

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація / освітня програма 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки
Криворучко О. В.
"7" листопада 2019 р.

Завдання на випускню кваліфікаційну роботу студентці

Давидовій Анні Вячеславівні

1. Тема випускної кваліфікаційної роботи: Розробка аналітичної системи обліку жителів кампусу.

Затверджена наказом ректора від «02» грудня 2019 р. № 4112

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи:

Мета роботи: дослідження існуючих аналітичних та 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 на 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. #Вастосування 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 жиетлів кампусу*
- 2.2. Постановка задачі*
- 2.3. Аналіз предмету розробки інформаційно-аналітичної системи обліку жителів кампсу*
- 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

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>		
2.	<i>Вступ та перелік літературних джерел</i>		
3.	<i>Розділ 1. Основні функціональні характеристики аналітичної системи обліку жителів кампусу та технічне завдання</i>		
4.	<i>Розділ 2. Предметна область розробки системи обліку жителів кампусу</i>		
5.	<i>Розділ 3. Розробка інформаційно-аналітичної системи обліку жителів кампусу</i>		
6.	<i>Висновки</i>		
7.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>		
8.	<i>Підготовка автореферату та презентації доповіді</i>		
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>		
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>		
11.	<i>Здача прожитої випускної кваліфікаційної роботи на кафедрі</i>		
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «__» _____ 20__ р.

8. Науковий керівник випускної кваліфікаційної роботи

Рзаєва С. Л.

9. Гарант освітньої програми

Цензура М. О.

10. Завдання прийняв до виконання студент

Давидова А. В.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(nidnuc, dama)

Рзаева С. Л.

(ПІБ, *nidnuc*, *data*)

Випускна кваліфікаційна робота студентки Давидової А.В. може бути допущена до захисту екзаменаційній комісії.

Цензура М. О.

Криворучко О. В.

20 p.

АНОТАЦІЯ

Дипломна робота на тему «Розробка аналітичної системи обліку жителів кампусу» містить 57 сторінок, 18 рисунків та 5 додатків. Перелік посилань налічує 29 найменувань.

Мета дослідження: роботи є дослідження існуючих аналітичних та 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 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 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" contains 57 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.

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка аналітичної системи обліку жителів кампусу»

Студентки 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

підпис студента

Давидової Анни
Вячеславівни

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Рзаєва Світлана
Леонідівна

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ — 2020

ЗМІСТ

ЗМІСТ	2
ВСТУП	4
РОЗДІЛ 1. ОСНОВНІ ФУНКЦІОНАЛЬНІ ХАРАКТЕРИСТИКИ АНАЛІТИЧНОЇ СИСТЕМИ 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 СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ	19
<i>2.1. Відбір інформації для інформаційно-аналітичної системи обліку жителів кампусу</i>	<i>19</i>
<i>2.2. Постановка задачі</i>	<i>22</i>
<i>2.3. Аналіз предмету розробки інформаційно-аналітичної системи обліку жителів кампусу</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 ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ	33
<i>3.1. Створення інформаційно-аналітичної системи у середовищі Visual Studio .Net33</i>	

					КНТЕУ 121 06-03.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Зміст	2	50
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М. О.							
Розроб.	Давидова А.В.				Зміст			

3.2. Опис функціональних характеристик інформаційно-аналітичної системи обліку жителів кампусу	43
3.3. Висновки до розділу 3	434
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ	51
<i>Додаток А. Модуль форми вітального вікна</i>	<i>51</i>
<i>Додаток Б. Модуль основної форми вибору користувача</i>	<i>512</i>
<i>Додаток В. Модуль форми коменданта та відділу студентського містечка</i>	<i>53</i>
<i>Додаток Г. Модуль форми відділу деканату факультетів.....</i>	<i>55</i>
<i>Додаток Д. Модуль створення бази даних</i>	<i>56</i>

					КНТЕУ 121 06-03.БР	Аркуш
						3
Зм.	Аркуш	№ докум	Підпис	Дата		

ВСТУП

Створення інформаційно-аналітичної системи обліку жителів кампусу є актуальним для кожного університету не тільки країни, але й світу. Розробка програмного забезпечення дозволить швидко та зручно керувати інформацією про студентів, котрі проживають на території університету. Кампус або територія студентського містечка (студмістечка), де знаходяться гуртожитки – є однією з найважливіших складових території університету, за неї несуть відповідальність безліч людей, ведеться купа документообігу. Задля забезпечення легкого керування цією інформацією пропонується розробити програмний продукт, що дозволить зручно та швидко управляти даними, буде аналізувати отриману інформацію.

У сучасному світі є дуже не зручним та не ергономічним ведення та зберігання інформації на паперових носіях або ж у табличних редакторах. Адже кількість даних збільшується з кожним днем і управляти нею стає дедалі важче. На практиці доведено, що це є затратно в часі та інших ресурсах, а також є не продуктивно. Саме тому автор пропонує для покращення ведення обліку жителів кампусу використовувати систему управління базами даних (СУБД). Одним із популярних видів СУБД є засоби обробки табличної інформації. За допомогою інструментальних засобів, що вони мають, СУБД дозволяють оброблювати великі об'єми інформації. Автоматизація процесів ведення інформації про студентів дозволить ефективніше управляти будь-якими об'єктами, зменшить ризик виникнення помилок, буде більш зручною для

					КНТЕУ 121 06-03.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри	Криворучко О.В.				Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Керівник	Рзаєва С.Л.					В	4	50
Гарант	Цензура М. О.					Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.	Давидова А.В.				Вступ			

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 інформаційно-аналітична система обліку жителів кампусу.

