мешканців.

3. Вимоги до системи

3.1. Вимоги до системи в цілому

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем

Програма «Сampus» підключена до загального серверу Державного торговельно-економічного університету, що дозволить мати постійний доступ до неї кожному з трьох головних користувачів, комендантові, дирекції студмістечку та деканату. Ця система є десктопною та має свою базу даних, яка зберігається на сервері.

3.1.1.1. Вимоги до способів і засобів інформаційного обміну між компонентами системи

Єдине, що потрібно для аналізу та вибору кращої кімнати для поселення нового мешканця – це анкета, котру студент заповнює та надсилає комендантові. Все решта знаходиться у базі даних системи, отже не виникає жодних проблем з паперовою волокниною та купою вкладень у табличному редакторі. Все разом, зручно, компактно та доступно.

3.1.1.2. Вимоги до режимів функціонування системи

База даних знаходиться на сервері університету, тому єдина вимога для повноцінного її функціонування – безперервне підключення до сервера. Але комендантові завжди можна створити локальну копію бази даних свого гуртожитку і при повторному підключенні до сервера, інформація оновиться.

3.1.1.3. Вимоги до діагностування системи

Ця система є сучасним додатком, тому важливим є оновлена операційна система. Але додаток працює для ОС Windows 7, 8 та 10.

									Аркуш	
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР					13

3.1.1.4. Вимоги до режимів управління системою

Доступ до системи «Сампус» мають три ланки: коменданти, дирекція студмістечку та деканати всіх факультетів. Під час відкриття програми користувач обирає, до якої з них він відноситься, та заходить у свій режим доступу.

3.1.2. Показники призначення

3.1.2.1. Параметри, що характеризують ступінь відповідності системи призначення

Додаток працює для ОС Windows 7, 8 та 10.

3.1.2.2. Вимоги до пристосовності системи до змін

Планується оновлення системи, оскільки потрібно весь час налаштовувати систему під сучасні операційні системи, а також програмне середовище.

3.1.2.3. Вимоги до збереження працездатності системи в різних ймовірних умовах

База даних інформаційно-аналітичної системи обліку жителів кампусу знаходиться на сервері університету, тому потрібне безперебійне підключення до нього. Але завжди можна створити локальну копію для власного користування, і при наступному підключенні до сервера система оновить свої дані.

3.1.3. Вимоги до надійності

3.1.3.1. Склад показників надійності до системи в цілому

База даних системи «Сампус» створює резервну копію раз на тиждень, але користувач також може робити це вручну частіше, або ж змінити у налаштуваннях частоту зберігання резервної копії. Завдяки інкапсуляції даних у програмуванні від користувача приховується можливість нашкодити безпеці

									Аркуш	
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР					14

зберігання інформації, саме тому він не зможе «випадково» видалити базу даних, або ж залишитися повністю без інформації.

3.1.3.2. Вимоги до надійності технічних засобів і програмного забезпечення

Найкраще програмний продукт працюватиме з постійним підключенням до сервера та заповненні всіх потрібних даних. Також основною операційною системою для найзручнішого користування додатком є Windows 10.

3.1.4 Вимоги до захисту інформації від несанкціонованого доступу

3.1.4.1. Вимоги до інформаційної безпеки

Оскільки система не має прямого виходу в мережу Інтернет, єдине від чого вона залежить – це сервер університету. Тому за його злому можливий доступ до системи. При наступному оновленні також планується створення авторизації користувачів.

3.1.4.2. Розмежування відповідальності ролей при доступі

Доступ до системи «Сампус» мають три ланки: коменданти, дирекція студмістечку та деканати всіх факультетів. Під час відкриття програми користувач обирає, до якої з них він відноситься, та заходить у свій режим доступу. Оскільки деканат може тільки переглядати відомості про студентів свого факультету, які проживають у гуртожитку, а дирекція студмістечка та комендант мають повне право змінювати, додавати та видаляти інформацію, а також виселяти та заселяти мешканців.

3.1.5. Вимоги до захисту від впливу зовнішніх факторів

Головною вимогою є не встановлювати програму на свої домашні комп'ютери та на будь-які носії поза університетом, адже таким чином поширюється доступ до серверу університету, не залишати відкриту програму на робочому місці, і головне виходити з програми після користування нею.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02мз-17.МР</i>				15

3.2. Вимоги до видів забезпечення

3.2.1. Вимоги до інформаційного забезпечення

3.2.1.1. Вимоги до складу, структури і способів організації даних в системі

База даних системи зберігається на сервері університету, доступ до якої мають три ланки користувачів, але право на редагування інформації мають лише дві з них – це комендант та дирекція студмістечка. Головною вимогою є перевірка даних перш ніж її змінювати, адже додаток постійно просить про це та змінювати частоту зберігання резервних копій, якщо це потрібно.

3.2.2.2. Вимоги до інформаційного обміну між компонентами системи

Сторонніми інформаційними даними для опрацювання системою інформації є:

- Документи мешканців.
- Квитанції про оплату.
- Анкета студентів.

Ця інформація зберігається особисто у коменданта, а у електронному вигляді її наявність заноситься до системи.

Дирекція студмістечка може повідомити коменданта про якісь зміни, також при редагуванні інформації кожен з користувачів отримує оновлення даних та повідомлення про нього.

3.2.2.3. Вимоги щодо застосування систем управління базами даних

Програмна платформа кампусу гуртожитків ЗВО «Сампус» полегшує роботу користувача з базою даних, адже її користування є абсолютно автоматичним, нічого не потрібно вводити вручну або прописувати, всі можливі запити вже були продумані автором програмного продукту, а отже робота з

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР			16

цією системою є набагато легшою ніж навіть з будь-яким табличним редактором

3.2.2.4. Вимоги до структури процесу збору, обробки, передачі даних в системі представлення даних

Вся необхідна паперова інформація про мешканців гуртожитку (така як, копії паспорту, фотографія, квитанції про сплату, тощо) зберігається особисто у коменданта, а у електронному вигляді її наявність заноситься до системи, будь-яка інформація, що зберігається у електронному вигляді, зберігається у системі.

Також завдяки тому, що система підключена до серверу, завжди існує система повідомлень, дирекція студмістечка може повідомити коменданта про зміни, а при редагуванні інформації кожен з користувачів отримує оновлення даних та повідомлення про нього.

3.2.2.4. Вимоги до захисту даних від руйнувань при аваріях і збоях в електроживленні системи

Програма створює резервні копії бази даних автоматично, тому можна не хвилюватися, що якісь дані будуть пошкоджені або втрачені. Також можна створювати локальну резервну копію, або налаштувати частоту збереження даних до резервної копії.

3.2.2.5. Вимоги до контролю, зберіганню, оновленню та відновленню даних

Завжди потрібно дозволяти програмі створювати резервні копії даних, а також вчасно її оновлювати для покращеної та полегшеної роботи з нею, та кращого захисту даних.

4. Вимоги до методичного забезпечення

Система є досить легкою у користуванні, легше ніж будь-який табличний редактор, має невелику кількість зручних та зрозумілих функцій і кожна дія її

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	17

користування має своє пояснення. Головне перевіряти введеною інформацією та розуміти як працюють програми з загальним доступом до серверу університету.

1.4. Висновки до розділу 1

Після аналізу отриманих даних стало очевидним, що наявна система ведення обліку жителів кампусу потребує оновлення. В деяких університетах вже існують успішні та зручні системи обліку студентів, які були розглянуті раніше. Вони мають багато переваг порівняно з традиційним табличним редактором і є значно ефективнішими у використанні та управлінні даними.

У першому розділі були проаналізовані програмні продукти, які дозволяють автоматизувати збір та управління інформацією про студентів, що проживають у гуртожитках. Зроблено висновок, що потрібна програмна платформа кампусу гуртожитків ЗВО, яка дозволить зберігати наявні дані, додавати та редагувати нові, надавати вільний доступ до даних уповноваженим особам, таким як коменданти, дирекція студентського містечка та деканати факультетів. Тому є потреба у створенні програмного продукту з метою аналізу даних про студентів та вдосконалення процесу підбору кімнати для нового жителя.

					<i>ДТЕУ 121 02мз-17.МР</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		18

РОЗДІЛ 2

ПРЕДМЕТНА ОБЛАСТЬ РОЗРОБКИ ПРОГРАМНОЇ ПЛАТФОРМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

2.1. *Відбір інформації для програмної платформи кампусу гуртожитків ЗВО*

Основною метою розробки програмної платформи кампусу гуртожитків ЗВО або додатку «Campus» є створення централізованої бази даних про жителів гуртожитків з можливістю додавання аналітичних функцій для ефективного підбору найкращої кімнати для нових мешканців.

Аналізуючи проблему поселення студентів у гуртожитки, яка виникає щороку, додаток «Campus» створює зручне зберігання та управління інформацією про студентів. Він пропонує аналітичні інструменти, які допомагають програмі визначати найкращу кімнату для нових мешканців на основі різних критеріїв та прогнозування.

Таким чином, додаток «Campus» забезпечує зручну інформаційну базу даних про жителів гуртожитків і надає аналітичні функції для підбору оптимального розташування студентів у кімнатах.

Для попередження конфліктів між мешканцями, автор пропонує використовувати маленьку анкету, яку студент повинен заповнити перед поселенням та передати коменданту гуртожитку.

