

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЧНОГО
НАДСИЛАННЯ ПРИВІТАНЬ З ДНЕМ НАРОДЖЕННЯ
З ВИКОРИСТАННЯМ PYTHON**

**DEVELOPMENT OF A MOBILE APPLICATION FOR
AUTOMATICALLY SENDING BIRTHDAY
GREETINGS USING PYTHON**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-41
Рудинець Роман Віталійович
Керівник:
Гордєєв Олександр Олександрович

Кваліфікаційну роботу
допущено до захисту

« 10 » 06 2025р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

[Signature]

« 20 » 12 2024 p.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Рудинцю Роману Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка мобільного додатку для автоматичного надсилення привітань з днем народження з використанням Python»

Керівник роботи: д.т.н., професор Гордєєв О. О.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра _____

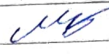
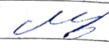
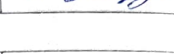
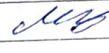
«10» червня 2025 р.

3. Вихідні дані до роботи: Python, SQLite, tkinter, методичні вказівки до виконання кваліфікаційної роботи бакалавра

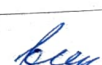
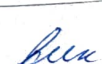
4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз актуальності теми, огляд існуючих рішень, визначення вимог до мобільного додатку, вибір інструментів та технологій розробки, реалізація мобільного додатку, тестування, налагодження й візуалізація роботи додатку.

5. Перелік графічного матеріалу: 12 рисунків, 1 таблиця

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	/ Гордєєв О. О.		
Специфікація вимог до розробленої системи	/ Гордєєв О. О.		
Розробка об'єкта проектування	/ Гордєєв О. О.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	2,44 %		
Академічна доброчесність	/ Гордєєв О. О.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	

Здобувач вищої освіти



Роман РУДИНЕЦЬ

/ Керівник кваліфікаційної роботи



Олександр ГОРДЄЄВ

АНОТАЦІЯ

Рудинець Р. В. Розробка мобільного додатку для автоматичного надсилання привітань з днем народження з використанням Python. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел.

У першому розділі проведено огляд аналогічних рішень, визначено актуальність, мету, об'єкт і предмет дослідження, а також сформульовано основні завдання.

У другому розділі здійснено вибір середовища розробки, обґрунтовано вибрані програмні технології, інструменти та методи реалізації мобільного додатку.

У третьому розділі представлено процес практичної реалізації проєкту: описано інтерфейс, реалізовані функціональні можливості, проведено тестування та налагодження системи, а також візуалізовано результати роботи додатку.

У висновках узагальнено результати роботи, підбито підсумки щодо досягнення мети та сформульовано перспективи подальшого розвитку додатка.

Ключові слова: мобільний додаток, автоматичне надсилання, Python, електронна пошта, привітання, інтерфейс, тестування, планувальник.

ABSTRACT

Rudynets R. Development of a Mobile Application for Automatically Sending Birthday Greetings Using Python. Manuscript. Bachelor's Qualification Thesis of the Educational Program "Software Engineering" in the specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions, and a list of references.

The first chapter provides a review of existing solutions, identifies the relevance of the topic, defines the aim, object, and subject of the research, and outlines the main objectives.

The second chapter describes the selection of the development environment and justifies the choice of software technologies, tools, and methods used to implement the mobile application.

The third chapter presents the process of practical implementation of the project: the user interface is described, the implemented functionality is outlined, testing and debugging of the system are conducted, and the results of the application are visualized.

The conclusions summarize the results of the work, highlight the achievement of the objectives, and propose directions for further development of the application.