Під час поселення у гуртожиток студенти заповнюють анкети та підписують контракти, вони оплачують проживання і вистояють довгі черги. Всі ці завдання можна спростити, замість довготривалих очікувань в черзі на поселення, можна заповнити анкету та отримати реквізити на сплату онлайн, надіслати документи електронною поштою, щоб комендант ввів усе до програми, яка зберігатиме всю потрібну інформацію. По-перше, це дозволить зекономити час, як комендантам гуртожитків, так і студентам. А, по-друге, дозволить університетові вести більш екологічну діяльність, оскільки зменшиться використання паперу. Цей програмний продукт спростить процес поселення студентів, адже проаналізувавши дані з анкети нового жителя, система обере кімнату найкращу для поселення саме цього студента. Тому тема «Розробка інформаційно-аналітичної системи обліку жителів кампусу» є актуальною.

Метою дослідження випускної кваліфікаційної роботи є дослідження існуючих інформаційно-аналітичних систем обліку жителів кампусу та

					КНТЕУ 121 06-03.БР	Аркуш
						5
Зм.	Аркуш	№ докум	Підпис	Дата		

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

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

Завдання дослідження:

- визначити основні поняття та терміни, котрі будуть використовуватися у роботі;
- розробка СУБД для інформаційно-аналітичній системі обліку жителів кампусу;
- проектування запитів для бази даних;
- створення програмного продукту, що надасть можливість зручно управляти даними про студентів-жителів кампусу;
- розробка програмного продукту «Інформаційно-аналітичної системи обліку жителів кампусу».

Метод дослідження – спостереження.

					КНТЕУ 121 06-03.БР	Аркуш
						6
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1.

ОСНОВНІ ФУНКЦІОНАЛЬНІ ХАРАКТЕРИСТИКИ АНАЛІТИЧНОЇ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

1.1. Загальні відомості про системи обліку жителів

Наразі кожен університет світу потребує сучасну систему обліку жителів студентського містечка. Аналізуючи те, як ведеться управління інформацією про студентів, котрі проживають у гуртожитках Київського національного торговельно-економічного університету (КНТЕУ), автор зрозуміла, що потрібно знайти більш сучасний спосіб ведення аналітичної інформації про мешканців.

Результат дослідження існуючого стану виявив, що для ведення обліку жителів у гуртожитку КНТЕУ досі використовується табличний редактор, а також багато паперових документів. По-перше, зрозуміло, що табличний редактор не дозволяє одночасний доступ до даних, наприклад, як комендантові, так і дирекції студентського містечка. Він є ненадійним та доволі не зручним у використанні. По-друге, у XXI столітті всі підприємства, організації та установи переводять свою діяльність у без паперовий спосіб, саме тому буде доцільним модернізувати систему обліку жителів кампусу для того, щоб зменшити всю паперову волокнину, а також знайти ефективне рішення з економією часу всіх залучених сторін.

Наразі деякі університети вже використовують автоматизовані системи обліку жителів кампусу. Наприклад, у Сполучених Штатах Америки студенти за бажання проживати у гуртожитку заповнюють всі дані он-лайн та

					КНТЕУ 121 06-03.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Р1	7	50
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М. О.							
Розроб.	Давидова А.В.							
					Основні функціональні характеристики інформаційної системи обліку жителів кампусу та технічне завдання			

надсилають на спеціальну електронну адресу університету. Такий спосіб подачі документів відразу економить час на дорогу, на спілкування, а також не має ніяких черг на подачу документів. Такий спосіб приводить до великої економії паперу, адже всі потрібні форми можна заповнити он-лайн. Також, наприклад, в університеті Отербайн (штат Огайо) існує програмне забезпечення, яке має в собі базу даних про всіх студентів, котрі живуть на території кампусу, і доступ до цієї інформації мають всі уповноважені особи. Саме цю базу даних автор взяла як приклад для розробки інформаційно-аналітичної системи обліку жителів кампусу для КНТЕУ.

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.

					КНТЕУ 121 06-03.БР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

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

Проаналізувавши ведення обліку жителів кампусу в різних університетах країн, автор вирішила розробити програмний продукт, котрий би дозволив зберігати всю інформацію про студентів, які проживають у гуртожитках на території університету, надати доступ до потрібної інформації усім уповноваженим особам, таким як комендант гуртожитку, дирекція студентського містечка та деканатам усіх факультетів. Також є важливим, щоб ця система не тільки зберігала дані та дозволяла керувати ними, основним завданням є – за допомогою аналітичних методів та найпростіших інструментів штучного інтелекту підбирати кімнату, яка б найкраще підходила для нового жителя, спростивши таким чином ручний розподіл поселення комендантами гуртожитків що відбувається кожного літа.

Для розробки будь-якого програмного забезпечення дуже важливо розібратися повністю у всіх його дрібницях, підготувати матеріали, опрацювати інформаційну базу, саме тому спочатку необхідно розробити технічне завдання для програмного продукту.

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 ефективне зберігання, обробка та аналіз даних. Технологічна платформа ІАС дозволяє підприємству (організації) здійснювати інтеграцію та координацію його бізнес-процесів. [23]

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

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 рішенням для управління інформацією. Є багато видів СУБД в залежності від того, з якою метою їх використовують, їх зв'язок з повним програмним забезпеченням, конфігурація з операційними системами, завантаженість пам'яті на комп'ютері користувача. СУБД дозволяє повноцінно завантажити та зберігати всю інформацію на сервері університету і надає доступ до неї кожній потрібній уповноваженій особі, що є досить зручним і не потребує багато знань для цього. Система управління базами даних – це численне програмне забезпечення, за допомогою якого можна створювати структури, які потрібні для роботи з базами даних та вводити в них необхідні елементи управління. Основна функція СУБД – це керування даними у зовнішній пам'яті. Вона включає збереження та ведення даних, перетворення даних за запитом на структурну інформацію для користувача.

Також до теоретичних положень теми можна віднести вивчення проблеми поселення студентів у гуртожитки, котра виникає кожного року, можемо зробити застосунок, що зробить зручнішим збереження та користування інформацією про студентів.