За досвідом, набутим в процесі поселення студентів закордоном, автор бажає поліпшити та спростити процес поселення для обох сторін. Саме тому була створена анкета, яка враховує курс, факультет та спеціальність

					<i>ДТЕУ 121 02мз-17.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>24.05.23</i>	<i>Розробка аналітичної системи обліку жителів кампусу</i>	<i>P2</i>	<i>19</i>	<i>51</i>
<i>Керівник</i>		<i>Рзаєва С.Л.</i>		<i>24.05.23</i>		<i>Факультет інформаційних технологій, 2 курс, 2мз група</i>		
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>24.05.23</i>				
<i>Розроб.</i>		<i>Удовця О.В.</i>		<i>24.05.23</i>				
					<i>Предметна область розробки системи обліку жителів кампусу</i>			

майбутнього мешканця. Крім того, вона також враховує базові звички, такі як чистота та тиша, які можуть значно впливати на конфліктні ситуації між мешканцями кімнати. Наприклад, якщо хтось має чутливий сон, то він, напевно, не захоче засипати під шум або гучну музику. За допомогою оцінки рівня чистоти, можна також враховувати, що мешкання охайної особи з недбалою не буде хорошою ідеєю.

Отже, включення такої анкети перед поселенням допомагає зменшити можливість конфліктів та підвищити задоволення мешканців від проживання у гуртожитку.

Для того, щоб не опрацьовувати ці анкети власноруч, адже тисячі студентів проживають у всьому студмістечку, програма «Campus» робить це для користувача за допомогою інструментів одного з елементарних типів штучного інтелекту. Програма проаналізує відповіді на певні питання, порівняє їх з іншими і поселить студента у кімнату, яка найкраще пасуватиме для нового мешканця.

Приклад анкети про студента:

ПІБ:

Стать: жіноча чоловіча

Факультет:

Курс:

Група:

Переваги сусідів по кімнаті (підкреслити): спортсмен, студент першого року, студент другого курсу, третього курсу, студент останнього курсу, магістр, міжнародний студент.

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР			20

Особистість (підкреслити): тиха та спокійна, посередині, галасна та товариська.

Відмітити чистоту від одного до п'яти:

Шум у кімнаті від одного до п'яти:

Лягаю спати о: __ (від десятої до третьої)

Підкреслити, що є для вас характерним:

- *Чутливий сон*
- *Можу спати з увімкненим нічником*
- *Потребую тиші*
- *Музика може бути увімкнена*
- *Міцно сплю*

Така коротка анкета дійсно допоможе зрозуміти звички нового мешканця та підібрати сусідів, які схожі за характером, щоб уникнути проблем та конфліктів у кімнаті. Хоча кожен студент, який проживає в гуртожитку, повинен дотримуватися чистоти та розпорядку дня, іноді забувають про це або можуть мати інші незначні недоліки, які можуть стати проблемою для інших мешканців. За допомогою анкети, комендант гуртожитку може попередньо підібрати сумісних мешканців, що полегшить йому роботу і допоможе уникнути конфліктів у кімнаті. Це особливо важливо, оскільки конфлікти можуть призвести до потреби перерозселення мешканців навіть під час навчального року, що є неприємним і трудомістким процесом.

Отже, враховуючи такі дрібниці та забезпечуючи підбір сумісних мешканців, адміністрація гуртожитків може запобігти конфліктам, полегшити роботу коменданту та забезпечити комфортне проживання для всіх студентів у гуртожитку.

					ДТЕУ 121 02мз-17.МР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

Отже, з метою зменшення використання паперу та зниження конфліктних ситуацій між мешканцями гуртожитків, а також для економії часу студентів та працівників університету, необхідна система обліку жителів. Програмний продукт «Campus» є ідеальним рішенням для збереження інформації про всіх мешканців гуртожитків, а також надає зручні інструменти для вибору найкращої кімнати для нових студентів у кампусі. Таким чином, «Campus» є важливим кроком у поліпшенні управління гуртожитками та створенні комфортного середовища для життя студентів.

2.2. Постановка задачі

Проаналізувавши зібрані дані, можна зробити такі висновки щодо постановки завдання на розробку програмної платформи кампусу гуртожитків ЗВО:

- налаштувати БД, яка буде не місткою за обсягом пам'яті, крос-платформною та дозволить зручно конфігурувати дані та легко суміщається з програмним забезпеченням;
- створити програмну платформу кампусу гуртожитків ЗВО, на основі проаналізованих даних, котра б відповідала всім запитам уповноважених осіб, які в подальшому будуть нею користуватися;
- використати для створення системи високорозвинену мову програмування, що дозволить створити початкові елементи штучного елементу для аналізу анкет мешканців та підбору найкращої кімнати;
- створити програмний продукт, що дозволить автоматизувати певні процеси ведення обліку, а також спростить процес поселення та мінімізує кількість дій коменданта до мінімуму;

						Аркуш
					<i>ДТЕУ 121 02мз-17.МР</i>	
Зм.	Аркуш	№ докум	Підпис	Дата		22

- розробити систему, завдяки користуванню якої зменшиться кількість часу на опрацювання поселення однієї людини, а отже і всіх студентів загалом, а також котра дозволить вести більш екологічну діяльність університету.

Таким чином, було поставлено завдання розробити сучасну програмну платформу кампусу гуртожитків ЗВО, яка буде доступна для усіх уповноважених осіб, таких як коменданти гуртожитків, дирекція студентського містечка та деканати факультетів. Головна мета системи полягатиме у забезпеченні зручного користування, автоматизації процесів ведення обліку мешканців та їх поселення, ефективному використанні часу, спрощенні ручного введення інформації та зменшенні використання паперової документації.

2.3. Аналіз предмету розробки програмної платформи кампусу гуртожитків ЗВО

Важливо підійти до вибору інструментів програмування та розробки з урахуванням ефективності програмного продукту для ведення обліку мешканців студентського містечка. Одним з ключових елементів буде база даних, яка повинна бути легкою у використанні та розміщена на сервері, забезпечуючи оптимальне використання ресурсів. Важливо також мати можливість регулярного резервного копіювання даних. Для розробки модернізованої системи необхідна мова програмування високого рівня, що дозволить створити аналітичний додаток з елементами штучного інтелекту.

Інструментами розробки автором цієї системи є створення компактної та не затратної у пам'яті бази даних СУБД SQLite, а також було обрано мову програмування високого рівня C#.

2.4. Опис алгоритму вибору кімнати для нового мешканця кампусу

						Аркуш
					ДТЕУ 121 02мз-17.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		23

Важливо правильно встановити пріоритети у анкеті нового мешканця кампусу, що має велике значення для аналітичної системи обліку кампусу. Кожен пункт анкети перевіряється обчислювальними елементами в кодї системи згідно з їх значущістю.

Наприклад, відомо, що хлопці та дівчата не можуть проживати разом у одній кімнаті. Першою перевіркою є стать нового мешканця, що автоматично зменшує кількість доступних кімнат наполовину.

Потім перевіряється приналежність до факультету, курсу та групи. Система автоматично шукає студентів, що мають близькі характеристики, і перевіряє ці три фактори одночасно.

Після вибору студентів з одного факультету та курсу, можливо, групу не завжди вдається знайти або кімната може бути заповнена. Це ще більше зменшує наш вибір. Система аналізує ці дані автоматично завдяки вбудованим елементам коду, і користувачу не потрібно вручну перевіряти нічого, окрім введених даних з анкети.

Наступним пріоритетом є чутливість до сну, оскільки він важливий у житті кожної людини. Після перевірки цього параметра ми перевіряємо, чи стануть сусіди нового мешканця людьми, що дбають про чистоту, або чи не засмучуватиме його забута чашка на столі.

Після проведення перевірок зазначених ознак кожного потенційного сусіда для нового студента, будуть отримані результати, які включатимуть кілька доступних кімнат. Система також перевіряє останній критерій - вподобання студента щодо спортсменів або людей з тихим характером серед його сусідів.

В результаті цих перевірок комендант отримує повідомлення, що для студента була обрана певна кімната. Цей процес обчислення та аналізу жителів

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	24

кампусу займає всього кілька секунд користувача і повністю автоматизує вибір кімнати для нового мешканця.

2.5. Створення інформаційної бази в СУБД SQLite

База даних для програмної платформи кампусу гуртожитків ЗВО розроблена у СУБД SQLite та складається з 14 таблиць, котрі були обрані базуючись на отриманих даних та на питаннях анкети, котра була розроблена для поселення мешканців у гуртожиток:

1. *Довідник міст* (код міста, назва міста);
2. *Довідник студентів* (id студента, прізвище, ім'я та по-батькові студента, id факультету, номер групи, номер курсу, дата народження, контактний телефон, код міста, адреса, id особистості, id звички, id типу навчання, дата заселення);
3. *Довідник факультетів* (id факультету, назва факультету);
4. *Тип навчання* (id типу навчання, тип навчання);
5. *Таблиця поселення* (id студента, id платежу, номер гуртожитку, стать, номер кімнати, дата сплати, дата виселення);
6. *Довідник типів платежів* (id платежу, тип платежу, сума сплати);
7. *Довідник кімнат* (номер кімнати, кількість місць, наявність меблів);
8. *Таблиця перевірки кімнат* (номер кімнати, бал за перевірку, дата перевірки);
9. *Довідник гуртожитків* (id гуртожитку, номер гуртожитку);
10. *Довідник звичок* (id звички, назва звички);
11. *Довідник особистостей* (id особистості, назва особистості);
12. *Довідник статей* (id статі, стать);
13. *Довідник факультетів* (id факультету, назва факультету);

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	25

14. Довідник уподобань сусідів (id уподобання, назва уподобання).

