

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ
ІНФОРМАЦІЇ ПРО СТУДЕНТІВ ЗАКЛАДУ ВИЩОЇ ОСВІТИ
SOFTWARE IMPLEMENTATION OF THE SYSTEM FOR MONITORING
INFORMATION ABOUT THE STUDENTS OF A HIGHER EDUCATION
INSTITUTION

Виконав: студент(ка) 4 курсу, групи КНТ-118сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Жидкова К.К.

(прізвище та ініціали)

Керівник Олійник А.О..

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

Жидковій Катерині Костянтинівні

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Програмна реалізація системи моніторингу інформації про студентів закладу вищої освіти

Software Implementation of the System for Monitoring Information about the Students of a Higher Education Institution

керівник проекту (роботи) Олійник Андрій Олександрович, д.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проекту (роботи) “5” травня 2021 року

3. Вихідні дані до проекту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Олійник А.О., доцент		
Нормоконтролер	Андрєєв М.О., асистент		

7. Дата видачі завдання “ 12 ” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент(ка)


(підпис)

Жидкова К.К.
(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Олійник А.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 88 с., 32 рис., 3 табл., 6 дод., 16 джерел.

ЗАКЛАДИ ВИЩОЇ ОСВІТИ, МОНІТОРИНГ, СТУДЕНТИ, ПРОГРАМНА РЕАЛІЗАЦІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОЕКТУВАННЯ.

Об'єкт дослідження – процес моніторингу інформації про студентів.

Предмет дослідження – методи моніторингу інформації про студентів.

Метою роботи є програмна реалізація системи моніторингу інформації про студентів.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C#, середовище розробки Microsoft Visual Studio, система керування базою даних MySQL.

Результати. Розроблено програмне забезпечення для моніторингу інформації про студентів.

Наукова новизна роботи полягає в розробці системи для моніторингу інформації про студентів, щоб спростила організаційні процеси у закладах вищої освіти.

Висновки. Розроблено програмне забезпечення для моніторингу інформації про студентів.

Галузь використання – заклади вищої освіти та їх структурні підрозділи.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 88 pages, 32 figures, 3 tables, 6 appendixes, 16 sources.

HIGHER EDUCATION INSTITUTIONS, MONITORING, STUDENTS, SOFTWARE IMPLEMENTATION, SOFTWARE, DESIGN.

The object of research is the process of monitoring information about students.

The subject of the research is methods of monitoring information about students.

The aim of the work is to programmatically implement a system for monitoring information about students.

Materials, methods, and technical tools: Object-Oriented Programming, C# programming language, Microsoft Visual Studio development environment, MySQL database management system.

Results. Software for monitoring student information has been developed.

The scientific novelty of the work is to develop a system for monitoring information about students in order to simplify organizational processes in higher education institutions.

Conclusions. Software for monitoring student information has been developed.

Field of usage: higher education institutions and their structural divisions.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Мета і задачі розробки автоматизованої системи обліку та аналізу даних про студентів закладу вищої освіти.....	11
1.2 Системи автоматизованого обліку студентів ВНЗ.....	12
1.2.1 Єдина державна електронна база з питань освіти	12
1.2.2 Система «Парус-Навчальний заклад».....	14
1.2.3 Система «Тауруна»	16
1.2.4 Пакет програм «Деканат» для вищих навчальних закладів III-IV рівня акредитації	18
1.2.5 Порівняння існуючих аналогів	18
1.3 Постановка завдання	20
1.3.1 Функціональність системи.....	20
1.3.2 Вимоги до інтерфейсу.....	21
1.3.3 Не функціональні вимоги до продукту.....	21
1.4 Висновки за розділом 1	22
2 Розробка архітектури програми	23
2.1 Вибір середовища проектування.....	23
2.1.1 Вибір мови програмування	23
2.1.2 Вибір середовища розробки.....	26
2.2 Загальна структура додатку.....	28
2.3 Проектування інтерфейсу	29
2.4 Проектування функціональної частини	30
2.5 Висновки за розділом 2	31
3 Основні рішення щодо реалізації компонентів системи	32
3.1 Проектування системи на основі предметної області	32
3.2 Обмін даними між системою та Excel-файлом	34

3.3 Функціонування системи	36
3.4 Експорт даних у документи Word.....	38
3.5 Висновки за розділом 3	40
4 Експлуатація, тестування та експериментальне дослідження програми	41
4.1 Опис застосування	41
4.2 Умови виконання програми.....	41
4.3 Інструкція по експлуатації програми	42
4.3.1 Звернення до програми.....	42
4.3.2 Вхідні й вихідні дані	42
4.4 Методика та результати тестування	42
4.4.1 Сценарій тестування	42
4.4.2 Хід виконання тестування.....	43
Висновки	47
Перелік джерел посилання	49
Додаток А Технічне завдання	51
Додаток Б Опис програми	55
Додаток В Керівництво програміста	59
Додаток Г Керівництво оператора	62
Додаток Д Текст програми	69
Додаток Е Слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

GUI – Graphical user interface;

MVC – Model-view-controller;

UML – Unified Modeling Language;

ЗВО – заклад вищої освіти;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СМІС – система моніторингу інформації про студентів.

ВСТУП

Комп'ютеризовані системи обробки даних вже досить довгий час допомагають виконувати безліч видів робіт у різноманітних сферах управління та виробництва. У будь-якому навчальному закладі виникає потреба створення системи для управління даними, що дозволила б збільшити ефективність роботи працівників кадрової служби та архівного підрозділу.

Основними факторами, що можуть зашкодити застосуванню або розробці різного спеціалізованого програмного забезпечення, як правило, є такі:

- специфічні особливості процесу обліку у даній організації, що є відмінними від стандартних та створюють складнощі при пошуку відповідного програмного забезпечення;
- складність та досить серйозна вартість розробки власного програмного продукту для обліку;
- проблематичність переведення інформації з паперових на цифрові носії, додавання до бази існуючих даних.

Тобто, актуальною є тематика розробки спеціалізованої системи для обліку та аналізу інформації про студентів навчального закладу із урахуванням конкретних вимог установи.

Таким чином, у даній випускній роботі розроблено програмні засоби, що дозволяють автоматизувати аналіз та обробку даних про студентів навчального закладу, створення інформаційно-довідкової та звітної документації на основі цих даних.

Основні задачі випускної роботи:

- проаналізувати існуючі програмні рішення для обробки даних про студентів, учнів навчальних закладів;
- виконати вибір середовища для створення програмного засобу;

- визначити основні вимоги до програмного засобу, які зумовлені необхідністю відповідних підрозділів навчального закладу;
- розробити функціональну та візуальну структуру програмного засобу;
- скласти керівництво програміста та навести порядок роботи з програмним засобом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Мета і задачі розробки автоматизованої системи обліку та аналізу даних про студентів закладу вищої освіти

З метою підвищення ефективності роботи архівної служби закладів вищої освіти (ЗВО) автоматизована система повинна вирішувати наступні задачі [1]-[4]:

- автоматизація роботи відділу кадрів та архіву ЗВО на етапі зарахування, переведення, відрахування студентів;
- пошук студентів та випускників за різними критеріями;
- виведення інформації про окремого студента;
- додавання інформації про студента;
- формування на основі інформації з програми документа – архівної довідки, довідки з місця навчання, службового листа із інформацією про випускника із використанням шаблону – бланку ЗВО.

Основна мета розробки даної системи – отримання можливості автоматизації роботи з архівними списками студентів коледжу, які мають паперовий вигляд. Такі списки складені за 1974 – 2011 роки. З 2011 року була введена система для систематизації даних про студентів усіх вищих навчальних закладів – Єдина державна база освіти (ЄДЕБО), а інформація про студентів, що навчалися у попередні роки, наявна тільки у паперовому вигляді, причому стан фізичного зносу більш старих книг обліку погіршується з кожним роком [1]-[4].

Автоматизована система обліку студентів повинна спростити роботу працівників відділу кадрів при складанні документації, а саме підвищити ефективність пошуку, перенесення інформації з електронних таблиць у документ, створений на підставі шаблону. Таким чином, функціонально програмний засіб дозволяє поєднати електронну таблицю у форматі .xls із документом .doc із використанням мови програмування C#.

1.2 Системи автоматизованого обліку студентів ВНЗ

В останні роки набули значного розповсюдження системи, що дозволяють автоматизувати облік як працівників підприємств, установ, організацій, так і учнів, студентів освітніх установ. Причому ці процеси стали активізуватись не тільки у приватних підприємствах, які більш швидко застосовують можливі технічні інновації, а і у державних установах. Хоча працівники, що мають справу з програмним забезпеченням у державних установах, обмежені вибором пропозицій тих розробників, які допущені до участі у тендері, або є державними. В останньому разі можна навіть казати про монополію державних програмних засобів у деяких випадках [1]-[4].

1.2.1 Єдина державна електронна база з питань освіти

Даний програмний комплекс розроблено з метою накопичення, оброблення, зберігання та захисту даних, у тому числі персональних, щодо закладів здійснюючих освітні послуги в Україні [5].

Дані з цієї бази використовуються під час виготовлення:

- документів про освіту державного зразка;
- документів про вчені звання та наукові ступені;
- ліцензій на надання освітніх послуг та сертифікатів про акредитацію;
- учнівських (студентських) квитків;
- для отримання інших відомостей інформаційного характеру та обробки статистичних даних в інтересах навчальних закладів, установ освіти для здійснення контролю та прийняття управлінських рішень.

Система має веб-інтерфейс та потребує захищеного з'єднання за допомогою спеціального ключа, що зберігається на змінному носії пам'яті. Відповідно до чинного законодавства України (Закон України «Про захист персональних даних» від 1 червня 2010 р.) для включення відомостей про

людину в електронну базу необхідно отримати її згоду. У навчальному закладі повинен бути забезпечений адміністративно-організаційний механізм збору та зберігання документів - персональні згоди на обробку відповідних даних в ЄДЕБО [5].

Вона спрямована на спрощення роботи навчальних закладів у наступних напрямках:

- верифікація даних щодо документації вступників (неможливо внести одну і ту ж особу до бази при вступі на бюджетну форму);
- створення єдиного рейтингу вступників;
- автоматизація замовлення навчальної документації;
- автоматизація складання різних типів звітів навчальних закладів;
- можливість виконання вибірки студентів за різними критеріями, експорту даних до табличних файлів.

Додання вступника

ПІБ * прізвище: Іванов, ім'я: Іван, по-батькові: Іванович

дата народження: 01.02.1990, стать: Чоловіча, ☒ Громадянин України

Адреса * ІВАНО-ФРАНКІВСЬКА ОБЛАСТЬ, ІВАНО-ФРАНКІВСЬК

Крайна: Україна, тип: вулиця, вулиця: Шевченка, будівля: 51, кв. 4

Поштовий індекс: 76018

Метричні дані(*) ☒ Паспорт ☐ Свідчення про народження ☐ Посвідка

серія: СТ, номер: 562363, дата видачі: 20.03.2006

Місце народження: , ІПН:

Медична довідка номер: , дата видачі:

Тип документа * Аттестат про повну загальну середню освіту

серія: АК, номер: 23658945, дата видачі: 01.06.2008

Документ * бал для вступу: 9.2, ☐ Некоректний документ, ☐ Іноземний взірць

нагороди: ☒ вивчалися іноземні мови: англійська

Контактна інформація телефон(*): 0935623665, мобільний(*): , електронна пошта:

Відомості про родичів батько (1-й опікун): , мати (2-й опікун):

контактний телефон: , контактний телефон:

☐ Дана згода на внесення персональних даних

* - обов'язкові дані, (*) - альтернативні дані

Створити Відмінити

Рисунок 1.1 – Зображення форми нового вступника

На рис. 1.1 наведено приклад форми, яку заповнює приймальна комісія про кожного абітурієнта, що подав документи для вступу до навчального закладу.

На рис. 1.2 наведено приклад звіту, який може бути сформований у системі на початок навчального року, подається до органів статистики.

І. Розподіл студентів за спеціальностями (напрямами підготовки) на 1 жовтня													2012	
Назва спеціальностей (напрямів підготовки)	Коди спеціальностей (напрямів підготовки) (№ рядка)	Рівні акредитації або підготовки	Прийнято				Із загальної кількості прийнятих (гр. 2-6) прийнято на початковий цикл навчання				Усього навчається	У тому числі жінок		
			усього (сума гр. 3-6)	у тому числі за рахунок коштів:			усього (сума гр. 8-11)	у тому числі за рахунок коштів:						
				державного бюджету	місцевих бюджетів	органів державної влади, юридичних осіб		фізичних осіб	державного бюджету	місцевих бюджетів			органів державної влади, юридичних осіб	фізичних осіб
A	Б	1	2	3	4	5	6	7	8	9	10	11	12	13
Бакалавр														
правознавство	6.030401	0	0	0	0	0	0	0	0	0	0	0	0	0
маркетинг	6.030507	4	85	30	0	15	40	85	30	0	15	40	245	128

Рисунок 1.2 – Приклад статистичного звіту

Також необхідно відмітити, що використання системи є платним, а зміни в установчих документах закладу викликають необхідність переоформлення ліцензій та ключа, що виконується тільки у установах, що знаходяться у м. Києві. Крім того, система має недоліки, наприклад, постійні зависання у роботі, довгий час очікування з'єднання із сервером. Також недоліком є те, що її впровадження було почато з вищих навчальних закладів І-ІІ рівня акредитації. Хоча логічніше було б почати зі шкіл, щоб дані про учнів у майбутньому не доводилося вносити повторно, як про абітурієнтів [5].

1.2.2 Система «Парус-Навчальний заклад»

Корпорація «Парус», вітчизняний розробник програмного забезпечення, теж пропонує свою власну систему для комплексного управління та аналізу діяльності навчального закладу, контролю фінансово-економічних процесів [6]. Дана система має наступні функціональні можливості:

- облік студентів різних форм навчання;
- зарахування студентів;

- збереження анкетних даних студентів;
- обробка наказів про зарахування, переведення, відрахування студентів;
- формування та роздрукування звітів;
- аналіз кількісних та якісних показників навчання студентів.

Крім того, система дозволяє виконувати також функції контролю, наприклад [6]:

- моніторинг відвідування студентами занять;
- організація пропускнуго режиму на території навчального закладу;
- автоматична реєстрація переміщення студентів;
- організація доступу в інші приміщення навчального закладу (бібліотека, гуртожиток);
- керування доступом до платних послуг;
- супроводження пропускнуго системи.

