

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«9» червня 2024 р.



Півень Н.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
 Кафедра _____ КІТАР _____
 Рівень вищої освіти _____ перший (бакалаврський) _____
 Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології _____
 Тип програми _____ Освітньо-професійна _____
 Освітня програма _____ Системна інженерія _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

« 4 » квітня 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Півню Назарію Павловичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи автоматизації електронного листування з використанням захищених локальних баз даних

Затверджена наказом по університету від 20.05.2024 р. № 478 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024 р.

3. Вихідні дані до роботи _____

3.1 Модель системи автоматизації електронного листування з використанням захищених локальних баз даних.

3.2 Реалізація моделі у вигляді програмного засобу _____.

3.3 Система автоматизації електронного листування _____

3.4 Система управління захищеною локальною базою даних _____

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ _____

4.2 Аналіз предметної області _____

4.3 Аналіз технічного завдання _____

4.4 Розробка концепції системи автоматизації електронного листування _____

4.5 Розробка програмного забезпечення системи автоматизації електронного листування _____

4.6 Охорона праці _____

4.7 Висновки _____

РЕФЕРАТ

Пояснювальна записка: 72 с., 2 табл., 26 рис., 2 дод., 45 джерел

АВТОМАТИЗАЦІЯ, КОМУНІКАЦІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА КЕРУВАННЯ.

Мета роботи – розробка системи автоматизації електронного листування з використанням захищених локальних баз даних, що включає в себе розробку та оптимізацію СУБД та модуля надсилання листів.

Об’єкт розробки – процес електронного листування.

Предмет розробки – система автоматизації електронного листування з використанням захищених локальних баз даних.

Під час виконання кваліфікаційної роботи було проведено аналіз актуальності роботи шляхом дослідження існуючих технологічних рішень та аналогів в сфері автоматизації систем електронного листування.

Було проведено аналіз предметної області, досліджено структуру систем листування та проаналізовано технічне завдання. Було розроблено концепцію програми завдяки використанню таких методологій моделювання роботи системи як IDEF0 і IDEF1, та побудові діаграм: C4, Use-case, Sequence та структурної схеми системи.

Виконано аналіз та обґрунтування вибору технологій розробки. Було розроблено програмне забезпечення, що виконує функції системи автоматизації електронного листування.

ABSTRACT

Explanatory note: 72 p., 2 tabl., 26 pic., 2 applications, 45 sources,

AUTOMATION, COMMUNICATION, SOFTWARE, CONTROL SYSTEM.

The purpose of the work is to develop a system for automating electronic correspondence using secure local databases, which includes the development and optimization of a DBMS and a module for sending letters.

The object of development is the process of electronic correspondence.

The subject of development is a system for automating electronic correspondence using protected local databases.

During the performance of the qualification work, an analysis of the relevance of the work was carried out by researching existing technological solutions and analogues in the field of automation of electronic correspondence systems.

An analysis of the subject area was carried out, the structure of correspondence systems was studied, and the technical task was analyzed. The concept of the program was developed thanks to the use of such methodologies for modeling system operation as IDEF0 and IDEF1, and the construction of diagrams: C4, Use-case, Sequence and structural diagram of the system.

The analysis and justification of the choice of development technologies was carried out. Software has been developed that performs the functions of an email automation system.

ЗМІСТ

Перелік скорочень.....	9
Вступ.....	11
1 Аналіз предметної області	13
1.1 Принцип роботи системи листування	15
1.2 Аналіз технічного завдання	18
1.3 Розгляд готових варіантів, їхніх архітектур, переваг та недоліків	20
1.4 Висновок до розділу 1	24
2 Розробка концепції системи автоматизації електронного листування	26
2.1 Концепція програми для вирішення поставлених вимог	27
2.2 Розробка концептуальної діаграми системи і її декомпозиція	29
2.3 Розробка діаграми відношень між сутностями.....	32
2.4 Розробка контекстної схеми системи.....	34
2.5 Розробка діаграми варіантів використання.....	35
2.6 Розробка діаграми послідовностей.....	36
2.7 Висновки до розділу 2.....	39
3 Розробка програмного забезпечення системи автоматизації електронного листування.....	41
3.1 Аналіз та обґрунтування вибору технологій розробки системи.....	41
3.2 Розробка захищеної локальної СУБД.....	43
3.3 Розробка дизайну користувацької складової системи.....	44
3.4 Висновок до розділу 3.....	55
4 Виведення цільової функції. Обчислення параметрів моделі системи.....	56
5 Охорона праці.....	62
5.1 Важливість організації робочого часу	62
5.2 Аналіз небезпек і шкідливих факторів.....	63
5.3 Заходи з охорони праці.....	64
5.4 Пожежна безпека.....	64

5.5 Висновки до п'ятого розділу.....	65
Висновки.....	66
Перелік джерел посилання.....	68
Додаток А Програмна частина проєкту.....	73
Додаток Б Демонстраційний графічний матеріал.....	86

ПЕРЕЛІК СКОРОЧЕНЬ

- БД – база даних;
- ЕЗ – електронні засоби;
- ЖЦ – життєвий цикл;
- ЗВО – заклад вищої освіти;
- КІТАР – кафедра комп’ютерно-інтегрованих технологій, автоматизації та робототехніки;
- ОС – операційна система;
- ПК – персональний комп’ютер;
- САУ – система автоматичного управління;
- СУБД – середовище управління базами даних;
- ТЗ – технічне завдання;
- ХНУРЕ – Харківський національний університет радіоелектроніки
- API – application programming interface;
- CRM – customer relationship management (система управління відносинами з клієнтами);
- CSV – comma separated values (файловий формат для представлення табличних даних);
- DNS – domain name system (система доменних імен);
- GIF – graphics interchange format;
- HTML – hypertext markup language (стандартизована мова розмітки веб-сторінок);
- IDEF – integrated definition for function modeling;
- IDEF0 – function modeling;
- IDEF1 – information modeling;
- IMAP – internet Mail Access Protocol;
- MX – mail exchanger (запис маршрутизації DNS до електронної пошти);
- POP – post office protocol (стандартний протокол доступу до електронної пошти)

SMTP – simple Mail Transfer Protocol;

SPAM – небажані електронні повідомлення комерційного характеру;

SQL – structured query language – мова програмування для роботи з БД;

UI – user Interface (користувацький інтерфейс);

UML – unified modeling language (уніфікована мова моделювання);

ВСТУП

У сучасному світі важливість автоматизації електронного листування набуває все більшого значення. Різноманітні сервіси та програмні рішення автоматизації процесу надсилання електронних листів активно використовуються підприємствами для підвищення продуктивності роботи шляхом організації робочих процесів завдяки економії часу, який витрачається на обробку повідомлень.

За допомогою використання автоматизованих систем електронного листування підприємства можуть досягти більшої ефективності у веденні комунікації з клієнтами, партнерами, та співробітниками, оскільки вони дозволяють створювати шаблони повідомлень та контролювати терміни надсилання.

Працівники кафедри КІТАР [1] потребують можливості постійного сповіщення своїх студентів та колег про заходи, протягом процесу навчання у ЗВО. Такими заходами можуть бути повідомлення про зміну у навчальному процесі, загальні термінові сповіщення для всіх студентів та викладачів, тощо.

Окрім того працівники ХНУРЕ [2] мають необхідність співпрацювати із іншими закладами вищої освіти (ЗВО) для закликання студентів, сповіщення адміністрації інших університетів про певні заходи, закликання на навчання студентів по обміну, тощо.

Відсутність інструментів, що вирішують вищеперелічені питання, у відкритому доступі посприяла на рішення розробити подібну систему.

Ці системи сприяють збереженню даних в безпеці з режимом доступу за ключем-паролем для авторизованих користувачів і як наслідок відбувається оптимізація комунікаційних процесів підприємства та забезпечується вищий рівень обслуговування клієнтів.

Дана кваліфікаційна робота присвячена розробці програмного забезпечення для системи автоматизації процесу надсилання повідомлень

електронною поштою на основі використання захищеної локальної бази даних SQLite [3] та поєднанню мов програмування С# [4] та Python [5].

Мета роботи – розробка ефективної та гнучкої системи автоматизації електронного листування, яка використовує захищену шифруванням локальну базу даних.

Така система включає в себе розробку та оптимізацію СУБД, а також створення умов керування затримкою на надсилання електронних листів з оригінальним вмістом модулем відправки повідомлень, що дозволить оптимізувати процеси комунікації підприємства.

Об'єктом розробки є автоматизація процесу електронного листування.

Предмет розробки – система автоматизації електронного листування з використанням захищених локальних баз даних.

Дану кваліфікаційну роботу виконано згідно ДСТУ 3008:2015 [6], керуючись навчальним посібником з оформлення кваліфікаційних робіт [7] та методичними вказівками [8].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Системи електронного листування є критично важливими інструментами для комунікації в наші дні, адже вони забезпечують надійний та швидкий обмін інформацією між користувачами.

Така система являє собою програмне забезпечення, що містить в собі користувацький інтерфейс, системи зберігання даних, взаємодію з серверами поштових повідомлень, які використовується для обміну повідомленнями електронною поштою та інші додаткові функції.

Функціонально система електронного листування має забезпечувати:

- надсилання та отримання листів;
- можливість створювати, редагувати та видаляти листи;
- пересилання листів та відповідей на них;
- можливість прикріплювати файли до листів;
- сортування та фільтрація отриманих листів;
- пошук листів за ключовими словами, датою, відправником тощо;
- автоматичне сповіщення про нові листи;
- зберігання та організація контактів;
- імпорт та експорт контактів;
- захист від спаму та вірусів.

Отже, у даній галузі важливими є: простий та зручний користувацький інтерфейс, стабільність системи зберігання даних, взаємодія з серверами поштових повідомлень, інтеграція з іншими сервісами, впровадження безпеки та конфіденційності, покращення алгоритмів захисту від спаму для роботи з електронними листами та базою адресатів.

Поштові клієнти надають користувачеві доступ до керування своїми електронними скриньками, дозволяючи їм надсилати, отримувати та керувати повідомленнями електронної пошти через веб-інтерфейси або за допомогою спеціальних програм для мобільних пристроїв або desktop [9].

На сьогоднішній день відомими є наступні поштові клієнти:

- Gmail – це безкоштовний поштовий сервіс [10], розроблений компанією Google [11]. Швидко набув популярності завдяки великому обсягу безкоштовного місця для зберігання та інтеграції з іншими сервісами Google;
- Outlook – це поштовий клієнт [12] від компанії Microsoft [13], що входить до складу пакету Microsoft Office [14]. Існують дві основні версії: Outlook.com [15] (безкоштовний веб-сервіс) та Microsoft Outlook [16] (десктопний клієнт).
- Yahoo Mail – це безкоштовний поштовий сервіс [17], розроблений компанією Yahoo [18].

У таблиці 1.1 наведено переваги та недоліки цих поштових сервісів.

Таблиця 1.1 – Переваги та недоліки відомих поштових сервісів

Сервіс	Переваги	Недоліки
Gmail	Великий обсяг безкоштовного місця для зберігання.	Реклама в інтерфейсі.
	Швидкий та ефективний пошук.	Монетизацію даних користувачів для рекламних цілей.
	Інтеграція з екосистемою Google.	
Outlook	Інтеграція з іншими продуктами Microsoft.	Деякі функції доступні лише в платній версії.
	Зручний інтерфейс для бізнес-користувачів.	Інколи складні налаштування для новачків.
	Надійний захист та безпека.	
Yahoo	Великий обсяг безкоштовного місця для зберігання.	Менша інтеграція з іншими популярними сервісами.
	Простий та зрозумілий інтерфейс.	Реклама в інтерфейсі.
	Надійний захист від спаму.	

1.1 Принцип роботи системи листування

Етапи життєвого циклу поштового клієнта:

- створення облікового запису;
- компонування вмісту електронного листа;
- відправлення електронного листа;
- пересилання електронного листа сервером електронної пошти;
- прийом електронного листа адресатом;
- керування та збереження електронних листів;
- безпека та захист від спаму.

В свою чергу, нефункціональні вимоги визначають якісні аспекти системи: надійність, безпека, продуктивність та інші. Наприклад:

- надійність: система повинна бути стабільною та надійною, з мінімальними випадками втрати даних або відмов;
- безпека: забезпечення конфіденційності, цілісності та доступності даних користувачів;
- швидкодія: система повинна працювати ефективно та оперативно, з мінімальними затримками у відправці та отриманні листів;
- сумісність: здатність взаємодіяти з іншими електронними поштовими системами та пристроями;
- легкість використання: інтуїтивний та зрозумілий інтерфейс користувача, який дозволяє швидко орієнтуватися та працювати з системою.

Для відправлення листів використовується протокол SMTP [19], що в перекладі з англійської «простий протокол передавання пошти» – комунікаційний протокол для пересилання електронної пошти.

Даний протокол синхронний, складається із серії команд з'єднання, що відбуваються за ініціативи відправника, яким зазвичай є поштовий клієнт або поштовий сервер і відповідей сервера на команди.

Оскільки SMTP є протокол доставки та транспортування - системи, де використовується даний протокол мають бути в активному стані.

Для отримання повідомлень клієнтські програми використовують різні протоколи інформації, такі як: POP [20] і IMAP [21], що є стандартами.

POP – стандартний протокол, що використовується клієнтом для доступу до електронної пошти з віддалених серверів на локальний клієнт. За замовчуванням цей протокол завантажує відкриті користувачем електронні листи та видаляє їх із сервера. Забезпечує також масове надсилання, зберігання та видалення повідомлень.

Internet Message Access Protocol (IMAP) – мережевий протокол, слугує для роботи з вхідними листами, має можливість пошуку за ключовим словом без збереження пошти в локальній пам'яті.

IMAP надає можливості роботи з поштовими скриньками на центральному поштовому сервері. Електронними листами можна керувати з клієнтського комп'ютера без постійного пересилання вмісту листа на сервер.

POP за функціональністю є біднішим за IMAP, не надаючи клієнт-інтерфейсу з маніпулювання папками на сервері, можливості завантаження заголовків листів або вибіркового отримання частин повідомлення.

Отже, принцип процесу роботи електронного листування наступний:

– крок перший – користувач створює обліковий запис в системі листування, вказуючи адресу електронної пошти і пароль. Після авторизації в поштовому клієнті користувач може компонувати повідомлення і обирати електронні адреси адресатів, вводити текст повідомлення і вкладати файли до листа. Натиснувши кнопку «Надіслати» поштовий клієнт обробляє і пересилає лист протоколом SMTP на локальний сервер обміну поштами і очікує на відправлення;

– крок другий – сервер обміну електронної пошти направляє повідомлення адресатам або до серверів їхніх поштових клієнтів. Адреса пошти для отримання електронних листів це рядок localaddress@sampledomain. До знаку «@» - локальна частина адреси, що зазвичай зберігається з іменем

користувача отримувача безпосередньо, а продовженн після знаку є іменем домену. Повідомлення приходять на поштові сервери передавання листів, що використовують доменне ім'я для визначення повного доменного імені сервера обміну листами адресатів.

– крок третій – сервер доменної системи імен (DNS) [22] надсилає у відповідь MX-записи [23], де перелічено сервери обміну пошти цього домену, тобто сервери інтернет-провайдера отримувача.

– крок четвертий – провайдер відправника використовуючи надсилає лист на сервер-провайдер отримувача використовуючи протокол SMTP і кладе його в скриню отримувача.

– крок п'ятий – коли отримувач ввійде в свій обліковий запис – зможе переглянути вміст повідомлення, відповісти чи виконати інші дії.