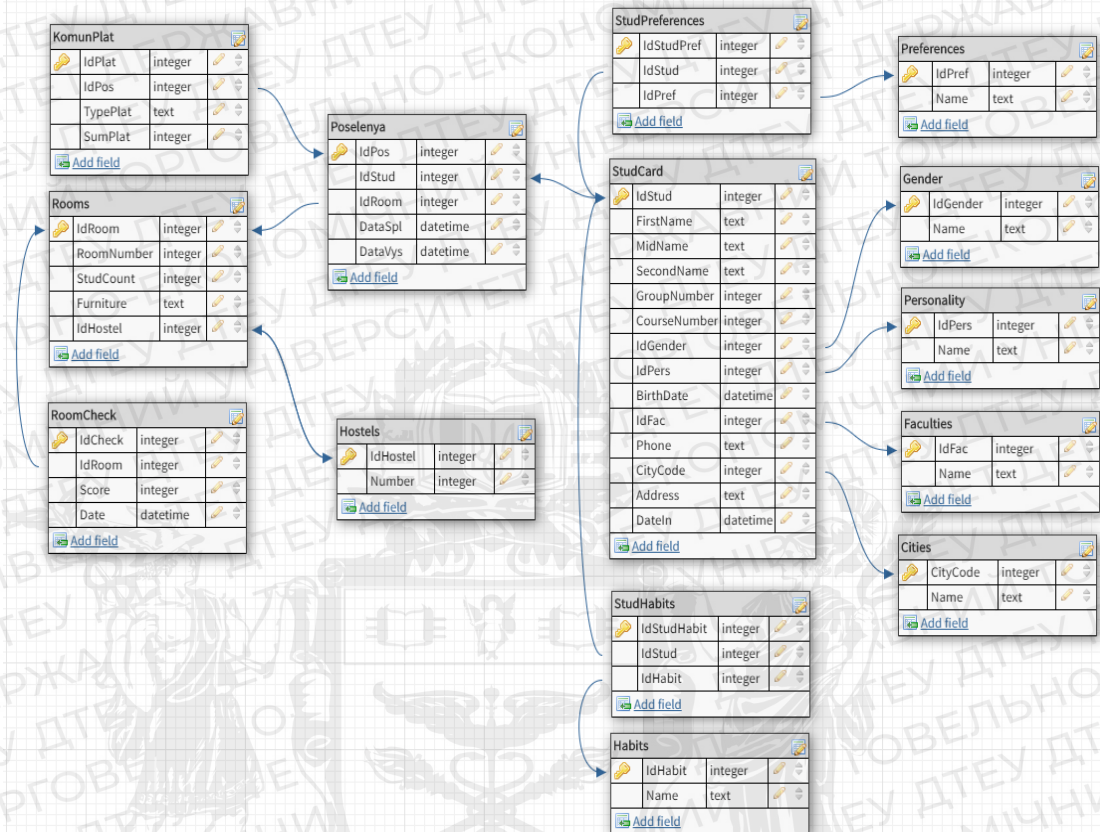


Рис. 2.1. Фізична модель бази даних програмної платформи кампусу гуртожитків ЗВО

Джерело: побудовано автором

Після планування схеми даних необхідно створити таблиці. Таблиця є основою будь-якої бази даних і представляє собою структурований набір пов'язаних даних, які зберігаються у вигляді стовпців та рядків. У реляційних базах даних таблиці використовують модель з вертикальними стовпцями (із унікальними іменами) та горизонтальними рядками.

Важливо зазначити, що дані в таблицях можуть не зберігатись фізично в базі даних. Існують також представлення, які функціонують як реляційні таблиці, але дані в них обчислюються під час виконання запиту.

Перед створенням таблиці необхідно розібратися, для яких цілей використовуються різні типи полів та даних.

Для введення числових даних рекомендується використовувати числові поля. Вони мають розмір та можуть бути різних типів, таких як цілі числа або числа з десятковою частиною. Для останніх також важливо вказати розмір десяткової частини поля.

Для введення даних дати або часу існує тип поля «Дата/Час».

Текстові поля призначені для запису текстових даних. Основна властивість текстового поля - його розмір. Проте варто зазначити, що текстові поля мають обмеження на кількість символів, яке зазвичай не перевищує 256. Якщо потрібно ввести довгий текст, більший за це обмеження, можна використовувати поле типу «МЕМО», яке дозволяє зберігати до 65535 символів. Зазначу, що дані цього типу фактично зберігаються в іншому місці, а поле містить лише посилання на розташування тексту.

Поле «Лічильник» - це автоматизоване поле для автоматичного нарощування значень. Якщо таке поле присутнє в базі даних, при додаванні нового запису значення цього поля автоматично збільшується на одиницю порівняно з попереднім записом. Це зручно для нумерації записів.

Таким чином, структура запису, яка включає назви полів, їх типи та властивості, визначені користувачем, визначає структуру таблиці. Це впливає на послідовність розташування даних у записі на фізичному носії та їх вигляд на екрані.

						Аркуш
					ДТЕУ 121 02мз-17.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		27

Відповідно до поставленого завдання та проведеного аналізу, було створено наступні таблиці:

1. Таблиця «Довідник міст» містить інформацію про код та назву міста. Ця таблиця має зв'язок лише з таблицею «Довідник студентів» за допомогою первинного ключа «Код міста».
2. Таблиця «Довідник студентів» включає поля для ід студента, прізвища, імені та по-батькові, ід факультету, номеру групи, дати народження, контактного телефону, коду міста, адреси, телефону батьків, ід типу навчання, дати заселення, ід звички, ід особистості, номеру кімнати, в яку був поселений студент, ід статі студента, а також номеру курсу, на якому навчається новий мешканець. Ця таблиця має зв'язки з іншими таблицями: з «Довідником міст» за допомогою ключа «Код міста»; з «Довідником факультетів» за ключем «ід факультету»; з таблицею «Поселення» за ключем «ід студента»; з таблицею «Особистість» за полем «ід особистості»; з таблицею «Звички» за ключем «ід звички»; з таблицею «Стать» за полем «ід статі»; а також з таблицею «Уподобання студентів» за ключем «ід студентського уподобання».
3. Таблиця «Довідник факультетів» містить тільки ід факультету та його назву та пов'язана з таблицею «Довідник студентів» по первинному ключу ід факультету.
4. Таблиця «Статі» містить тільки ід статі та її назву та пов'язана з таблицею «Довідник студентів» по первинному ключу ід статі.
5. Таблиця «Звички студентів» містить в собі такі поля, а саме: ід студентської звички, ід звички та ід студента. Вона зв'язана з таблицею «Довідник студентів» за полем ід студентської звички, а з таблицею «Звички» за ключем ід звички. Окремо існує таблиця «Звички», котра містить тільки ід

						Аркуш
					ДТЕУ 121 02мз-17.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		28

звички та саму звичку та пов'язана з таблицею «Звички студентів» по первинному ключу id звички.

6. Таблиця «Особистості» містить тільки id особистості та її характеристику та пов'язана з таблицею «Довідник студентів» по первинному ключу id особистості.

7. Існує таблиця «Уподобання студентів», котра містить такі поля: id уподобання студента, id уподобань, id студента. Вона пов'язана з «Довідником студентів» за полем id студента, а з таблицею «Уподобання» за ключем id уподобань.

8. Таблиця «Уподобання» містить лише id уподобання та його характеристику.

9. Таблиця «Довідник типів платежів» містить такі поля: id платежу, код поселення, тип платежу, сума сплати. Вона сполучається з таблицею «Поселення» по полю код поселення.

10. Таблиця «Поселення» містить в собі код поселення, id студента, id платежу, номер кімнати, дата сплати, дата виселення. Вона сполучається з таблицями «Довідник студентів», «Довідник типів платежів» та «Довідник кімнат» по зовнішнім ключам id студента, код поселення, id платежу та «Номер кімнати» відповідно.

11. Таблиця «Довідник кімнат» має в собі первинний ключ «Номер кімнати», а також кількість місць у ній та поле наявності меблів. Вона пов'язана з двома таблицями: «Поселення» та «Перевірка кімнат» по полю «Номер кімнати».

12. Таблиця «Гуртожитки» має в собі первинний ключ id Гуртожитку та номер гуртожитку і зв'язаний з таблицею «Довідник кімнат» по ключу id Гуртожитку.

					<i>ДТЕУ 121 02мз-17.МР</i>	<i>Аркуш</i>
						29
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

13. Таблиця перевірки кімнат містить номер кімнати, бал за перевірку та дату її проведення. Пов'язана лише з однією таблицею – це «Довідник кімнат», з якої ми беремо номер кімнати.