На рисунку 1.3 наведено приклад вікна програми.

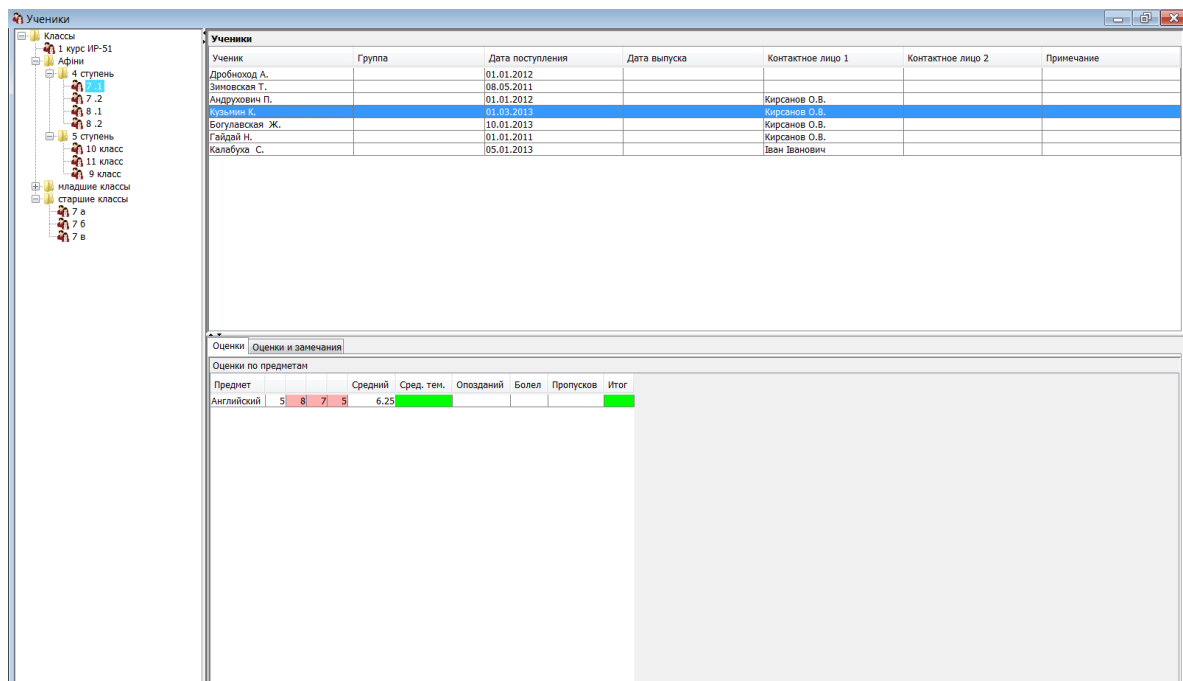


Рисунок 1.3 – Вікно «Учні» системи «Парус-Учебное заведение»

Таким чином, дана система більше спрямована на реалізацію функцій управління та контролю за діяльністю навчального закладу та буде доречною у роботі працівників, які відповідають за виховні функції та безпеку навчального закладу [6].

1.2.3 Система «Тауруна»

Цей розробник програмного забезпечення пропонує дві окремі системи, які призначені відповідно для обліку вступників та студентів навчального закладу [7].

Система «Абітурієнт» призначена покращити якість роботи приймальної комісії: фіксація даних студента без помилок, перегляд статистики, переліку документів, що подають вступники. Також дана система дозволяє формувати екзаменаційні групи, планувати графік іспитів, вести облік робочого часу екзаменаторів. В системі реалізовано заповнення картки абітурієнта, внесення до неї великої кількості особової інформації. Це захищає систему від помилок, які може допустити сам вступник при заповненні анкет та заяв вручну. Система дозволяє виконати перевірку результатів тестування – сертифікат про кількість балів може бути перевірений на достовірність за допомогою звірки із державною базою.

Автоматичне роздрукування розписки про отримані документи від приймальної комісії.

Можливість експорту звітів у програми MS Excel и Adobe PDF. Ця статистика може розміщуватись як на сайті закладу, так і на дошці оголошень, тому що не потребує додаткової обробки і форматування.

Також програма надає можливість підготовки договорів про контрактне навчання.

Дані з системи «Абітурієнт» легко переносяться у систему «Студент», у цьому випадку деканат отримує необхідну інформацію про вступників і звільнюється від необхідності створення такої бази з нуля [7].

Система «Студент» дозволяє вести облік учнів, студентів, враховуючи моніторинг успішності, облік динаміки, результати семестрових сесій. Ці дані слугують джерелом для заповнення відповідних звітів, зведених відомостей за підсумками року, а також формування додатків до дипломів.

З цією системою можуть працювати не тільки працівники деканатів, але і самі викладачі.

Основні функціональні можливості та переваги системи: облік динаміки руху студентів (переведення, відпустки, відрахування). У базі відображена тільки актуальна інформація щодо статусу конкретного студента.

Пошук та відбір даних про студента за різними параметрами, що значно спрощує облік та звітність навчального закладу, зокрема, подачу звітів до установ освіти та військкоматів. Єдина база успішності заповнюється викладачами, що дозволяє бачити актуальну та достовірну картину. На підставі цих відомостей формується зведена відомість групи, курсу, факультету, що дозволяє уникнути дублювання інформації та зайвої роботи. Як наслідок, спрощується робота під час підготовки до сесії: допуск студентів, інформування про наявність заборгованостей [7].

Також спрощено оформлення додатків до дипломів: підсумкові оцінки формуються автоматично на підставі зведених відомостей.

Документообіг значно спрощено завдяки спеціальним формам, що оптимізовані для роздрукування. Крім того, кожен навчальний заклад має можливість самостійно редагувати, змінювати зовнішній вигляд цих форм та зберігати архів документації.

Також у системі є можливість ведення обліку студентів заочної та вечірньої форми навчання, різних видів курсів.

Передбачена можливість роботи системи одночасно на декількох робочих станціях.

1.2.4 Пакет програм «Деканат» для вищих навчальних закладів

Даний програмний пакет призначений для використання у вищих навчальних закладах III-IV рівня акредитації та дозволяє побудувати систему із наступними можливостями [8]:

- внесення структури навчального закладу;
- формування даних про навантаження викладачів, що дозволяє виконати підготовку відповідної звітної документації;
- розклад роботи викладачів;
- збереження інформації про студентів;
- облік навчальної діяльності студентів (успішність, відвідування занять, переведення, відрахування, випуск);
- використання аудиторій, розклад.

Пакет складається з цілої низки окремих програм, призначених для виконання конкретних функцій управління навчальним процесом. Програми працюють із використанням веб-інтерфейсу, підтримується взаємодія сервер-клієнт, тому немає необхідності роботи за закріпленим робочим місцем. Спеціаліст може виконувати роботу з системою, знаходячись за ПК, що під'єднаний до мережі Інтернет. Недоліком такої можливості є можливі небезпеки для передачі даних мережею Інтернет.

1.2.5 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). У таблиці будуть порівнюватися попередні аналоги із системою, що розробляється – система моніторингу інформації про студентів (СМІС). Серед критеріїв виділимо наступні:

- доступ до інформації про здобувачів освіти – чи надає програмне забезпечення (ПЗ) доступ до інформації про здобувачів вищої освіти;

- періодичність оновлення інформації – Як часто поновлюється інформація про здобувачів;
- складність розгортання та підтримки – важливий критерій при оцінці коштовності встановлення та подальшої підтримки системи в ЗВО;
- збереження архівної інформації – чи зберігає система архівну інформацію, наприклад про випускників;
- платний функціонал – чи наявний платний функціонал в системі.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	ЄДЕБО	Парус	Тауруна	Деканат	СМІС
Доступ до інформації про здобувачів освіти	Частково	Надає	Надає	Надає	Надає
Періодичність оновлення інформації	Щорічно	За потреби	За потреби	За потреби	За потреби
Складність розгортання та підтримки	Віддалена база	Середня	Висока	Висока	Низька
Збереження архівної інформації	Частково	Не зберігає	Не зберігає	Не зберігає	Зберігає
Платний функціонал	Немає	Платний сервіс	Платний сервіс	Платний сервіс	Безкоштовна

На підставі аналізу можна зробити висновок, що розробка подібних систем є досить актуальним питанням, особливо при врахуванні специфіки конкретного навчального закладу. Крім того, необхідно додати, що практично усі системи призначені для обліку даних щодо студентів, що навчаються зараз у даному навчальному закладі. Але про необхідність збереження інформації про випускників теж не можна забувати з декількох причин:

- необхідність створення звітної та статистичної документації, що порівнює успішність, кількість учнів за декілька років;
- обробка заяв громадян, що були студентами навчального закладу, необхідність складання архівних довідок на підставі цих заяв;
- обробка запитів державних установ, які виконують перевірку осіб, що приймаються на роботу, виконуючи відправку відповідних запитів до вищих навчальних закладів.

1.3 Постановка завдання

Таким чином, відділ кадрів або архівний підрозділ навчального закладу потребує системи, яка дозволить виконувати пошук та вибірку інформацію як про існуючих, так і про колишніх студентів. Зокрема, це актуально і з іншої причини: найчастіше дані про колишніх студентів, особливо, якщо навчальний заклад існує 20 і більше років, зберігаються на паперових носіях, фізичний стан яких з часом погіршується. Створення відповідних автоматизованих систем для збереження таких даних є можливістю для ведення історії закладу та створення сприятливих умов для обробки запитів громадян та державних установ.

1.3.1 Функціональність системи

Виходячи з проведеного аналізу існуючих засобів було виявлено основні можливості, які повинна надавати система, яка забезпечує збереження, обробку і аналіз даних про студентів навчального закладу:

- можливість додавання інформації про студентів, як існуючих, так і відрахованих з навчального закладу з різних причин;
- реалізація пошуку студента за різними критеріями: дата народження, прізвище, спеціальність, та ін.;
- проведення статистичного аналізу на підставі існуючих даних;

- автоматизоване створення документів на бланках навчального закладу, що забезпечується імпортом з табличного процесора до шаблону відповідного документа: листа, довідки, звіту.

1.3.2 Вимоги до інтерфейсу

Для забезпечення ергономічності, зручності та зрозумілості інтерфейс розроблюваного ПЗ повинен:

- бути зручним та інтуїтивно-зрозумілим;
- при першому користуванні продуктом користувач отримує підказки;
- пароль відображається зірочками.

1.3.3 Не функціональні вимоги до продукту

Для нормального функціонування ПЗ необхідно виконання технічних умов.

Вимоги до складу серверної частини:

- сервер, що працює на базі однієї з лінійки операційних систем (ОС) Windows Server;
- вільне місце на накопичувачі – від 5 Гб;
- оперативна пам'ять стандарту DDR4 об'ємом від 16 Гб;
- мережева карта.

Вимоги до клієнтської частини:

- IBM-сумісний персональний комп'ютер (ПК) під управлінням ОС з лінійки Windows не старше 7 версії;
- вільне місце на накопичувачі – від 1 Гб;
- оперативна пам'ять стандарту DDR3 об'ємом від 4 Гб;
- мережева карта.

1.4 Висновки за розділом 1

Даний розділ містить у собі аналіз існуючих систем, які дозволяють виконувати облік та аналіз інформації про студентів та учнів навчальних закладів, полегшують відбір даних, а також роздрукування найчастіше використовуваних документів.

Таким чином, на підставі проведеного аналізу засобів, методів та технологій ведення обліку студентів навчальних закладів, було виділено основні функціональні можливості, які повинна мати система обліку студентів.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

У попередньому розділі було проведено аналіз засобів та систем, що дозволяють оперувати списками студентів навчального закладу, розглянуто основні можливості таких засобів. Виходячи з проведеного аналізу, було сформульовано основні можливості, якими повинна оперувати система, що проектується.

Що стосується технічних та програмних можливостей, то система повинна забезпечувати імпорт даних про студентів з таблиць у форматі *.xls, виконувати їх обробку, та передавати відповідні дані у документ *.doc для формування відповідних документів стосовно випускників та студентів навчального закладу. Система для такої взаємодії розроблена із використанням мови програмування C#, базуючись на вбудованих класах мови, що дозволяють працювати з офісними додатками – Microsoft.Office.Interop.Excel, Microsoft.Office.Interop.Word.

Даний розділ містить опис питань, які стосуються безпосередньо структури системи, що розробляється, вибору середовища проектування, а також визначаються основні та специфічні архітектурні рішення, які має система.

2.1 Вибір середовища проектування

2.1.1 Вибір мови програмування

C# (вимовляється Сі-шарп) – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. C# – дана мова програмування є об'єктно-орієнтованою мовою та дозволяє максимально зекономити час при розробці програмного продукту. Швидкодія програм написаних на C# досить висока. Перевагою даної мови є те, що розмір результуючих програмних продуктів є дуже малий і не потребує додаткових бібліотек. Для виконання програм написаних на мові C#

необхідно щоб на комп'ютері було інстальоване Framework 3.5. Також Перевагою C# програм є виконання на різних платформах, для яких розроблений Net Framework. Програма написана на C# не залежить ні від апаратури комп'ютера, ні від платформи [9].

Коли говорять C#, нерідко мають на увазі технології платформи. net (Windows Forms, WPF, ASP.NET, Xamarin). І, навпаки, коли говорять. Net, нерідко мають на увазі C#. Однак, хоча ці поняття пов'язані, ототожнювати їх невірно. Мова C# була створена спеціально для роботи з фреймворком. Net, проте саме поняття. Net дещо ширше [10].

Підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд з C# це також VB.NET, C++, F#, а також різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi.NET. При компіляції код на будь-якому з цих мов компілюється в збірку на загальному мовою CIL (Common Intermediate Language) – свого роду асемблер платформи .NET. Тому за певних умов ми можемо зробити окремі модулі одного додатка на окремих мовах [9], [10].

Кроссплатформність. Net є портативною платформою (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент – .Net 5 підтримується на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі. Net, можна розробляти додатки на мові C # для самих різних платформ-Windows, MacOS, Linux, Android, iOS, Tizen [9], [10].