– крок шостий – електронні листи отримано завдяки протоколу IMAP, закріплено за користувачами, зберігаються вони системами листування, листи можна переміщувати, завантажити, відмітити як прочитані, видалення, тощо. Задіяно механізми безпеки та фільтрації від спаму та небажаних повідомлень.

На рисунку 1.1 зображено послідовність роботи системи листування [24].

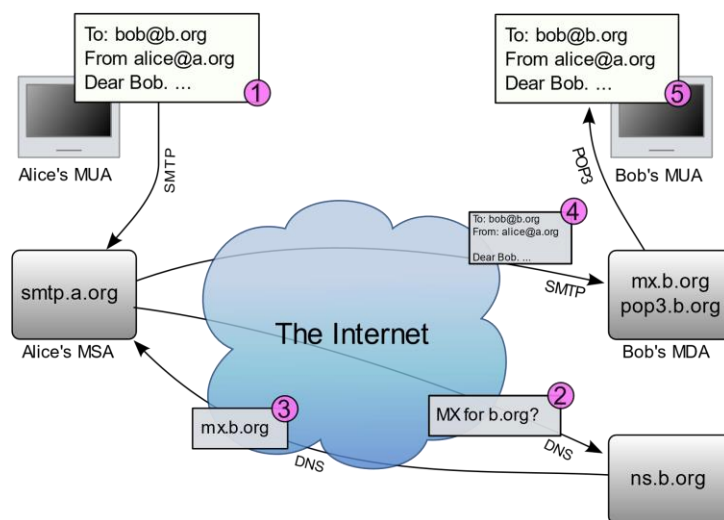


Рисунок 1.1 – Послідовність роботи системи листування

1.2 Аналіз технічного завдання

Розсилка пропозицій співпраці між навчальними закладами та їхніми представниками є важливим інструментом для залучення більшої кількості людей до навчального процесу в Україні. Цей процес має значний вплив на розвиток освіти, суспільства та економіки, оскільки він дозволяє навчальним закладам підвищувати рівень професійної комунікації, завдяки чому, в подальшому, персоналізована інформація щодо навчальних програм, умов вступу та переваг навчання в конкретному закладі стає ще доступнішою. Це в свою чергу сприяє підвищенню обізнаності потенційних студентів і допомагає їм зробити інформований вибір.

Розсилки можуть бути спрямовані на певні групи партнерів, що дозволяє навчальним закладам ефективніше використовувати ресурси для досягнення своїх цільових аудиторій, що важливо в умовах високої конкуренції між навчальними закладами.

Навчальні закладами можуть співпрацювати між собою для проведення спільних заходів, таких як дні відкритих дверей, семінари та конференції. Це дозволяє абітурієнтам дізнатися більше про можливості вищої освіти. Співпраця між навчальними закладами дозволяє обмінюватися ресурсами та досвідом, що сприяє підвищенню якості освіти. Це включає спільні дослідницькі проєкти, обмін викладачами та студентами, а також спільне використання навчальних матеріалів.

Спільні проєкти та розробки допомагають впроваджувати нові технології та методики в навчальний процес, що робить його більш сучасним та ефективним. Це, в свою чергу, сприяє підготовці кваліфікованих спеціалістів, які можуть успішно працювати в різних галузях економіки.

Розсилка пропозицій та інформації про українські навчальні заклади за кордон допомагає залучати іноземних студентів, що не лише збільшує кількість студентів, але й сприяє розвитку культурного обміну та міжнародного співробітництва.

Університети можуть брати участь у міжнародних освітніх програмах та обмінах, що відкриває нові можливості для студентів та викладачів. Це підвищує репутацію українських навчальних закладів на міжнародній арені.

Згідно технічного завдання (ТЗ) необхідно розробити проєкт локального програмного засобу системи автоматизації процесу електронного листування між навчальними закладами та їхніми представниками під назвою «Emailer».

Метою проєкту є покращення стану професійної комунікації навчального закладу шляхом створення гнучкої та інтуїтивно зрозумілої системи, що дозволить користувачеві ефективно та безпечно керувати даними і автоматизувати процес розсилки оригінальних листів з запобіганням потрапляння в спам.

«Emailer» – гнучкий інструмент для автоматизованої розсилки персоналізованих електронних листів навчальним закладам та їхнім представникам, що має забезпечувати такі функціональні вимоги:

- інтуїтивно зрозумілий інтерфейс користувача;
- безпечне зберігання, управління та обробка даних в СУБД;
- сегментація списків отримувачів за критеріями;
- імпорт та експорт списків отримувачів в форматі CSV [25];
- персоналізація листів за допомогою змінних (ім'я, прізвище і тд.);
- форматування листів за допомогою текстового процесору;
- надсилання готових HTML-листів [26];
- завантаження вкладень до листів;
- налаштування затримок надсилання листів за формулами;
- запобігання потраплянню листів в спам.

Нефункціональні вимоги:

- висока надійність та стабільність роботи;
- можливість масштабування системи.

1.3 Розгляд готових варіантів, їхніх архітектур, переваг та недоліків

Автоматизація відправлення електронних листів може бути комплексним завданням, особливо коли йдеться про велику кількість потенційних клієнтів підприємства.

Існуючі засоби автоматизації, такі як Mailchimp [27], Constant Contact [28] та Sendinblue [29], дозволяють розсилати рекламні листи, що збільшує ефективність професійної комунікації.

Ці інструменти надають детальну аналітику про відкриття листів, клікабельність посилань та інші метрики, що дозволяють оцінити ефективність рекламних компаній та вдосконалювати їх у майбутньому.

Засоби автоматизації дозволяють значно спростити процеси створення, розсилки та відстеження рекламних листів, що зменшує необхідність в ручній роботі та збільшує ефективність.

Багато з них надають можливості для сегментації аудиторії за різними параметрами, такими як інтереси, поведінка або демографічні характеристики, що дозволяє надсилати спеціалізовані пропозиції різним групам клієнтів.

Деякі засоби автоматизації можуть інтегруватися з іншими системами маркетингу, CRM [30] або аналітики, що дозволяє більш ефективно використовувати дані та координувати різні маркетингові ініціативи, що допомагає підприємству ефективно взаємодіяти зі своїми клієнтами.

Отже, популярні інструменти для масового розсилання пропонують різноманітні функції для створення, надсилання та аналізу електронних листів, що допомагає компаніям ефективно керувати своїми маркетинговими кампаніями.

Розглянемо наявні автоматизовані засоби відправки електронних листів більш детально:

- Mailchimp – один з найпопулярніших сервісів для автоматизації електронної пошти, що пропонує широкий спектр інструментів для створення, надсилання та аналізу кампаній електронної пошти. Цей сервіс використовує розподілену архітектуру для своєї системи, їхні сервери та сервіси розташовані на різних вузлах, що забезпечує масштабованість, надійність та ефективність. Завдяки використанню мікросервісної архітектури, де різні компоненти програмного забезпечення працюють як незалежні мікросервіси, що пов'язані одним API, реакція на зміни залишається швидкою та легко впровадити новий функціонал.

Переваги:

- інтуїтивний інтерфейс: Mailchimp пропонує зручний інтерфейс, який дозволяє швидко створювати та налаштовувати розсилки;
- безкоштовний план: Для початківців є безкоштовний план з обмеженнями, що дозволяє використовувати основні функції;
- автоматизація: Mailchimp надає можливості автоматизованої розсилки, включаючи привітання нових підписників та нагадування.

Недоліки:

- обмеження на безкоштовному плані: обмеження на кількість підписників та листів, що можуть бути відправлені щомісяця;
- складніші налаштування: Для складних автоматизованих кампаній може знадобитися додаткове вивчення документації.

- Constant Contact – сервіс, орієнтований на малі та середні підприємства, який пропонує прості у використанні інструменти для електронного маркетингу. Даний сервіс також використовує розподілену архітектуру для своєї системи. Його сервери та сервіси розташовані на різних вузлах, що дозволяє їм забезпечувати масштабованість та надійність. Також використовується мікросервісна архітектуру та інші інструменти

децентралізації для забезпечення ефективності та масштабованості своїх систем.

Переваги:

- легкість використання: Constant Contact відомий своїм інтуїтивним інтерфейсом, що робить його привабливим для початківців;
- шаблони для створення: Платформа пропонує широкий вибір шаблонів для створення різноманітних листів;
- підтримка: Constant Contact надає хорошу підтримку користувачам, включаючи онлайн-ресурси та живу підтримку;

Недоліки:

- обмежені функції безкоштовного плану: Безкоштовний план Constant Contact має обмежені можливості порівняно з платними планами;
- вищі ціни: Деякі користувачі можуть вважати, що ціни на Constant Contact є дещо вищими порівняно з іншими платформами.
- Brevo (Sendinblue) – потужний сервіс для автоматизації електронного маркетингу з розширеними можливостями для сегментації та персоналізації. Цей сервіс також використовує розподілену та мікросервісну архітектуру в своїй системі, для забезпечення ефективності та гнучкості своєї платформи.

Переваги:

- потужність автоматизації: Sendinblue має широкі можливості автоматизації, включаючи сегментацію аудиторії та відправку спеціальних пропозицій;
- доступна ціна: ціни на Sendinblue є конкурентоспроможними, особливо для бізнесів з великим обсягом розсилок;
- простий API: Sendinblue надає зручний доступ до інтеграцій в власних цілях.

Недоліки:

- інтерфейс: може бути складнішим для новачків;
- обмежена підтримка: підтримка може бути менш оперативною.

– SendGrid [31] – платформа для електронного маркетингу та транзакційних листів, що пропонує масштабовані рішення для відправки електронної пошти.

Переваги:

- масштабованість: працює з великими об'ємами даних;
- потужні інструменти для інтеграції через API;
- детальна аналітика та звіти;
- висока доставка листів завдяки хорошій репутації відправника.

Недоліки:

- складність для новачків;
- платні плани з обмеженнями.

В таблиці 1.2 наведено порівняння популярних сервісів розсилання електронних листів.

Таблиця 1.2 – Порівняння популярних сервісів розсилки листів

Сервіс	Безкоштовний план	Ціни в (\$/міс.)	Переваги	Недоліки
Mailchimp	До 500 контактів	13	Інтуїтивний інтерфейс	Висока вартість для великих баз
			Шаблони	
Constant Contact	60 днів пробного періоду	10	Простота використання	Дизайн шаблонів
			Підтримка	
Brevo	До 300 листів на день	25	Ціноутворення,	Складний інтерфейс
			Автоматизація	
SendGrid	До 100 листів на день	15	Масштабованість	Складність
			API	Обмежена підтримка
			Аналітика	

1.4 Висновок до розділу 1

У даному розділі було проаналізовано предметну область системи електронного листування.

Було розібрано основні поняття, актуальність розробки, принципи роботи, технічне завдання, структури подібних систем електронного листування, виведено їхню порівняльну характеристику.

Системи електронного листування – це програмні засоби, які дозволяють створювати, відправляти та керувати електронними листами з метою підвищення рівня комунікації, підтримки клієнтів, а також для транзакційних повідомлень.

З розвитком цифрових технологій та зростанням конкуренції в онлайн-просторі, ефективна комунікація з клієнтами стала ключовою складовою успіху підприємства.

Для того аби підвищити залученість клієнтів, ефективніше керувати ресурсами та поліпшувати маркетингову стратегію підприємства необхідно розробити систему автоматизації електронного листування, яка задовольнятиме вимогам ТЗ.

Згідно вимогам ТЗ, вибір припадає на створення власної автоматизованої системи електронного листування для задоволення потреб підприємства.

Було сформовано завдання для виконання кваліфікаційної роботи:

- провести аналіз матеріальної бази, технічного та методичного забезпечення бази для підготовки кваліфікаційної роботи;
- провести аналіз структури систем листування;
- визначитися з метою, об'єктом і предметом розробки; – ознайомитися з основними публікаціями, нормативними, довідковими матеріалами за темою роботи;
- провести аналіз існуючих аналогів, аналіз предметної області;
- розробити контекстну, функціональну і структурну схему системи;

- обрати технології для системи з обґрунтуванням;
- розробити користувальницьку складову системи листування;
- розробити програмний код модулю листування;
- розробити захищену локальну базу даних для системи листування.

Така система слугуватиме оптимальним рішенням завдяки тому, що вона безкоштовна, за своєю суттю працює як система автоматизованого управління (САУ), не потрапляє в спам, захищена і на довгу перспективу масштабована.

2 РОЗРОБКА КОНЦЕПЦІЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ЕЛЕКТРОННОГО ЛИСТУВАННЯ

Розробка концепції програми є ключовим етапом у процесі створення програмного забезпечення, адже вона закладає основу для подальшого проектування та реалізації системи.

Використання різних методологій та діаграм дозволяє всебічно охопити всі аспекти системи.

Методології IDEF0 та IDEF1 [32] забезпечують чітке розуміння функціональних процесів і структур даних. IDEF0 моделює функції системи та їхні взаємозв'язки, тоді як IDEF1 фокусується на інформаційних структурах.

Діаграми C4 (Context, Container, Component, Code) [33] допомагають візуалізувати архітектуру системи на різних рівнях деталізації, починаючи з контексту і закінчуючи конкретними компонентами та їх взаємодіями.

Use-case-діаграми [34] демонструють взаємодію користувачів із системою, висвітлюючи всі можливі сценарії використання.

Sequence-діаграми [35], у свою чергу, деталізують послідовність обміну повідомленнями між об'єктами системи у конкретних сценаріях, що дозволяє зрозуміти динаміку роботи системи.

Синтез цих підходів забезпечує комплексний і багаторівневий аналіз, який є критичним для успішної розробки програмної концепції.

Проаналізувавши предметну область було визначено основні процеси системи листування і стало можливим розробити власну концепцію програми для досягнення вимог ТЗ.

2.1 Концепція програми для вирішення поставлених вимог

Відмінні особливості системи електронного листування з використанням захищених локальних баз даних.

Дана інформаційна система створюється під задачу розсилання повідомлень навчальним закладам та їхнім представникам шляхом використання оригінальних та повністю керованих рекламних листів без потрапляння в спам.

На рисунку 2.1 зображено структурну схему модулів системи автоматизації електронного листування.