Приблизно 2,5 тисячі студентів проживають у гуртожитках Київського національного торговельно-економічного університету. Наразі університет має 5 гуртожитків для студентів бакалаврського та магістерського ступеня. У коменданта гуртожитку має бути повна інформація про кожного з проживаючих у певному гуртожитку. Також до даної інформації повинні мати доступ дирекція студмістечку та деканати факультетів. Для того, щоб синхронізувати

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

всі дані та зменшити паперову волокнину автор цієї роботи вирішила зробити програмний застосунок, що допоможе зберігати всю інформацію в одному місці.

У кожному з гуртожитків проживають студенти з різних факультетів, і навіть в кімнатах можуть зустрітися дівчата/хлопці абсолютно різних курсів чи навіть магістри різних факультетів та спеціальностей. Іноді таке поєднання може призвести до конфлікту між співмешканцями. Саме тому, хочеться зберігати інформацію так, щоб при поселенні нових студентів була можливість направити їх в кімнату, де вже є хтось з їх факультету та курсу, а якщо пощастить то й з групи.

Отже, потрібно зробити програмний застосунок, котрий зберігатиме інформацію по кімнаті та проживаючим у ній особам, надасть можливість зручно конфігурувати та управляти даними, дозволить автоматизувати деякі процеси ведення обліку та спростить поселення у кімнати.

1.3. Технічне завдання

1. Загальні відомості

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

1.1. Найменування системи

1.1.1. Повне найменування системи

«Інформаційно-аналітична система обліку жителів кампусу «Campus»»

1.1.2. Скорочене найменування системи

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

					КНТЕУ 121 06-03.БР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

1.2. Головний бенефіціар та потенційні користувачі системи

Для початку «інформаційно-аналітична система обліку жителів кампусу «Сампус»» розробляється як програма для внутрішнього користування тільки працівниками Київського національного 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 XXI 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 мешканцями, завдяки автоматизації багатьох процесів та інструментів аналітичних функцій для кращого ведення обліку всіх мешканців.

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

3. Вимоги до системи

3.1. Вимоги до системи в цілому

3.1.1. Вимоги до структури та функціонування системи, перелік підсистем

Програма «Сампус» підключена до загального серверу Київського національного 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.

3.1.1.4. Вимоги до режимів управління системою

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		13

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

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

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.

3.2. Вимоги до видів забезпечення

3.2.1. Вимоги до інформаційного забезпечення

					КНТЕУ 121 06-03.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

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

Інформаційно-аналітична система «Campus» 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 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

					КНТЕУ 121 06-03.БР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

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

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

1.4. Висновки до розділу 1

Таким чином проаналізувавши отримані дані, зрозуміло, що наявна система ведення обліку жителів кампусу потребує оновлення. В деяких університетах є приклади продуктивних та зручних систем обліку студентів, які розглянуті вище, вони мають багато переваг над звичайним табличним редактором і є набагато ефективнішими у користуванні та управлінні даними.

У першому розділі були проаналізовані програмні продукти, які дозволяють автоматизувати збір та управління інформацією про студентів, проживаючих у гуртожитках. Зроблено висновки, що потрібна інформаційно-аналітична система обліку жителів кампусу, яка дозволить зберігати вже наявні дані, додавати та редагувати нові, надасть вільний доступ до даних уповноваженим особам, тобто комендантам, дирекції студмістечка та деканатам факультетів. Тому існує потреба у створенні програмного продукту з метою аналізу даних про студентів та кращого підбору кімнати для нового жителя.

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

РОЗДІЛ 2.

ПРЕДМЕТНА ОБЛАСТЬ РОЗРОБКИ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

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

Головною метою розробки інформаційно-аналітичної системи обліку жителів кампусу або розробка додатку «Campus» є створення інформаційної бази даних про жителів гуртожитків, з можливістю додавання аналітичних функцій задля підбору найкращої кімнати для нового мешканця.

Проаналізувавши проблему поселення у гуртожитки, котра виникає кожного року, можемо зробити програмний додаток, що зробить зручнішим збереження та користування інформацією про студентів.

Саме тому додаток «Campus» пропонує свої аналітичні інструменти, за допомогою яких програма вираховує в яку ж кімнату краще поселити нового мешканця.

Для запобігання конфліктам мешканців між собою автор також пропонує перед поселенням заповнити маленьку анкету студентіві, яку потрібно буде передати комендантові.

Беручи досвід поселення студентів закордоном, автор хоче покращити цей процес та полегшити його для обох сторін. Тому було створено таку анкету, яка по-перше, враховує курс та факультет, спеціальність майбутнього мешканця, а також вона враховує базові звички такі як, схильність до чистоти та тиші, котрі можуть помітно впливати на конфліктні ситуації поміж жителями кімнати.

					КНТЕУ 121 06-03.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					Р2	19	50
Керівник	Рзаєва С.П.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М. О.							
Розроб.	Давидова А.В.				Предметна область розробки системи обліку жителів кампусу			

Адже, якщо у когось дуже чутливий сон, то навряд чи він захоче засинати під галасу чи під гучну музику. А також, відмітивши рівень чистоти для себе, можна зрозуміти, що охайного студента поселити з неохайним навряд чи буде хорошою ідеєю.

Для того, щоб не опрацьовувати ці анкети власноруч, адже тисячі студентів проживають у всьому студмістечкові, програма «Campus» робить це для вас 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.

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:

					КНТЕУ 121 06-03.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

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

Така коротка анкета дозволить зрозуміти звички нового мешканця та підібрати схожих за характером сусідів для того, щоб не перетворити життя у кімнаті у випробування на виживання. Звичайно ж кожен студент, котрий проживає у гуртожитку, повинен підтримувати чистоту та дотримуватися розпорядку дня, але більшість студентів можуть забути помити чашку, засидітися допізна з книжкою, або дуже голосно розсміятися на жарт. Хтось може проігнорувати такі дрібниці, а для когось вони можуть стати величезною проблемою, врахувавши такі дрібниці, комендант полегшить собі роботу, адже в подальшому йому не доведеться вирішувати конфлікти у кімнаті. Адже іноді вони доходять до того, що мешканці хочуть роз'їхатися. І посеред навчального року починається пошук вільних кімнат та розселення мешканців, що є зовсім неприємною справою.