Потужна бібліотека класів. Net представляє єдину для всіх підтримуваних мов бібліотеку класів. І яке б додаток ми не збиралися писати на C# - текстовий редактор, чат або складний веб-сайт - так чи інакше ми задіємо бібліотеку класів .NET [9], [10].

Різноманітність технологій. Загальномовне середовище виконання CLR і Базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків. Наприклад,

для роботи з базами даних в цьому стеку технологій призначена технологія ADO.NET і Entity Framework Core. Для побудови графічних додатків з багатим насиченим інтерфейсом-технологія WPF і UWP, для створення більш простих графічних додатків-Windows Forms. Для розробки мобільних додатків-Xamarin. Для створення веб-сайтів і веб-застосунків - ASP.NET і т. д. [9], [10].

До цього варто додати активній розвивається і набирає популярність Blazor-фреймворк, який працює поверх.NET і який дозволяє створювати веб-застосунки як на стороні сервера, так і на стороні клієнта. А в майбутньому буде підтримувати створення мобільних додатків і, можливо, десктоп-застосунків [9], [10].

Продуктивність. Згідно ряду тестів веб-застосунки на .NET 5 в ряді категорій сильно випереджають веб-застосунки, побудовані за допомогою інших технологій. Додатки на .NET 5 в принципі відрізняються високою продуктивністю.

Також ще слід відзначити таку особливість мови C# і фреймворку .NET, як автоматична збірка сміття. А це означає, що нам в більшості випадків не доведеться, на відміну від C++, піклуватися про звільнення пам'яті. Вищезгадана загальномовна середовище CLR сама викличе збирання сміття і очистить пам'ять [9], [10].

Проведемо порівняння мов програмування для реалізації мобільного застосунку, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні:

- наявність літератури та простота вивчення – сукупний критерій для визначення спільноти розробників та популярності мови програмування;
- стандартні бібліотеки шаблонів – наявність таких бібліотек може спростити розробку;
- кроссплатформність – чи може розроблене ПЗ бути кроссплатформним;

- мультипарадигмальність – використання декількох парадигм програмування може спростити розробку;
- масштабованість – чи легко можуть бути масштабовані розроблені застосунки;
- паралельне програмування – чи підтримує мова програмування паралельність виконання;
- простота розроблення графічного інтерфейсу користувача (GUI)
- наскільки просте розроблення GUI при роботі із мовою програмування.

Таблиця 2.1 – Порівняння мов програмування

	C#	C++	Java
Наявність літератури та простота вивчення	++	++	++
Стандартні бібліотеки шаблонів			
Кроссплатформність	±		++
Мультипарадигмальність	++	++	±
Масштабованість			
Паралельне програмування	++	++	
Простота розроблення GUI	++	±	

За результатами порівняння можна зробити висновок, що C# залишається кращою альтернативою, адже має і розширену спільноту, і набір розширень, що спрощують розробку. Для написання нашого застосунку за обраною темою розробки цілком достатньо вибрати мову C#.

2.1.2 Вибір середовища розробки

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки ПЗ та ряд інших інструментальних засобів [11]. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що

підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Варто зазначити, що з 2015 року Visual Studio стала відкритим середовищем розробки, завантажити яке може кожен охочий з офіційного сайту. Обов'язковою є тільки реєстрація на сайті для вільного користування.

Також суттєвою зміною стала підтримка багатьох цільових платформ: окрім базової Windows з'явилась можливість будувати проекти для IOS та Android. Для розробників комп'ютерних ігор була додана підтримка фреймворку Unity. Був оновлений механізм автентифікації: користувач під час запуску Visual Studio синхронізується з єдиним аккаунтом Microsoft [11].

Перший нестабільний випуск наступної версії – Visual Studio 2017, під умовною назвою «15» побачив світ 30 березня 2016 року. Основними змінами стали інтерфейс інсталятора та численні незначні покращення у різних компонентах середовища розробки. Очікується підтримка мови програмування Solidity [11].

Порівняльна характеристика найпоширеніших середовищ розробки для мови C# наведена у таблиці 2.2. Серед характеристик для порівняння виділімо:

- кроссплатформність – чи є можливість використовувати середовище розробки на різних ОС;
- швидкість модернізації – як швидко оновлюється середовище в контексті нових стандартів мов програмування;
- наявність генератора сценаріїв складання – чи підтримує середовище генератор сценаріїв складання;
- підтримка репозиторіїв – чи є вбудований механізм роботи із репозиторіями.

За результатами порівняння перевагу було надано саме Microsoft Visual Studio.

Таблиця 2.2 – Порівняння середовищ розробки

	Microsoft Visual Studio	Code::Blocks
Кроссплатформність		
Швидкість модернізації		±
Наявність генератора сценаріїв складання		±
Підтримка репозиторіїв		±

2.2 Загальна структура додатку

Для побудови архітектурного каркасу ПЗ буде використано шаблон проектування Model-view-controller (MVC). Ця схема використання декількох шаблонів проектування, за допомогою яких модель даних програми, користувацький інтерфейс і взаємодія з користувачем розділені на три окремих компоненти [12].

View – відповідає за збір даних від користувачів в різних формах та обробку їх у відповідності зі специфікацією.

Controller – обробляє дані, отримані від користувача для передачі на наступний рівень.

Model – сукупність правил, принципів, залежностей поведінки об'єктів предметної області. Це реалізація предметної області в інформаційній системі.

На рисунку 2.1 показано, який вигляд має архітектура ПЗ на основі предметної області з використанням шаблону проектування MVC.

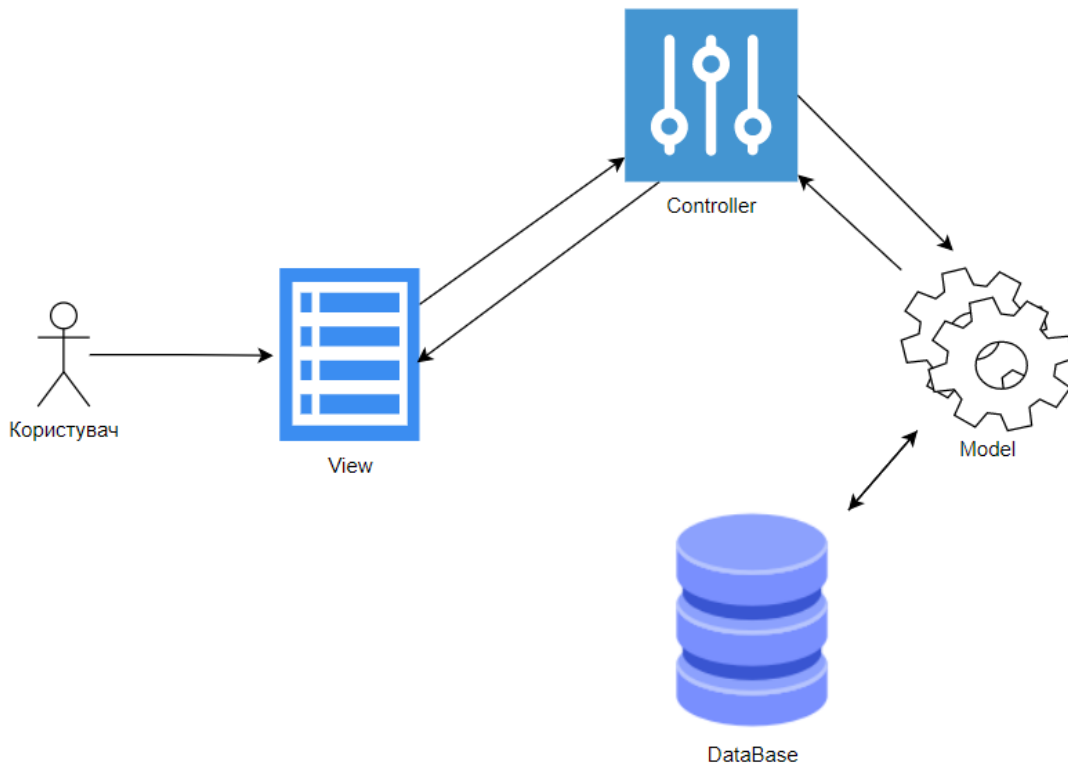


Рисунок 2.1 – Архітектура системи «СМІС» на основі предметної області

2.3 Проектування інтерфейсу

Блок візуалізації призначений для взаємодії користувача з системою. Основні складові інтерфейсу програмного засобу наступні [12]:

- вкладка «Пошук», що містить кнопку «Імпорт файлу» та дозволяє завантажити до програми табличний файл; таблицю, куди буде розміщено дані з файлу *.xls; текстовий рядок, який дозволяє виконувати пошук відповідних значень у таблиці;
- вкладка «Друк документів», яка призначена для виведення на друк інформації з таблиць, а саме: створення документів – архівної довідки, довідки про навчання, а також листа із підтвердженням інформації про студента;
- вкладка «Статистика» дозволяє отримати деяку статистичну інформацію щодо випускників: загальну кількість випускників у базі,

кількість виданих дипломів з відзнакою, кількість випускників певної спеціальності, кількість випускників за роками.

2.4 Проектування функціональної частини

Автоматизована система обробки даних виконує наступні функції:

- імпорт файлу *.xls до програми;
- виведення інформації з таблиці на відповідний елемент форми;
- пошук студента за різними критеріями;
- виведення інформації на друк;
- підрахунок різних статистичних показників.

Імпорт даних з програми виконується за допомогою механізму, що забезпечується класом `Microsoft.Interop.Office.Excel`. Об'єктна модель додатка Excel надає доступ до різних типів, які можна використовувати для автоматизації роботи із файлами, створеними у програмі Microsoft Office Excel. Наприклад, є можливість програмно формувати аркуші книги, створювати діаграми, аркуші, отримувати та змінювати значення діапазонів та окремих комірок. Крім того, у цій моделі доступні і можливості програми із виконання розрахунків. Для розробки рішень, які використовує Microsoft Office Excel, необхідна взаємодія із об'єктами, що надаються об'єктною моделлю Excel. Найважливішими об'єктами моделі є наступні:

- `Microsoft.Office.Interop.Excel.Application`;
- `Microsoft.Office.Interop.Excel.Workbook`;
- `Microsoft.Office.Interop.Excel.Worksheet`;
- `Microsoft.Office.Interop.Excel.Range`.

Аналогічна об'єктна модель використовується і для експорту отриманих даних у файли Microsoft Word, яка надає наступні можливості:

- створення документа із використанням шаблону;
- відкриття закриття документа;
- збереження документа під поточним або новим іменем;

- роздрукування документа;
- попередній перегляд документа;
- використання функцій вбудованих діалогових вікон без їх відображення;
- додавання різних об'єктів;
- перевірка орфографії;
- додавання колонтитулів до документа;
- видалення приміток.

2.5 Висновки за розділом 2

Таким чином, архітектура програмного засобу дозволяє вирішувати задачі, що були поставлені перед системою, пов'язані із переглядом, пошуком даних про студентів, а також виведенням інформації на друк. Засоби, що використовуються, забезпечують реалізацію функцій програми.

Для програмної реалізації можливостей програмного засобу, що реалізує обробку та аналіз даних про студентів навчального закладу, виділених під час аналізу у попередньому розділі, проведено вибір засобів та обґрунтовано структуру системи, що проектується.

Аналізуючи обрану структуру програмного засобу, можна зробити висновок, що вона є лаконічною, але, разом з тим, досить працездатною, використовує ресурси ПК раціонально. Також програмний засіб є нескладним у використанні, обране середовище програмування дозволяє у майбутньому розширити та доповнити можливості та інтерфейс системи.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

В даному розділі розглядаються основні рішення, що пов'язані із реалізацією компонентів системи. Для цього аналізується взаємодія програмного засобу з додатками з Excel, виконання пошуку та аналізу даних, експорт інформації у Word.

Виходячи з аналізу можливих варіантів взаємодії, розроблено алгоритм функціонування програмного засобу, а також реалізовані необхідні функції системи із урахуванням особливостей роботи з даними із додатків Microsoft.

3.1 Проектування системи на основі предметної області

На попередньому етапі було сформовано архітектуру системи на основі шаблону розробки MVC. Тепер розглянемо основні функціональні вимоги до системи у вигляді діаграми прецедентів, що продемонстрована на рисунку 3.1 [13], [14].



Рисунок 3.1 – Діаграма прецедентів

Виходячи із попереднього проектування з використанням паттерну MVC та функціоналу описаного у вигляді діаграми прецедентів сформуємо структурну схему системи, що розробляється (див. рис. 3.2) [15], [16].

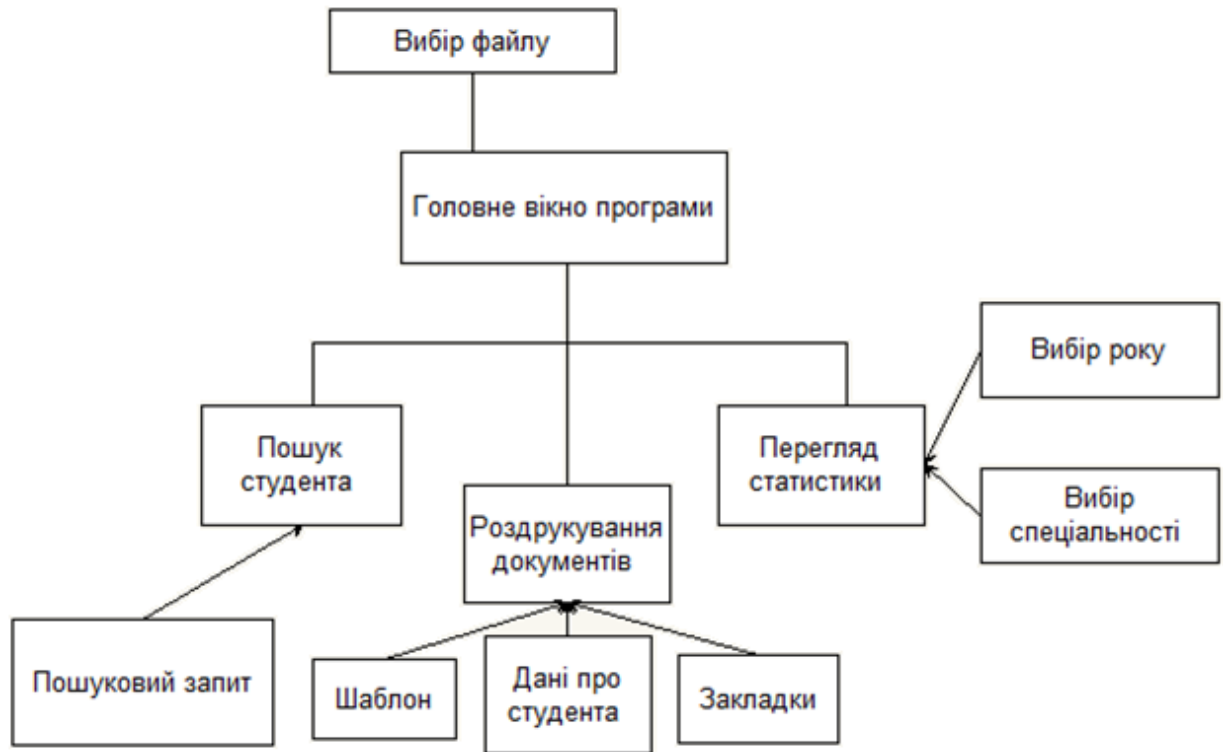


Рисунок 3.2 – Структурна схема програми

Тобто виходячи із структурної схеми програми порядок роботи з підсистемою можна описати наступним чином:

- користувач запускає програму, з’являється головне вікно програми;
- користувач обирає пункт меню “Пошук студента” та вводить пошуковий запит;
- користувач переміщується до роздрукування документа, обирає потрібний шаблон, корегує дані про студента, передає дані у закладки;
- користувач переходить до вкладники «Статистика» та переглядає статистичні дані;
- користувач виходить з програми.