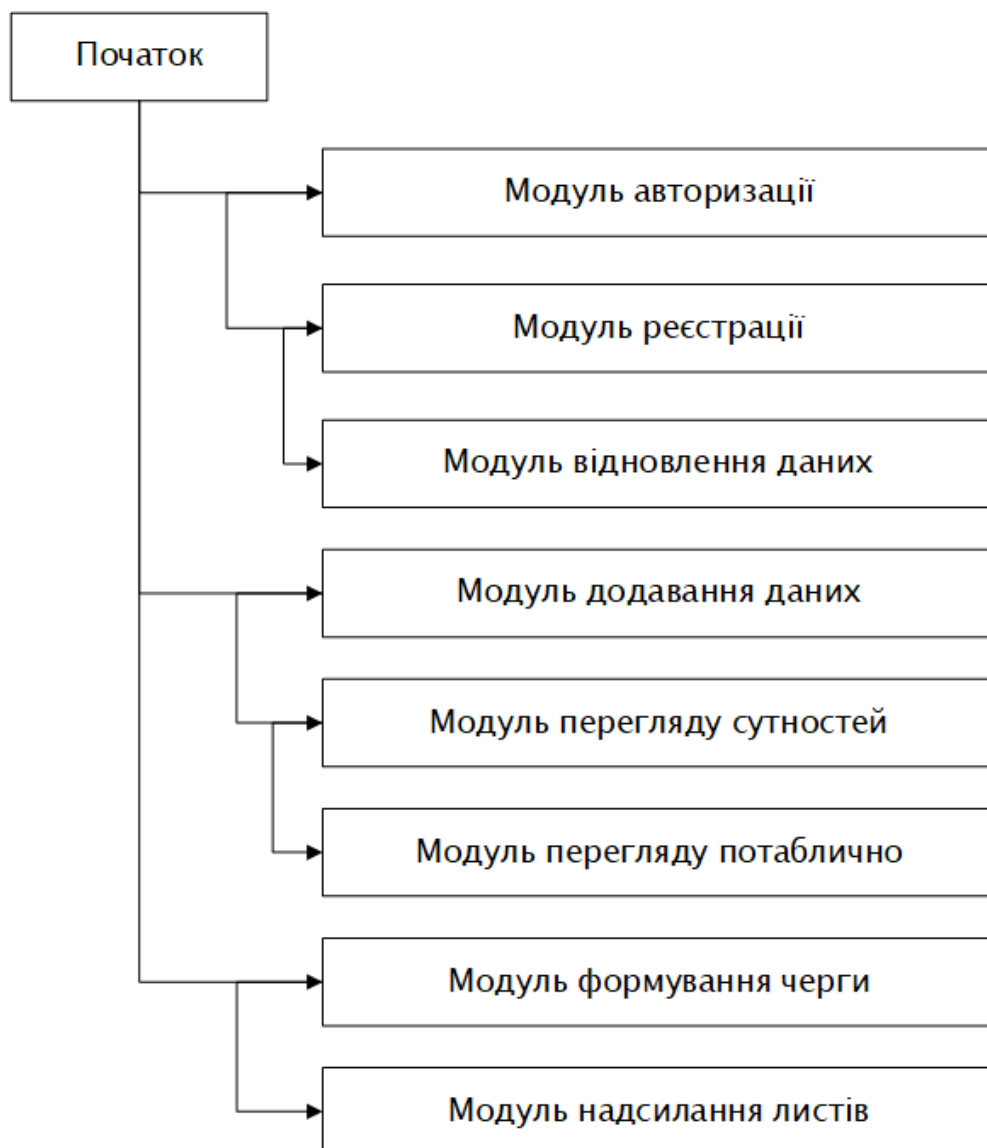


Рисунок 2.1 – Структурна схема модулів системи автоматизації

Ця система має складатися з установника [36] для операційної системи Windows [37] , що встановлює виконуваний файл для ОС Windows, який містить два модуля: СУБД і модуль надсилання електронних листів поштовим клієнтом.

Установником є виконуваний файл, що дозволяє компактно розподілити ресурси системи в файлових директоріях користувача.

Захищена локальна база даних має бути ефективним інструментом для збереження та обробки даних програми користувачем.

Модуль надсилання листів, містить в собі набір для роботи з вмістом електронного листа, керування чергою адресатів та алгоритм керування затримкою на надсилання електронних листів.

Робота з вмістом електронного листа керується текстовим процесором для перетворення тексту на HTML-лист з використанням тегів, що в свою чергу працюють з обов'язковими комірками бази даних.

Алгоритм керування затримками на надсилання листів буде реалізовано шляхом використання формул, які вносить особисто користувач.

Авторизація в якій реалізовано як мікросервіс, дозволяє гнучко масштабувати модуль окремо від інших компонентів системи.

СУБД являє собою програмний засіб написаний мовою програмування C# для повного керування захищеною базою даних SQLite, а саме: додавання, редагування, видалення та перегляд інформації потаблично та присуттєво про навчальні заклади та їхніх представників.

Програмний засіб має містити в собі логування, перевірку на цілісність даних, функції експорту наявної версії та імпорту попередніх версій баз даних для адміністрування.

Необхідно реалізувати створення файлу CSV-формату для формування подальшої черги адресатів з бази даних для поштового клієнта.

Авторизація має відбуватися за допомогою електронної пошти та зформованого аккаунтом gmail.com паролем додатку для прямої авторизації через програмний код.

Алгоритм формування черги працюватиме завдяки тому, що обрана черга отримувачів листів з СУБД прораховується поадресно для формування затримки черги.

Текстовий процесор має дозволяти працювати безпосередньо з вмістом листа, створювати оригінальні листи з використанням персоналізованих імен адресатів, додавати файли до 25 мегабайт та більше до листів і також обробляти завантаження зображень та GIF-анімацій з буферу обміну в трансформовані або готові HTML-листи.

Алгоритм надсилання листів є набором редагованих адміністратором параметрів як: черга адресатів для листів, встановлення періодичної затримки на відправку листів, прикріплення файлів, підтвердження надсилання тестових електронних листів, підтвердження надсилання оригінальних електронних листів, зупинка та продовження надсилання і запобігання потраплянню в спам.

Розроблення стабільного проекту системи автоматизації процесів комунікації підприємства як високоорганізованої, гнучкої системи можна подати як типовий процес, який починається з визначення головної специфічної мети функціонування системи та передбачає її типову диспозицію та ринкову межу.

2.2 Розробка концептуальної діаграми системи і її декомпозиція

IDEF (Integration Definition for Function Modeling) – це методологія для моделювання та аналізу функціональних аспектів систем.

IDEF0 та IDEF1 важливі у розробці концепції, оскільки вони дозволяють чітко візуалізувати та аналізувати функції системи та інформаційні потоки, що спрощує виявлення взаємозв'язків, оптимізацію процесів та прийняття обґрунтованих рішень, що особливо критично на етапі концептуального проектування складних систем.

IDEF0 застосовується для проектування, аналізу та документування складних систем та процесів, допомагаючи виявити функціональні вимоги та покращити управління процесами.

Основний процес – надсилати електронний лист.

Було розроблено контекстну IDEF0 діаграму A-0 для системи електронного листування, що зображена на рисунку 2.2.

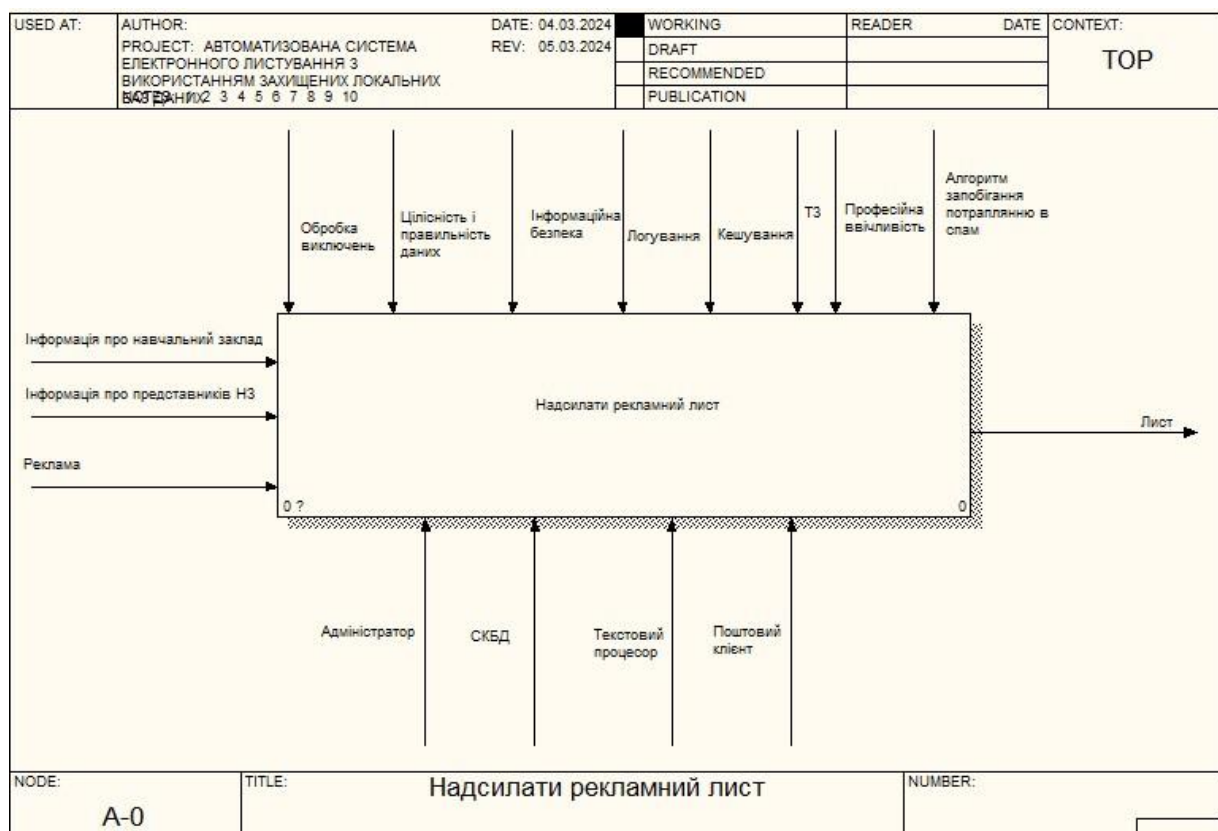


Рисунок 2.2 – Контекстна діаграма системи електронного листування

Декомпозиція складних систем є важливим інструментом для досягнення більшої ефективності, гнучкості та надійності програмного забезпечення, що відповідає сучасним вимогам до розробки і підтримки інформаційних систем.

IDEF1 використовується для розробки баз даних, проектування інформаційних систем та забезпечення цілісності та узгодженості даних.

Основною метою є моделювання інформаційних структур та даних з упором на опис даних, їхньої структури та взаємозв'язків. IDEF1 декомпозує IDEF0 на сутності, атрибути та відносини між ними.

Декомпозицію основного процесу на три інші процеси, зображено на рисунку 2.3.

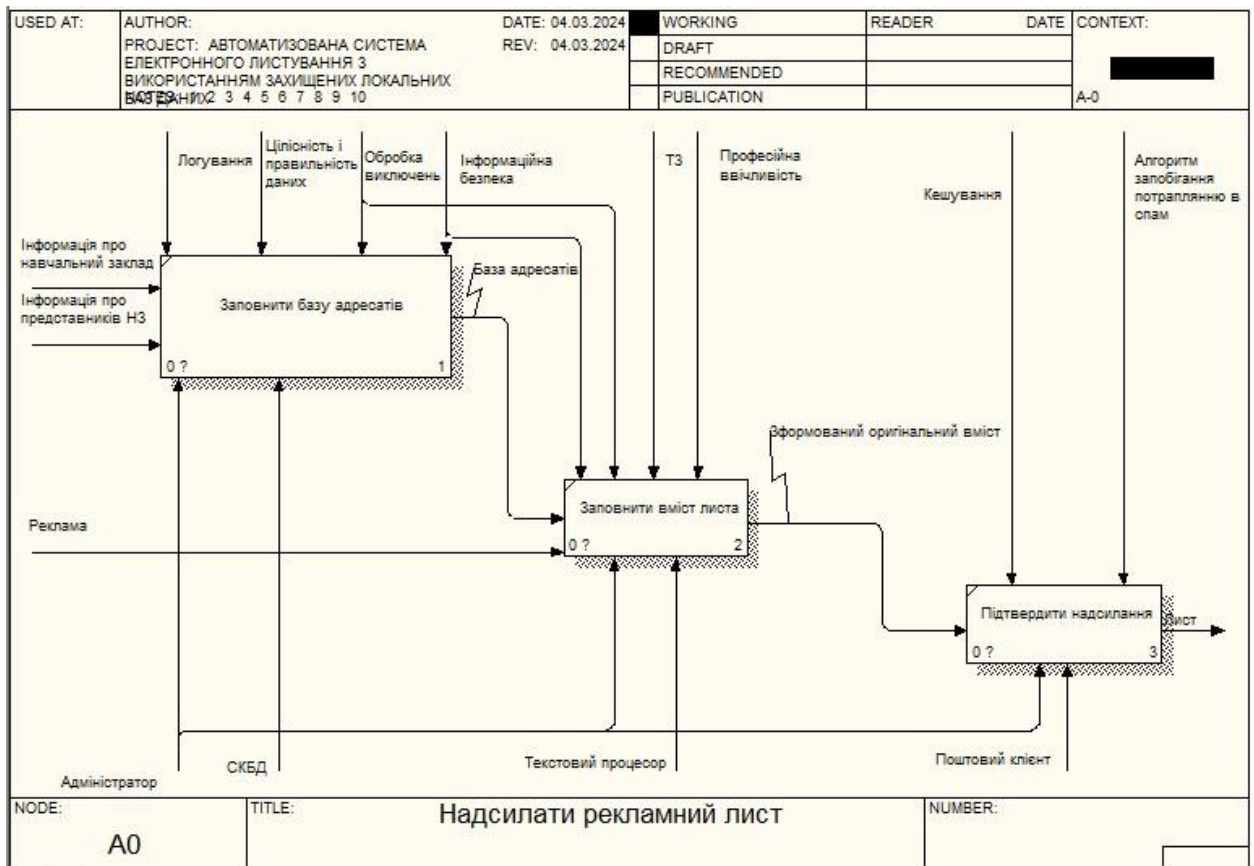


Рисунок 2.3 – Декомпозована модель основного процесу

IDEF1 відрізняється від IDEF0 своєю спрямованістю на моделювання даних, а не функцій.

2.3 Розробка діаграми відношень між сутностями

Проаналізувавши вхідні дані IDEF1 та технічне завдання було виділено чотири основні сутності для коректного виконання завдань проєкту. Такі характеристики сутностей стали основою для структури бази даних.

Було розроблено модель відношення даних для роботи системи як зображено на рисунку 2.4.

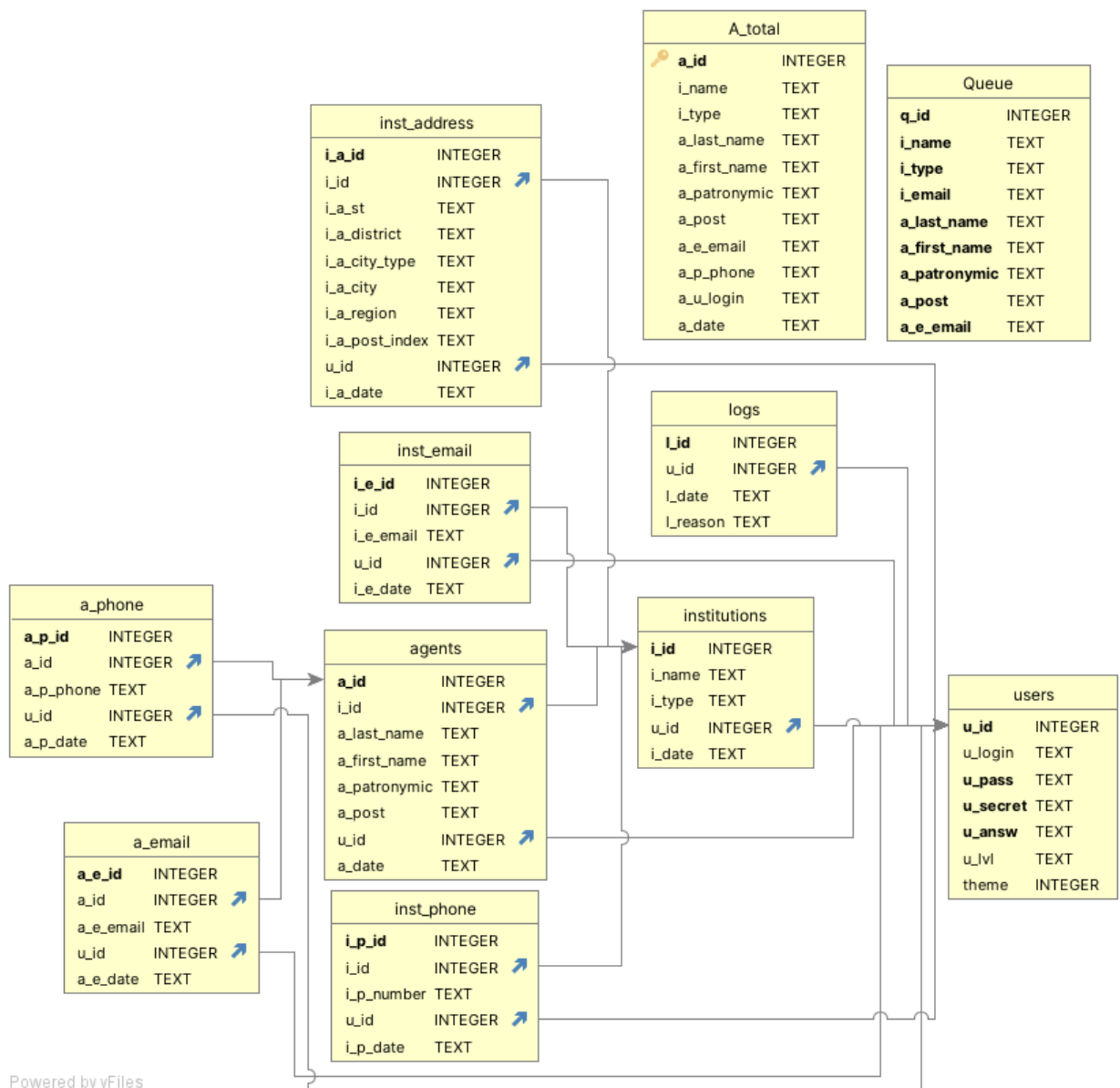


Рисунок 2.4 – Модель відношень між описаними сутностями

Розглянемо чотири сутності: адміністратор, навчальний заклад і представники навчального закладу.