Всі таблиці автор прописувала одночасно і разом, щоб відразу їх сполучити за первинними ключами – це значно полегшує процес розробки бази даних, аніж написання кожної таблиці окремо.

```
72 CREATE TABLE Poselenya (  
73     IdPos integer PRIMARY KEY AUTOINCREMENT,  
74     IdStud integer,  
75     IdRoom integer,  
76     DataPos datetime,  
77     DataVys datetime  
78 );  
79  
80 CREATE TABLE KomunPlat (  
81     IdPlat integer PRIMARY KEY AUTOINCREMENT,  
82     IdPos integer,  
83     TypePlat text,  
84     SumPlat integer  
85 );  
86  
87 CREATE TABLE Rooms (  
88     IdRoom integer PRIMARY KEY AUTOINCREMENT,  
89     RoomNumber integer,  
90     StudCount integer,  
91     Furniture text,  
92     IdHostel integer  
93 );
```

Рис. 2.2. Фрагмент програмного коду зі створення таблиць у СУБД SQLite під час розробки інформаційно-аналітичної системи обліку жителів кампусу

Джерело: побудовано автором

Підключення бази даних до системи відбувається покроково:

- 1) спочатку база даних прив'язується до Microsoft Visual Studio;
- 2) прописується код, що відповідає за обмін інформацією між таблицями користувачів;
- 3) проходить вибір інформації, яка буде потрібною для показу користувачеві на екрані (приклад – рис. 2.3.).

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР				30


```

public class Room
{
    public long Id { get; }
    public long Number { get; }
    public long Count { get; }

    public List<Resident> Residents { get; }

    public Room(long Id, long Number, long Count)
    {
        this.Id = Id;
        this.Number = Number;
        this.Count = Count;
        this.Residents = new List<Resident>();
    }
}

```

Рис. 2.3. Приклад вибору даних для показу таблиці на екрані користувача

Джерело: побудовано автором

На екран виводяться тільки основні дані про жителя кампусу для користувача. Адже більшість питань з анкети не несуть своєї важливості для коменданта у діяльності обліку кімнат, окрім процесу поселення. Тому дані зберігаються в базі даних, але в таблицю користувача вони не виводяться з метою розробки системи без зайвого навантаження для користувача.

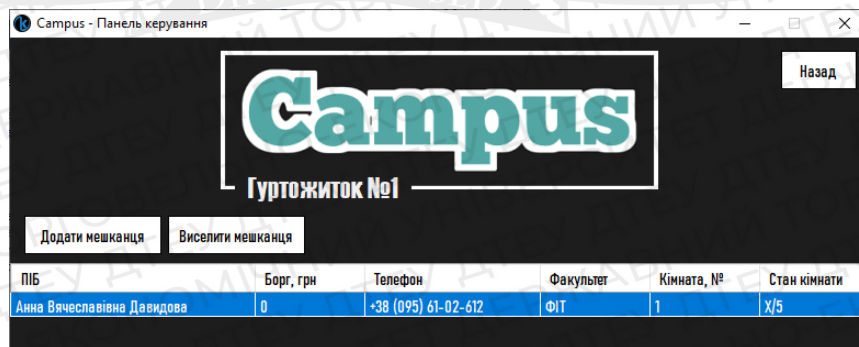


Рис. 2.4. Приклад відображення таблиці про жителів кампусу для коменданта у аналітичній системі обліку жителів кампусу

Джерело: побудовано автором

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

ДТЕУ 121 02мз-17.МР

Отже, у цій частині було визначено основні поняття з розробки БД та розглянуто реляційну модель даних, а саме СУБД SQLite, в якій описано таблиці та їх наповнення для програмного продукту «Campus». Також у цьому підрозділі можна прочитати детальний опис таблиць та їх вміст, побачити схему даних потрібних саме для системи обліку жителів кампусу, яка буде використана для подальшої розробки програмної платформи обліку студентів, котрі проживають на території студентського містечка.

2.6. Висновки до розділу 2

Отже, для розробки аналітичної системи обліку жителів кампусу та її функції автоматичного поселення було визначено предметну область та інструменти, такі як середовище Visual Studio .NET з мовою програмування C# високого рівня та СУБД SQLite. Вивчаючи анкети, використані в інших університетах зі схожими системами обліку, був обраний найкращий та найзручніший приклад анкети для жителів студентського містечка. Поставивши перед собою завдання, було детально описано основні функції, такі як зберігання даних та надання доступу до них трьом різними видами користувачів з відповідними правами, а також повне автоматизоване поселення нових жителів завдяки використанню елементарних компонентів штучного інтелекту для аналізу наявних даних. Важливо ретельно продумати зв'язки між інформацією в базі даних. Створення, редагування та зміна типів даних в базі даних досить прості до підключення до програмного продукту. У цьому розділі була наведена схема даних та описані таблиці, які будуть використовуватися для ведення обліку жителів студентського містечка.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 02мз-17.МР</i>				32

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОЇ ПЛАТФОРМИ КАМПУСУ ГУРТОЖИТКІВ ЗВО

3.1. Створення програмної платформи кампусу гуртожитків ЗВО

Одним з головних завдань проекту є розробка аналітичної системи управління даними. Створення форм є необхідною частиною цього завдання, оскільки форми спрощують та прискорюють роботу з базою даних, дозволяють створити зручний та дружелюбний користувацький інтерфейс і є основним засобом взаємодії користувача з даними, а також мають багато переваг та функцій.

Для зручного управління та користування програмної платформи кампусу гуртожитків необхідні різні форми, такі як вітальна форма, форма коменданта для повного керування даними, форма для дирекції студмістечка, окрема форма для деканатів факультетів, а також основна форма для вибору користувача системи. З використанням програмної платформи Visual Studio .NET 2017 можна створити елемент «Форма» одним натисканням миші, і всі поля з базового джерела даних автоматично розміщуються на формі. При створенні форми її можна використовувати безпосередньо або змінювати відповідно до потреб додатку у режимі конструктора.

					<i>ДТЕУ 121 02мз-17.МР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.		06.09.23	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		06.09.23		РЗ	33	51
Гарант		Котенко Н.О.		06.09.23		Факультет інформаційних технологій, 2 курс, 2мз група		
Розроб.		Удовця О.В.		06.09.23				

```

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

namespace campus
{
    public partial class Main : Form
    {
        public Main()
        {
            InitializeComponent();
        }
    }
}

```

Рис. 3.1. Приклад автоматичного підключення бібліотек під час створення форми у Visual Studio .Net

Джерело: побудовано автором

Форма є зручною для користування, у ній немає нічого складного для розуміння, немає жодних зайвих кнопок чи вікон, котрі могли б відвернути увагу від потрібних завдань при роботі з даними та базою даних.

Розробка інтерфейсу є важливою складовою, і автор намагалася створити його дуже простим та доступним у розумінні для кожного з користувачів. Для створення програмної платформи кампусу гуртожитків було визначено чотири різних Windows форми, які з'єднані між собою.

Для того, щоб створити нову форму потрібно:

1. У Visual Studio .Net натиснути «Файл» - «Створити» - «Проект» або комбінацію клавіш Ctrl+Shift+N;
2. Обрати Додаток «Windows Forms (.Net Framework)». Важлива саме ця бібліотека, адже вона має повний ресурс для створення користувацького інтерфейсу десктопної версії програмного продукту;
3. Дати назву проекту та обрати місце для зберігання та натиснути «ОК».

Після цих кроків перед розробником відкривається дизайнер форм. Спочатку форма буде пустою (рис.3.2), її треба наповнити наперед

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

ДТЕУ 121 02мз-17.МР

підготовленими іконками користувачів та кнопками. Іконки можна додати обравши на панелі «Tools» категорію «Label». Кнопки на форму додаються за допомогою інструменту «Button».

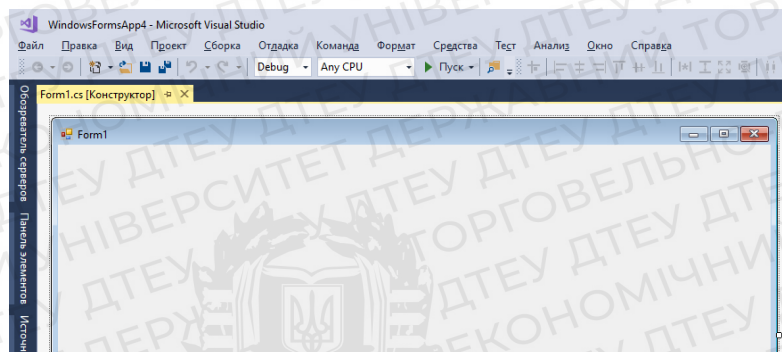


Рис. 3.2. Видгляд стандартної Windows Form до редагування та наповнення

Джерело: побудовано автором

Буває форма може без кнопок. Наприклад, для програмної платформи кампусу гуртожитків ЗВО було створено вітальну форму, котра має лічильник та тільки іконку привітання. (рис.3.3.)