3.2 Обмін даними між системою та Excel-файлом

Для виконання аналізу даних про студентів навчального закладу необхідно виконати імпорт табличних даних з файлу *.xls. Для цього у програмі використовується створення нового екземпляру додатку Excel та виконується ініціалізація діапазону комірок, з якого формуються елементи списку студентів. В цьому списку є такі категорії інформації про студентів:

- номер студента;
- рік вступу;
- рік закінчення;
- прізвище, ім'я, по батькові;
- дата народження;
- дата вступу;
- наказ про вступ;
- дата відрахування;
- наказ про відрахування;
- причина відрахування.

Таким чином, для відкриття файлу та ініціалізації додатку, робочого аркушу, використовується оголошення змінних та відповідна функція:

```
public static string DB_PATH = @"\";

public static BindingList<Student> StudList = new
BindingList<Student>();

// Створення списку студентів для завантаження даних с файлу
private static Excel.Workbook MyBook = null; // Ініціалізація робочої
книги

private static Excel.Application MyApp = null; // Ініціалізація додатку
private static Excel.Worksheet MySheet = null; // Ініціалізація робочого
аркушу книги

OpenFileDialog openFileDialog = new OpenFileDialog();
openDialog.Filter = "Файл Excel|*.XLSX;*.XLS";
```

```

openDialog.ShowDialog();

try
{
    ObjExcel = new Microsoft.Office.Interop.Excel.Application();
    //Книга.
    ObjWorkBook = ObjExcel.Workbooks.Open(openDialog.FileName);
    //Таблиця.
    ObjWorkSheet = ObjExcel.ActiveSheet as
Microsoft.Office.Interop.Excel.Worksheet;
    Microsoft.Office.Interop.Excel.Range rg = null;
    Int32 row = 1;
    dataGridViewMain.Rows.Clear();
    List<String> arr = new List<string>();
    while (ObjWorkSheet.get_Range("a" + row, "a" + row).Value != null)
    {
        // Читання з комірки
        rg = ObjWorkSheet.get_Range("a" + row, "m" + row);
        foreach (Microsoft.Office.Interop.Excel.Range item in rg)
        {
            try
            {
                arr.Add(item.Value.ToString().Trim());
            }
            catch { arr.Add(""); }
        }
        dataGridViewMain.Rows.Add(arr[0], arr[1], arr[2], arr[3], arr[4], arr[5],
arr[6], arr[7], arr[8],arr[9], arr[10], arr[11], arr[12]);arr.Clear();row++;
    }
}

```

Таким чином, дані з файлу Excel завантажуються та відображаються у елементі DataGridView.

3.3 Функціонування системи

Після завантаження даних у комірки DataGridView у системі є можливість виконати пошук даних, використовуючи список стовбців таблиці, а також текстовий рядок, куди вводиться пошуковий запит.

Для цього у програмі реалізовано наступну функцію:

```
public static List<Student> FilterstudList(string searchValue, string
searchExpr)
{
    List<Student> FilteredList = new List<Student>();
    switch (searchValue.ToUpper())
    {
        case "ID":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.ID.ToLower().Contains(searchExpr));
            break;
        case "YearJoin":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.YearJoin.ToLower().Contains(searchExpr));
            break;
        case "YearExpel":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.YearExpel.ToLower().Contains(searchExpr));
            break;
        case "Group":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.Group.ToLower().Contains(searchExpr));
```

```

        break;
        case "Spec":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.Spec.ToLower().Contains(searchExpr));
        break;
        case "Name":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.Name.ToLower().Contains(searchExpr));
        break;
        case "DateBirth":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.DateBirth.ToLower().Contains(searchExpr));
        break;
        case "DateJoin":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.DateJoin.ToLower().Contains(searchExpr));
        break;
        case "OrderJoin":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.OrderJoin.ToLower().Contains(searchExpr));
        break;
        case "DateExpel":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.DateExpel.ToLower().Contains(searchExpr));
        break;
        case "OrderExpel":
            FilteredList = StudList.ToList().FindAll(stud =>
stud.OrderExpel.ToLower().Contains(searchExpr));
        break;
        case "ReasonExpel":

```

```

        FilteredList          =          StudList.ToList().FindAll(stud          =>
stud.ReasonExpel.ToLower().Contains(searchExpr));
        break;
        case "Other":
        FilteredList          =          StudList.ToList().FindAll(stud          =>
stud.Other.ToLower().Contains(searchExpr));
        break;
        default:
        break;
    }
    return FilteredList;

```

Пошук даних використовує метод роботи зі списком FindAll, якому у якості аргументів передається назва стовпця, яку користувач обирає зі списку, а також пошуковий запит, який вводиться у текстове поле.

Виконання статистичних розрахунків теж виконується за допомогою методів роботи зі списками, наприклад, загальна кількість студентів у списку реалізована наступним чином:

```
textBox6.Text = Convert.ToString(MyExcel.StudList.Count());
```

У даному випадку використовується метод Count, який повертає загальну кількість елементів списку.

3.4 Експорт даних у документи Word

Після того, як було знайдено потрібного студента, його ID використовується для використання даних про нього для формування відповідних документів. Для цього було створено три шаблони у програмі Microsoft Word, до яких було додано відповідні закладки, призначені для додавання інформації з програми.

Ініціалізація додатку Word у програмі реалізовано наступним чином:

```

Microsoft.Office.Interop.Word.Application wordapp = new
Microsoft.Office.Interop.Word.Application();// ініціалізація додатку
wordapp.Visible = true;
Microsoft.Office.Interop.Word.Document doc =
wordapp.Documents.Add("doc", Type.Missing, Type.Missing, Type.Missing);//
створення нового документа із використанням шаблону із назвою Doc.

```

Для того, щоб відповідні значення зі списку студентів було передано до створеного документа, використовується функція, що дозволяє заповнити створені закладки відповідними даними з діапазону комірок. Наприклад, для створення архівної довідки на підставі даних з таблиці використовується наступна частина програми:

```

doc.Bookmarks["FIO"].Range.Text = textBox4.Text;
string buv;
if(ObjWorkSheet.Cells[i, "M"] = 1) {buv = "була зарахована";} else {buv
= "був зарахований"};
doc.Bookmarks["BUV"].Range.Text = buv;
doc.Bookmarks["D1"].Range.Text = buv;
doc.Bookmarks["N1"].Range.Text = ObjWorkSheet.Cells[i, "H"];
doc.Bookmarks["PR"].Range.Text = comboBox2.Text;
doc.Bookmarks["D3"].Range.Text = ObjWorkSheet.Cells[i, "J"];
doc.Bookmarks["N2"].Range.Text = ObjWorkSheet.Cells[i, "K"];
doc.Bookmarks["G1"].Range.Text = ObjWorkSheet.Cells[i, "B"];
doc.Bookmarks["G2"].Range.Text = ObjWorkSheet.Cells[i, "C"];
doc.SaveAs2("spravka", "doc");

```

Єдиним недоліком в даному випадку є те, що дані, отримані з книг обліку студентів за 70-90-ті роки, були російською мовою, відповідно, таблиці теж заповнені російською мовою. Тому у програмі передбачено можливість перед передачею даних у документ відкорегувати прізвище, ім'я, по-батькові студента, а також причину відрахування.

Як бачимо, остання функція виконує збереження файлу із встановленим ім'ям та розширенням.

3.5 Висновки за розділом 3

Таким чином, даний розділ розглядає питання функціонування системи, що полягає у налаштуванні тристоронньої взаємодії Таблиця Excel – Інтерфейс користувача – документ Word. На підставі проведеного аналізу можливих програмних засобів реалізації такої взаємодії було обрано оптимальні засоби, які дозволили виконати поставлені задачі. Таким чином, було використано об'єктні моделі роботи із додатками Microsoft Office, а саме Microsoft.Office.Interop.Word, Microsoft.Office.Interop.Excel.

Запропоновано схему функціонування системи в цілому. Ця схема відповідає архітектурі взаємодії користувача із програмою за допомогою інтерфейсу, а також дозволяє виконати усі функції, які покладені на систему, найголовніші з них – пошук студента, створення відповідних документів на підставі знайдених даних.

Крім того, розглянуто основні механізми функціонування програмного засобу.

Запропоновані засоби функціонування як системи в цілому, так і її частин, дозволяють забезпечити логічну та раціональну роботу програмного засобу для аналізу та обробки даних про студентів вищого навчального закладу.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування

Розроблений програмний засіб створено з метою підвищення ефективності роботи архівної та кадрової служби навчального закладу, тому автоматизована система повинна вирішувати наступні задачі:

- автоматизація роботи відділу кадрів та архіву ЗВО на етапі зарахування, переведення, відрахування студентів;
- пошук студентів та випускників за різними критеріями;
- виведення інформації про окремого студента;
- додавання інформації про студента;
- формування на основі інформації з програми документа – архівної довідки, довідки з місця навчання, службового листа із інформацією про випускника із використанням шаблону – бланка ЗВО.

4.2 Умови виконання програми

Для нормального функціонування ПЗ необхідно виконання технічних умов.

Вимоги до складу серверної частини:

- сервер, що працює на базі однієї з лінійки операційних систем (ОС) Windows Server;
- вільне місце на накопичувачі – від 5 Гб;
- оперативна пам'ять стандарту DDR4 об'ємом від 16 Гб;
- мережева карта.

Вимоги до клієнтської частини:

- IBM-сумісний персональний комп'ютер (ПК) під управлінням ОС з лінійки Windows не старше 7 версії;
- вільне місце на накопичувачі – від 1 Гб;

- оперативна пам'ять стандарту DDR3 об'ємом від 4 Гб;
- мережева карта.

4.3 Інструкція по експлуатації програми

4.3.1Звернення до програми

Для запуску програмного продукту необхідно запустити файл програми students.exe. При згортання програми, кнопка відновлення відображається в панелі завдань.

4.3.2Вхідні й вихідні дані

Вхідними даними для програми є: табличні файли, що містять списки студентів, команди, пошукові запити користувача.

Вихідними даними програми є наступні: результати пошуку даних, результати розрахунків, створені документи.

4.4 Методика та результати тестування

4.4.1Сценарій тестування

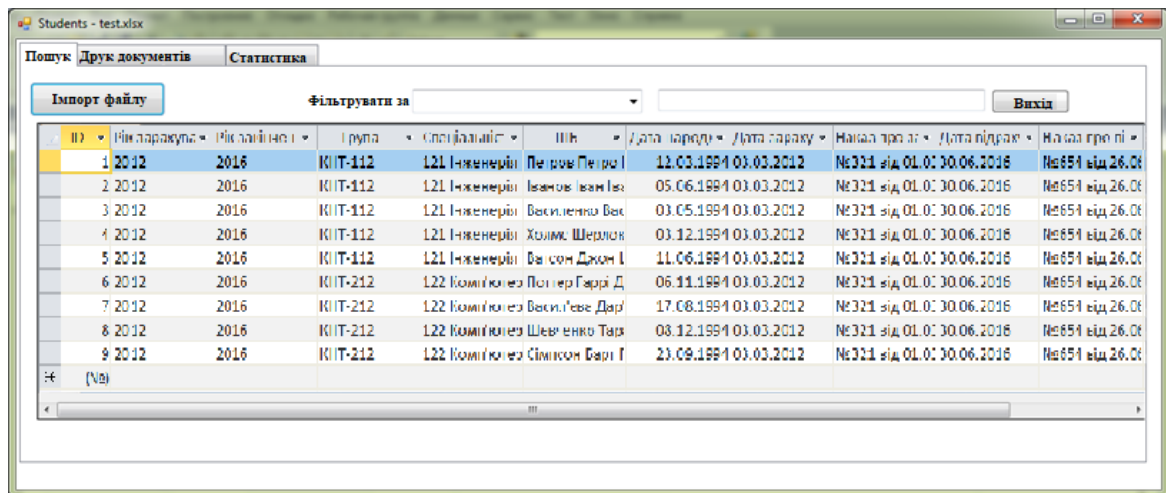
Основний сценарій тестування складемо на основі порядку роботи із системою наступним чином:

- користувач запускає програму, з'являється головне вікно програми;
- користувач обирає пункт меню «Пошук студента» та вводить пошуковий запит;
- користувач переміщується до роздрукування документа, обирає потрібний шаблон, корегує дані про студента, передає дані у закладки;
- користувач переходить до вкладники «Статистика» та переглядає статистичні дані;

– користувач виходить з програми.

4.4.2 Хід виконання тестування

Для того, щоб почати роботу із системою, необхідно запустити застосунок Students.exe. Потім потрібно відкрити файл, що містить список студентів, на першій вкладці буде відображено список студентів з цього файлу (рис 4.1). При натисненні кнопки «Імпорт» з'являється відповідне вікно (рис. 4.2).