Таким чином, для того, щоб зменшити паперову волокнину, а також забезпечити меншу кількість конфліктних ситуацій між мешканцями гуртожитків, зекономити час студентів та працівників університету і потрібна система обліку жителів. Саме тому програмний продукт «Campus» є не тільки чудовим рішенням для зберігання інформації по всіх мешканцях у гуртожитках, а також є дійсним способом для кращого вибору кімнати для поселення нових жителів кампусу.

					КНТЕУ 121 06-03.БР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

2.2. Постановка задачі

Проаналізувавши зібрані дані, можна зробити такі висновки щодо постановки завдання на розробку інформаційно-аналітичної системи жителів кампусу:

- налаштувати Базу Даних, яка буде не місткою за обсягом пам'яті, крос-платформною та дозволить зручно конфігурувати дані та легко суміщається з програмним забезпеченням;
- створити інформаційно-аналітичну систему обліку жителів кампусу, 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;
- 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.

Отже стає зрозумілим, що перед автором постає завдання розробити модернізовану інформаційно-аналітичну систему обліку жителів кампусу, котра буде доступною для всіх уповноважених осіб, котрі повинні мати доступ до інформації про студентів, а тобто для комендантів гуртожитків, дирекції студентського містечка та деканатів факультетів, також вона повинна бути

					КНТЕУ 121 06-03.БР	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

зручною у користуванні, автоматизувати певні процеси ведення обліку мешканців та їх поселення, бути економною в часі, спрощувати дії на введення інформації вручну, зменшувати обсяг використання та ведення паперової документації.

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

Оскільки автор хоче розробити програмний продукт, котрий буде дійсно ефективним рішенням для ведення обліку мешканців студмістечка, з особливою важливістю потрібно підійти до вибору інструментів програмування та розробки.

Для ведення обліку студентів знадобиться база даних, котра буде легкою у використанні, яку можна буде розмістити на сервері, та яка не потребуватиме забагато ресурсів, резервні копії якої можна буде спокійно зберігати час від часу. Також для розробки модернізованої системи потрібна мова програмування високого рівня, за допомогою інструментів якої можна буде створити аналітичний застосунок з інструментами елементарного 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

Важливим є саме правильно розставити пріоритети у анкеті нового мешканця кампусу. Обчислювальні елементи в коді аналітичної системи обліку кампусу перевіряє кожен пункт анкети згідно до їх значення.

					КНТЕУ 121 06-03.БР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

Наприклад, зрозумілим є те, що хлопці і дівчата не можуть проживати в одній кімнаті. Найперша перевірка – це стать нового мешканця. Кількість кімнат вже зменшується наполовину автоматично, просто завдяки аналізу за статтю людини.

Після цього ми перевіряємо приналежність до того чи іншого факультету, курсу та групи. Оскільки система автоматично шукає студентів близьких за духом, ці три чинники перевіряються одночасно.

Після того, як ми обрали студентів з одного факультету та курсу, групу можна не завжди знайти ту саму, або кімната може бути вже повною – ми скоротили нашу вибірку ще в рази. Система аналізує ці дані автоматично завдяки прописаним елементам коду, користувачу навіть не потрібно нічого перевіряти, окрім введених даних з анкети.

Наступним пріоритетом буде чутливість сну, адже у житті кожного сон грає важливу роль. Після перевірки за цим параметром, ми перевіряємо чи будуть сусіди нового мешканця чистюлями, чи його не особливо нервуватиме забута чашка на столі.

Після того як виконається перевірка попередньо наведених ознак кожного з потенційних сусідів для нового студента, буде отримано результат з декількома кімнатами. Система перевіряє останній критерій на вподобання жителем своїх сусідів чи мають вони бути спортсменами чи тихі за характером.

І в результаті, після цієї перевірки користувач, а саме, комендант отримує повідомлення, що для студента було обрано певну кімнату.

Весь цей процес обчислення та аналізу жителів кампусу займає всього лишень декілька секунд користувача та повністю автоматизує процес вибору кімнати для нового жителя.

					КНТЕУ 121 06-03.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

2.5. Створення інформаційної бази в СУБД SQLite

База даних для інформаційно-аналітичної системи обліку жителів кампусу 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);
14. *Довідник уподобань сусідів* (id уподобання, назва уподобання).