Keywords: mobile application, automatic sending, Python, email, greeting, interface, testing, planner.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	9
1.1 Аналіз сучасного стану проблеми.....	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновок до розділу 1.....	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	15
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	15
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання...	19
Висновок до розділу 2.....	22
РОЗДІЛ 3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЧНОГО...	22
3.1 Практична реалізація об'єкта проектування.....	23
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	28
3.3 Візуалізація роботи додатка.....	31
Висновок до розділу 3.....	36
ВИСНОВКИ.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39

ВСТУП

Актуальність теми. У сучасному цифровому світі значну частину міжособистісного спілкування здійснюють за допомогою електронних засобів комунікації. Привітання з Днем народження є однією з найбільш поширених форм соціальної взаємодії, яка має не лише емоційне, а й етичне значення. Водночас у повсякденному житті, коли обсяг інформації зростає, а темп життя пришвидшується, люди дедалі частіше забувають вчасно вітати своїх друзів, колег чи родичів. Це може призводити до зниження рівня довіри, зіпсованих відносин або просто втрати приємної нагоди підтримати контакт.

Метою цієї кваліфікаційної роботи є створення мобільного додатку з використанням мови програмування Python, який здатен самостійно виявляти іменинників за датою, формувати персоналізоване привітання на основі шаблону, і надсилати його за вказаними контактними даними у форматі електронного листа.

Об'єктом виступає процес автоматизації надсилення електронних привітань у контексті персональної комунікації користувача.

Предметом є розробка архітектури та реалізація мобільного додатку з графічним інтерфейсом, здатного забезпечувати щоденну перевірку дат народження, редагування шаблонів повідомлень, відправку привітань через SMTP та зберігання історії надсилення.

Для досягнення поставленої мети були поставлені наступні завдання:

- провести детальний аналіз предметної області, розглянути існуючі рішення на ринку;
- сформулювати функціональні та нефункціональні вимоги, розробити UML-діаграму для візуального відтворення роботи системи;
- проаналізувати інструменти, які існують на ринку для розробки мобільних додатків та обрати стек технологій для реалізації майбутньої системи;

- розробити мобільний додаток для автоматичних надсилань привітань з днем народженням;
- провести детальне тестування та налагодити систему;
- детально описати роботу додатка, використовуючи відповідні зображення.

Таким чином, розробка мобільного застосунку, дозволить автоматизувати процес привітань. Такий додаток має не лише функціональне значення, а й соціальну цінність, адже сприяє збереженню та підтримці міжособистісних зв'язків.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасному цифровому середовищі мобільні додатки відіграють важливу роль у вирішенні повсякденних завдань, включаючи комунікацію, планування, нагадування та організацію особистого часу. Одна з актуальних потреб сучасного користувача – автоматизація процесу привітань з днем народження. У ритмі сьогодення важко запам'ятати дати всіх близьких людей, не забути своєчасно надіслати теплі слова чи подарунок, особливо коли мова йде про велику кількість контактів у професійному або особистому колі.

Існуючі застосунки частково вирішують цю проблему, пропонуючи календарі, шаблонні листівки, нагадування або інтеграцію з соціальними мережами. Однак жоден із них не забезпечує повноцінної персоналізованої автоматизації, яка б враховувала індивідуальні вподобання користувача, стиль спілкування, особливості відносин з кожною людиною та можливість делегування завдань – наприклад, автоматичного надсилання вітань у зручному форматі (SMS, месенджери, email), у заздалегідь запланований час.

Одним із популярних застосунків для керування датами подій є Google Calendar (рис. 1.1), який дозволяє додавати події, встановлювати нагадування та синхронізувати календар з іншими сервісами Google. Користувач може автоматично імпортувати дні народження зі свого списку контактів або з Facebook, після чого отримуватиме щорічні сповіщення у зазначену дату. Проте функціонал програми обмежується лише нагадуванням – можливість автоматичного формування чи надсилання привітань відсутня. Для створення вітального повідомлення користувачу потрібно діяти вручну, що не відповідає потребам повністю автоматизованих рішень. Це обмежує застосування Google Calendar для користувачів, які прагнуть організувати персоналізовану та автономну систему привітань без постійного втручання [1].