Сутність навчальний заклад має наступні властивості: назва, тип закладу, вулиця, район, тип міста, місто, область, поштовий індекс, електронна пошта, номер телефону.

У навчального закладу може бути декілька електронних адрес та номерів телефонів.

Сутність представників навчального закладу має наступні властивості: прізвище, ім'я, по-батькові, посада, навчальний заклад, електронна пошта, номер телефону.

У представників навчального закладу може бути декілька адрес і номерів телефону.

Для адміністрування системи було створено сутність адміністратор, що містить в собі: логін, пароль, питання для відновлення, відповідь на питання для відновлення, рівень доступу.

Для формування черги було створено таблицю-буфер для тимчасового збереження даних черги, що являє собою дані отримувача, обов'язкові до заповнення.

В подальшому, для масштабування безпеки проєкту, було додано таблицю логуювання, що містить в собі дату та подію.

Такі складові моделі даних дозволяють коректно працювати в ситуації, коли необхідно повертати декілька електронних адрес або номерів телефонів одержувачів під час формування черги за критеріями, обраними користувачем.

2.4 Розробка контекстної схеми системи

Будь-яка система працює у певному контексті — оточенні. Насамперед, це користувачі та інші системи. Користувачі можуть мати різні ролі, такі як автор контенту, читач, адміністратор. Також вони можуть бути внутрішніми чи зовнішніми. Інші програми можуть бути джерелом даних для системи або отримувати інформацію від неї.

Важливо розуміти цей контекст, щоб правильно спроектувати систему та явно виразити необхідність інтеграції із зовнішніми системами.

Контекстна діаграма C4 – це набір ієрархічних абстракцій системи: програмні системи, контейнери, компоненти та код.

На рисунку 2.5 зображено контекстну діаграму C4 системи листування.

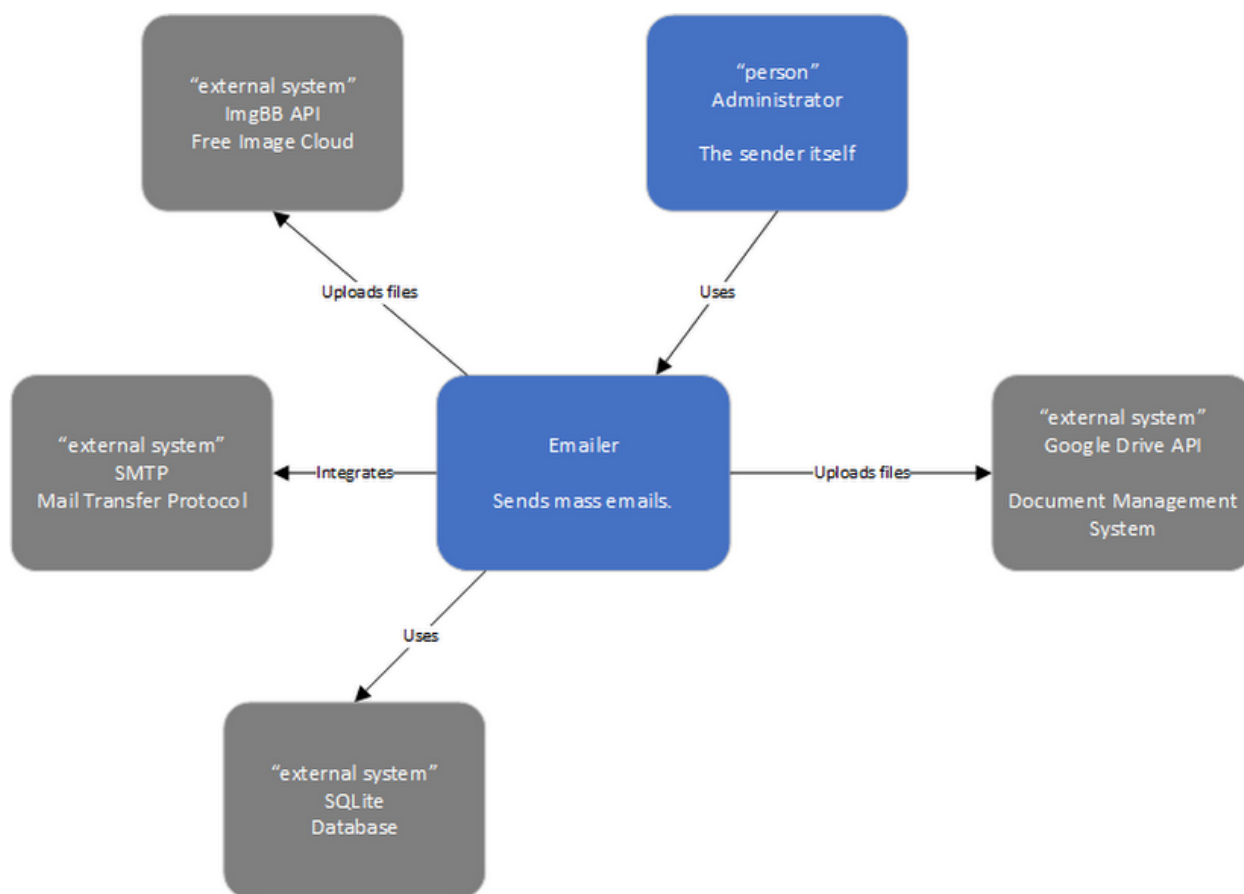


Рисунок 2.5 – Контекстна діаграма системи

2.5 Розробка діаграми варіантів використання

Діаграма варіантів використання – діаграма з класу UML більш спрямована на моделювання бізнес-процесів, оскільки вона показує взаємодію між елементами системи високорівнево. Самі варіанти використання прийнято розглядати як бізнес-можливості.

На рисунку 2.6 зображено діаграму використання системи з урахуванням контексту та вимог технічного завдання.

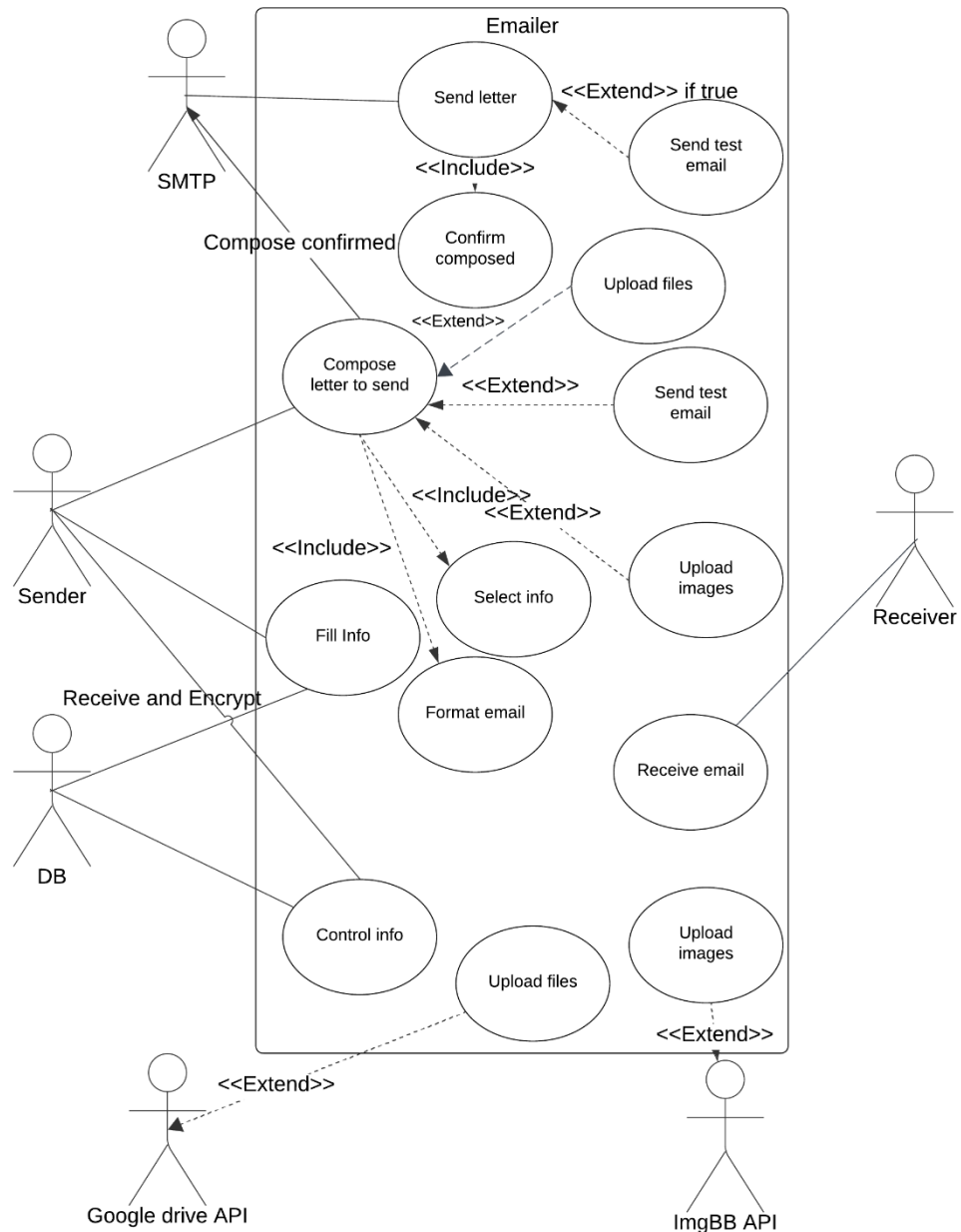


Рисунок 2.6 – Діаграма варіантів використання системи

2.6 Розробка діаграми послідовностей

Діаграма послідовностей показує як елементи системи співвідносяться один до одного. Ця діаграма демонструє, які процеси відбуваються в системі та яким чином відбувається взаємодія

На рисунку 2.7 зображено діаграму послідовностей роботи системи.

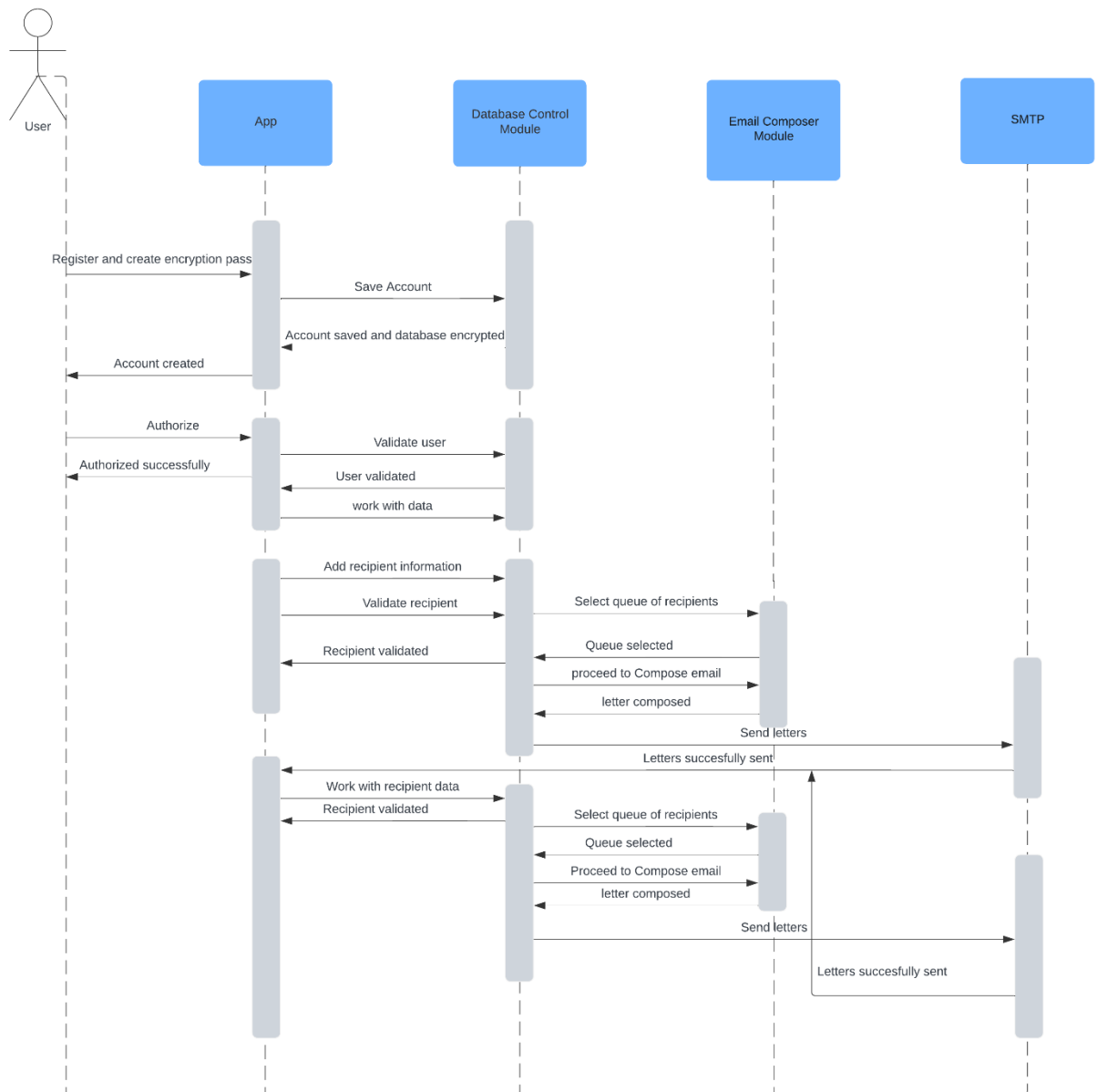


Рисунок 2.7 – Діаграма послідовностей

Вгорі ми бачимо взаємодіючі об'єкти: користувачі, окремі модулі, зовнішні системи, сервіси та сховища даних. Кожен об'єкт має вертикальну лінію внизу. Взаємодія між сервісами є горизонтальними стрілками між вертикальними лініями.