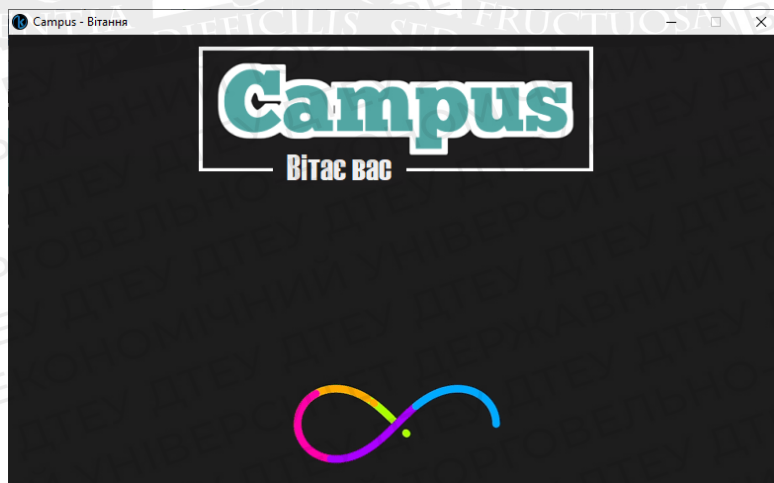


Рис. 3.3. Зображення вітальної форми

Джерело: побудовано автором

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		35

ДТЕУ 121 02мз-17.МР

Після такого як було виконано зовнішнє оформлення форми, подвійним кліком в будь-якій області форми ми переходимо до середовища, де прописуємо код цієї форми. Для завантаження цієї форми був використаний такий код.(рис. 3.4) Найважливіша її частина – це таймер, котрий відраховує період показу форми на екрані.

```
private void Welcome_Load(object sender, EventArgs e)
{
    const int kLoadTime = 4300; // Час в мілісекундах на протязі якого відображається вітальне вікно.
    CenterToScreen();
    new System.Threading.Timer(LoadingComplete, null, kLoadTime, 0);
}

private void LoadingComplete(object state)
{
    if (InvokeRequired)
        Invoke(new LoadingCompleteCallback(LoadingComplete), state);
    else
    {
        Hide();
        new Main().ShowDialog(this);
        Close();
    }
}
```

Рис. 3.4. Код вітальної форми

Джерело: побудовано автором

Коли зникає вітальна форма, на екрані з'являється форма вибору користувача. Є основні три користувачі цим програмним продуктом: комендант, дирекція студентського містечка, відділ деканатів факультетів. Кожен з них має різні обмеження на право редагування інформації.

Комендант має право додавати нових мешканців, виселяти жителів, а також редагувати інформацію. Дирекція студентського містечка має право на додавання квитанцій на оплату для одного жителя або для усієї кімнати, а також проводить оцінку кімнат, дані про перевірку вони також вносять до цієї ж системи. Деканат факультетів має право лише на перегляд інформації, оскільки

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

ДТЕУ 121 02мз-17.МР

головна їх мета – це ведення обліку їх студентів і визначення, хто з студентів їх факультету проживає у гуртожитку. Наведена далі таблиця повністю характеризує всі дії та права на виконання них різними користувачами в залежності від посади уповноважених осіб.

Таблиця 3.1.

НАДАННЯ ПРАВ КОРИСТУВАЧАМ

Користувач	Право на додавання нового мешканця	Право на виселення мешканця	Редагування даних про жителя	Оцінка кімнат	Додавання квитанції на сплату	Перегляд даних
1	2	3	4	5	6	7
Комендант	Так	Так	Так	Ні	Ні	Так
Дирекція студмістечку	Ні	Ні	Ні	Так	Так	Так
Деканат	Ні	Ні	Ні	Ні	Ні	Так

Джерело: побудовано автором

З наведеної таблиці зрозуміло, що потрібно створити застосунок, де в кожного користувача будуть певні права та обмеження. Саме тому після вітального вікна програми, перед користувачем з'являється основна форма вибору уповноваженої особи, хто в даний момент користується цим програмним продуктом (рис.3.5).

					<i>ДТЕУ 121 02мз-17.МР</i>	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

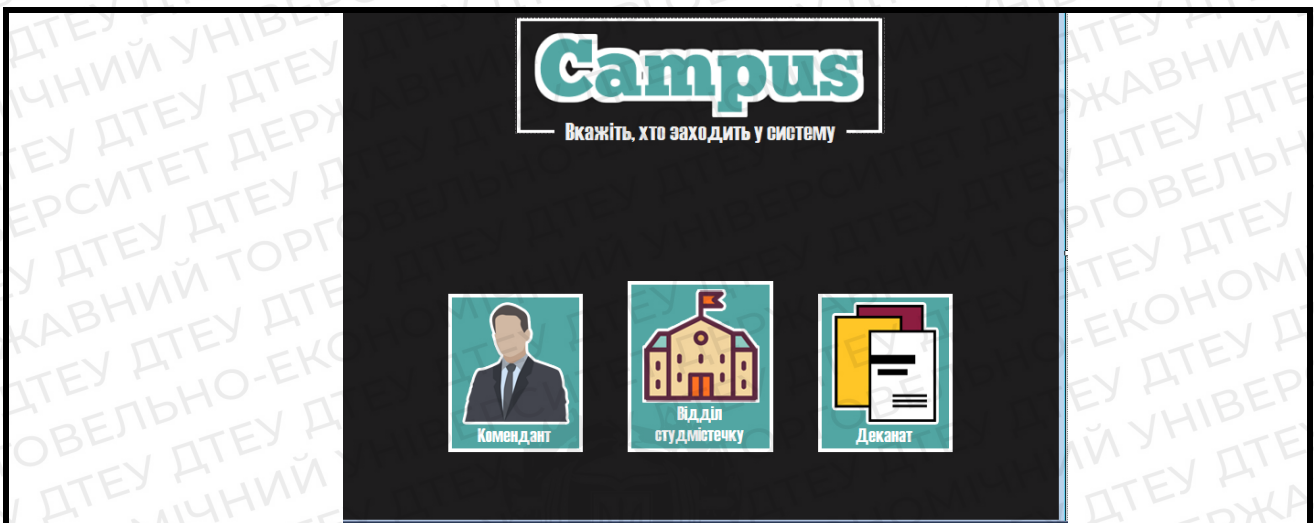


Рис. 3.5. Зображення основної форми вибору користувача

Джерело: побудовано автором

Дана форма була наповнена однією іконкою інструмента «Label» для лого програми «Campus». А також трьома кнопками, функції яких наведені нижче (рис.3.6.).

```
private void komendant_button_Click(object sender, EventArgs e)
{
    new HostelSelector(ControlPanel.Role.Komendant).ShowDialog(this); //кнопка коменданта для відкриття вибору гуртожитка
}

private void campus_button_Click(object sender, EventArgs e)
{
    new HostelSelector(ControlPanel.Role.StudCity).ShowDialog(this); //кнопка студмістечка для вибору гуртожитку
}

private void decans_office_button_Click(object sender, EventArgs e)
{
    new FacultySelector().ShowDialog(this); //кнопка деканату для вибору факультету
}
```

Рис. 3.6. Код функціоналу вибору користувачів

Джерело: побудовано автором

Даний код виконує таку функцію: залежно від вибору користувача перед особою з'являється наступна форма з випадаючим списком на запит факультету

						Аркуш
					ДТЕУ 121 02мз-17.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		38

(для відділу деканату) (рис.3.7) або ж з вибором гуртожитку (для відділу студентського містечка та коменданта) (рис.3.8).

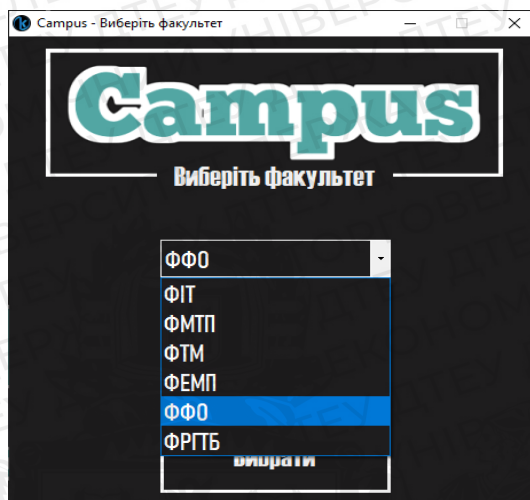


Рис. 3.7. Зображення форми вибору факультету для користувача з відділу деканату факультету

Джерело: побудовано автором

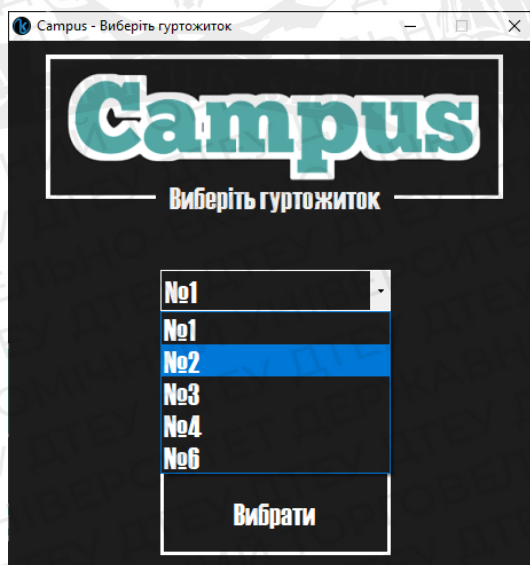


Рис. 3.8. Форма вибору гуртожитку

Джерело: побудовано автором

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	39

Такий розподіл значно полегшує управління даними та є дружелюбним до користувача. Розподіл прав та форм значно полегшує конфігурацію даних та запобігає введенню великої кількості помилок, «випадковому» редагуванню даних або ж взагалі видаленню їх, що робить значно комфортнішим ведення обліку даних про жителів студентського містечка, а також робить дуже зручним їх поселення адже наведена анкета у розділі 2.1 існує в електронному вигляді у профілі коменданта (рис. 3.9).