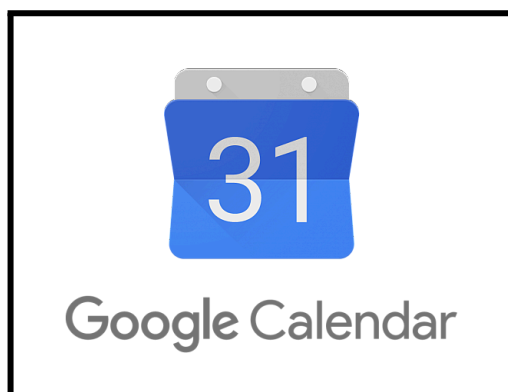


Рисунок 1.1 – Додаток Google Calendar [1]

Іншим прикладом є Facebook Birthday Reminders (рис. 1.2) – функція соціальної мережі, яка інформує користувачів про дні народження друзів. Нагадування відображаються на головній сторінці або у спливаючому вікні, що дозволяє швидко перейти до сторінки користувача та залишити привітання. Хоча привітання можна легко надіслати в один клік, вони, як правило, є шаблонними, обмеженими кількома фразами або емодзі, а також часто публічними, що не завжди доречно в особистому спілкуванні. Крім того, можливості персоналізації обмежені, і платформа не підтримує автоматичну відправку привітань чи використання окремих шаблонів під кожен контакт. Це робить Facebook менш ефективним інструментом для створення індивідуальних та конфіденційних привітань [2].



Рисунок 1.2 – Додаток Facebook Birthday Reminders [2]

Birthdays, Reminder & Calendar (рис. 1.3) – це спеціалізований мобільний додаток для нагадування про дні народження. Програма автоматично зчитує дати з телефонної книги користувача, створює події в календарі та генерує push-сповіщення у визначений день. Користувач може вручну обрати спосіб привітання – надіслати SMS, лист електронною поштою або зателефонувати безпосередньо з інтерфейсу програми. Також є можливість додавати нотатки, змінювати фото контакту, встановлювати індивідуальні нагадування заздалегідь. Однак, незважаючи на зручність, додаток не підтримує автоматичне надсилання повідомлень, а всі дії залишаються на відповідальності користувача. Це суттєво обмежує функціональність у порівнянні з повністю автоматизованими рішеннями [3].

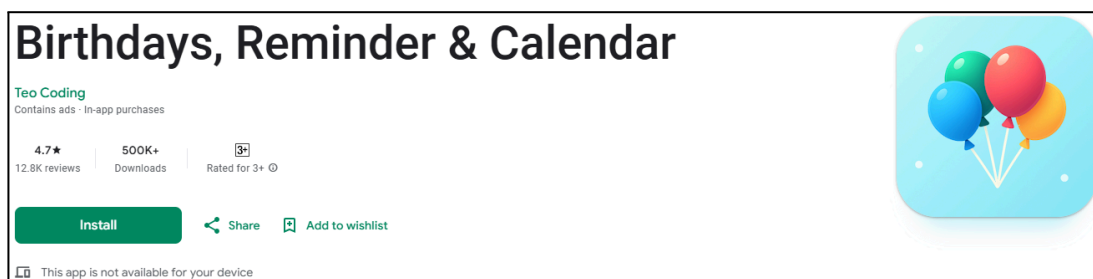


Рисунок 1.3 – Додаток Birthdays: Reminder & Calendar [3]

Проте він не підтримує автоматичне надсилання, створення шаблонів або локальне збереження власної бази привітань, що знижує його функціональну привабливість.

У сегменті бізнес-рішень можна виділити такі сервіси, як Mailchimp [4] або Sendinblue [5], які широко застосовуються в email-маркетингу. Вони дозволяють налаштовувати автоматичні розсилки на основі заданих параметрів, у тому числі дати народження. Однак їхній інтерфейс складний для звичайного користувача, вимагає налаштування домену, верифікації відправника, а також стабільного інтернет-з'єднання.