Ці стрілки можуть бути різних типів залежно від того, чи це синхронна операція, чи асинхронна. Сірі прямокутники показують, що процес займає деякий час, і довжина повинна вказувати на тривалість: чим довше прямокутник, тим більше часу.

Детальний опис:

- створення облікового запису і шифрування бази даних за допомогою пароля: користувач реєструє обліковий запис, програма зберігає дані облікового запису в базі даних, додаток шифрує пароль користувача. Додаток підтверджує успішне створення облікового запису;
- авторизація користувача: користувач намагається авторизуватися, програма перевіряє облікові дані користувача в базі даних, додаток підтверджує успішну авторизацію користувача;
- після успішної авторизації користувач: додає інформацію про одержувачів: програма зберігає інформацію про одержувача в модулі керування базою даних, програма підтверджує, що інформацію про одержувача додано;
- якщо необхідно змінити інформацію про одержувача: за потреби користувач змінює інформацію про одержувача, програма оновлює інформацію про одержувача в модулі керування базою даних, додаток підтверджує, що інформацію про одержувача оновлено;
- створення черги електронних листів: користувач створює чергу електронної пошти, програма отримує необхідні параметри з модуля керування базою даних;
- програма підтверджує створення черги електронної пошти: користувач підтверджує чергу електронних пошт в модулі керування базою даних;

- модуль керування базою даних підтверджує чергу адресатів, програма повідомляє користувача про підтвердження черги, користувач переходить до модуля створення електронного листа;
- модуль формування вмісту електронних листів отримує підтвердження надсилання, надсилає електронні листи через протокол SMTP до сервісу електронної пошти;
- сервіс електронної пошти надсилає електронні листи, по завершенню виводить повідомлення про завершення надсилання електронних листів.

Ця діаграма послідовності тепер точно відображає взаємодію зі службою електронної пошти для надсилання листів після підтвердження черги в модулі керування базою даних і проходження через модуль створення електронної пошти.

Мікросервіс – архітектурний стиль розробки програмного забезпечення, в якому додаток складається з невеликих часток, що функціонують як окремі модулі.

Розбиття складних систем на частки є ключовою стратегією в сучасному програмуванні, особливо в сфері електронного листування, коли використовується мікросервісна архітектура. Такий підхід дозволяє розбити складні задачі на простіші, що полегшує роботу та підтримку окремих модулів програмного забезпечення.

Мікросервісна архітектура зосереджена на створенні окремих сервісів, що функціонують незалежно один від одного, поки виконують свої завдання. Однією з валивих переваг підходу розбиття складної системи на модулі є збільшення масштабованості системи, що дозволяє підтримати високу надійність та продуктивність при підвищенні даних.

Залучення мікросервісів сприяє розширенню та впровадженню інновацій як відповідь на постійне оновлення бізнес-логіки програм

2.7 Висновки до розділу 2

Ключовим етапом у процесі створення програмного забезпечення є розробка концепції програми, так як вона закладає основу для подальшого проєктування та реалізації системи. Використання різних методологій моделювання процесів та діаграм дозволяє всебічно охопити всі аспекти системи, забезпечуючи комплексний підхід до аналізу та проєктування складних систем.

IDEF0 та IDEF1 відіграють важливу роль у створенні чіткого розуміння функціональних процесів і структур даних.

IDEF0 моделює функції системи та їхні взаємозв'язки, дозволяючи виявити ключові елементи та взаємодії між ними, що забезпечує загальний огляд системи та дозволяє виявити можливі точки покращення. IDEF1, фокусуючись на інформаційних структурах, забезпечує детальне розуміння даних, які використовуються в системі, що є критичним для побудови надійної та ефективної бази даних.

Методологія C4 (Context, Container, Component, Code) допомагає візуалізувати архітектуру системи на різних рівнях деталізації. Від контексту, який показує загальну картину системи та її взаємодію із зовнішніми елементами, до конкретних компонентів та їх взаємодій, ці діаграми надають всебічний огляд структури системи. Це дозволяє не лише зрозуміти, як система функціонує, але й ідентифікувати можливі проблемні зони та оптимізувати архітектуру.

Use-case-діаграми демонструють взаємодію користувачів із системою, висвітлюючи всі можливі сценарії використання. Для визначення вимог користувачів та забезпечення, щоб система відповідності потребам користувачів використовується Use-case діаграма. Ця діаграма демонструє взаємодію користувачів із системою Вони також допомагають виявити всі можливі шляхи взаємодії з системою, що дозволяє більш точно проєктувати інтерфейси та функціональні можливості.

Діаграми послідовностей відображають послідовність обміну повідомленнями між об'єктами системи у конкретних сценаріях, що дозволяє зрозуміти динаміку роботи системи, виявити можливі проблеми з продуктивністю та забезпечити, що всі компоненти працюють гармонійно. Послідовність дій є критичною для оптимізації процесів та забезпечення безперебійної роботи системи.

Загалом, ці підходи забезпечують комплексний і багаторівневий аналіз, який є критичним для успішної розробки програмної концепції. Проаналізувавши предметну область, було визначено основні процеси системи листування, що стало основою для розробки власної концепції програми. Дана концепція дозволяє забезпечити відповідність вимогам ТЗ та розпочати проєктування реалізації системи.

Таким чином, використання різних методологій та діаграм для розробки концепції програми дозволяє створити детальне та комплексне розуміння системи, що є критичним для досягнення успішного результату, бо кожен з підходів додає свій внесок у загальну картину, забезпечуючи всебічний аналіз та оптимізацію процесів, що в кінцевому результаті сприяє створенню ефективного, надійного та функціонального програмного забезпечення.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ЕЛЕКТРОННОГО ЛИСТУВАННЯ

3.1 Аналіз та обґрунтування вибору технологій розробки системи

Для вирішення даної задачі було використано Microsoft Visual Studio 2022 [38], а саме – підрозділ C# Windows Forms [39]. Завдяки легкості у застосуванні інтерфейсу для роботи з базою адресатів та вмістом листів. Щоб ця програма працювала коректно зберігаючи дані авторизуватися знадобиться середовище роботи з SQL-базами даних [40] – SQLite і його інтеграція в проєкт СУБД.

Основними вимогами до програмного продукту є інтуїтивно зрозумілий інтерфейс, який відображає загально-стильову концепцію і дозволяє без зайвих зусиль заповнювати базу адресатів та вмістом електронних листів.

Інформаційна система складається з СУБД та модуля надсилання листів.

СУБД зберігає інформація про сутності

Для СУБД було обрано технологію розробки C# Windows Forms, бо C# є статично типізована мова програмування, що дозволяє їй перевіряти типи даних в методах та звести до мінімуму помилки, пов'язані із невірним типом даних.

Для модуля відправлення повідомлень було обрано Python для інтеграції його в інтерфейс користувача Windows Forms. Ці мови програмування постійно розвиваються, що дозволяє зручно розробляти не лише код застосунку, а й інтерфейс.

Для обраних технологій найзручнішим середовищем розробки є Visual Studio Community 2022, завдяки своїй ергономічності.

Так як, обраною мовою програмування для розробки інформаційної системи є C# і Python версії 3.8.0 – було вирішено в якості сховища даних використовувати СУБД SQLite, що є компактною бібліотекою, написаною на мові програмування C [41], що дозволяє їй реалізовувати надійний двигун реляційних баз даних SQL. При всіх включених функціях розмір бібліотеки може бути менше 600 КБ. Важливою відмінністю від інших рішень є її автономність: для роботи SQLite не потрібна наявність сторонніх бібліотек або служб.

СУБД легко інтегрується з різним програмним забезпеченням, підтримує достатній набір команд SQL і транзакцій. На відміну від клієнт-серверних баз даних, SQLite призначена для локального зберігання даних окремих додатків і пристроїв.

SQLite – однофайлове СУБД. Вся база, включаючи визначення, таблиці, індекси і дані, зберігається в одному файлі на комп'ютері або пристрої, де виконується програма. Формат файлу бази даних є кросплатформним, тобто його можна копіювати між різними архітектурами, 32-розрядними і 64-розрядними системами.

Python – це динамічно типізована мова програмування, що означає, що типи даних визначаються автоматично на етапі виконання програми. У Python немає необхідності явно вказувати типи даних при оголошенні змінних або параметрів функцій. Python задовольняє потреби системи в роботі з SMTP, дуже просто інтегрується в C#, що дозволяє значно простіше розробити інтерфейс користувача.

3.2 Розробка захищеної локальної СУБД

Базу даних було створено у середовищі розробки БД SQLite. Було використано браузер для SQLite баз даних під назвою «DB browser SQLite». Розроблену базу даних вважається захищеною через те, що і файл і запити є шифрованими за допомогою використання AES-256 [42].

AES-256 – реалізація одного з найпотужніших алгоритмів шифрування. Дана технологія використовує блоковий шифр із ключем довжиною 256 біт, що забезпечує високий рівень безпеки для зашифрованих даних.

База даних, розроблена протягом цього розділу, підключена до СУБД та модулю листування за допомогою використання модальних класів, що відображають характеристики сутності, так само як зображено на рисунку 2.4, що відображає відношення між сутностями.

Приклад реалізації характеристики сутності навчального закладу в вигляді таблиці бази даних зображено на рисунку 3.1.

	i_id	i_name	i_type	u_id	i_date
	Філ...	Фільтр	Фільтр	Філ...	Фільтр
1	1	ХНУРЕ	Університет	1	24.06.2024 20:44:03
2	2	ХПКК	Коледж	1	24.06.2024 21:36:43
3	3	ХАДІ	Університет	1	24.06.2024 21:37:10
4	4	ХПІ	Інститут	1	24.06.2024 20:49:53

Рисунок 3.1 – Таблиця в базі даних

Приклад реалізації модального класу для функціоналу в мові програмування С# зображено на рисунку 3.2.

```

public class Logs
{
    private int l_id;
    private int u_id;
    private string l_date;
    private string l_reason;

    Свойство 0
    public int L_id
    {
        get { return l_id; }
    }

    Свойство 0
    public int U_id
    {
        get { return u_id; }
        set { u_id = value; }
    }

    Свойство 0
    public string L_date
    {
        get { return l_date; }
        set { l_date = value; }
    }

    Свойство 0
    public string L_reason
    {
        get { return l_reason; }
        set { l_reason = value; }
    }

    Свойство 0
    public Logs()
    {
        l_id = 0;
        u_id = 0;
        l_date = " ";
        l_reason = " ";
    }

    Свойство 0
    public Logs (string info)
    {
        info = info.Trim();
        if (info.Length > 2)
        {
            string[] val = info.Split('|');
            l_id = Convert.ToInt32(val[0]);
            u_id = Convert.ToInt32(val[1]);
            l_date=val[2];
            l_reason=val[3];
        }
    }
}

```

Рисунок 3.2 – Програмний код модального класу сутності Логування

3.3 Розробка дизайну користувацької складової системи

Для забезпечення повного доступу до інформації для користувача було розроблено програмний код користувацької складової системи електронного листування в середовищі розробки Visual Studio, використовуючи Windows Forms С#. Були розміщені основні користувацькі елементи для авторизації, реєстрації, відновлення вхідних даних, додавання, редагування, опрацювання даних для формування черги адресатів.

Це в свою чергу значно поліпшує користувацький досвід та дозволяє користувачеві повністю налаштувати програму під потреби розсилання листів.

Користувальницький інтерфейс для вікна авторизації було розроблено з використанням простого дизайну як зображено на рисунку 3.3.

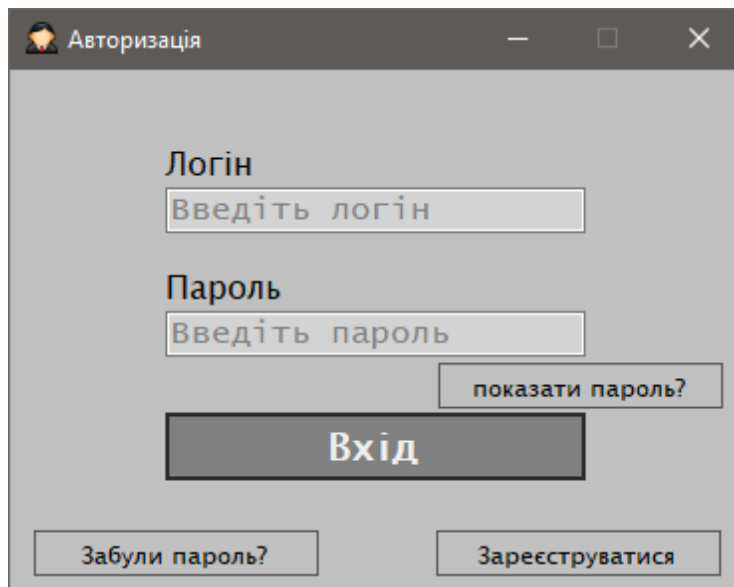


Рисунок 3.3 – Реалізація вікна авторизації

Вікно реєстрації користувача зображено на рисунку 3.4.

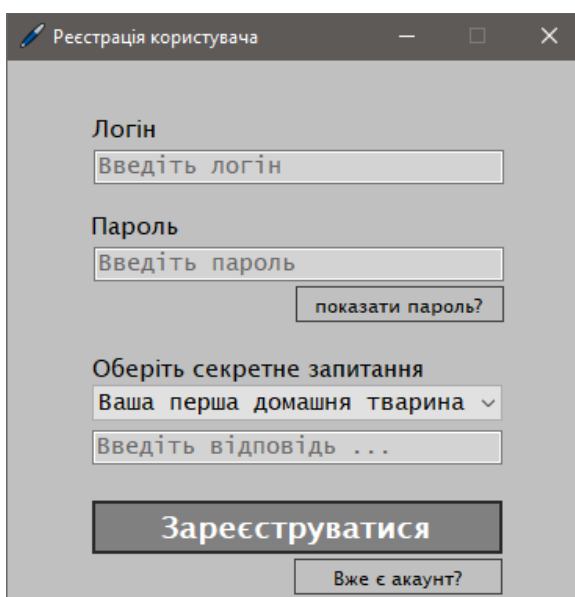


Рисунок 3.4 – Вікно реєстрації користувача

На рисунку 3.5 зображено вікно відновлення даних для входу.

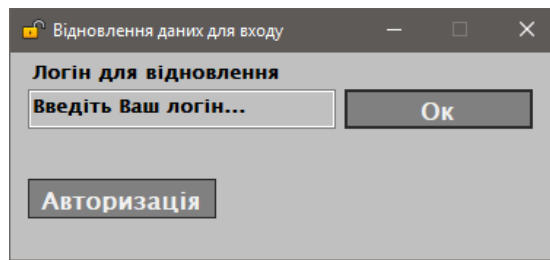


Рисунок 3.5 – Вікно відновлення даних для входу

Вікно додавання інформації працює одразу з двома сутностями - спочатку навчальний заклад, після нього – представник.

На рисунку 3.6 зображено вікно додавання інформації про навчальний заклад.