The screenshot shows a window titled "Students - test.xlsx" with a menu bar containing "Пошук", "Друк документів", and "Статистика". Below the menu is a toolbar with "Імпорт файлу" and "Вихід". A search filter "Фільтрувати за:" is present. The main area displays a table with the following data:

№	Рік зарахування	Рік закінчення	Група	Спеціальність	Ім'я	Дата народження	Дата зарахування	Назва групи	Дата закінчення
1	2012	2016	КІПТ-112	Інженерія	Петров Петро І	12.03.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
2	2012	2016	КІПТ-112	Інженерія	Іванов Іван І	05.06.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
3	2012	2016	КІПТ-112	Інженерія	Васильченко Вас	03.05.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
4	2012	2016	КІПТ-112	Інженерія	Холмак Шерлок	03.12.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
5	2012	2016	КІПТ-112	Інженерія	Васильченко Вас	11.06.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
6	2012	2016	КІПТ-212	Комп'ютерна	Поптер Гаррі Д	06.11.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
7	2012	2016	КІПТ-212	Комп'ютерна	Васильченко Вас	17.08.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
8	2012	2016	КІПТ-212	Комп'ютерна	Шевченко Тар	08.12.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016
9	2012	2016	КІПТ-212	Комп'ютерна	Сімпсон Барі Г	23.09.1994	03.03.2012	№321 від 01.01.2016	№654 від 26.06.2016

Рисунок 4.1 – Вкладка «Пошук» із результатом завантаження файлу

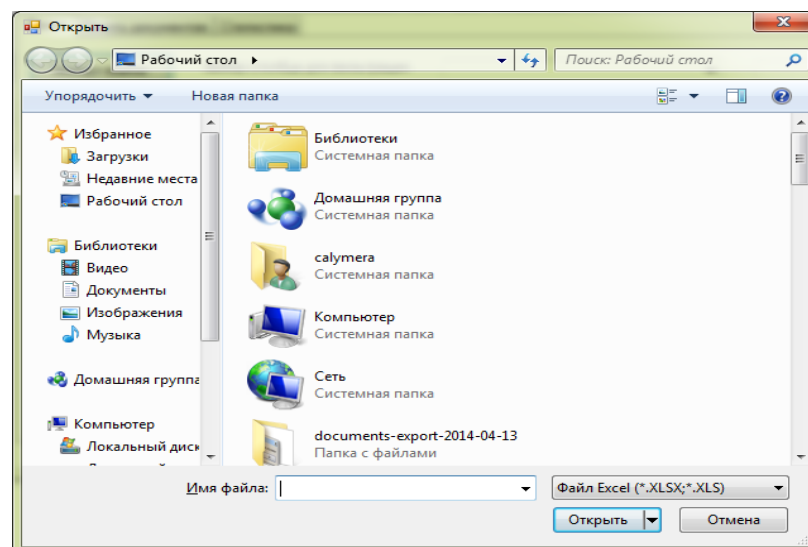


Рисунок 4.2 – Діалогове вікно відкриття табличного файлу

Для того, щоб знайти інформацію про певного студента, необхідно ввести інформацію у текстове поле, наприклад, перші літери прізвища, або дату народження. Наприклад, якщо потрібна інформація про студента Петрова, необхідно обрати стовбець «ПІБ» та ввести у поле для пошуку «Петров». В результаті у таблиці буде відображено інформацію про даного студента (рис. 4.3).

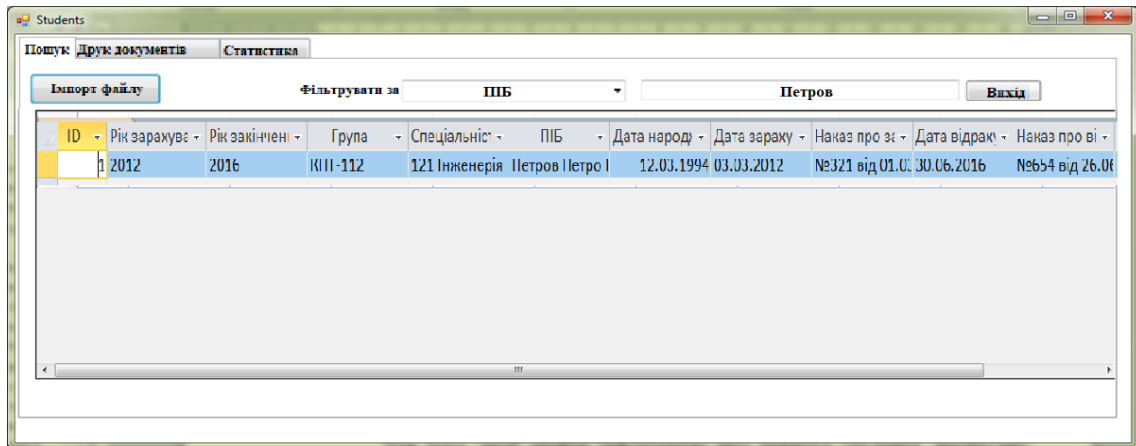


Рисунок 4.3 – Приклад вікна із результатом пошуку

Для того, щоб створити необхідний документ на підставі цих даних, необхідно перейти на вкладинку «Друк документів». У полі «Вибір випускника» буде відображено номер студента у списку – 2. Також автоматично відображається його прізвище, ім'я, по батькові і причина відрахування (рис. 4.4).

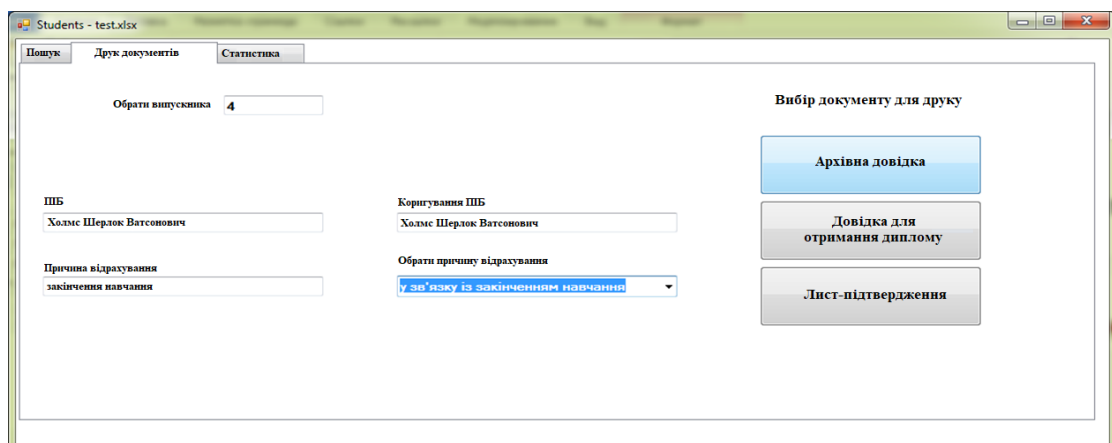


Рисунок 4.4 – Вкладка «Печать документов»

Ці дані необхідно відкорегувати у відповідних полях для передачі в програму MS Word. Після цього користувач натискає кнопку «Архівна довідка» для отримання довідки за інформацією про студента.

Приклад довідки наведено на рисунку 4.5

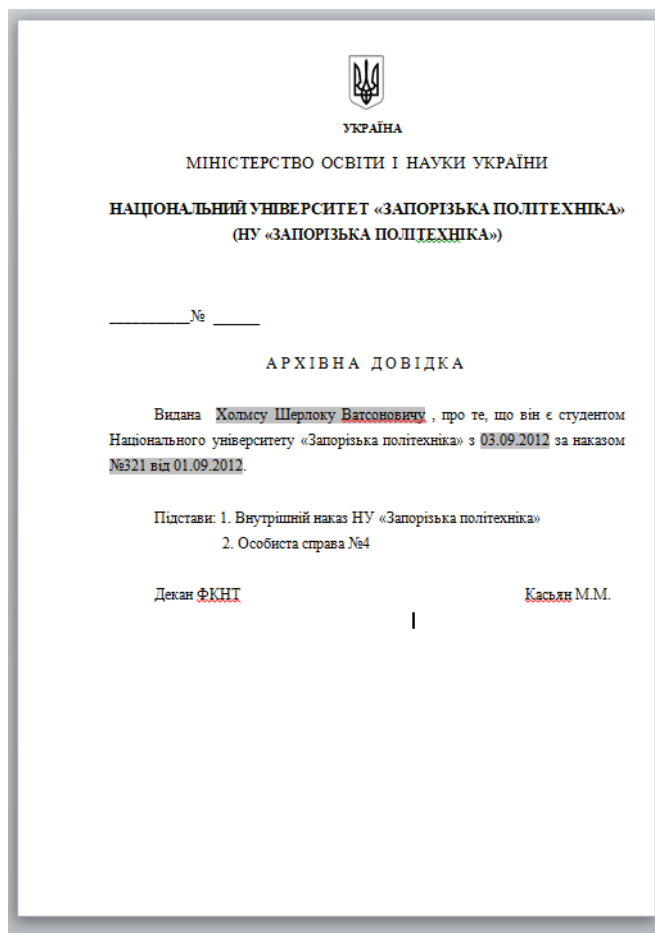


Рисунок 4.5 – Приклад створеного документу

Для перегляду інформації про те, скільки випускників було випущено в цьому ж році, необхідно перейти на вкладку «Статистика» (рис. 4.6), де буде відображено інформацію про відповідний рік, який можна обрати зі списку. Також можна побачити, скільки було випущено студентів за певною спеціальністю.

Students - test.xlsx

Пошук Друк документів Статистика

Загальна кількість студентів в системі

Показати

Диплом з відзнакою 21

Кількість випускників за роком

2012 158

Кількість випускників за спеціальностями

ID	Назва спеціальності	Кількість випускників
1	Інженерія програмного	52
2	Комп'ютерні науки	49
3	Системний аналіз	57
*	(No)	

Вивести дані

Рисунок 4.6 – Приклад вікна зі статистикою

ВИСНОВКИ

В ході виконання даної випускної роботи було проведено огляд та аналіз програмних засобів для обробки даних про студентів ЗВО. Відібрані матеріали були систематизовані, проаналізовані, узагальнені та надані у першому розділі пояснювальної записки.

Створено технічне завдання, в якому вказано мету розробки та її призначення, а також основні вимоги до програмного засобу, порядок її функціонування.

Розроблений програмний продукт має структуру та виконує функції, необхідні для виконання обробки та аналізу даних про студентів ЗВО.

Для програмної реалізації ПЗ було проаналізовано сучасні засоби для проектування та розробки застосунків під управлінням ОС Windows на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної випускної роботи. Було виявлені необхідні засоби. Отже, для розробки практичної частини була розроблена за допомогою: мова програмування C#, середовище розробки Microsoft Visual Studio; система керування базами даних MySQL.

В результаті розробки дипломної кваліфікаційної роботи, було розроблено ПЗ з наступними можливостями:

- можливість додавання інформації про студентів, як існуючих, так і відрахованих з навчального закладу з різних причин;
- реалізація пошуку студента за різними критеріями: дата народження, прізвище, спеціальність, та ін.;
- проведення статистичного аналізу на підставі існуючих даних;
- автоматизоване створення документів на бланках навчального закладу, що забезпечується імпортом з табличного процесора до шаблону відповідного документа: листа, довідки, звіту.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції, для моніторингу інформації про студентів закладу вищої освіти.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Василів, В. Науково-практичні підходи до управління проектами автоматизації обліку результатів навчальної роботи студентів [Текст] / В. Василів, Р. Костюкевич, І. Хоменко // Нова педагогічна думка. – 2013. – № 1.1. – С. 234
2. Дворецька, С. В. Інформаційна система обліку та аналітичної обробки результатів контролю знань студентів [Текст] / С. В. Дворецька // Наукові праці [Чорноморського державного університету імені Петра Могили комплексу "Києво-Могилянська академія"]. Серія : Комп'ютерні технології. – 2014. – Т. 250, Вип. 238. – С. 38-43.
3. Горобець, Л. С. Використання новітніх інформаційних технологій навчання в процесі практичної підготовки студентів за напрямом "Економіка та підприємництво" (на прикладі викладання дисципліни "Бухгалтерський облік в організації") [Текст] / Л. С. Горобець // Вісник Луганського національного університету імені Тараса Шевченка. Педагогічні науки. – 2014. – № 5(2). – С. 173-184.
4. Діденко, О. В. Електронний журнал обліку успішності слухачів (курсантів, студентів) як засіб раціоналізації навчально-виховного процесу [Текст] / О. В. Діденко, Д. А. Купрієнко // Інформаційні технології і засоби навчання. – 2015. – Т. 47, вип. 3. – С. 110-123.
5. Коломоєць, Г. П. Програма підготовки до друку дипломів про вищу освіту за даними ЄДЕБО із використанням сучасних технологій Java [Текст] / Г. П. Коломоєць // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2018. – № 30-31. – С. 57-65.
6. Парус-Навчальний заклад [Електронний ресурс]. – Режим доступу : <http://www.parus.ua/ua/78/>
7. Тауруна [Електронний ресурс]. – Режим доступу : <https://tauruna.ru/>

8. Пакет програм "Деканат" для вищих навчальних закладів III-IV рівнів акредитації [Електронний ресурс]. – Режим доступу : <http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery34>
9. C# documentation [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/dotnet/csharp/>
10. Мова C# і платформа. Net Core [Електронний ресурс]. – Режим доступу : <https://metanit.com/sharp/tutorial/1.1.php>
11. Visual Studio [Електронний ресурс]. – Режим доступу : <https://visualstudio.microsoft.com/ru/>
12. Федорович, О. Є. Забезпечення якості розробки програмних систем з використанням компонентного проектування та технології R-Modeling Framework [Текст] / О. Є. Федорович, О. В. Бабич, О. В. Бабич // Радіоелектронні і комп'ютерні системи. – 2019. – № 1. – С. 66–73.
13. Постіл, С.Д. UML. Уніфікована мова моделювання інформаційних систем [Текст] : навч. посіб. / С. Д. Постіл ; Ун-т держ. фіск. служби України. - Ірпінь : Ун-т держ. фіск. служби України, 2019. – 321 с.
14. Blaha, M. Object-oriented modeling and design with UML [Text] / Michael Blaha, James Rumbaugh. – 2nd ed. – Upper Saddle River, New Jersey : Pearson Prentice Hall, 2005. – XVII, 475 p.
15. Гудзовата, О. О. Використання уніфікованої мови візуального моделювання UML (Unified Modeling Language) як інструменту підтримки проектування інформаційних систем [Текст] / О. О. Гудзовата, В. І. Костирко, І. В. Артищук // Підприємництво і торгівля. – 2019. – Вип. 24. – С. 108-114.
16. Chili, N. S. The challenges of developing small tourism enterprises in townships: case of Umlazi, South Africa [Текст] / N. S. Chili, S. L. Mabaso // Problems and perspectives in management. – 2016. – Vol. 14, Iss. 1(contin). – P. 201-205.