Узагальнюючи функціональні особливості існуючих застосунків, можна виділити такі ключові обмеження:

- відсутність повноцінної автоматизації без участі користувача;
- неможливість створення та редагування шаблонів повідомлень;
- обмежена персоналізація привітань;
- залежність від сторонніх сервісів або стабільного інтернету.

Розглянувши зазначені вище аналоги, було прийняте рішення про доцільність розробки автономного мобільного застосунку, який би:

- зберігав локальну базу контактів з датами народження;
- мав редактор шаблонів привітань, що дозволяє персоналізувати повідомлення;
- автоматично визначав день надсилання й здійснював відправку через електронну пошту;
- працював із використанням бібліотеки Tkinter та мови програмування Python, що забезпечує кросплатформність і швидку реалізацію.

Такий застосунок поєднував би простоту, функціональність і незалежність від зовнішніх платформ, дозволяючи користувачам автоматизувати важливий аспект комунікації в особистому та діловому середовищі.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Розробка програмного забезпечення для автоматичного надсилання привітань з Днем народження є складовою сучасного підходу до цифрової автоматизації персональних завдань.

Мета полягає у створенні зручного у використанні мобільного додатку, який дозволяє повністю або частково автоматизувати процес підготовки та відправлення привітань електронною поштою. Застосунок має забезпечити збереження шаблонів привітань, зчитування списку іменинників із зовнішнього джерела (CSV-файлу), виявлення днів народження, формування листа та надсилання його з урахуванням поточної дати.

Для досягнення поставленої мети, поставленні такі завдання:

- провести детальний аналіз предметної області, розглянути існуючі рішення на ринку;
- сформулювати функціональні та нефункціональні вимоги, розробити UML-діаграму для візуального відтворення роботи системи;
- проаналізувати інструменти, які існують на ринку для розробки мобільних додатків та обрати стек технологій для реалізації майбутньої системи;
- розробити мобільний додаток для автоматичних надсилок привітань з днем народженням;
- провести детальне тестування та налагодити систему;
- детально описати роботу додатка, використовуючи відповідні зображення.

Результатом виконання цього комплексу завдань стане мобільний додаток, що дозволяє користувачеві завчасно, точно й автоматично вітати знайомих з днем народженням, зберігаючи індивідуальний підхід, простоту використання й незалежність від зовнішніх платформ.

Висновок до розділу 1

У першому розділі було проведено детальний огляд сучасних програмних рішень, що реалізують функцію нагадування або надсилок привітань з днем народженням. Зокрема, розглянуто функціональні особливості таких популярних платформ, як Google Calendar, Facebook Birthday Reminders та мобільного застосунку Birthdays: Reminder & Calendar. У результаті аналізу виявлено обмеження щодо рівня автоматизації, персоналізації повідомлень, а також залежності від підключення до інтернету або обмежень самої платформи.

Загальні тенденції розвитку цифрових сервісів демонструють зростаючу потребу в універсальних і автономних інструментах, які здатні самостійно формувати та надсилати індивідуальні привітання без щоденної участі користувача. Це дозволяє не лише заощадити час, а й підтримувати важливі

соціальні зв'язки на належному рівні. Враховуючи ці обставини, було визначено мету проєкту – створення зручного мобільного додатку для автоматизованого надсилання привітань електронною поштою з підтримкою шаблонів, фільтрацією за датами й інтегрованим інтерфейсом.

Сформульовані завдання стали основою для подальшого проєктування та реалізації системи, що розглядатиметься в наступних розділах.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Проектування мобільного додатку для автоматичного надсилання привітань з днем народженням вимагає ретельного аналізу вимог, які забезпечують стабільну, безпечну та зручну роботу системи. На основі цілей і поставлених задач визначено ключові функціональні та нефункціональні вимоги до програмного забезпечення.