Додавання інформації

Додавання навчального закладу Перегляд Пошук Авторизація

Додавання навчального закладу

Назва закладу Введіть назву закладу

Тип закладу Введіть тип закладу

Адреса * Введіть вулицю

Район * Введіть район

Тип міста * Введіть тип міста

Місто * Введіть місто

Область * Введіть область

Поштовий індекс * Введіть поштовий індекс

Телефон закладу * Введіть номер телефону закладу (один на рядок)
(один на рядок)

E-mail закладу Введіть електронну пошту закладу (одну на рядок)
(один на рядок)

* – Необов'язкове поле

Додати заклад

Рисунок 3.6 – Вікно додавання інформації про навчальний заклад

На рисунку 3.7 зображено вікно додавання інформації, але про представника навчального закладу.

Додавання інформації

Додавання представника Перегляд Пошук Авторизація

Додавання представника

Прізвище

Ім'я

По-батькові

Заклад

Пост представника

Телефон представника *
(один на рядок)

E-mail представника
(один на рядок)

* - Необов'язкове поле

Додати представника

Рисунок 3.7 – Вікно додавання інформації про представника навчального закладу

Також було розроблено вікно перегляду сутностей навчального закладу та представника.

На рисунку 3.8 зображено вікно перегляду сутностей.

The screenshot shows a window titled 'Перегляд інформації' (Information View). It contains a table with the following columns: №, Назва закладу, Тип, Вулиця, Район, Тип міста, Місто, Область, Поштовий індекс, Номер телефону, Ел. адреса, Ким додано, and Дата додавання. The table lists five entries, all of which are 'Університет' (University) type institutions.

№	Назва закладу	Тип	Вулиця	Район	Тип міста	Місто	Область	Поштовий індекс	Номер телефону	Ел. адреса	Ким додано	Дата додавання
3	123	Університет	-	-	-	-	-	-	-	i73415441@g...	1	24.06.2024 19:...
3	123	Університет	-	-	-	-	-	-	-	nzprvn@gmail...	1	24.06.2024 19:...
4	XATT	Університет	-	-	-	-	-	-	-	mr.pumpkinfee...	1	11.06.2024 4.5...
4	XATT	Університет	-	-	-	-	-	-	-	pivenazari@g...	1	11.06.2024 4.5...
4	XATT	Університет	-	-	-	-	-	-	-	pivennazari@g...	1	11.06.2024 4.5...

Рисунок 3.8 – Вікно перегляду сутностей навчального закладу

На рисунку 3.9 зображено вікно потабличного перегляду бази даних.

The screenshot shows a window titled 'Перегляд бази даних потаблично' (Table View of Database). It contains a table with the following columns: № закладу, Назва, Тип, Редаговано, and Дата. The table lists four entries, with the first entry highlighted in blue.

№ закладу	Назва	Тип	Редаговано	Дата
1	NURE	Університет	0	24.06.2024 18:54:32
2	sdfsdaf	Коледж	0	24.06.2024 18:59:47
3	123	Університет	1	24.06.2024 19:05:02
4	XATT	Університет	1	11.06.2024 4:58:22

Рисунок 3.9 – Вікно потабличного перегляду бази даних

Вікно формування черги дозволяє користувачеві обрати інформацію, яка заповнюється обов'язково аби по-перше надсилати персоналізовані листи, по друге-для забезпечення сегментації отримувачів за обраними факторами.

На рисунку 3.10 зображено вікно пошуку інформації для формування черги з отримувачів за певними критеріями.

Шукати за: Очистити все

Назва закладу: -

Тип закладу: -

Тип міста: -

Місто: -

Район: -

Область: -

Шукати

Умови: Видалити рядок Очистити умови

Назва закладу	Тип закладу	Тип міста	Місто	Район	Область
ХНУРЕ	-	-	-	-	-
ХПКК	-	-	-	-	-

Повертати:

- Назва закладу
- Тип закладу
- Тип міста
- Місто
- Вулиця
- Район
- Область
- Поштовий індекс
- Електронна пошта закладу
- Номер телефону закладу
- ПІБ Представника
- Пост
- Електронна пошта
- Номер телефону представника

Експорт бази даних

Поточну базу даних в папку версій

Імпорт бази даних

Вивести чергу в файл CSV

Черга:

Назва закладу	Тип закладу	Електронна пошта закладу	Номер телефону закладу	Прізвище	Ім'я	По-батькові	Пост	Електронна пошта	Номер телефону
ХНУРЕ	Університет	officialnazar13...	-	не додано	не додано	не додано	не додано	не додано	не додано
ХНУРЕ	Університет	officialnazar13...	-	Іваненко	Віктор	Олександров...	Завідуючий к...	gtx650i2gboc...	-
ХНУРЕ	Університет	officialnazar13...	-	Шевченко	Олег	Віталійович	Викладач	nazar015minu...	Введіть номе...
ХНУРЕ	Університет	officialnazar13...	-	Шевченко	Олег	Віталійович	Викладач	nazar015minu...	Введіть номе...
ХНУРЕ	Університет	officialnazar13...	-	Шевченко	Олег	Віталійович	Викладач	pnazar554@g...	Введіть номе...
ХНУРЕ	Університет	officialnazar13...	-	Олександров	Роман	Олегович	Викладач	pivennazar67...	123
ХПКК	Коледж	sikouseer@g...	-	не додано	не додано	не додано	не додано	не додано	не додано

Вивести чергу в БД Перейти до формування листів

Рисунок 3.10 – Вікно формування черги з отримувачів

Зверху вікна використовується таблиця для відображення умов пошуку, знизу – черга для надсилання за обраними категоріями.

3.3 Розробка модуля надсилання листів

Було розроблено програмну складову до модуля надсилання листів в середовищі розробки VS Community 2022, використовуючи мову програмування C# та Python. Основою для даної розробки є бібліотека: Python.Interop [43].

Дана бібліотека була обрана оскільки вона надає можливість інтегрувати Python-код в C# та керувати їхньою взаємодію.

В ситуації поєднання цих двох технологій C# Windows Forms обрано як класичний інструмент для створення програми з стабільно працюючим користувальницьким інтерфейсом.

Python являє собою мову програмування, що постійно розвивається та підтримує інтеграцію з усіма необхідними сервісами для задоволення вимог ТЗ.

На рисунку 3.11 зображено вікно керування вмістом листів.

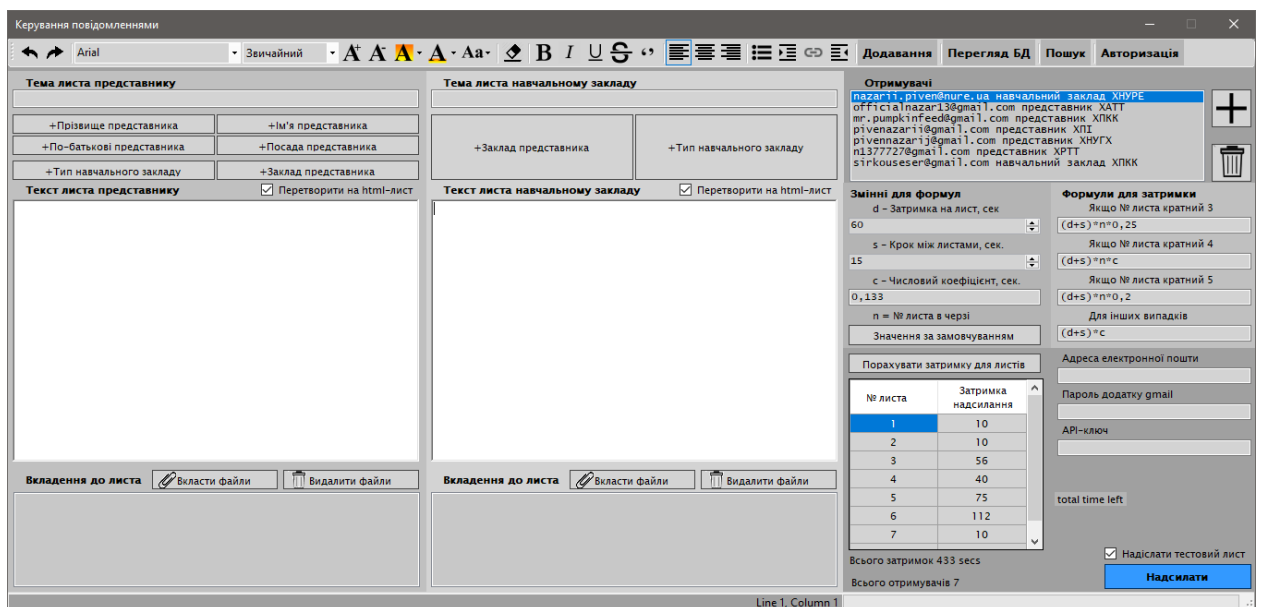


Рисунок 3.11 – Вікно керування електронними листами

У вікні керування електронними листами міститься текстовий процесор для двох варіантів листів: як для навчального закладу так і до представника.

Текстовий процесор необхідний для керуванням вмістом листа, робота текстового процесору дозволяє також впроваджувати в лист теги, що забезпечують адресне використання даних для листа отримувачу

За замовчуванням можна розсилати як готові html-листи, так і форматувати текст в елементі RichTextBox на HTML натиснувши «Перетворити на html-лист», щоб потім формувати його в готову програму, що зображено на рисунку 3.12.

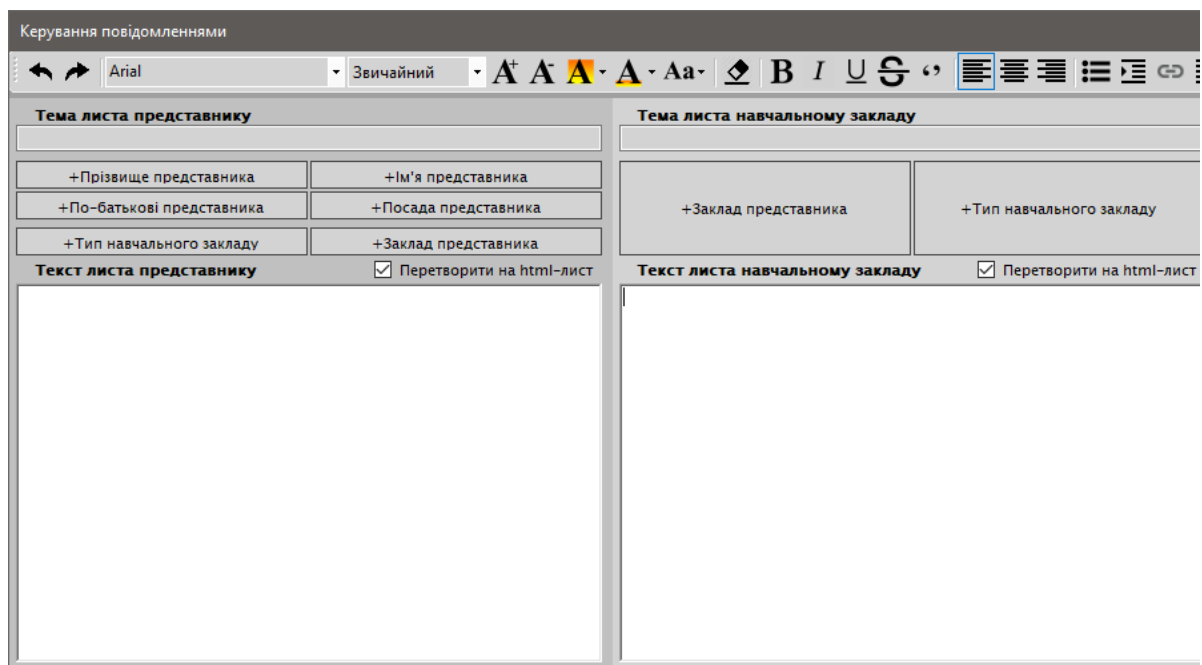


Рисунок 3.12 – Текстовий процесор

На рисунку 3.13 зображено керування чергою та затримками

The screenshot shows a software interface with the following sections:

- Отримувачі (Recipients):** A list of email addresses and their corresponding organizations, such as 'nazarii.piven@nure.ua навчальний заклад ХНУРЕ' and 'officialnazar13@gmail.com представник ХАТТ'. There are icons for adding (+) and deleting (trash) recipients.
- Змінні для формул (Variables for formulas):**
 - d - Затримка на лист, сек (Delay per page, sec):** 60
 - s - Крок між листами, сек (Step between pages, sec):** 15
 - c - Числовий коефіцієнт, сек (Numerical coefficient, sec):** 0,133
 - n = № листа в черзі (n = Page number in queue):** Значення за замовчуванням (Default values)
- Формули для затримки (Formulas for delays):**
 - Якщо № листа кратний 3 (If page number is multiple of 3): $(d+s) * n * 0,25$
 - Якщо № листа кратний 4 (If page number is multiple of 4): $(d+s) * n * c$
 - Якщо № листа кратний 5 (If page number is multiple of 5): $(d+s) * n * 0,2$
 - Для інших випадків (For other cases): $(d+s) * c$
- Порахувати затримку для листів (Calculate delay for pages):** A button that triggers the calculation of delays.
- Таблиця затримок (Delay table):**

№ листа (Page No)	Затримка надсилання (Delay)
1	10
2	10
3	56
4	40
5	75
6	112
7	10
- Адреса електронної пошти (Email address):** Input field.
- Пароль додатку gmail (Gmail app password):** Input field.
- API-ключ (API key):** Input field.
- total time left:** A label indicating the remaining time.
- Надіслати тестовий лист (Send test email):** A checked checkbox.
- Надіслати (Send):** A blue button.
- Summary:** Всього затримок 433 secs (Total delays 433 secs), Всього отримувачів 7 (Total recipients 7).

Рисунок 3.13 – Керування чергою та затримками

Користувач може редагувати чергу отримувачів, а також вводити змінні для затримок на кожен лист кратний трьом, чотирьом, п'яти та іншим. Та вводити власноруч створену формулу для обчислення затримки на переключення до наступної адреси.

Нижче в формі є контейнери для завантаження файлів з комп'ютера відправника. У поштових сервісів діють обмеження пов'язані з розмірами-лімітами, переходячи за які поштовий сервіс повертає помилку «завеликий зміст». Тому було зручно інтегровано механізм для завантаження файлів з контейнеру, за умови, що об'єм файлів на лист більше 25Мб (25600 кб) на

Google drive та автоматичне повертання посилання для його впровадження в текст листа.

На рисунку 3.14 зображено завантаження світлин більших за об'ємом ніж 25 Мб. Після вибору файлу-ключа сервісного акаунту Google Drive API.

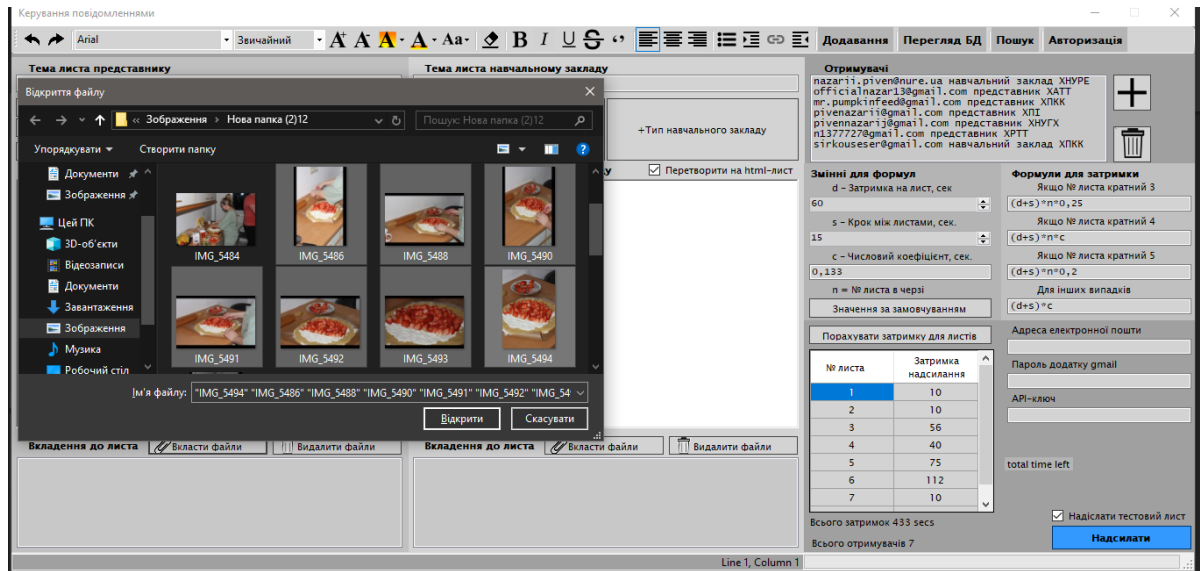


Рисунок 3.14 – Завантаження файлів більше 25 Мб.