ДОДАТОК А
Технічне завдання

Вступ

Програмний засіб «Система моніторингу інформації про студентів – СМІС» призначена для забезпечення ефективної роботи працівників архівної та кадрової служби навчального закладу та дозволяє виконувати обробку списків студентів закладу, формування необхідної документації.

А.1 Підстава для розробки

Підставою для розробки слугує завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація системи моніторингу інформації про студентів закладу вищої освіти», затверджене наказом Національного університету «Запорізька політехніка» № 103 від 30 березня 2021 р.

А.2 Призначення розробки

Програмний засіб «Система моніторингу інформації про студентів» призначена для забезпечення ефективної роботи працівників архівної та кадрової служби навчального закладу та дозволяє виконувати обробку списків студентів закладу, формування необхідної документації.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмний продукт повинен виконувати такі основні функції:

- автоматизація роботи відділу кадрів та архіву ЗВО на етапі зарахування, переведення, відрахування студентів;
- пошук студентів та випускників за різними критеріями;
- виведення інформації про окремого студента;

- додавання інформації про студента;
- формування на основі інформації з програми документа – архівної довідки, довідки з місця навчання, службового листа із інформацією про випускника із використанням шаблону – бланка ЗВО.

Користувацький інтерфейс повинен забезпечити просте та ефективне спілкування користувача із програмним продуктом.

А.3.2 Вимоги до надійності

Програма повинна коректно відображати робочі форми, своєчасно відповідати на натискання клавіш та рухи миші, не призводити до «зависання» комп'ютерів спеціалістів, точно розраховувати усі статистичні дані та коректно відображати документи, які необхідно підготувати.

А.3.3 Умови експлуатації

Програма може зберігатися та переноситися на будь-яких існуючих носіях інформації.

Експлуатація програми здійснюється згідно експлуатаційної документації на розроблену програму, яка відповідає стандартам і містить інформацію, необхідну для її освоєння та експлуатації.

А.3.4 Вимоги до складу та параметрів технічних засобів

Для нормального функціонування ПЗ необхідно виконання технічних умов.

Вимоги до складу серверної частини:

- сервер, що працює на базі однієї з лінійки операційних систем (ОС) Windows Server;
- вільне місце на накопичувачі – від 5 Гб;

- оперативна пам'ять стандарту DDR4 об'ємом від 16 Гб;
- мережева карта.

Вимоги до клієнтської частини:

- IBM-сумісний персональний комп'ютер (ПК) під управлінням ОС з лінійки Windows не старше 7 версії;
- вільне місце на накопичувачі – від 1 Гб;
- оперативна пам'ять стандарту DDR3 об'ємом від 4 Гб;
- мережева карта.

A.3.5 Вимоги до маркування і пакування

Програмний продукт повинен поставлятися на носії інформації з обов'язковим назвою програми на упаковці – «Система моніторингу інформації про студентів».

A.3.6 Вимоги до транспортування й зберігання

Загальний початковий розмір файлу програми складає близько 500 Мб, що дозволяє записувати його на будь-який сучасний носій інформації. Відповідно до вибраного носієм визначаються і вимоги до зберігання і транспортування.

A.3.7 Вимоги до програмної документації

Програмний продукт повинен поставлятися у наступному складі:

- технічне завдання;
- опис програми;
- керівництво програміста;
- керівництво оператора;
- текст програми.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програмний продукт «Система моніторингу інформації про студентів» призначений для організації роботи спеціалістів архівної та кадрової служби навчального закладу.

Б.2 Функціональне призначення

Програмний продукт повинен виконувати такі основні функції:

- автоматизація роботи відділу кадрів та архіву ЗВО на етапі зарахування, переведення, відрахування студентів;
- пошук студентів та випускників за різними критеріями;
- виведення інформації про окремого студента;
- додавання інформації про студента;
- формування на основі інформації з програми документа – архівної довідки, довідки з місця навчання, службового листа із інформацією про випускника із використанням шаблону – бланка ЗВО.

Б.3 Опис логічної структури

Структуру роботи підсистеми можна представити у вигляді схеми зображеної на рисунку Б.1



Рисунок Б.1 – Структурна схема системи

Порядок роботи з підсистемою можна описати наступним чином:

- користувач запускає програму, з'являється головне вікно програми;
- користувач обирає пункт меню «Пошук студента» та вводить пошуковий запит;
- користувач переміщується до роздрукування документа, обирає потрібний шаблон, корегує дані про студента, передає дані у закладки;
- користувач переходить до вкладки «Статистика» та переглядає статистичні дані;
- користувач виходить з програми.

Б.4 Використані технічні засоби

Для нормального функціонування ПЗ необхідно виконання технічних умов.

Вимоги до складу серверної частини:

- сервер, що працює на базі однієї з лінійки операційних систем (ОС) Windows Server;
- вільне місце на накопичувачі – від 5 Гб;
- оперативна пам'ять стандарту DDR4 об'ємом від 16 Гб;
- мережева карта.

Вимоги до клієнтської частини:

- IBM-сумісний персональний комп'ютер (ПК) під управлінням ОС з лінійки Windows не старше 7 версії;
- вільне місце на накопичувачі – від 1 Гб;
- оперативна пам'ять стандарту DDR3 об'ємом від 4 Гб;
- мережева карта.

Б.5 Виклик і завантаження

Для запуску програмного комплексу необхідно запустити виконуваний файл students.

Б.6 Вхідні та вихідні дані

Вхідними даними для програми є: табличні файли, що містять списки студентів, команди, пошукові запити користувача.

Вихідними даними програми є наступні: результати пошуку даних, результати розрахунків, створені документи.

ДОДАТОК В
Керівництво програміста

В.1 Призначення та умови застосування програми

Розроблений програмний продукт призначений для виконання обробки та аналізу даних про студентів вищого навчального закладу.

Для роботи програмного продукту необхідно використання IBM-сумісного ПК з наступного програмно-апаратною платформою:

Для нормального функціонування ПЗ необхідно виконання технічних умов.

Вимоги до складу серверної частини:

- сервер, що працює на базі однієї з лінійки операційних систем (ОС) Windows Server;
- вільне місце на накопичувачі – від 5 Гб;
- оперативна пам'ять стандарту DDR4 об'ємом від 16 Гб;
- мережева карта.

Вимоги до клієнтської частини:

- IBM-сумісний персональний комп'ютер (ПК) під управлінням ОС з лінійки Windows не старше 7 версії;
- вільне місце на накопичувачі – від 1 Гб;
- оперативна пам'ять стандарту DDR3 об'ємом від 4 Гб;
- мережева карта.

В.2 Характеристика програми

Розробка програмного продукту «Автоматизована система обробки та аналізу інформації про студентів вищого навчального закладу» проводилась із використанням середовища розробки Microsoft Visual Studio 2016 Express, ОС Windows Seven. Тестування також проводилось на ОС Windows Eight з використанням MS Office 2010.

Програма має дружній інтерфейс, створені компоненти керування полегшують взаємодію користувача з програмою.

В.3 Звернення до програми

Для запуску програмного продукту необхідно запустити файл програми `students.exe`. При згортання програми, кнопка відновлення відображається в панелі завдань.

В.4 Вхідні та вихідні дані

Вхідними даними для програми є: табличні файли, що містять списки студентів, команди, пошукові запити користувача.

Вихідними даними програми є наступні: результати пошуку даних, результати розрахунків, створені документи.

В.5 Повідомлення програмісту

Реалізована програмна система гарантує її стабільну роботу. Виключення можуть виникати в тому випадку, коли користувач буде намагатись завантажити до програми файл, у разі виникнення помилки йому буде відображене відповідне повідомлення: «Ошибка при считывании excel файла».

Таким чином, створена система має широкі функціональні можливості. Розроблений програмний засіб забезпечує стійку роботу та обробку виключень.

ДОДАТОК Г
Керівництво оператора

Г.1 Призначення програми

Розроблений програмний засіб створено з метою підвищення ефективності роботи архівної та кадрової служби навчального закладу, тому автоматизована система повинна вирішувати наступні задачі:

- автоматизація роботи відділу кадрів та архіву ЗВО на етапі зарахування, переведення, відрахування студентів;
- пошук студентів та випускників за різними критеріями;
- виведення інформації про окремого студента;
- додавання інформації про студента;
- формування на основі інформації з програми документа – архівної довідки, довідки з місця навчання, службового листа із інформацією про випускника із використанням шаблону – бланка ЗВО.

Г.2 Інтерфейс користувача

Зараз жодна організація не може працювати без автоматизованої інформаційної системи. Зазвичай, користувачі (в нашому випадку це спеціалісти державного центру зайнятості та безробітні) інформаційних систем - це люди далекі від розуміння тонкощів обчислювальної техніки, тому розробка зручного та інтуїтивно-зрозумілого інтерфейсу користувача для роботи з системою – це одна з найважливіших задач при створенні програмного засобу для підприємства.

Під інтерфейсом користувача розуміється комплекс засобів, які надаються програмою для керування нею та перегляду результатів її роботи. Для даної програми інтерфейс складається з форм вводу й редагування даних, а також документів, які після перегляду на екрані можуть бути надруковані.

Проектування форм для введення та редагування даних повинно відповідати наступним вимогам:

- логічно обґрунтовані угруповання та послідовності полів;
- логічно обґрунтовані угруповання та послідовності полів;
- візуально привабливий вигляд вікна форми або поля звіту;
- погоджену термінологію та скорочення;
- погоджене використання кольорів;
- візуальне виділення простору та границь полів введення даних;
- зручні засоби переміщення курсору;
- засоби виводу повідомлень про помилки при введенні неприпустимих значень.

Згідно до вимог було розроблено інтерфейс користувача для даного програмного засобу. Приклади інтерфейсу наведені нижче, (рис. Г.1-Г.6).

Після завантаження програми завантажується форма із активною першою вкладинкою (рис. Г.1), вона дозволяє завантажити до системи файл, який містить список студентів для обробки та аналізу. При натисненні кнопки «Імпорт» відкривається вікно вибору файлу (рис. Г.2).

Після цього користувач бачить результат імпорту та може знайти відповідного студента, для цього йому необхідно ввести відомі йому дані у текстовий блок, пошук буде виконаний після введення кожного символу.

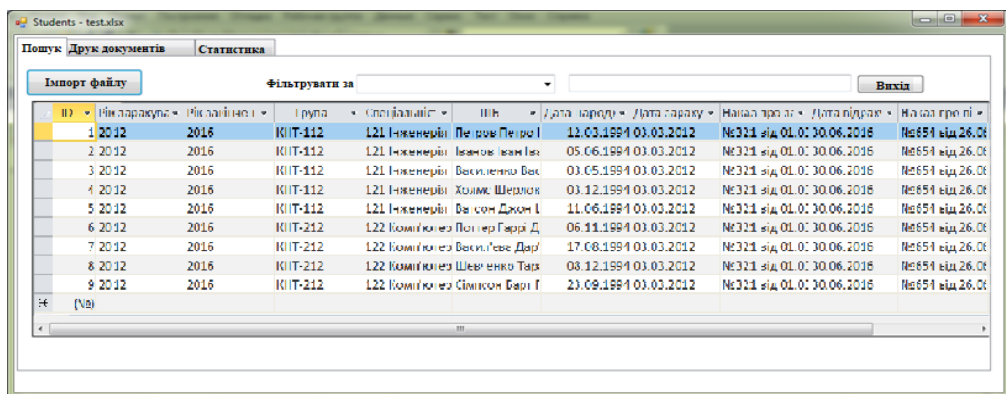
Для роздрукування інформації про знайденого студента користувач переходить до вкладки «Друк документу», у разі необхідності вносить корективи у особисті дані, це необхідно через невідповідність мови, на якій заповнені дані з архівних книг, мові, якою повинні бути складені документи. Після цього користувач обирає потрібний йому документ натисненням відповідної кнопки. Буде завантажено документ Microsoft Word, створений на основі заданого шаблону із вставленими даними з діапазону комірок таблиці.

Для отримання статистичної інформації з даного файлу необхідно перейти на вкладнику «Статистика». На цій вкладниці (рис. Г.3) відображається загальна кількість студентів у списку, кількість дипломів з

відзнакою, а також статистична інформація про кількість випускників, які закінчили навчальний заклад у певному році, або отримали певну спеціальність.

Г.3 Виконання програми

Для того, щоб почати роботу із системою, необхідно запустити застосунок Students.exe. Потім потрібно відкрити файл, що містить список студентів, на першій вкладці буде відображено список студентів з цього файлу (рис Г.1). При натисненні кнопки «Імпорт» з'являється відповідне вікно (рис. Г.2).