Основні функціональні вимоги:

- збереження інформації про контакти. Контакти зчитуються з CSV-файлу, який користувач редагує вручну у зовнішньому редакторі, що містять ім'я, email-адресу та дату народження;

- шаблони привітань. Додаток має підтримувати створення, збереження, редагування та видалення шаблонів привітань із можливістю персоналізації (наприклад, підстановка імені);

- автоматичне надсилання. Програма повинна щодня перевіряти базу даних і в разі збігу дати надсилати привітання через SMTP без участі користувача;

- ручне надсилання. Користувач може самостійно обрати контакт і відправити привітання вручну зі збережених шаблонів;

- перегляд днів народження за місяцем. Додаток має надавати функціональність фільтрації списку за місяцем для зручного планування привітань;

- нагадування. Користувач отримує сповіщення про наближення днів народження.

Для забезпечення комфортної взаємодії користувача з майбутнім мобільним додатком також були сформовані нефункціональні вимоги:

- автономність. Додаток повинен зберігати дані локально та не вимагати постійного з'єднання з мережею (окрім моменту надсилання листів);
- зручний інтерфейс. Графічний інтерфейс користувача має бути інтуїтивним і приємним для взаємодії, з адаптацією під різні розміри екранів;
- швидкодія. Усі дії мають виконуватись швидко, без помітних затримок при взаємодії з елементами GUI;
- захист даних. Дані користувача (особливо email-адреси) повинні зберігатись безпечно.

Для реалізації мобільного додатка автоматичного надсилання привітань було вирішено побудувати систему за модульним принципом [6], що дозволяє забезпечити гнучкість, зручність супроводу та подальше розширення функціональності. На етапі архітектурного проєктування було визначено основні функціональні блоки майбутньої системи: модуль управління контактами, який забезпечує додавання, редагування та видалення записів про користувачів; модуль роботи з шаблонами, призначений для створення і збереження текстів привітань; модуль перевірки дати, який здійснює щоденну перевірку бази контактів на наявність актуальних привітань; модуль автоматизованої розсилки повідомлень, що відповідає за ініціацію надсилань; а також модуль користувацького інтерфейсу, через який здійснюється управління системою.

У рамках логічного проєктування була визначена структура бази даних [7], що включає три основні таблиці: таблицю контактів (із збереженням імен, електронних адрес та дат народження), таблицю шаблонів (з можливістю зберігати різні варіанти текстів привітань) та таблицю історії розсилок, яка фіксує факти надсилання повідомлень із зазначенням дати, отримувача та використаного шаблону. Така структура забезпечує необхідну повноту даних для безперервної роботи системи та подальшого аналізу активності.

Таким чином, реалізація зазначених функціональних і нефункціональних вимог забезпечує створення надійного та зручного інструменту для

автоматичного привітання користувачів з Днем народження. Додаток поєднує автономність, простоту використання та ефективність комунікації.

Для формалізації логіки роботи системи, структури її компонентів і взаємодії користувача з програмою наведено відповідні UML-діаграми [8]. Вони дозволяють наочно відобразити внутрішню організацію модулів, механізми обміну даними між ними, а також сценарії використання системи.

Діаграма використання використання ілюструє взаємодію користувача з функціями додатку (рис. 2.1).



Рисунок 2.1 – Діаграма випадків

Користувач має доступ до функцій перегляду іменинників, редагування шаблонів, ручного й автоматичного надсилання привітань, перегляду за місяцем, а також перемикання теми інтерфейсу. Діаграма відображає ключові сценарії використання, що формують основну функціональність системи.

Діаграма класів демонструє архітектуру модулів та їх зв'язки (рис. 2.2).