На рисунку 3.15 зображено результат автоматичного завантаження файлів в хмарне сховище Google Drive [44] у вигляді посилання та перевірки посилання шляхом переходу на нього.

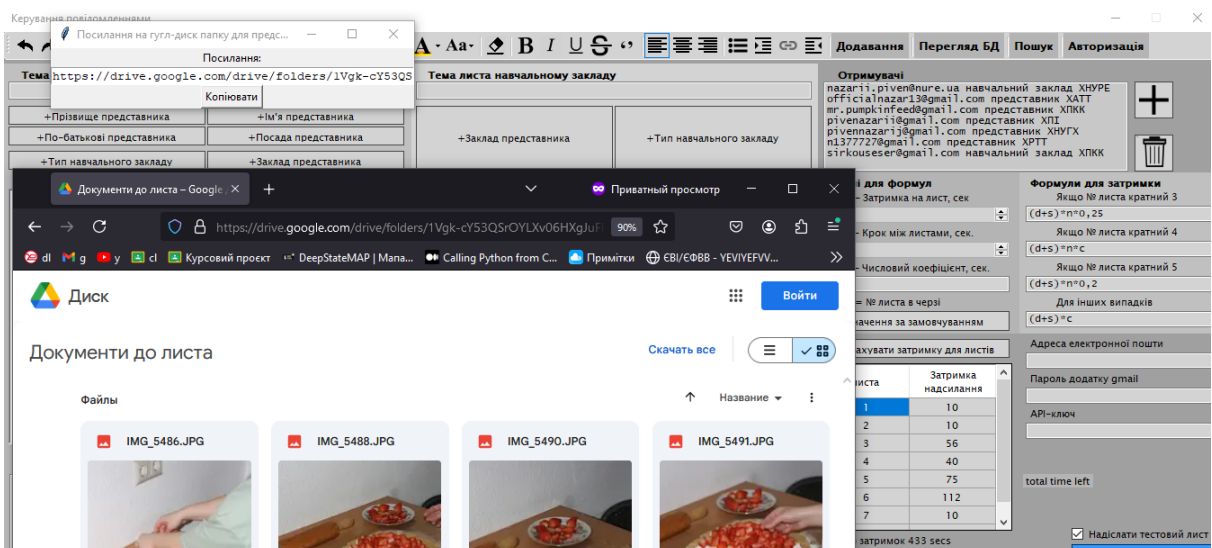


Рисунок 3.15 – Результат завантаження

Це стало можливим завдяки використанню мови програмування Python та її потужній інтеграції з Google Drive API.

Авторизація користувача в акаунт електронної пошти відбувається за допомогою введення паролю додатку Google.

Надсилання повідомлення на електронну адресу працює за рахунок наявності електронної адреси та паролю додатку Google.

Вставка зображень різних форматів в лист відбувається зазвичай відбувається завдяки використанню сервісу ImgBB [45] та їхнього API для зберігання світлин на їхньому безкоштовному хмарному сховищі, авторизація в яке та повертання посилання відбувається автоматично.

Реалізація забезпечення надсилання персоналізованих листів полягає в використанні розмітки за допомогою тегів всередині тексту листа. Цей процес зображено на рисунку 3.16.

```

delayer_else = delayer + 1000;

originalRtf = rtbe_Body_agents.Rtf;
rtbe_buffer.Rtf = originalRtf;

ReplaceTextInRichTextBox(rtbe_buffer, "INST_NAME", queue[em].I_name);
ReplaceTextInRichTextBox(rtbe_buffer, "INST_TYPE", queue[em].I_type);
ReplaceTextInRichTextBox(rtbe_buffer, "A_FIRSTNAME", queue[em].A_first_name);
ReplaceTextInRichTextBox(rtbe_buffer, "A_LASTNAME", queue[em].A_last_name);
ReplaceTextInRichTextBox(rtbe_buffer, "A_PATRONYMIC", queue[em].A_patronymic);
ReplaceTextInRichTextBox(rtbe_buffer, "A_POST", queue[em].A_post);

string instname_formatted_subject = subject_agent_text.Replace("INST_NAME", queue[em].I_name);
string insttype_subject = instname_formatted_subject.Replace("INST_TYPE", queue[em].I_type);
string firstname_formatted_subject = insttype_subject.Replace("A_FIRSTNAME", queue[em].A_first_name);
string lastname_formatted_subject = firstname_formatted_subject.Replace("A_LASTNAME", queue[em].A_last_name);
string patronymic_formatted_subject = lastname_formatted_subject.Replace("A_PATRONYMIC", queue[em].A_patronymic);
string post_formatted_subject = patronymic_formatted_subject.Replace("A_POST", queue[em].A_post);
string subject_formatted_text_agent = post_formatted_subject;

subject_formatted_text = subject_formatted_text_agent;
formattedpathways = send_to_receiver.Replace("files_PATH", files_pathway_agent);
formattedsenderemail = formattedpathways.Replace("email_sender", smtp_sender_email);
string formattedreceiveremail = formattedsenderemail.Replace("email_receiver", receiver_email);
formattedsmtppass = formattedreceiveremail.Replace("password_smtp", smtp_app_pass);
formattedsubject = formattedsmtppass.Replace("email_Subject", subject_formatted_text);
formattedpicture_count = formattedsubject.Replace("picture_count", (picture_count_agent + 1).ToString());
py_code = formattedpicture_count;

await MyMethodAsync();
pb_sent_emails.Minimum = 0;
pb_sent_emails.Maximum = 100;

// Calculate the progress percentage
int progressPercentage = (int)((double)(em + 1) / lb_emails.Items.Count) * 100;

// Update the progress bar value

```

Рисунок 3.16 – Приклад розмітки за допомогою тегів всередині тексту

3.4 Висновок до розділу 3

У третьому розділі було успішно розроблено систему електронного листування, яка інтегрує інтерфейс користувача на базі технології Windows Forms C#, модуль надсилання листів за допомогою мови використання мови програмування Python та бази даних SQLite для зберігання інформації. Інтерфейс Windows Forms забезпечує зручність та інтуїтивність взаємодії користувача з системою, дозволяючи легко створювати, надсилати та відстежувати електронні листи.

Модуль Python реалізує функціонал надсилання листів через SMTP-сервер, що забезпечує надійність та ефективність процесу передачі даних. Використання бази даних SQLite дозволяє зберігати необхідну інформацію про користувачів, листи та їх статуси в локальному сховищі, забезпечуючи швидкий доступ до даних та їхню цілісність.

Всі компоненти системи тісно інтегровані між собою, що забезпечує цілісність та узгодженість роботи системи в цілому.

Реалізація такої комплексної системи електронного листування демонструє ефективність використання сучасних технологій в поєднанні з класичними рішеннями складних систем та інструментів для вирішення завдань автоматизації процесів комунікації.

4 ВИВЕДЕННЯ ЦІЛЬОВОЇ ФУНКЦІЇ. ОБЧИСЛЕННЯ ПАРАМЕТРІВ МОДЕЛІ СИСТЕМИ

В розробленій системі автоматизації електронного листування є набір параметрів, які являють собою вихідні дані програми. Для оптимізації цих параметрів необхідно визначити цільову функцію, тобто умови задоволення задачі оптимізації.

$$v_k, k = 1, 2, \dots, N_v$$

$$0 \leq v_k \leq 1,$$

де, v_k – вихідні дані програми;

N_v – завдання користувача дорівнює 2, бо у користувача 2 завдання.

Та набір параметрів, що являє собою налаштування програми.

$$u_k, k = 1, 2 \dots N_U,$$

де u_k – налаштування програми;

N_U – параметри управління дорівнює 9, бо 9 параметрів управління.

На рисунку 4.1 зображено схематична модель як об'єкта автоматизації

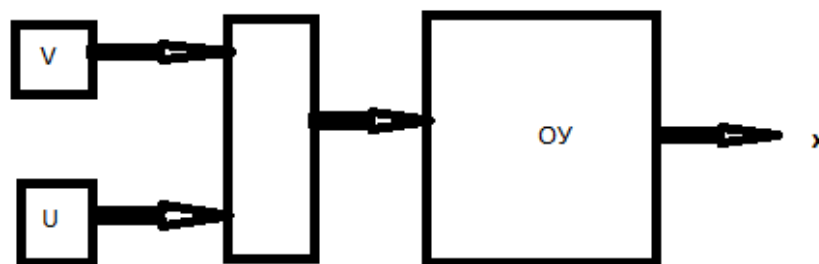


Рисунок 4.1 – Схематична модель програми як об'єкта автоматизації

Модель програми як об'єкта автоматизації розраховується за (4.1):

$$x = \sum_{k=1}^{N_v} a_k v_k + \sum_{k=1}^{N_u} b_k u_k = \sum_{k=1}^2 a_k v_k + \sum_{k=1}^9 b_k u_k \quad (4.1)$$

де, a_k, b_k – параметри моделі;

x – час виконання завдання програми

Основна ідея полягає в підборі параметрів u_k таким чином, щоб час виконання завдання час (x) досягав заданого значення.

Для побудови математичної моделі програми як об'єкта автоматизації використовуємо статичну ідентифікацію на основі даних тестувань програми.

$$x^{(j)} \begin{cases} V_k^{(j)} & k = 1, 2, \dots, N_v \\ U_k^{(j)} & k = 1, 2, \dots, N_u, \\ j = 1, 2, \dots, n \end{cases} \quad (4.2)$$

$$x^{(j)} = \sum_{k=1}^2 a_k v_k^{(j)} + \sum_{k=1}^9 b_k u_k^j \quad (4.3)$$

де n – число випробувань;

$V_k^{(j)}, U_k^{(j)}$ - умови випробування j ;

$x^{(j)}$ – час випробування j .

Щоб знайти параметри моделі використовується метод найменших квадратів:

$$s = \sum_{j=1}^n \left(\sum_{k=1}^9 a_k u_k^j + \sum_{k=1}^2 b_k v_k^j - x^j \right)^2 \quad (4.3)$$

$$\frac{ds}{da_i} = 0, i = 1, 2, \dots, 9 \quad (4.4)$$

$$\frac{ds}{db_i} = 0, i = 1, 2 \quad (4.5)$$

$$\frac{ds}{da_i} = \sum_{j=1}^n (\sum_{i=1}^{n_u} a_i u_i^j + \sum_{i=1}^{n_v} b_i v_i^{(j)} - x^j) * u_i^j, i=1, 2, \dots, N_u \quad (4.6)$$

$$\frac{ds}{db_i} = \sum_{j=1}^n (\sum_{i=1}^{n_u} a_i u_i^j + \sum_{i=1}^{n_v} b_i v_i^{(j)} - x^j) * v_i^j, i=1, 2, \dots, N_v \quad (4.7)$$

$$\frac{ds}{da_k} = \sum_{k=1}^{n_u} (\sum_{j=1}^n u_i^j \cdot u_k^j) a_k + \sum_{k=1}^{n_v} (\sum_{j=1}^n u_i^j v_k^j) b_k - \sum_{j=1}^n u_i^j x^j \quad (4.8)$$

$i=1, 2, \dots, N_u$

$$\frac{ds}{db_i} = \sum_{k=1}^{N_u} (\sum_{j=1}^n v_i^j * u_k^j) a_k + \sum_{k=1}^{n_v} (\sum_{j=1}^n v_i^j v_k^j) b_k - \sum_{j=1}^n v_i^j x^j \quad (4.9)$$

$i=1, 2, \dots, N_v$

Введемо умовні позначення:

$$A_{ik} = \sum_{j=1}^n u_i^j u_k^j \quad (4.10)$$

$$; B_{ik} = \sum_{j=1}^n v_i^j v_k^j \quad (4.11)$$

$$; C_{ik} = \sum_{j=1}^n u_i^j v_k^j; \quad (4.12)$$

$$f_i^u = \sum_{j=1}^n u_i^j x^j; \quad (4.13)$$

$$f_i^v = \sum_{j=1}^n v_i^j x^j; \quad (4.14)$$

З урахуванням позначень (4.8) та (4.9) отримані рівняння (4.4) та (4.5) дорівнюють нулю. Таким чином отримуємо систему рівнянь, наведену в (4.15).

$$\sum_{k=1}^9 A_{ik} a_k + \sum_{k=1}^2 c_{ik} b_k = f_i^u; \quad i = 1, 2, \dots, 9 \quad (4.15)$$

$$\sum_{k=1}^9 C_{ik} a_k + \sum_{k=1}^2 B_{ik} b_k = f_i^v; \quad i = 1, 2$$

Дану систему лінійних рівнянь необхідно вирішити методом Гауса.

$$\begin{pmatrix} A & C_u \\ C_v & B \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} f_u \\ f_v \end{pmatrix},$$

Заповнюємо кожну матрицю для розв'язання.

$$A = \begin{vmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{vmatrix} \quad C = \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{vmatrix}$$

$$C_v = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

$$B = \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix}$$

$$f_u = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

$$f_v = \begin{vmatrix} 1 & 0 \end{vmatrix}$$

Для формування розширеної матриці об'єднуємо всі матриці у одну розширену матрицю:

$$M = \begin{vmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 2 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & 2 \end{vmatrix}$$

Додаємо вектор правих частин f:

$$f = \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{vmatrix}$$

Для розв'язання системи рівнянь методом Гауса використовуємо часткове вибирання головного елемента для уникнення числової нестабільності.

Програмний код для розв'язання системи рівнянь такого роду, написаний мовою Python, наведено в додатку А.

Після виконання програмного коду, було виведено результати, зображені на рисунку 4.2.

```

Вектор рішення a:
[ 0.20138889 -0.13888889  0.09722222 -0.20833333  0.0625   -0.20833333
 0.09722222 -0.13888889  0.20138889]

Вектор рішення b:
[0.45833333 0.57638889]

PS C:\Users\9>

```

Рисунок 4.2 – Результат обчислення параметрів a_k та b_k

$$a_k =$$

0.20138889
-0.13888889
0.09722222
-0.20833333
0.0625
-0.20833333
0.9722222
-0.13888889
0.20138889

$$b_k =$$

0.45833333
0.57638889

Тепер, коли ми маємо масив a_k та b_k елементів – можна шукати параметри для управління u_k через v_k , щоб час виконання завдання x дорівнював x -цільовому.

Цільова функція матиме наступний вигляд:

$$f(u) = |x_{target} - (\sum_{k=1}^{N_v} a_k v_k + \sum_{k=1}^{N_u} b_k u_k)|$$

5 ОХОРОНА ПРАЦІ

Інформаційні технології є невід'ємною частиною нашого повсякденного життя. Правила роботи з ними впроваджуються в наше життя завдяки збільшенню ергономічності систем з точки зору користувацького досвіду як наслідку постійного розвитку технологій.