Аналіз усіх критеріїв та полів даної анкети дозволить обрати кімнату для нового студента, та спростить ручний аналіз даних, так як це відбувається кожного року під час поселення студентів комендантами гуртожитків та дирекцією студмістечка.

The screenshot shows a web browser window with the title "Campus - Додати мешканця". The main heading is "Campus" in a large, stylized font, followed by "Інформація про мешканця". The form contains several sections:

- Personal Information:** Fields for "Прізвище" (Surname), "Ім'я" (Name), and "По-батькові" (Patronymic).
- Gender:** Radio buttons for "Чоловіча" (Male), "Жіноча" (Female), and "Не визначено" (Not specified).
- Academic Info:** Fields for "Факультет" (Faculty), "Курс" (Course), and "Група" (Group), along with a "ФІТ" (FIT) dropdown.
- Preferences:** Two columns of checkboxes. The left column is "Переваги в сусідах" (Advantages in neighbors) with options: "Спорсмен" (Sportsman), "Студент першого курсу" (First-year student), "Студент другого курсу" (Second-year student), and "Студент третього курсу" (Third-year student). The right column is "Особливості" (Features) with options: "Чутливий сон" (Sensitive to noise), "Можу спати з увімкненим нічним" (Can sleep with night light on), "Потребую тиші" (Need quiet), and "Музика може бути увімкнена" (Music can be turned on).
- Character:** Radio buttons for "Тихий та спокійний" (Calm and quiet), "Нейтральний" (Neutral), and "Галасний та товариський" (Loud and friendly).
- Location:** Fields for "Місто" (City) with a dropdown set to "Київ" (Kyiv) and "Адреса" (Address).
- Contact Info:** Fields for "Телефон" (Phone) with a "+38" prefix and "Дата народження" (Date of birth) set to "28 March 2020".

Рис. 3.9. Електронний вигляд анкети студента, яку комендант заповнює при поселенні нового мешканця

Джерело: побудовано автором

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
						40

Наприклад, система аналізує статтю студента і, якщо новий мешканець – це студентка, то запропонує кімнату, у якій проживають дівчата (рис.3.10).

```
private void gender_checkedListBox_SelectedIndexChanged(object sender, EventArgs e)
{
    var checked_list_box = (CheckedListBox)sender;
    for (int ix = 0; ix < checked_list_box.Items.Count; ++ix)
        if (ix != checked_list_box.SelectedIndex)
            checked_list_box.SetItemChecked(ix, false);
}
```

Рис. 3.10. Фрагмент коду, що перевіряє стать нового жителя гуртожитку

Джерело: побудовано автором

Всі ці критерії додаються разом таким чином, якщо студент обирає у запитанні «Переваги в сусідах» поле «Спортсмен», то програма шукатиме вже існуючих жителів-спортсменів і перевірятиме чи є місце у кімнаті і паралельно також перевірить звички та особистість студентів, чи вони галасні та яка в них чутливість під час сну (рис.3.11). Вся ця інформація є досить важливою для запобігання конфліктів між сусідами у кімнаті.

```
command = new SQLiteCommand("INSERT INTO StudCard (FirstName, MidName, SecondName, " +
    "GroupNumber, CourseNumber, IdGender, " +
    "IdPers, BirthDate, IdFac, Phone, " +
    "CityCode, Address, DateIn) " +
    "VALUES (@FirstName,@MidName,@SecondName, " +
    "@GroupNumber,@CourseNumber,@IdGender, " +
    "@IdPers,@BirthDate,@IdFac,@Phone, " +
    "@CityCode,@Address,datetime())", sql_con);
command.Parameters.AddWithValue("FirstName", f_name_textBox.Text);
command.Parameters.AddWithValue("MidName", m_name_textBox.Text);
command.Parameters.AddWithValue("SecondName", s_name_textBox.Text);
command.Parameters.AddWithValue("GroupNumber", Int64.Parse(group_textBox.Text));
command.Parameters.AddWithValue("CourseNumber", Int64.Parse(course_textBox.Text));
command.Parameters.AddWithValue("IdGender", gender_id);
command.Parameters.AddWithValue("IdPers", personality_id);
```

Рис. 3.11. Фрагмент коду штучного інтелекту, що самостійно перевіряє параметри та характеристики студентів

Джерело: побудовано автором

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
						41

Частина «Command.Parameters» перевіряє певну задану характеристику кожного студента, наприклад, курс навчання і обирає той, який обрав студент для своїх сусідів. Так само система працює з іншими параметрами. Такий шлях є найкращим, адже код аналізу даних ми прописуємо один раз, а потім лише додаємо його до кожного з характеристик, обраних в CheckBox анкети жителя кампусу.

Також програма обов'язково перевіряє наявність всіх внесених даних до анкети, і якщо дані внесені невірно або некоректно, або було пропущене поле, то система виводить повідомлення про помилку на екран (рис. 3.12). Також, щоб звернути увагу користувача вони також відмічені червоною зірочкою.

```

else if (faculty_combo_box.SelectedItem == null || String.IsNullOrEmpty(course_textBox.Text) || String.IsNullOrEmpty(group_textBox.Text))
    MessageBox.Show("Будь-ласка, вкажіть факультет, курс та групу мешканця", "Заповніть всі поля", MessageBoxButtons.OK, MessageBoxIcon.Error);
else if (pers_checkedListBox.CheckedItems.Count == 0)
    MessageBox.Show("Будь-ласка, вкажіть характер мешканця", "Заповніть всі поля", MessageBoxButtons.OK, MessageBoxIcon.Error);
else if (city_combo_box.SelectedItem == null || String.IsNullOrEmpty(address_textBox.Text))
    MessageBox.Show("Будь-ласка, вкажіть місто та адресу мешканця", "Заповніть всі поля", MessageBoxButtons.OK, MessageBoxIcon.Error);

```

Рис. 3.12. Фрагмент коду перевірки коректності введення даних

Джерело: побудовано автором

Також обов'язково перед видаленням/виселенням студента система обов'язково запитає ще раз чи дійсно користувач хоче виконати цю дію, щоб переконатися, що кнопка була натиснута не випадково, а навмисно.

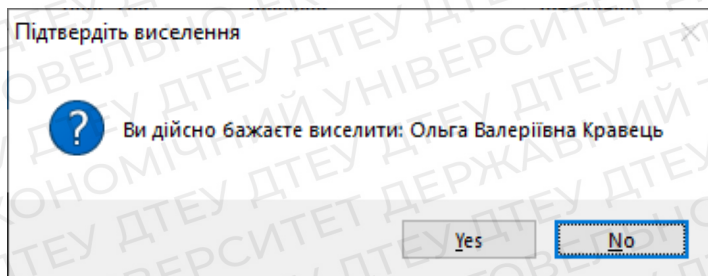


Рис. 3.13. Приклад вікна підтвердження про видалення/виселення студента

Джерело: побудовано автором

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
						42

Також варто зауважити, що дані зникають з вікна користувача, але у базі даних вони ще залишаються, для ведення звітності про студентів, котрі проживали та проживають на кампусі університету (рис.3.14).

```

button_wrapper((Button)sender, e, () => {
    if (students_grid_view.SelectedRows.Count < 1)
        MessageBox.Show("Виділіть студента з таблиці котрого ви бажаєте виселити.", "Помилка виселення", MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        if (MessageBox.Show("Ви дійсно бажаєте виселити: " + students_grid_view.SelectedRows[0].Cells["FullName"].Value.ToString(),
            "Підтвердіть виселення",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question,
            MessageBoxDefaultButton.Button2)
            == System.Windows.Forms.DialogResult.No)
            return;
        using (SqlConnection sql_con = new SqlConnection("Data Source=resources/campus.db; Version=3.0"))
    }
}

```

Рис. 3.14. Фрагмент коду, що відповідає за виселення студента

Джерело: подубовано автором

База даних була ініціалізована у програмному продукті за допомогою інструменту DataGridView, котрий дозволяє за допомогою запитів показати тільки потрібні поля кожному користувачу та повноцінно взаємодіяти з базою даних.

По закінченню розробки користувачам буде представлений зручний у користуванні програмний додаток, що допоможе збирати, редагувати, організувати та аналізувати дані про студентів, котрі проживають на кампусі.

Отже, таким чином за допомогою Visual Studio .Net можна створити повноцінну програмну платформу кампусу гуртожитків ЗВО, котра є повним програмним додатком, котрий може бути легким та зручним для використання кожному з уповноважених осіб університету для ведення даних про студентів, котрі проживають на території студентського містечка.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
					43	

3.2. Опис функціональних характеристик програмної платформи кампусу гуртожитків ЗВО

Програмна платформа кампусу гуртожитків ЗВО містить функції для повноцінного ведення всієї інформації про студентів, які проживають на території студентського містечка. А саме: редагування, додавання нових даних, зберігання та конфігурація вже наявних, аналіз існуючих даних при додаванні нових та автоматичне вирішення запиту на поселення нового мешканця, оскільки система має певні інструменти аналізу і дозволяє в разі спростити роботу комендантам гуртожитків під час поселення студентів. Система «Campus» містить певні обмеження прав для кожного з користувача, чим запобігає внесенню помилок чи випадковому видаленню даних. Завдяки інкапсуляції даних, одному з принципів об'єктно-орієнтованого програмування, система повністю захищена від будь-яких дій користувача всередині коду програми чи зміні бази даних поза дозволеними межами.