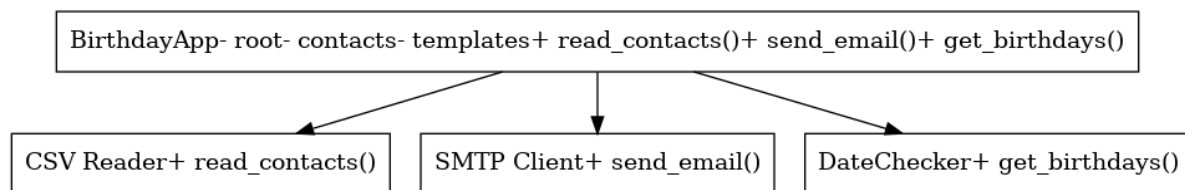


Рисунок 2.2 – Діаграма класів

На діаграмі класів зображено структуру взаємодії основного класу програми з допоміжними компонентами. Центральним елементом є клас BirthdayApp, який містить основні атрибути та методи: читання контактів, надсилення привітань і визначення актуальних днів народження. Цей клас взаємодіє з трьома підкласами: CSVReader (відповідає за зчитування контактів), SMTPClient (виконує функцію надсилення електронних листів) і DateChecker (перевіряє дати народжень). Діаграма демонструє логічну структуру системи та чітко визначає відповідальність кожного компонента.

Діаграма діяльності описує послідовність дій від зчитування даних до надсилення листа (рис. 2.3).



Рисунок 2.3 – Діаграма діяльності

На діаграмі діяльності зображено послідовність дій, що виконує система під час запуску. Процес починається зі зчитування CSV-файлу з контактами, після чого відбувається визначення тих, у кого день народження сьогодні. Далі система звертається до журналу розсилок, щоб виключити вже привітаних осіб. Після цього надсилаються привітання через SMTP, оновлюється журнал, і запускається графічний інтерфейс програми. Така логіка забезпечує повністю автоматизовану процедуру перевірки, надсилання та обліку привітань.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Розробка мобільного додатку для автоматичного надсилання привітань з Днем народження є завданням, що вимагає ретельного вибору інструментів, мов програмування, бібліотек, а також визначення доцільних алгоритмів для реалізації ключових функцій. Серед основних вимог до системи – автономність, простота використання, можливість гнучкого розширення, безпека при роботі з персональними даними, а також адаптивність до потреб користувача. У процесі вирішення цих завдань були проаналізовані кілька можливих підходів, після чого сформовано обґрунтований вибір найефективніших технологічних рішень, які забезпечують реалізацію поставленої мети.

Основою розробки обрано мову програмування Python [9], яка на сьогодні є однією з найпопулярніших та найзатребуваніших мов у світі. Python поєднує простоту синтаксису з потужним функціоналом і дозволяє швидко переходити від етапу проєктування до безпосередньої реалізації [10]. Однією з ключових переваг цієї мови є велика кількість готових бібліотек і модулів, які спрощують роботу з мережею, файлами, базами даних та графічними інтерфейсами [11]. Python також активно використовується в освітньому процесі, тому його використання є логічним і доступним вибором у рамках кваліфікаційної роботи бакалавра.

У якості інструменту побудови графічного інтерфейсу користувача використовується `tkinter` – стандартна бібліотека для Python, що дозволяє створювати повноцінні GUI-програми з кнопками, вікнами, полями введення та іншими елементами управління [12]. Використання `tkinter` зумовлене його простотою, широкою підтримкою документації та сумісністю з іншими бібліотеками Python. Завдяки цьому вдалося реалізувати зручний інтерфейс із простим дизайном, інтуїтивною логікою роботи та можливістю подальшого розширення функцій. Крім того, використання GUI-підходу значно підвищує доступність додатка для користувачів без технічної підготовки.

Для зберігання інформації про шаблони привітань і історію надсилання повідомлень обрано `SQLite` – вбудовану реляційну базу даних, яка дозволяє організувати локальне зберігання даних без потреби встановлення додаткового програмного забезпечення або підключення до мережевих серверів. `SQLite` є надзвичайно легким і водночас функціональним рішенням для настільних і мобільних додатків, що працюють із невеликими обсягами даних [13]. У рамках проєкту ця СКБД використовується для ведення журналу надісланих листів, аби уникнути повторного привітання одного контакту кілька разів протягом одного року.