					КНТЕУ 121 06-03.БР	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

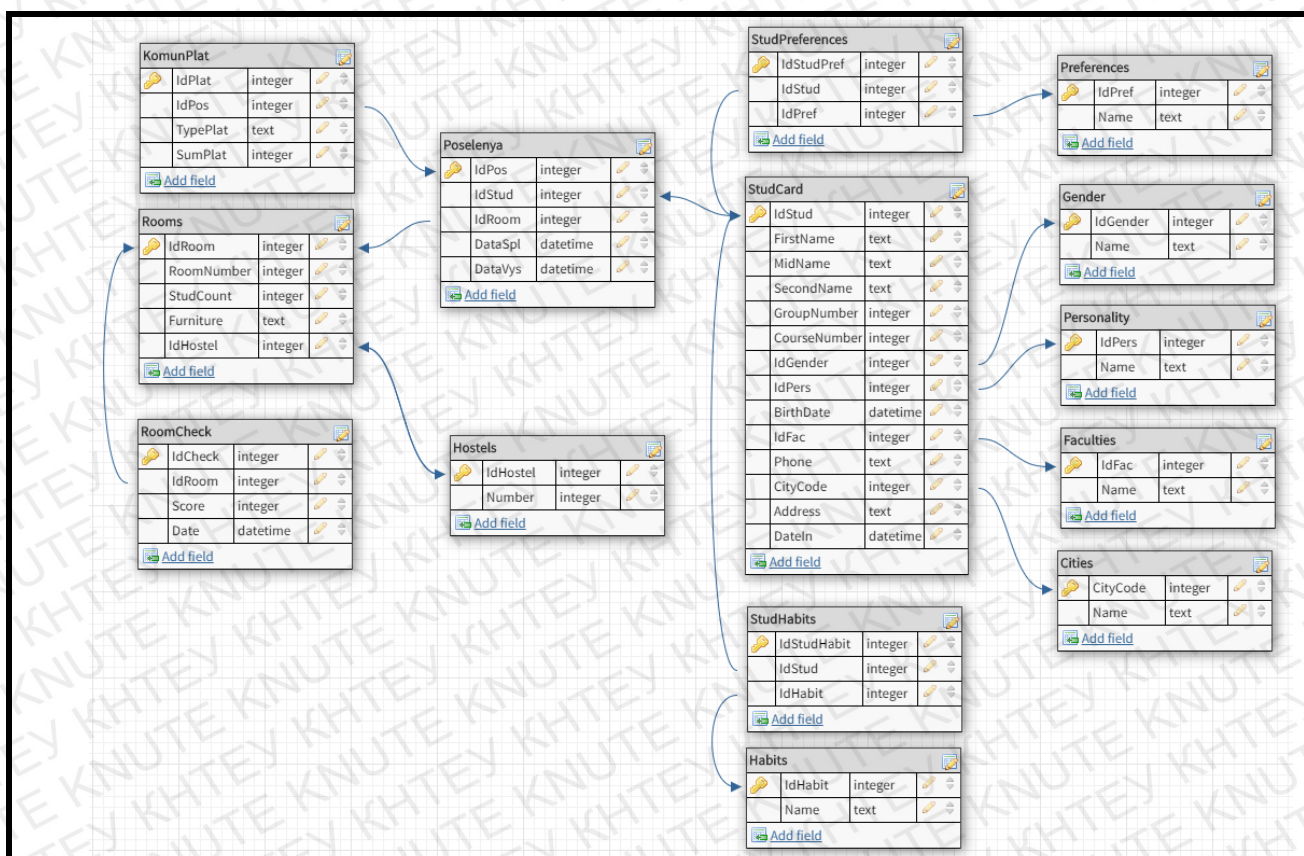


Рис. 2.1. Фізична модель бази даних інформаційно-аналітичної системи обліку жителів кампусу

Після проектування схеми даних потрібно створити таблиці. Таблиця – це основа будь-якої бази даних, це сукупність пов’язаних даних, що зберігаються в структурованому вигляді в базі даних. Вона складається з стовпців та рядків. В реляційних базах даних таблиця – це набір елементів даних (значень), котрі використовують модель вертикальних стовпців (ті, що мають унікальне ім’я) та горизонтальних рядків.

Дані в таблицях не обов’язково фізично зберігаються в базі даних. Представлення також функціонують, як реляційні таблиці, але їх дані вираховуються під час виконання запиту.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-03.БР	26

Перед тим, як створювати таблицю потрібно розібратися для чого призначаються різні типи полів та даних.

Для введення числових даних слід використовувати числові поля. Вони мають розмір, також бувають різними, наприклад для введення цілочисельних та десяткових значень. Для останніх потрібно також ввести крім розміру поля ще й розмір десяткової частини.

Для введення даних дати чи часу існує тип поля Дата/Час.

Текстові поля призначені для запису текстових даних, як зрозуміло. Основна властивість текстового поля – розмір, з ним є пов'язаний недолік – це обмеження не більше 256 символів). Якщо потрібно ввести довгий текст, що перевищує цю кількість символів, слід використовувати поле типу МЕМО (цей тип дозволяє зберігати до 65535 символів). Але цей тип має певну особливість, вона полягає в тому, що реально ці дані зберігаються в іншому місці, а не в полі (в ньому зберігається тільки покажчик на те, де розташований текст).

Поле Лічильник – це автоматизоване поле для нарощування кількості, тобто, якщо воно є у базі даних, то про додаванні нового запису в нього автоматично вводиться число на одиницю більше, ніж значення з попереднього запису. Воно є зручним для нумерації записів.

Отже, структура запису (сукупність назв полів, їх типів та властивостей, визначених користувачем під час надання конкретного завдання) – це структура таблиці. Вона визначає послідовність розташування даних у записі на фізичному носії і вигляд даних на екрані.

За допомогою набутих знань та аналізу поставленого завдання були створені потрібні таблиці:

					КНТЕУ 121 06-03.БР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

«Довідник міст» містить інформацію з кодом та назвою міста. Ця таблиця пов'язана тільки з таблицею «Довідник студентів» по первинному ключу «Код міста».

«Довідник студентів» включає в себе ід студента, його прізвище, ім'я та по-батькові студента, ід факультету, номер групи, дату народження, контактний телефон, код міста, адресу, телефон батьків, ід типу навчання, дату заселення, ід звички, ід особистості, номер кімнати, в яку був поселений студент, ід статі студента, а також номер курсу, на якому навчається новий мешканець. Ця таблиця зв'язана з багатьма таблицями: про «Довідник міст» вже було написано вище; з таблицею «Довідник факультетів» по ключу «ід факультету»; з таблицею «Поселення» по ключу «ід студента»; з «Особистістю» по полю «ід особистості»; з таблицею «Звички» по ключу «ід звички», а також з таблицею «Стать» за полем «ід статі»; з таблицею «Уподобання студентів» пов'язана за ключем ід студентського вподобання.

Таблиця «Довідник факультетів» містить тільки ід факультету та його назву та пов'язана з таблицею «Довідник студентів» по первинному ключу ід факультету.