Програмна платформа має дружелюбний до користувача інтерфейс і є дуже легкою в користуванні. Вона не потребує особливих знань, і на думку автора, ведення обліку жителів саме в програмному продукті «Campus» є набагато зручнішим, ніж в будь-якому табличному редакторі чи зберіганні всієї інформації в паперовому вигляді.

Програмна платформа кампусу гуртожитків ЗВО зменшує витрати часу користувачів під час створення звітів, ведення інформації, друк паперової документації, пошуку потрібних даних, адже вся потрібна інформація по кожному з жителів знаходиться під рукою у програмі з дружелюбним користувацьким інтерфейсом. Система також дозволяє користувачу зменшити обсяги використання паперу і таким чином вести більш екологічну діяльність, що є дуже важливим та популярним у сучасному світі.

						Аркуш
					ДТЕУ 121 02мз-17.МР	
Зм.	Аркуш	№ докум	Підпис	Дата		44

3.3. Висновки до розділу 3

За допомогою обраних інструментів СУБД SQLite та середовища розробки VisualStudio .Net було створено повноцінну програмну платформу кампусу гуртожитків ЗВО задля використання її в університеті такими користувачами: коменданти гуртожитків, відділ студентського містечка університету, а також відділи деканатів факультету. За допомогою інструментів бази даних система дозволяє додавати, редагувати та видаляти дані, робити аналіз інформації та спрощувати процеси поселення нових мешканців, а також ведення обліку жителів, система аналізує наявну інформацію та обирає автоматично за допомогою елементарних інструментів штучного інтелекту кімнату для поселення нового мешканця. СУБД SQLite дозволила розробити не містку базу даних, котра легко конфігурується з усім програмним забезпеченням. А середовище розробки дозволило створити програмний продукт зі зручним користувацьким інтерфейсом, котрий полегшить процеси обліку даних та дозволить зекономити час користувачам цієї системи.

SCIENTIA DIFFICILIS SED FRUCTUOSA

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР			45

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Отже, було проведено наукове дослідження та опрацьовано літературні джерела з розробки програмної платформи кампусу гуртожитків ЗВО. В ході дослідження було зібрано та проаналізовано дані про різні підходи та методи ведення подібної інформації в університетах.

На основі цих досліджень був розроблений програмний продукт «Campus», який відповідає потребам комендантів гуртожитків у ефективному управлінні даними. Ця система надає можливість створити цифровий простір для роботи з даними, дозволяє переглядати, редагувати та додавати інформацію про мешканців гуртожитків. Важливо зазначити, що система «Campus» автоматично аналізує та порівнює анкети мешканців, що дозволяє автоматично визначати найкращу кімнату для поселення.

Таким чином, програмна платформа кампусу гуртожитків «Campus» вирішує проблему організації даних для комендантів гуртожитків та надає повний набір інструментів для управління даними та автоматичного аналізу анкет.

За час виконання випускної кваліфікаційної роботи були виконані такі завдання:

- було розроблено та наповнено БД для програмної платформи кампусу гуртожитків ЗВО;
- було розроблено алгоритму аналізу даних штучним інтелектом системи;

Зм.	Аркуш	№ докум	Підпис	Дата			
Зав. кафедри		Криворучко О.В.		01.11.23	<i>ДТЕУ 121 02мз-17.МР</i> Розробка аналітичної системи обліку жителів кампусу Висновки та пропозиції		
Керівник		Рзаєва С.Л.		01.11.23			
Гарант		Котенко Н.О.		01.11.23			
Розроб.		Удовця О.В.		01.11.23			
					Стадія	Аркуш	Аркушів
					ВП	46	51
					Факультет інформаційних технологій, 2 курс, 2мз група		

- було проведено аналіз особливостей схожих систем дізналася про особливості розробки програмної платформи кампусу гуртожитків ЗВО та використано їх під час створення програмного продукту «Campus»;
- було використано інструментів бібліотеки .Net для написання елементів штучного інтелекту, що використовується в обчисленні найзручнішої кімнати для нового мешканця аналітичною системою обліку жителів кампусу;
- було розроблено програмний додаток з дружельним інтерфейсом та підключення попередньо створеної бази даних.

На початок виконання випускної кваліфікаційної роботи було поставлене завдання створити програмний додаток, що дозволив би додавати, видаляти, редагувати дані по жителям кампусу, а також який буде аналізувати ці дані для кращого та спрощеного вибору кімнати для нового жителя гуртожитків. За час роботи над додатком були створені база даних, таблиці та запити до них, форми з користувацьким інтерфейсом, а також елементи аналізу анкет мешканців для вибору кімнати незалежно від коменданта, тобто система сама обирає кращий варіант для нового мешканця. Розроблена база даних стала автоматизованою, за рахунок підключення її до Windows.Forms, а отже є зручною та доступною для користувача, за допомогою елементарних інструментів штучного інтелекту та аналізу наданої інформації за бази даних система автоматично оброблює дані та обирає кімнату для нового жителя, замість ручної обробки даних комендантом.

Завдяки формі, створеній у середовищі розробки Visual Studio .NET, вдалося встановити зв'язок з таблицями бази даних та налаштувати користувацький інтерфейс. Використовуючи інструменти бібліотеки .NET, були створені кодові елементи, що дозволяють системі автоматично вирішувати, в яку кімнату найкраще поселити студента, враховуючи надані в анкеті інформаційні дані. База даних повністю функціонує за допомогою програмного

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	
					47	

додатку «Campus», тому можна вважати, що поставлене завдання було успішно виконано.

В майбутньому планується оновлення програмного додатку та розширення його можливостей та функціональності для ще більш комфортного ведення обліку жителів кампусу за допомогою цієї інформаційно-аналітичної системи.



									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР				48

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основний

1. R. Miles. C# Programming Yellow Book Ed.8.1. / R. Miles. Minnesota, 2016 – 216 p.
2. Дж. Оппель, Пол Н. Вайнберг. Інструкція по SQL: пер. с англ. / Дж. Оппель, Пол Н. Вайнберг. Київ, 2018 – 960с.
3. Ентоні Молінаро. SQL. Збірник рецептів: пер. з англ. / Ентоні Молінаро. Київ, 2009 – 672 с.
4. Sams. Teach Yourself SQL in 10 Minutes (3rd Edition). / Sams, 2004 – 256p.
5. Asaad, Renas. (2020). Object Oriented Programming C-Sharp Programming C# provides full support for object-oriented programming including encapsulation, inheritance, and polymorphism.
6. Бен Форма. Опануй самотійно SQL. 10 хвилин на урок, 3-є видання. / Бен Форма. Київ, 2005 – 287с.
7. Б. Карвін. Програмування баз даних SQL. Типові помилки та їх усунення. / Б. Карвін. Рід Груп, 2012 – 379с.
8. Ржеуцька С.Ю. Бази Даних. Мова SQL. / Ржеуцька С.Ю. ВоГТУ, 2010 – 159с.
9. А. Стиллімен. Вивчаємо C#. / А. Стиллімен. Київ, Рід Груп, 2012.-704с.
10. Eric Freeman, Elisabeth Robson. Head First Design Patterns, 2nd Edition, December 2020, O'Reilly Media, Inc., ISBN: 9781492078005.

					<i>ДТЕУ 121 02мз-17.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>		<i>01.11.23</i>	<i>Розробка інформаційно-аналітичної системи обліку жителів кампусу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Рзаєва С.П.</i>		<i>01.11.23</i>		<i>СВД</i>	<i>49</i>	<i>51</i>
<i>Гарант</i>		<i>Котенко Н.О.</i>		<i>01.11.23</i>		<i>Факультет інформаційних технологій, 2 курс, 2мз група</i>		
<i>Розроб.</i>		<i>Удовіця О.В.</i>		<i>01.11.23</i>				

11. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно орієнтованого програмування C#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, - 190с.

12. М. Грубер. Розуміння SQL. / М. Грубер. Київ, 1999 – 279с.

13. Rishabh Verma, Neha Shrivastava, Ravindra Akella. Parallel Programming with C# and .NET Core: Developing Multithreaded Applications Using C# and .NET Core 3.1 from Scratch (English Edition). BPB Publications (June 26, 2020). ISBN-13: 978-9389423327

14. Elsmasri, Navathe. Fundamentals of Database Systems. / Elsmasri. Navathe, 2010 – 1201р.

15. Stephane Faroult. The art of SQL. / Stphane Faroult. Nevada, 2006 – 372р.

16. Simon Kendal. Object Oriented Programming using C#, 2011, Simon Kendal & Ventus Publishing ApS. ISBN 978-87-7681-814-2

17. Benjamin Perkins, Jon D. Reid. Beginning C# and .NET, 2021 Edition ISBN: 978-1-119-79578-0

21. Ланде Д.В. Новітні підходи й технології інформаційно-аналітичної підтримки прийняття рішень. // Національна безпека: український вимір: щокв. наук. зб. / Рада нац. безпеки і оборони України, Ін-т пробл. нац. безпеки; - К., 2008. - Вип. 1-2 (20-21). - С. 87-105.