Адміністратори – відповідальні та дуже зосереджені на вирішенні проблем люди. Навіть під час вільного часу від роботи такі люди зазвичай думають про вирішення проблеми підприємства.

У ролі адміністратора на підприємстві, особливо при роботі з електронною системою листування, надзвичайно важливо не лише забезпечувати ефективну роботу систем та підлеглих, але й правильно організувати власний робочий час та відпочинок.

Постійне перебування у стані мозкового штурму може призвести до вигорання, зниження продуктивності та негативно вплинути на загальну ефективність роботи.

5.1 Важливість організації робочого часу

– **планування завдань:** планування робочого дня дозволяє краще розподілити час та ресурси, уникнути перевантаження та зменшити стрес. Важливо визначити пріоритетні завдання і розподілити час на їх виконання;

– **тайм-менеджмент:** використання методів тайм-менеджменту, таких як техніка Pomodoro (робота інтервалами по 25 хвилин з короткими перервами) або метод ABC (розподіл завдань за важливістю та терміновістю), допомагає підвищити продуктивність та зосередженість;

– **делегування обов'язків:** розподіл завдань між підлеглими не лише знижує навантаження на адміністратора, але й сприяє розвитку команди. Важливо навчитися довіряти підлеглим та ефективно розподіляти обов'язки;

– регулярні перерви: перерви під час роботи допомагають зняти напругу та відновити працездатність. Рекомендується робити короткі перерви кожні 60 хвилин, під час яких можна виконати прості фізичні вправи або пройтися.

При роботі за комп'ютером важливо забезпечити відповідність умов праці державним стандартам України, зокрема ДСТУ ISO 45001:2019 "Системи менеджменту охорони здоров'я та безпеки праці".

Основні вимоги охорони праці включають забезпечення безпечних умов роботи на робочому місці, організацію належного освітлення, вентиляції, ергономіки робочих місць та мінімізації шкідливих факторів, пов'язаних з тривалим перебуванням за комп'ютером.

5.2 Аналіз небезпек і шкідливих факторів

Основні небезпеки та шкідливі фактори при роботі з автоматизованою електронною системою листування включають:

- фізичні фактори: тривале сидіння за комп'ютером, неправильна постава, неадекватне освітлення робочого місця;
- психофізіологічні фактори: стрес, пов'язаний з великою кількістю інформації та швидкими термінами виконання завдань;
- електромагнітне випромінювання: випромінювання від комп'ютерної техніки, яке може впливати на здоров'я працівників;
- втома очей: постійне навантаження на зір через тривале використання моніторів.

Запобігання цим небезпекам та шкідливим факторам вимагає впровадження відповідних заходів, таких як ергономічна організація робочого місця, регулярні перерви під час роботи, використання спеціальних програм для зменшення навантаження на очі та психоемоційну підтримку працівників.

5.3 Заходи з охорони праці

Для забезпечення безпеки працівників необхідно впровадити наступні заходи з охорони праці:

- організація робочого місця: відповідно до ДСанПіН 3.3.2-007-98, робоче місце повинно бути організоване з урахуванням ергономіки, забезпечено регульованими меблями та правильним розташуванням обладнання;
- освітлення: відповідно до ДБН В.2.5-28-2006, освітлення робочого місця повинно бути не менше 300 люкс, щоб зменшити напруження очей;
- вентиляція: відповідно до ДБН В.2.5-67:2013, необхідно забезпечити належну вентиляцію приміщень для підтримки свіжого повітря;
- перерви та вправи: введення регулярних перерв кожні 60 хвилин для виконання вправ для очей та спини;
- навчання: проведення регулярних інструктажів з охорони праці та навчання працівників правилам безпечного використання комп'ютерної техніки.

5.4 Пожежна безпека

Для забезпечення пожежної безпеки в приміщеннях, де використовується автоматизована система електронного листування, необхідно дотримуватись наступних вимог:

- засоби пожежогасіння: наявність первинних засобів пожежогасіння (вогнегасників) відповідно до ДСТУ 3675-98;
- система оповіщення: встановлення системи пожежної сигналізації та оповіщення про пожежу відповідно до ДСТУ EN 54-2:2003;
- евакуаційні шляхи: забезпечення вільного доступу до евакуаційних шляхів та виходів, які повинні бути позначені та освітлені згідно з ДБН В.1.1-7:2016;

5.5 Висновки до п'ятого розділу

Організація робочого часу та відпочинку є критично важливою для адміністратора на підприємстві. Ефективне планування завдань, використання методів тайм-менеджменту, делегування обов'язків, а також регулярні перерви, фізична активність, релаксаційні техніки, збалансоване харчування та якісний сон допомагають підтримувати високу продуктивність, знижувати стрес та запобігати професійному вигоранню. Забезпечення власного здоров'я та добробуту є ключем до успішного виконання професійних обов'язків та загального процвітання підприємства.

Забезпечення охорони праці при роботі з електронною системою листування є критично важливим для підтримання здоров'я та безпеки працівників.

Дотримання державних стандартів України, таких як ДСТУ ISO 45001:2019, ДСанПіН 3.3.2-007-98, ДБН В.2.5-28-2006, ДБН В.2.5-67:2013, ДСТУ 3675-98 та ДБН В.1.1-7:2016, дозволяє мінімізувати вплив шкідливих факторів, запобігти травмам та аварійним ситуаціям.

Впровадження відповідних заходів охорони праці та пожежної безпеки забезпечить комфортні та безпечні умови роботи для всіх працівників, що сприятиме підвищенню продуктивності та ефективності роботи організації.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи була розроблена система автоматизації процесу надсилання електронних листів, що базується на поєднанні мов програмування C# та Python із використанням захищеної локальної бази даних SQLite.

В першу чергу було проведено аналіз матеріальної бази та технічного і методичного забезпечення, що використовувались під час проходження практики та підготовки кваліфікаційної роботи. Це дозволило виявити основні вимоги та визначити оптимальні підходи до реалізації системи.

Було здійснено детальний аналіз структури існуючих систем листування, що дало змогу сформулювати цілі, об'єкт і предмет розробки. Ознайомлення з основними публікаціями та нормативними документами дозволило визначити сучасні тенденції та вимоги у сфері автоматизації електронного листування.

На основі аналізу існуючих аналогів та предметної області було розроблено контекстну, функціональну та структурну схеми системи. Ці схеми дозволили візуалізувати основні процеси та взаємодії між компонентами системи.

Для реалізації системи були обрані технології, які забезпечують ефективність, надійність та безпеку обробки даних. Зокрема, використання мови програмування C# для розробки інтерфейсу користувача на основі Windows Forms дозволило створити зручний та інтуїтивно зрозумілий інтерфейс.

Модуль написаний мовою програмування Python реалізує функціонал надсилання листів через SMTP-сервер, що забезпечує надійність та ефективність передачі даних. Локальна база даних SQLite забезпечує безпеку зберігання інформації завдяки використанню шифрування.

У рамках розробки були створені умови керування затримкою на надсилання електронних листів з оригінальним вмістом, що дозволяє оптимізувати процеси комунікації підприємства.

Система забезпечує можливість створення шаблонів повідомлень, контроль термінів надсилання та збереження даних в безпеці, що значно підвищує ефективність ведення комунікації з клієнтами, партнерами та співробітниками.

Завдяки цьому досягається оптимізація комунікаційних процесів підприємства та забезпечується вищий рівень обслуговування клієнтів.

В результаті виконання роботи була розроблена ефективна та гнучка система автоматизації електронного листування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tapr.nure.ua: Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки [Електронний ресурс]. Tapr.nure.ua – Головна. Текстові дані Режим доступу: <https://tapr.nure.ua/> – 20.04.2024.
2. Nure.ua: Харківський національний університет радіоелектроніки [Електронний ресурс]. Nure.ua – Про університет. Текстові дані Режим доступу: <https://nure.ua/universitytet/pro-universitytet> – 20.04.2024.
3. Sqlite.org: Сайт SQLite. [Електронний ресурс] Documentation. Текстові дані. – Режим доступу <https://www.sqlite.org/docs.html> – 20.04.2024.
4. C# Guide - .NET managed language. *Microsoft Learn: Build skills that open doors in your career.* – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/> – 25.06.2024.
5. Welcome to Python.org. *Python.org.* – Режим доступу: <https://www.python.org/about> – 25.06.2024.
6. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
7. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, О. І. Филипенко, О. В. Токарева, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2023. – 151 с.
8. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Системна інженерія» / Упоряд.: І.Ш. Невлюдов, О.М. Цимбал, О.В. Токарева, А.І. Бронніков. Харків: ХНУРЕ, 2022. 66 с.

9. A Beginner's Guide to Desktop App Development. *Medium.com*. – Режим доступу: https://medium.com/@marketing_18039/a-beginners-guide-to-desktop-app-development-115e6cc6d6ba – 26.06.2024.

10. Gmail: Електронна пошта від Google. *Gmail*. – Режим доступу: <https://www.gmail.com/mail/help/intl/uk/about.html?ua>. – 25.06.2024.

11. Our mission is to organize the world's information and make it universally accessible and useful. *www.google.com*. – Режим доступу: <https://about.google/> – 25.06.2024.

12. Outlook for everyday email and calendars. <https://www.microsoft.com>. Режим доступу: <https://www.microsoft.com/en-us/microsoft-365/outlook/email-and-calendar-software-microsoft-outlook> – 25.06.2024.

13. Unlock your potential with Microsoft Copilot due to several reasons. *www.microsoft.com*. Режим доступу: <https://www.microsoft.com/en-us/> – 25.06.2024.

14. Office is now Microsoft 365. *www.microsoft.com*. Режим доступу: <https://www.microsoft.com/en-us/microsoft-365/microsoft-office> – 25.06.2024.

15. Connect. Organize. Get things done. <https://outlook.live.com>. Режим доступу: https://outlook.live.com/mail/about/index_en.html – 25.06.2024.

16. Outlook for business. <https://www.microsoft.com>. Режим доступу: <https://www.microsoft.com/en-us/microsoft-365/outlook/outlook-for-business> – 26.06.2024.

17. Yahoo Пошта – організована електронна пошта. *Yahoo Mail*. Режим доступу: <https://mail.yahoo.com/?lang=uk-UA> – 25.06.2024.

18. Yahoo. *Yahoo*. Режим доступу: <https://login.yahoo.com/?intl=ua&.src=yms> – 25.06.2024.

19. Hughes, L (1998). Internet E-mail: Protocols, Standards and Implementation. Artech House Publishers. ISBN 978-0-89006-939-4.

20. Unlock your potential with Microsoft Copilot due to several reasons. *www.microsoft.com*. Режим доступу: <https://www.microsoft.com/en-us/> – 25.06.2024.

21. Відомості про протоколи IMAP і POP - Підтримка від Microsoft. *Microsoft Support*. Режим доступу: <https://support.microsoft.com/uk-ua/office/відомості-про-протоколи-imap-i-pop-ca2c5799-49f9-4079-aefe-ddca85d5b1c9> – 25.06.2024.

22. Reviewing DNS Concepts. *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/reviewing-dns-concepts> – 25.06.2024.

23. Gather the information you need to create DNS records - Microsoft 365 admin. *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/microsoft-365/admin/get-help-with-domains/information-for-dns-records?view=o365-worldwide> – 25.06.2024.

24. Wikimedia.org: Vector version of File: How e-mail works.png [Електронний ресурс]. File: Email.svg. Текстові дані – Режим доступу: <https://commons.wikimedia.org/wiki/File:Email.svg#mw-jump-to-license> – 20.04.2024.

25. Створення й редагування файлів CSV для імпорту до Outlook - Підтримка від Microsoft. *Microsoft Support*. Режим доступу: <https://support.microsoft.com/uk-ua/office/створення-й-редагування-файлів-csv-для-імпорту-до-outlook-4518d70d-8fe9-46ad-94fa-1494247193c7> – 25.06.2024.

26. Microsoft. HTML Programming with Visual Studio Code. *Visual Studio Code - Code Editing. Redefined*. Режим доступу: <https://code.visualstudio.com/docs/languages/html> – 25.06.2024.

27. Mailchimp.com: Email marketing [Електронний ресурс]. Turn emails into revenue. Текстові дані Режим доступу: <https://mailchimp.com/> – 20.04.2024.

28. Constantcontact.com: Digital and Email Marketing Platform [Електронний ресурс]. Constant Contact Текстові дані. Режим доступу: <https://www.constantcontact.com/> – 20.04.2024.

29. Brevo.com: Formerly sendinblue [Електронний ресурс]. Brevo. Текстові дані. Режим доступу: <https://www.brevo.com/en/> – 20.04.2024.

30. What is CRM? | Microsoft Dynamics 365. *Business Applications / Microsoft Dynamics 365*. Режим доступу: <https://dynamics.microsoft.com/en-us/crm/what-is-crm/> – 25.06.2024.

31. SendGrid Email API and Email Marketing Campaigns | SendGrid. *SendGrid*. Режим доступу: <https://sendgrid.com/en-us> – 25.06.2024.

32. Microsoft.com: Сайт компанії Microsoft [Електронний ресурс] Create IDEF0 diagrams Текстові дані. Режим доступу: <https://support.microsoft.com/en-us/office/create-idef0-diagrams-ea7a9289-96e0-4df8-bb26-a62ea86417fc> - 20.04.2024.

33. The C4 model for visualising software architecture. *The C4 model for visualising software architecture*. Режим доступу: <https://c4model.com/> – 25.06.2024.

34. Rational Software Architect 9.6.1. *IBM - United States*. Режим доступу: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case> – 25.06.2024.

35. IBM Developer. *IBM Developer*. Режим доступу: <https://developer.ibm.com/articles/the-sequence-diagram/> – 25.06.2024.

36. Visual Studio Installer Projects and .NET. *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/deployment/installer-projects-net-core?view=vs-2022> – 25.06.2024.

37. Experience the Power of Windows 11 OS, Computers, & Apps | Microsoft. *Windows*. Режим доступу: <https://www.microsoft.com/en-us/windows/> – 25.06.2024.

38. Microsoft.com: Сайт компанії Microsoft [Електронний ресурс] GitHub Copilot і Visual Studio 2022. Текстові дані. Режим доступу: <https://visualstudio.microsoft.com/> – 20.04.2024.

39. Create a Windows Forms app with C# - Visual Studio (Windows). *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022> – 25.06.2024.

40. SQL Language | Oracle. *Oracle / Cloud Applications and Cloud Platform*. Режим доступу: <https://www.oracle.com/database/technologies/appdev/sql.html> – 25.06.2024.

41. C docs - get started, tutorials, reference. *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/cpp/c-language/?view=msvc-170> – 25.06.2024.

42. Encryption risks and protections. *Microsoft Learn: Build skills that open doors in your career*. Режим доступу: <https://learn.microsoft.com/en-us/purview/office-365-encryption-risks-and-protections> – 25.06.2024.

43. GitHub - pythonnet/pythonnet: Python for .NET is a package that gives Python programmers nearly seamless integration with the .NET Common Language Runtime (CLR) and provides a powerful application scripting tool for .NET developers. *GitHub*. Режим доступу: <https://github.com/pythonnet/pythonnet> – 25.06.2024.

44. Простий і надійно захищений доступ до вашого вмісту. Google. <https://www.google.com/>. Режим доступу: <https://www.google.com/intl/uk/drive/> – 25.06.2024.

45. Upload Image – Free Image Hosting. *ImgBB*. Режим доступу: <https://imgbb.com/> – 25.06.2024.