The screenshot shows a window titled 'Students - test.xlsx' with a menu bar containing 'Поиск', 'Друк документів', and 'Статистика'. Below the menu is a toolbar with 'Імпорт файлу' and 'Вивести'. A search bar is also present. The main area displays a table with the following data:

№	Рік народження	Рік закінчення	Група	Спеціальність	ІМІ	Дата народження	Дата закінчення	Назва проєкту	Дата проєкту
1	2012	2016	КІПТ-112	Інженерія	Петров Петро І	12.03.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
2	2012	2016	КІПТ-112	Інженерія	Іванов Іван І	05.06.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
3	2012	2016	КІПТ-112	Інженерія	Васильченко Вас	01.05.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
4	2012	2016	КІПТ-112	Інженерія	Хоміж Шерлок	03.12.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
5	2012	2016	КІПТ-112	Інженерія	Васильченко Вас	11.06.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
6	2012	2016	КІПТ-212	Інженерія	Попов Петро Д	06.11.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
7	2012	2016	КІПТ-212	Інженерія	Васильченко Вас	17.08.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
8	2012	2016	КІПТ-212	Інженерія	Шевченко Тарас	08.12.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016
9	2012	2016	КІПТ-212	Інженерія	Сімпсон Барі Г	23.09.1994	03.03.2012	№311 від 01.01.2016	№651 від 26.06.2016

Рисунок Г.1 – Вкладка «Пошук» із результатом завантаження файлу

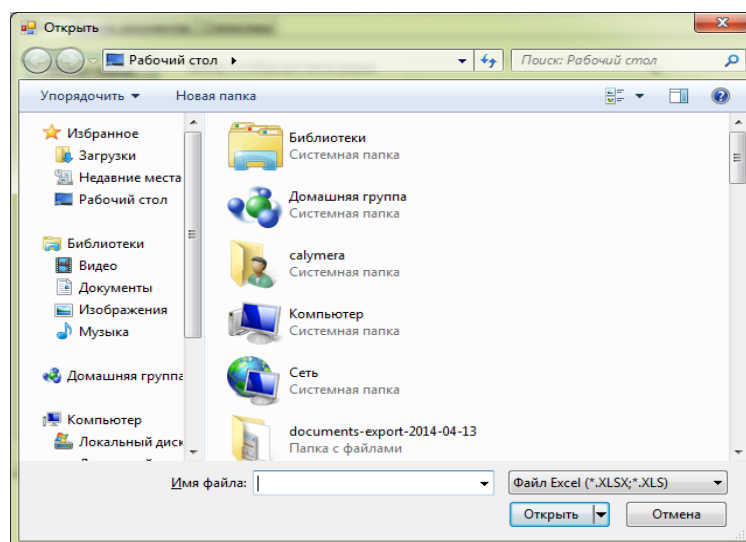


Рисунок Г.2 – Діалогове вікно відкриття табличного файлу

Для того, щоб знайти інформацію про певного студента, необхідно ввести інформацію у текстове поле, наприклад, перші літери прізвища, або дату народження. Наприклад, якщо потрібна інформація про студента Петрова, необхідно обрати стовбець «ПІБ» та ввести у поле для пошуку «Петров». В результаті у таблиці буде відображено інформацію про даного студента (рис. Г.3).

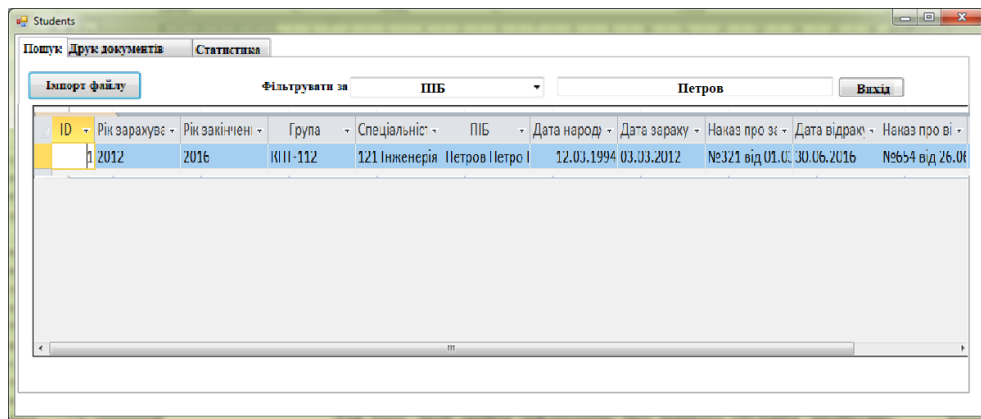


Рисунок Г.3 – Приклад вікна із результатом пошуку

Для того, щоб створити необхідний документ на підставі цих даних, необхідно перейти на вкладинку «Друк документів». У полі «Вибір випускника» буде відображено номер студента у списку – 2. Також автоматично відображається його прізвище, ім'я, по батькові і причина відрахування (рис. Г.4).

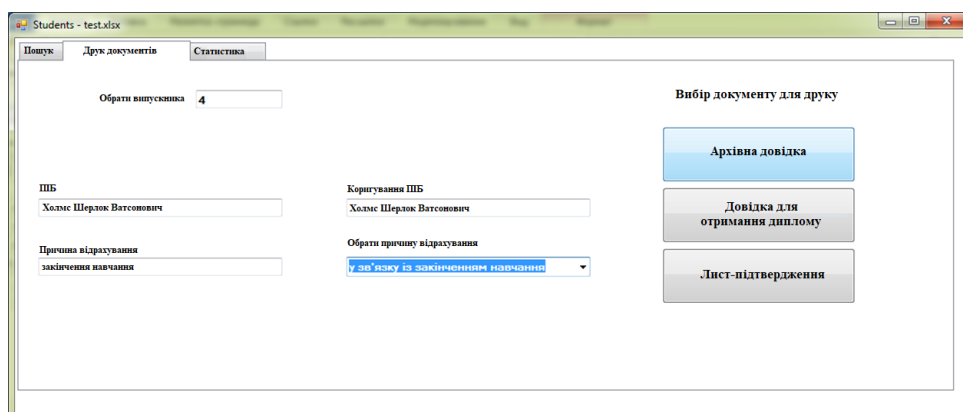


Рисунок Г.4 – Вкладка «Печать документов»

Ці дані необхідно відкорегувати у відповідних полях для передачі в програму MS Word. Після цього користувач натискає кнопку «Архівна довідка» для отримання довідки за інформацією про студента.

Приклад довідки наведено на рисунку Г.5.

УКРАЇНА
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»
(НУ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»)

№ _____

АРХІВНА ДОВІДКА

Видана Холмєу Шеремєу Василєвичу, про те, що він є студентом
Національного університету «Запорізька політехніка» з 03.09.2012 за наказом
№321 від 01.09.2012.

Підстави: 1. Внутрішній наказ НУ «Запорізька політехніка»
2. Особиста справа №4

Декан ФКНТ _____ Касьян М.М. _____

Рисунок Г.5 – Приклад створеного документу

Для перегляду інформації про те, скільки випускників було випущено в цьому ж році, необхідно перейти на вкладку «Статистика» (рис. Г.6), де буде відображено інформацію про відповідний рік, який можна обрати зі списку. Також можна побачити, скільки було випущено студентів за певною спеціальністю.

ID	Назва спеціальності	Кількість випускників
1	Інженерія програмного	52
2	Комп'ютерні науки	49
3	Системний аналіз	57
*	(Не)	

Рисунок Г.6 – Приклад вікна зі статистикою

Г.4 Повідомлення оператору

У разі виникнення помилок у програмі передбачено відображення стандартних діалогових вікон. Подібні повідомлення можуть виникати під час виникнення небажаних ситуацій, до яких можна віднести помилки при введенні даних, пошкодження файлу, відкриття неіснуючого файлу.

Приклад діалогового вікна, яке з'явиться у випадку помилки при відкритті файлу, наведено на рис. Г.7.

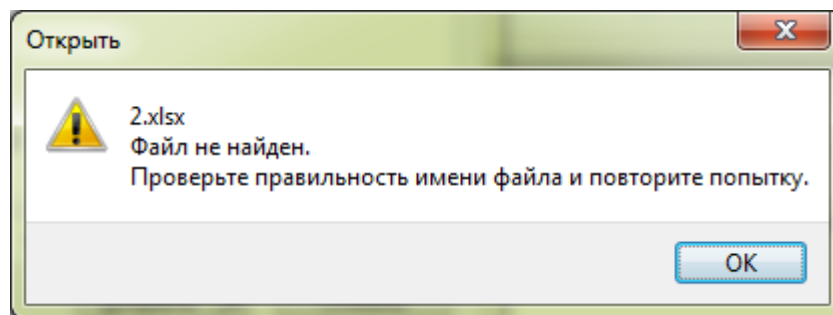


Рисунок Г.7 – Діалогове вікно

ДОДАТОК Д
Текст програми

```

package com.example.chatapp;
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 Microsoft.Office.Interop.Excel;
using System.Net.Mail;
using System.Net;
using System.Threading;
using System.Data.SqlClient;
using DataGridViewAutoFilter;
using Microsoft.Office.Interop.Word;

namespace DemoExcel
{
    public partial class Form1 : Form
    {
        private Microsoft.Office.Interop.Excel.Application
ObjExcel;
        private Microsoft.Office.Interop.Excel.Workbook
ObjWorkBook;
        private Microsoft.Office.Interop.Excel.Worksheet
ObjWorkSheet;
        int i;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void buttonExport_Click(object sender,
EventArgs e)
        {
            try
            {

```

```

ObjExcel = new
Microsoft.Office.Interop.Excel.Application();
//Книга.
ObjWorkBook =
ObjExcel.Workbooks.Add(System.Reflection.Missing.Value);
//Таблица.
ObjWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ObjWorkBook.Sheets[1];

for (int i = 0; i <
dataGridViewMain.Rows.Count; i++)
{
    DataGridViewRow row =
dataGridViewMain.Rows[i]; // строки

    for (int j = 0; j < row.Cells.Count;
j++) //цикл по ячейкам строки
    {
        ObjExcel.Cells[i + 1, j + 1] =
row.Cells[j].Value;
    }
}

//ObjWorkBook.SaveAs(fileName);
}
catch (Exception ex) {
MessageBox.Show(ex.Message, "Error");}
finally
{
    {
        ObjWorkBook.Close();
        // Заккрытие приложения Excel.
        ObjExcel.Quit();
        ObjWorkBook = null;
        ObjWorkSheet = null;
        ObjExcel = null;
        GC.Collect();
    }
}

private void buttonImport_Click(object sender,
EventArgs e)
{
    OpenFileDialog openFileDialog = new
OpenFileDialog();
    openFileDialog.Filter = "Файл Excel|*.XLSX;*.XLS";
    openFileDialog.ShowDialog();

    try
    {

```

```

        ObjExcel = new
Microsoft.Office.Interop.Excel.Application();
        //Книга.
        ObjWorkBook =
ObjExcel.Workbooks.Open(openDialog.FileName);
        //Таблица.
        ObjWorkSheet = ObjExcel.ActiveSheet as
Microsoft.Office.Interop.Excel.Worksheet;
        Microsoft.Office.Interop.Excel.Range rg =
null;

        Int32 row = 1;
        dataGridViewMain.Rows.Clear();
        List<String> arr = new List<string>();
        while (ObjWorkSheet.get_Range("a" + row,
"a" + row).Value != null)
        {
            // Читаем данные из ячейки
            rg = ObjWorkSheet.get_Range("a" + row,
"m" + row);

            foreach
(Microsoft.Office.Interop.Excel.Range item in rg)
            {
                try
                {
arr.Add(item.Value.ToString().Trim());
                }
                catch { arr.Add(""); }
            }
            dataGridViewMain.Rows.Add(arr[0],
arr[1], arr[2], arr[3], arr[4], arr[5], arr[6], arr[7], arr[8],
arr[9], arr[10], arr[11], arr[12]);
            arr.Clear();
            row++;
        }

        MessageBox.Show("Файл успешно считан!",
"Считывания excel файла", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    catch (Exception ex) { MessageBox.Show("Ошибка:
" + ex.Message, "Ошибка при считывании excel файла",
MessageBoxButtons.OK, MessageBoxIcon.Error); }

    {

    }
}

```

```

        this.Text = this.Text + " - " +
openDialog.SafeFileName;

    }

    private void textBox2_TextChanged(object sender,
EventArgs e)
    {
        i = Convert.ToInt16(textBox2.Text);
        textBox3.Text = ObjWorkSheet.Cells[i,
"F"].Value.ToString();
        textBox5.Text = ObjWorkSheet.Cells[i,
"L"].Value.ToString();

    }

    private void button1_Click(object sender, EventArgs
e)
    {
        Microsoft.Office.Interop.Word.Application
wordapp = new Microsoft.Office.Interop.Word.Application();
        wordapp.Visible = true;

        Microsoft.Office.Interop.Word.Document doc =
wordapp.Documents.Add("doc", Type.Missing, Type.Missing,
Type.Missing);

        doc.Bookmarks["FIO"].Range.Text =
textBox4.Text;
        string buv;
        if(ObjWorkSheet.Cells[i, "M"] = 1) {buv =
"була зарахована";} else {buv = "був зарахований"};
        doc.Bookmarks["BUV"].Range.Text = buv;
        doc.Bookmarks["D1"].Range.Text = buv;
        doc.Bookmarks["N1"].Range.Text =
ObjWorkSheet.Cells[i, "H"];
        doc.Bookmarks["PR"].Range.Text =
comboBox2.Text;
        doc.Bookmarks["D3"].Range.Text =
ObjWorkSheet.Cells[i, "J"];
        doc.Bookmarks["N2"].Range.Text =
ObjWorkSheet.Cells[i, "K"];
        doc.Bookmarks["G1"].Range.Text =
ObjWorkSheet.Cells[i, "B"];
        doc.Bookmarks["G2"].Range.Text =
ObjWorkSheet.Cells[i, "C"];
        doc.SaveAs2("spravka", "doc");

    }

```

```

        private void button4_Click(object sender, EventArgs
e)
        {
            ObjExcel.Quit();
            System.Windows.Forms.Application.Exit();
        }

        private void button5_Click(object sender, EventArgs
e)
        {
            OpenFileDialog ExcelDialog = new
OpenFileDialog();
            ExcelDialog.Filter = "Excel Files (*.xlsx) |
*.xlsx";
            ExcelDialog.InitialDirectory = @"C:\";
            ExcelDialog.Title = "Select your team excel";
            if (ExcelDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
            {
                MyExcel.DB_PATH = ExcelDialog.FileName;
                MyExcel.InitializeExcel();
                MyExcel.ReadMyExcel();
            }
            textBox6.Text =
Convert.ToString(MyExcel.StudList.Count());
        }
    }

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Office.Interop.Excel;
using System.Data.OleDb;
using Excel = Microsoft.Office.Interop.Excel;
using System.ComponentModel;

namespace DemoExcel
{
    class MyExcel
    {
        public static string DB_PATH = @"";
        public static BindingList<Student> StudList = new
BindingList<Student>();
        private static Excel.Workbook MyBook = null;
        private static Excel.Application MyApp = null;
        private static Excel.Worksheet MySheet = null;
    }
}