Таблиця «Статі» містить тільки ід статі та її назву та пов'язана з таблицею «Довідник студентів» по первинному ключу ід статі.

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

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

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

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

Таблиця «Уподобання» містить лише ід уподобання та його характеристику.

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

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

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».

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

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.

					КНТЕУ 121 06-03.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

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

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.).

					КНТЕУ 121 06-03.БР	Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата		

```

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

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.

ПІБ	Борг, грн	Телефон	Факультет	Кімната, №	Стан кімнати
Анна Вячеславівна Давидова	0	+38 (095) 61-02-612	ФІТ	1	X/5

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

					КНТЕУ 121 06-03.БР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

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 інформаційно-аналітичної системи обліку студентів, котрі проживають на території студентського містечка.

2.6. Висновки до розділу 2

Отже, для написання елементів штучного інтелекту для аналітичної системи обліку жителів кампусу та її функції автоматичного поселення жителів кампусу було визначено предметну область та її інструменти, а саме: середовище Visual Studio .NET з мовою програмування високого рівня C#, СУБД SQLite. Дослідивши анкети на поселення інших університетів, котрі володіють схожими системами обліку, було обрано найкращий та найзручніший приклад анкети для заповнення мешканцями студентського містечка. При постановці задачі було описано всі основні завдання та функції, а саме зберігання даних, доступ до них трьома різними видами користувачів з різними правами, а також повна автоматизація поселення нових жителів кампусу завдяки використанню елементарних частин штучного інтелекту для аналізу наявних даних. Перш за все важливо продумати усі зв'язки інформації у базі даних. Базу даних можна легко створити, відредагувати та змінити певні типи, але до моменту підключення її до програмного продукту. В даному розділі було наведено схему даних та описано таблиці, що будуть використовуватися для ведення обліку жителів студмістечка.

					КНТЕУ 121 06-03.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

РОЗДІЛ 3.

РОЗРОБКА ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ОБЛІКУ ЖИТЕЛІВ КАМПУСУ

3.1. Створення інформаційно-аналітичної системи у середовищі Visual Studio .Net

Одне з основних завдань проекту – створення аналітичної системи управління даними. Тому створення форм є невід’ємною частиною цього ж завдання, тому що форми спрощують та пришвидшують роботу з базою даних, дозволяють створити зручний та дружлюбний користувацький інтерфейс, є основним способом взаємодії користувача з даними та мають безліч переваг та функцій.

Для зручного управління та користування базою даних інформаційно-аналітичної системи обліку жителів кампусу потрібні такі форми: вітальна форма, форма коменданта для повного керування даними, форма для дирекції студмістечка, окрема форма для деканатів факультетів, а також основна форма попереднього вибору користувача системи. Програмна платформа Visual Studio .NET 2017 дозволяє створення елемента «Форма» одним натисканням миші, і вже автоматично всі поля з базового джерела даних розміщуються на формі.(рис.3.1) Щойно створену форму можна використовувати одразу або змінювати відповідно до потреб застосунку у режимі конструктора.

					КНТЕУ 121 06-03.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.					РЗ	33	50
Керівник	Рзаєва С.Л.					Факультет інформаційних технологій, 4 курс, 6 група		
Гарант	Цензура М. О.							
Розроб.	Давидова А.В.				Розробка інформаційно-аналітичної системи обліку жителів кампусу			

```

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

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 інформаційно-аналітичної системи обліку жителів кампусу 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

					КНТЕУ 121 06-03.БР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

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

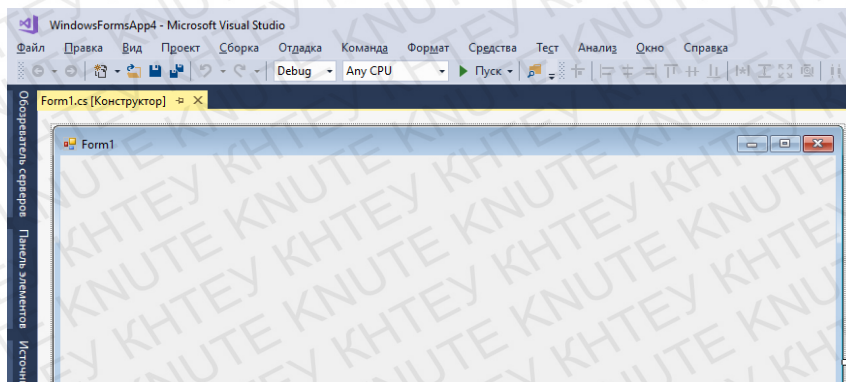


Рис.3.2. Вигляд стандартної Windows Form до редагування та наповнення

Буває форма може без кнопок. Наприклад, для інформаційно-аналітичної системи обліку жителів кампусу було створено вітальну форму, котра має лічильник та тільки іконку привітання. (рис.3.3.)



Рис.3.3. Зображення вітальної форми

Після такого як було виконано зовнішнє оформлення форми, подвійним кліком в будь-якій області форми ми переходимо до середовища, де прописуємо

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-03.БР	35

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(LoadComplete, null, kLoadTime, 0);
}

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

Рис. 3.4. Код вітальної форми

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 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.

					КНТЕУ 121 06-03.БР	Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата		

Таблиця 3.1.

НАДАННЯ ПРАВ КОРИСТУВАЧАМ

Користувач	Право на додавання нового мешканця	Право на виселення мешканця	Редагування даних про жителя	Оцінка кімнати	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

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).