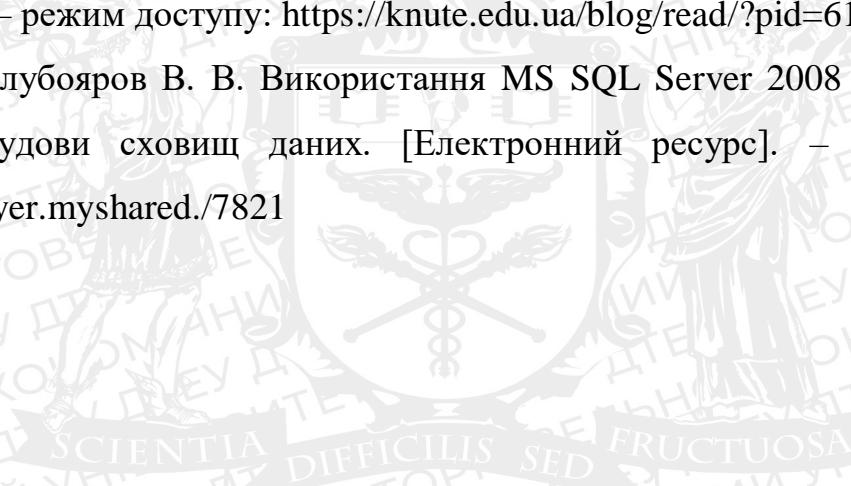
Інтернет-ресурси

1. Об'єктно-орієнтоване програмування [Електронний ресурс]. – режим доступу: <http://programming.in.ua/programming/basisprogramming/25-oop.html>

2. Середовище Visual Studio .NET [Електронний ресурс]. – режим доступу: http://life-prog.ru/ukr/1_4216_seredovishche-Visual-Studio-NET.html

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 02мз-17.МР	50

3. developer.microsoft.com [Електронний ресурс]. - режим доступу:
<https://msdn.microsoft.com/en-us/>
4. Онлайн-курси coursera від IBM [Електронний ресурс]. – режим доступу:
<https://coursera.org/learn/sql-data-science>
5. sqlite.org [Електронний ресурс]. – режим доступу:
<https://www.sqlite.org/about.html>
6. Бібліотека конгресу США [Електронний ресурс]. – режим доступу:
<https://www.loc.gov/preservation/resources/rfs/data.html>
7. Список документів на поселення у гуртожитки КНТЕУ. [Електронний ресурс]. – режим доступу: <https://knute.edu.ua/blog/read/?pid=6157&uk>
8. Полубоярів В. В. Використання MS SQL Server 2008 Analysis Services для побудови сховищ даних. [Електронний ресурс]. – режим доступу:
<http://player.myshared./7821>



						<i>ДТЕУ 121 02мз-17.МР</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докum</i>	<i>Підпис</i>	<i>Дата</i>			<i>51</i>

ДОДАТКИ

Додаток А

Модуль форми вітального вікна

```
namespace campus {
    public partial class Welcome : Form {
        delegate void LoadingCompleteCallback(object
state);
        public Welcome() {
            InitializeComponent();
        }
        private void Welcome_Load(object sender,
EventArgs e) {
            const int kLoadTime = 4300; // Час в
мілісекундах протягом якого відображається вітальне
вікно.
            CenterToScreen();
            new System.Threading.Timer(LoadingComplete,
null, kLoadTime, 0);
        }
        private void LoadingComplete(object state) {
            if (InvokeRequired)
                Invoke(new
LoadingCompleteCallback(LoadingComplete), state);
            Else {
                Hide();
                new Main().ShowDialog(this);
                Close();
            }
        }
    }
}
```


Модуль основної форми вибору користувача

```
namespace campus {  
    public partial class Main : Form  
    {  
        public Main() {  
            InitializeComponent();  
        }  
        private void Main_Load(object sender, EventArgs  
e) {  
            CenterToParent();  
        }  
        private void komendant_button_Click(object  
sender, EventArgs e) //вибір користувача коменданта  
{  
            new  
HostelSelector(ControlPanel.Role.Komendant).ShowDialog(th  
is); //відкриття форми вибір гуртожитку  
        }  
        private void campus_button_Click(object sender,  
EventArgs e) //вибір користувача відділ студмістечка  
{  
            new  
HostelSelector(ControlPanel.Role.StudCity).ShowDialog(thi  
s); //відкриття форми вибір гуртожитку  
        }  
        private void decans_office_button_Click(object  
sender, EventArgs e) //вибір користувача - відділ  
деканату {  
            new  
FacultySelector().ShowDialog(this);  
            //відкриття форми вибору факультету  
        }  
    }  
}
```

Модуль форми коменданта та відділу студентського містечка

```

namespace campus {
    public partial class ControlPanel : Form {
        public enum Role { Komendant, StudCity }; //якщо
це комендант
        private Role role { get; }
        private long hostel_id { get; } //зчитування
інформації про номер гуртожитку
        public ControlPanel(Role role, long hostel_id) {
            InitializeComponent();
            this.role = role;
            this.hostel_id = hostel_id;
        }
        private void LoadStudents() {
            students_grid_view.Rows.Clear(); //запит полів до бази
даних на показ інформації про студентів
            using (SQLiteConnection sql_con = new
            SQLiteConnection("Data Source=resources/campus.db;
            Version=3;")) {
                try {
                    sql_con.Open(); // Get students of the hostel
                    var cmd = new SQLiteCommand("SELECT *
FROM StudCard " +
                    "INNER JOIN Poselenya ON StudCard.IdStud
= Poselenya.IdStud " +
                    "INNER JOIN Rooms ON Poselenya.IdRoom =
Rooms.IdRoom " +

```



```

students_grid_view.Rows.Add(result["IdStud"],
result["FirstName"] + " " + result["MidName"] + " " +
result["SecondName"], debt, MessageBoxIcon.Error);
        Else {
            if (MessageBox.Show("Ви дійсно бажаєте виселити: " +
students_grid_view.SelectedRows[0].Cells["FullName"].Value
e.ToString() "Підтвердіть виселення",
MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2)
new AddKomunPlat(new
Resident((long)students_grid_view.SelectedRows[0].Cells["
ID"].Value)).ShowDialog(this);
                LoadStudents();
            });
        }
private void students_grid_view_KeyDown(object sender,
KeyEventArgs e) {
    var grid_view = (DataGridView)sender;
    if (e.KeyCode == Keys.Escape)
        grid_view.ClearSelection();
    }
private void ControlPanel_KeyDown(object sender,
KeyEventArgs e) {
    students_grid_view_KeyDown(students_grid_view, e);
    } } }

```

Модуль форми відділу деканату факультетів

```

namespace campus {
    public partial class Students : Form {
        private long faculty_id { get; set; }
        public Students(long faculty_id) {
            this.faculty_id = faculty_id;
            InitializeComponent();
        }
        private void Students_Load(object sender,
            EventArgs e) //запит до бази даних по запрошених полях
            для отримання інформації у додатку {
            CenterToParent();
            using (SQLiteConnection sql_con = new
            SQLiteConnection("Data Source=resources/campus.db;
            Version=3;")) {
                try {
                    sql_con.Open(); // Get faculty name
                    Poselenya.IdStud " +
                catch (SQLiteException ex) {
                    MessageBox.Show(ex.Message, "Database
                    error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            private void back_button_Click(object sender,
            EventArgs e) {
                Hide();
                new Main().ShowDialog(this);
                Close();
            }
        }
    }
}

```


Модуль створення бази даних

```
CREATE TABLE StudCard (  
    IdStud integer PRIMARY KEY AUTOINCREMENT,  
    FirstName text,  
    MidName text,  
    SecondName text,  
    GroupNumber integer,  
    CourseNumber integer,  
    IdGender integer,  
    IdPers integer,  
    BirthDate datetime,  
    IdFac integer,  
    Phone text,  
    CityCode integer,  
    Address text,  
    DateIn datetime  
);  
  
CREATE TABLE StudPreferences (  
    IdStudPref integer PRIMARY KEY AUTOINCREMENT,  
    IdStud integer,  
    IdPref integer  
);  
  
CREATE TABLE StudHabits (  
    IdStudHabit integer PRIMARY KEY AUTOINCREMENT,  
    IdStud integer,  
    IdHabit integer );  
  
CREATE TABLE Faculties (  
    IdFac integer PRIMARY KEY AUTOINCREMENT,
```

```
Name text
);
INSERT INTO "Faculties" ("IdFac","Name") VALUES
('1','ФІТ');
INSERT INTO "Faculties" ("IdFac","Name") VALUES
('2','ФМП');
CREATE TABLE Cities (
    CityCode integer PRIMARY KEY AUTOINCREMENT,
    Name text
);
INSERT INTO "Cities" ("CityCode","Name") VALUES
('1','Київ');
INSERT INTO "Cities" ("CityCode","Name") VALUES
('2','Кропивницький');
INSERT INTO "Cities" ("CityCode","Name") VALUES
('3','Суми');
INSERT INTO "Cities" ("CityCode","Name") VALUES
('4','Вінниця');
INSERT INTO "Cities" ("CityCode","Name") VALUES
('24','Рівне');
CREATE TABLE Poselenya (
    IdPos integer PRIMARY KEY AUTOINCREMENT,
```