```

```

private static int lastRow = 0;

public static void InitializeExcel()
{
    MyApp = new Excel.Application();
    MyApp.Visible = false;
    MyBook = MyApp.Workbooks.Open(DB_PATH);
    MySheet = (Excel.Worksheet)MyBook.Sheets[1]; //
    Explicit cast is not required here
    lastRow =
    MySheet.Cells.SpecialCells(Excel.XlCellType.xlCellTypeLastCell).
    Row;
}

public static BindingList<Student> ReadMyExcel()
{
    StudList.Clear();
    for (int index = 2; index <= lastRow; index++)
    {
        System.Array MyValues =
        (System.Array)MySheet.get_Range("A" + index.ToString(), "L" +
        index.ToString()).Cells.Value;
        StudList.Add(new Student
        {
            ID = (MyValues.GetValue(1,
1).ToString()),
            YearJoin = MyValues.GetValue(1,
2).ToString(),
            YearExpel = MyValues.GetValue(1,
3).ToString(),
            Group = MyValues.GetValue(1,
4).ToString(),
            Spec = MyValues.GetValue(1,
5).ToString(),
            Name = MyValues.GetValue(1,
6).ToString(),
            DateBirth = MyValues.GetValue(1,
7).ToString(),
            DateJoin = MyValues.GetValue(1,
8).ToString(),
            OrderJoin = MyValues.GetValue(1,
9).ToString(),
            DateExpel = MyValues.GetValue(1,
10).ToString(),
            OrderExpel = MyValues.GetValue(1,
11).ToString(),
            ReasonExpel = MyValues.GetValue(1,
12).ToString()

        });
    }
    return StudList;
}

```

```

    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DemoExcel
{
    class Student
    {
        string ID;
        string YearJoin;
        string YearExpel;
        string Group;
        string Spec;
        string Name;
        string DateBirth;
        string DateJoin;
        string OrderJoin;
        string DateExpel;
        string OrderExpel;
        string ReasonExpel;
        string Other;
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Students_registration.CLASSES
{
    class Student_reg
    {
        //properties or attributes
        public int id;
        public string Name;
        public string sur_name;
        public int age;
        public string gender;
        public string address;

        //constructor
        public Student_reg() {
            this.id = 0;
            this.Name = "";
            this.sur_name = "";
            this.age = 0;
            this.gender = "";
        }
    }
}

```

```

        this.address = "";
    }

    //save method
    public void save() {
        string sql = "";
        sql = "INSERT INTO registered_students
VALUES (NULL, '" +
            this.Name + "', '" + this.sur_name + "', '" +
            this.age + "', '" + this.gender + "', '" +
this.address + "')";
        try {
            GLOBAL_VARS.d.executeNonQuery(sql);
        }
        catch (Exception e) {
            MessageBox.Show(e.Message);
        }
    }

    public void loadStudents(LstVlew lsv) {
        lsv.Items.Clear();
        string sql = "SELECT * FROM
registered_students";
        GLOBAL_VARS.d.execute(sql);
        if (GLOBAL_VARS.d.reader.HasRows) {
            while (GLOBAL_VARS.d.reader.Read()) {
                int i = lsv.Items.Count;

lsv.Items.Add(GLOBAL_VARS.d.reader["Id"].ToString());

lsv.Items[i].SubItems.Add(GLOBAL_VARS.d.reader["Name"].ToString(
));

lsv.Items[i].SubItems.Add(GLOBAL_VARS.d.reader["sur_name"].ToStr
ing());

lsv.Items[i].SubItems.Add(GLOBAL_VARS.d.reader["age"].ToString()
);

lsv.Items[i].SubItems.Add(GLOBAL_VARS.d.reader["gender"].ToStrln
g());

lsv.Items[i].SubItems.Add(GLOBAL_VARS.d.reader["address"].ToStrl
ng());

            }
        }
        //GLOBAL_VARS.d.reader.Close();
    }

    public void loadStudent(int id) {
        string sql = "SELECT * FROM
`students`.`registered_students` "+
            "WHERE Id="+id+"";
    }

```

```

        try
        {
            GLOBAL_VARS.d.execute(sql);
            if (GLOBAL_VARS.d.reader.HasRows)
            {
                GLOBAL_VARS.d.reader.Read();
                thls.ld =
Convert.ToInt32(GLOBAL_VARS.d.reader["ld"].ToString());
                thls.Name =
GLOBAL_VARS.d.reader["Name"].ToString();
                thls.sur_name =
GLOBAL_VARS.d.reader["sur_name"].ToString();
                thls.age =
Convert.ToInt32(GLOBAL_VARS.d.reader["age"].ToString());
                thls.gender =
GLOBAL_VARS.d.reader["gender"].ToString();
                thls.address =
GLOBAL_VARS.d.reader["address"].ToString();
            }
        }
        catch (Exception e) {
            MessageBox.Show(e.Message);
        }
        finally{
            GLOBAL_VARS.d.reader.Close();
        }
    }

    public void updateStudent() {
        string sql = "UPDATE
`students`.`reglstered_students` "+
            "SET Name='"+thls.Name+"', "+
            "sur_name='"+thls.sur_name+"', "+
            "age='"+thls.age+"', "+
            "gender='"+thls.gender+"', "+
            "address='"+thls.address+"' "+
            "WHERE ld='"+thls.ld+"";
        GLOBAL_VARS.d.executeNonQuery(sql);
    }

    public void deleteStudent() {
        string sql = "DELETE FROM reglstered_students
"+
            "WHERE ld='"+thls.ld+"";
        GLOBAL_VARS.d.executeNonQuery(sql);
    }

    public void searchStudent(string keyword, ListView
lsv) {
        lsv.Items.Clear();
        string sql = "SELECT * FROM
`students`.`reglstered_students` "+

```

```

        "where Name llke '"+keyword+"%' OR
"+
        "sur_name llke '" + keyword + "%'
OR "+
        "address llke '" + keyword +
"%'";

GLOBAL_VARS.d.execute(sql);

if (GLOBAL_VARS.d.reader.HasRows)
{
    while (GLOBAL_VARS.d.reader.Read())
    {
        int l = lsv.ltems.Count;

lsv.ltems.Add(GLOBAL_VARS.d.reader["ld"].ToString());

lsv.ltems[l].Subltems.Add(GLOBAL_VARS.d.reader["Name"].ToString(
));

lsv.ltems[l].Subltems.Add(GLOBAL_VARS.d.reader["sur_name"].ToStr
lng());

lsv.ltems[l].Subltems.Add(GLOBAL_VARS.d.reader["age"].ToString()
);

lsv.ltems[l].Subltems.Add(GLOBAL_VARS.d.reader["gender"].ToStrln
g());

lsv.ltems[l].Subltems.Add(GLOBAL_VARS.d.reader["address"].ToStrl
ng());

    }
}
GLOBAL_VARS.d.reader.Close();
}
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace Students_registration.CLASSES
{
    class Db
    {
        //local variables
        public string _server = "";
        public string _user = "";
        public string _pw = "";
        public string _db = "";
        public MySqlDataReader reader;
    }
}

```

```

        public MySqlConnection conn = new
        MySqlConnection();

        public bool connect() {
            string server_string;
            server_string = "server = " + this._server +
                            ";username = " + this._user +
                            ";password = " + this._pw +
                            ";database = " + this._db;
            this.conn.ConnectionString = server_string;

            try
            {
                conn.Open();
                MessageBox.Show("database connected");
                return true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
            }
        }

        public void executeNonQuery(string sql) {
            try
            {
                MySqlCommand comm = new MySqlCommand();
                comm.Connection = this.conn;
                comm.CommandText = sql;
                comm.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }

        public MySqlDataReader execute(string sql)
        {
            try
            {
                MySqlCommand comm = new MySqlCommand();
                comm.Connection = this.conn;
                comm.CommandText = sql;
                this.reader = comm.ExecuteReader();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
            return this.reader;
        }

```

}
}
}

ДОДАТОК Е
Слайди презентації

Міністерство освіти та науки України
Національний Університет «Запорізька політехніка»

**Програмна реалізація системи
моніторингу інформації про студентів
закладу вищої освіти**

Жидкова Катерина Костянтинівна

ст. групи КНТ-118сп

Олійник Андрій Олександрович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Е.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес моніторингу інформації про студентів.

Предмет дослідження – методи моніторингу інформації про студентів.

Метою роботи є програмна реалізація системи моніторингу інформації про студентів.

2

Рисунок Е.2 – Слайд 2

Постановка завдання

Основні можливості, які повинна надавати система, яка забезпечує збереження, обробку і аналіз даних про студентів навчального закладу:

- 1) можливість додавання інформації про студентів, як існуючих, так і відрахованих з навчального закладу з різних причин;
- 2) реалізація пошуку студента за різними критеріями: дата народження, прізвище, спеціальність, та ін.;
- 3) проведення статистичного аналізу на підставі існуючих даних;
- 4) автоматизоване створення документів на бланках навчального закладу, що забезпечується імпортом з табличного процесора до шаблону відповідного документа: листа, довідки, звіту.

3

Рисунок Е.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	ЕДЕБО	Парус	Тауруна	Деканат	СМІС
Доступ до інформації про здобувачів освіти	Частково	Надає	Надає	Надає	Надає
Періодичність оновлення інформації	Щорічно	За потреби	За потреби	За потреби	За потреби
Складність розгортання та підтримки	Віддалена база	Середня	Висока	Висока	Низька
Збереження архівної інформації	Частково	Не зберігає	Не зберігає	Не зберігає	Зберігає
Платний функціонал	Немає	Платний сервіс	Платний сервіс	Платний сервіс	Безкоштовна

4

Рисунок Е.4 – Слайд 4

Порівняння мов програмування

	C#	C++	Java
Наявність літератури та простота вивчення	++	++	++
Стандартні бібліотеки шаблонів	+	+	+
Кроссплатформність	±	+	++
Мультипарадигмальність	++	++	±
Масштабованість	+	+	+
Паралельне програмування	++	++	+
Простота розроблення GUI	++	±	+

5

Рисунок Е.5 – Слайд 5

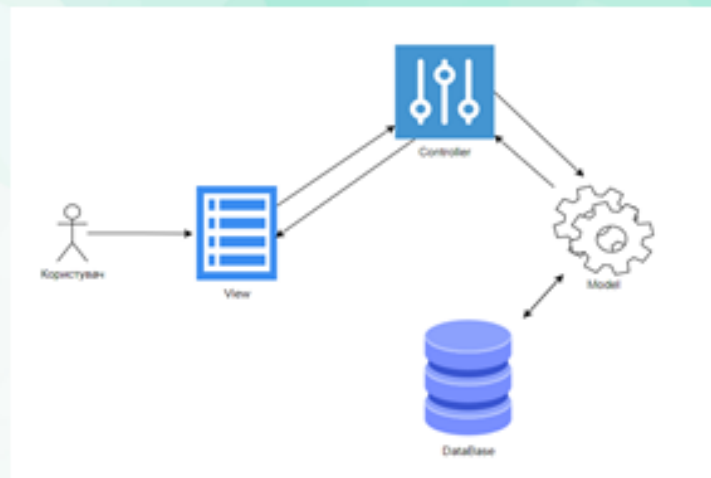
Порівняння середовищ розробки

	Microsoft Visual Studio	Code::Blocks
Кроссплатформність	+	+
Швидкість модернізації	+	±
Наявність генератора сценаріїв складання	+	±
Підтримка репозиторіїв	+	±

6

Рисунок Е.6 – Слайд 6

Архітектура системи «СМІС» на основі предметної області



7

Рисунок Е.7 – Слайд 7

Діаграма прецедентів



8

Рисунок Е.8 – Слайд 8

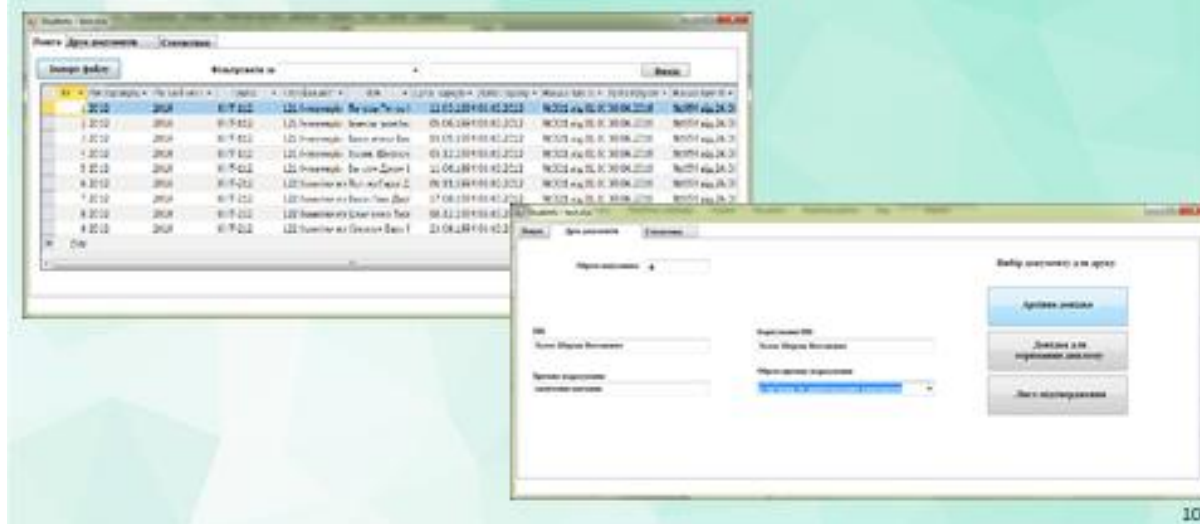
Структурна схема програми



9

Рисунок Е.9 – Слайд 9

Приклади інтерфейсу системи



10

Рисунок Е.10 – Слайд 10

Висновки

В результаті розробки випускної роботи, було розроблено ПЗ з наступними можливостями:

- 1) можливість додавання інформації про студентів, як існуючих, так і відрахованих з навчального закладу з різних причин;
- 2) реалізація пошуку студента за різними критеріями: дата народження, прізвище, спеціальність, та ін.;
- 3) проведення статистичного аналізу на підставі існуючих даних;
- 4) автоматизоване створення документів на бланках навчального закладу, що забезпечується імпортом з табличного процесора до шаблону відповідного документа: листа, довідки, звіту.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції, для моніторингу інформації про студентів закладу вищої освіти.

11

Рисунок Е.11 – Слайд 11

Дякую за увагу

12

Рисунок Е.12 – Слайд 12