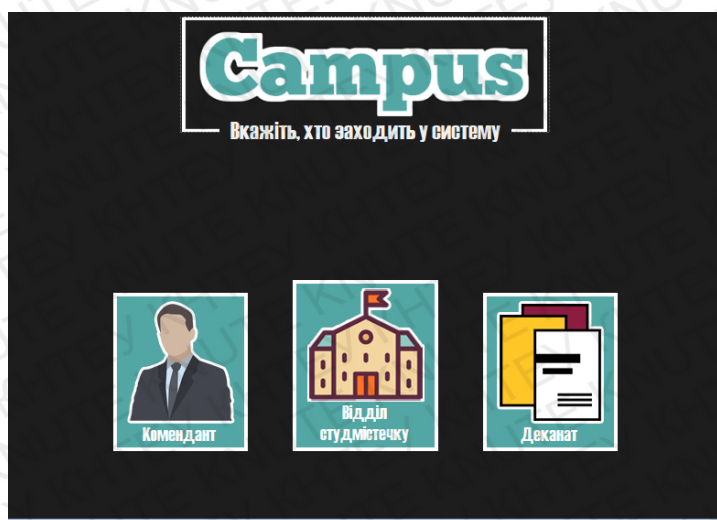


Рис. 3.5. Зображення основної форми вибору користувача

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-03.БР	37

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

Даний код виконує таку функцію: залежно від вибору користувача перед особою з'являється наступна форма з випадаючим списком на запит факультету (для відділу деканату) (рис.3.7) або ж з вибором гуртожитку (для відділу студентського містечка та коменданта) (рис.3.8).

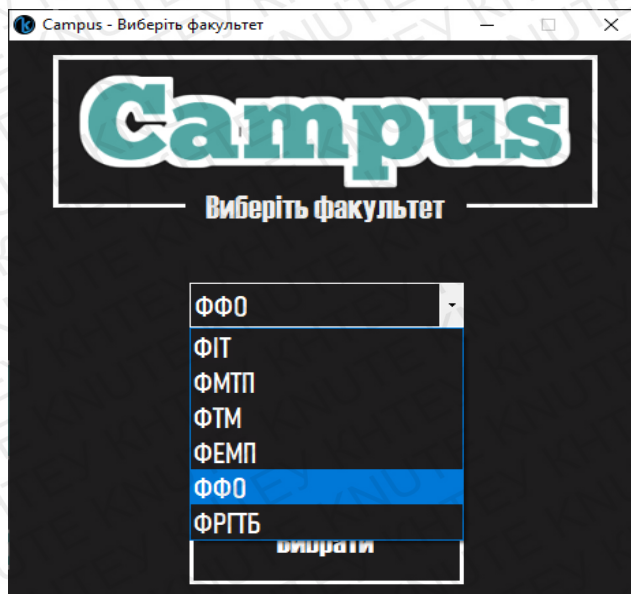


Рис. 3.7. Зображення форми вибору факультету для користувача з відділу деканату факультету

					КНТЕУ 121 06-03.БР	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

Рис.3.8. Форма вибору гуртожитку

Такий розподіл значно полегшує управління даними та є дружельюбним 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.

					КНТЕУ 121 06-03.БР	Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата		

Campus

Інформація про мешканця

Прізвище *

Ім'я *

По-батькові *

Стать *

☒ Чоловіча☐ Жіноча☐ Не визначено

Факультет *

Курс *

Група *

ФІП

Переваги в сусідах

☒ Спортсмен☐ Студент першого курсу☐ Студент другого курсу☐ Студент третього курсу

Особливості

☒ Чутилий сон☐ Можу спати з увімкненим нічникком☐ Потребую тиші☐ Музика може бути увімкнена

Характер *

☒ Тихий та спокійний☐ Нейтральний☐ Галасний та товариський

Місто *

Київ

Адреса *

Телефон *

+38 () - - -

Дата народження *

28 March 2020

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

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

Всі ці критерії додаються разом таким чином, якщо студент обирає у запитанні «Переваги в сусідах» поле «Спортсмен», то програма шукатиме вже

					КНТЕУ 121 06-03.БР	Архив
						40
Зм.	Архив	№ докум	Підпис	Дата		

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", pers_personality_id);
```

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

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_comboBox.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

					КНТЕУ 121 06-03.БР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

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.

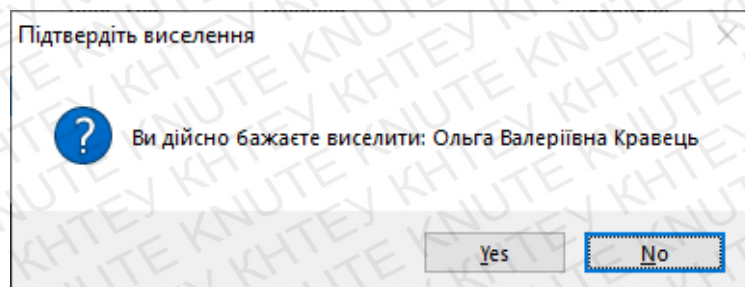


Рис. 3.13. Приклад вікна підтвердження про видалення/виселення студента

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

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, організовувати та аналізувати дані про студентів, котрі проживають на кампусі.

Отже, таким чином за допомогою Visual Studio .Net можна створити повноцінну інформаційно-аналітичну систему обліку жителів кампусу, котра є

					КНТЕУ 121 06-03.БР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

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.

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

Інформаційно-аналітична система обліку жителів кампусу 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 ООП, система повністю захищена від будь-яких дій користувача всередині коду програми чи зміні бази даних поза дозволеними межами.

Інформаційно-аналітична система 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.

Інформаційно-аналітична система обліку жителів кампусу зменшує витрати часу користувачів під час створення звітів, ведення інформації, друк

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						43
Зм.	Аркуш	№ докум	Підпис	Дата		

паперової документації, пошуку потрібних даних, адже вся потрібна інформація по кожному з жителів знаходиться під рукою у програмі з дружельюбним 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.

3.3. Висновки до розділу 3

За допомогою обраних інструментів СУБД SQLite та середовища розробки VisualStudio .Net було створено повноцінну інформаційно-аналітичну систему обліку жителів кампусу 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.

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Отже, автором було опрацьовано теоретичний матеріал з розробки інформаційно-аналітичної системи обліку жителів кампусу, такий як списки документів на поселення студентів у гуртожитки, а також літературу для написання інформаційно-аналітичних систем та створення баз даних, було проведено дослідження даних про інші способи ведення подібної інформації в різних університетах країн, проаналізовано отриману інформацію та на її базі розроблено повноцінний програмний продукт.

Майже кожен комендант гуртожитку має потребу в організації даних без використання безлічі друкованих або написаних від руки списків для аналізу інформації та створення звітів. На думку автора, інформаційно-аналітична система обліку жителів «Campus» є чудовим рішенням цієї проблеми. Адже в ній є можливість створити інформаційний простір для роботи, дозволяє переглядати, редагувати вже існуючі та вносити нові дані про мешканців гуртожитків, тобто повністю надає всі способи задля управління даними, автоматично аналізує та порівнює анкети мешканців і автоматично обирає найкращу для поселення кімнату.

За час виконання випускної кваліфікаційної роботи були виконані такі завдання:

- було розроблено та наповнено Базу Даних для інформаційно-аналітичної системи обліку жителів кампусу;
- було розроблено алгоритму аналізу даних штучним інтелектом системи;

					КНТЕУ 121 06-03.БР		
Зм.	Аркуш	№ докум	Підпис	Дата			
Зав. кафедри	Криворучко О.В.				Розробка аналітичної системи обліку жителів кампусу	Стадія	Аркуш
Керівник	Рзаєва С.П.					ВП	45
Гарант	Цензура М. О.					Факультет інформаційних технологій, 4 курс, 6 група	
Розроб.	Давидова А.В.						
					Висновки та пропозиції		

- було проведено аналіз особливостей схожих систем дізналася про особливості розробки інформаційно-аналітичної системи та використала 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 дружельюбним інтерфейсом та підключення попередньо створеної бази даних.

На початок виконання випускної кваліфікаційної роботи перед автором 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.

За допомогою форми у середовищі розробки Visual Studio .NET було створено зв'язок з таблицями бази даних та налагоджений користувацький інтерфейс, завдяки інструментам бібліотеки .NET були створені елементи коду, що дозволять системі самостійно вирішувати, в яку кімнату краще селити студента, беручи за основу інформаційні дані, наведені в анкеті. База даних

					КНТЕУ 121 06-03.БР	Аркуш
						46
Зм.	Аркуш	№ докум	Підпис	Дата		

функціонує повноцінно через програмний додаток «Campus», тому можна вважати, що поставлена задача була виконана.

Звичайно в майбутньому плануються оновлення програмного додатку та розширення можливостей та функцій для ще більш комфортної організації процесу ведення обліку жителів кампусу через цю інформаційно-аналітичну систему.

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основний

1. R. Miles. C# Programming Yellow Book Ed.8.1. / R. Miles. Minnesota, 2016 – 216 р.
2. Дж. Оппель, Пол Н. Вайнберг. Інструкція по SQL: пер. с англ. / Дж. Оппель, Пол Н. Вайнберг. Киев, 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 – 256р.
5. Агуров П. С#. Разработка компонентов в MS Visual Studio. / П.Агуров. – СПб.: БХВ-петербург, 2008. – 480с.
6. Бен Форма. Опануй самостійно 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. Стилмен. Изучаем С#. / А. Стилменн. Київ, Рід Груп, 2012.-704с.
10. Б. Албахари. С# 6.0. Полное описание языка / Б. Албахари. Дж.Албахари. Вільямс, 2016 – 1040с.
11. Хэндерсон К. Профессиональное руководство по SQL Server. / Хэндерсон К. Київ, 2005 – 560с.
12. М. Грубер. Розуміння SQL. / М. Грубер. Київ, 1999 – 279с.

					КНТЕУ 121 06-03.БР		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Зав. кафедри</i>		Криворучко О.В.			<i>Розробка інформаційно-аналітичної системи обліку жителів кампусу</i>	<i>Стадія</i>	<i>Аркуш</i>
<i>Керівник</i>		Рзаєва С.Л.				<i>СД</i>	<i>Аркушів</i>
<i>Гарант</i>		Цензура М. О.					
<i>Розроб.</i>		Давидова А.В.					
					<i>Список використаних джерел</i>	Факультет інформаційних технологій, 4 курс, 6 група	

13. Хомоненко А. Д., Мальцев М. Г., Цыганков В. М. Базы данных. Учебник для высших. / Хомоненко А. Д., Мальцев М. Г., Цыганков В. Корона-Век, М. 2009 – 734с.
14. Elmasri, Navathe. Fundamentals of Database Systems. / Elmasri. Navathe, 2010 – 1201p.
15. Stephane Faroult. The art of SQL. / Stphane Faroult. Nevada, 2006 – 372p.
16. К. Дж. Дейт. Введение в системы баз данных / Пер. с англ. — 8-е изд. — М.: Вільямс, 2005. — 1328 с.
17. Кевин Клайн. SQL. Справочник. /Кевин Клайн. Кудиц-Образ, 2006. — 832 с.
18. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. — 8-е изд. / Дейт К. Дж. Вильямс, 2005. — 1328 с.
19. Кузнецов С. Д. Основы баз данных. — 2-е изд. / Кузнецов С.Д. Интернет-Университет Информационных Технологий; Лаборатория знаний, 2007. — 484 с.
20. Васвани В. Zend Framework. Разработка веб-приложений на PHP / В. Васвани. Киев, 2012.-432 с.
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>

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		49

2. Середовище Visual Studio .NET [Електронний ресурс]. – режим доступу: http://life-prog.ru/ukr/1_4216_seredovishche-Visual-Studio-NET.html
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. Полубояров В.В. Використання MS SQL Server 2008 Analysis Services для побудови сховищ даних. [Електронний ресурс]. – режим доступу: <http://player.myshared.ru/7821M>

					<i>КНТЕУ 121 06-03.БР</i>	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

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.
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,

```