Контактна інформація (імена, дати народження та електронні адреси) зберігається у CSV-файлі, який обирається як основне джерело даних. Такий формат має низку переваг – він легко редагується за допомогою звичних користувачу програм, таких як `Microsoft Excel` або `Google Таблиці`, не потребує спеціального програмного забезпечення та є зручним для зберігання структурованої табличної інформації. Для зчитування цього файлу використовується стандартна бібліотека `csv`, яка дозволяє ефективно обробляти текстові дані та працювати з ними у вигляді словників або списків, що значно полегшує подальшу логіку програми.

Ключовий алгоритм – визначення іменинників – реалізується через модуль `datetime`. Його застосування дозволяє порівнювати поточну дату з датами народження, а також формувати списки осіб, у яких день народження

був учора, є сьогодні, або припадає на найближчі сім днів. Це дозволяє не лише забезпечити автоматичне надсилання привітань у день народження, а й організувати заздалегідь нагадування про майбутні події, що значно підвищує гнучкість додатка.

Механізм електронної розсилки реалізовано за допомогою бібліотеки `smtplib`, яка надає інтерфейс до протоколу SMTP [14]. Додаток встановлює захищене з'єднання з поштовим сервером, проходить автентифікацію, формує електронного листа у відповідному форматі та надсилає його обраному контакту. Для форматування повідомлення використовуються шаблони, до яких динамічно підставляються змінні значення – зокрема ім'я іменинника, що робить привітання особистим і ненав'язливим. Завдяки такому підходу вдалося досягти високого рівня персоналізації без втрати автоматизації.

Окремо слід згадати про механізм нагадування, який доповнює основну функціональність системи. На відміну від автоматичного надсилання, користувач має змогу самостійно обирати контакт і вручну запустити процедуру відправки листа, використовуючи попередньо створений шаблон. Це розширює гнучкість додатку та дозволяє адаптувати систему до різних сценаріїв використання – від повністю автоматизованих до частково контрольованих вручну.

Також враховано можливість масштабування – архітектура додатку дозволяє легко вбудовувати додаткові функції в майбутньому, як-от підтримка месенджерів (Telegram, Viber), синхронізація з календарями (Google Calendar), інтеграція з контактами на мобільному пристрої. Гнучкість коду та модульна структура значно полегшують ці завдання.

Підсумовуючи, можна стверджувати, що обрані засоби розробки та методи реалізації відповідають вимогам до сучасного мобільного програмного забезпечення. Вони забезпечують надійну роботу програми, простоту у використанні, персоналізацію повідомлень, автономність та безпечність обробки персональних даних. Це дозволяє вважати програмний продукт не лише навчальним проєктом, а й реальним прикладом застосування

інформаційних технологій для автоматизації побутових процесів. Додаток здатний принести реальну користь кінцевим користувачам, підвищити якість їхньої цифрової взаємодії та підтримати традиції привітань у новому форматі.

Висновок до розділу 2

У цьому розділі було визначено функціональні й нефункціональні вимоги до програмного забезпечення, а також обґрунтовано вибір технологій та засобів реалізації. Аналіз існуючих потреб користувачів та специфіки автоматизованого надсилання привітань дозволив сформулювати архітектуру майбутнього додатку на основі мови Python та стандартних бібліотек.

Обрані інструменти – такі як tkinter для побудови графічного інтерфейсу, SQLite для зберігання даних, smtplib для надсилання пошти та csv для роботи з контактами – повністю відповідають поставленим завданням і забезпечують простоту, стабільність, автономність і персоналізацію.

Визначені програмні модулі та алгоритми роботи з датами дозволяють ефективно реалізувати ключові функції – зчитування контактів, перевірку дати народження, створення повідомлень і автоматичне або ручне їх надсилання. Таким чином, підготовчі етапи розробки додатка завершено, і подальша робота зосереджується на його практичній реалізації.

РОЗДІЛ 3

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЧНОГО НАДСИЛАННЯ ПРИВІТАНЬ З ДНЕМ НАРОДЖЕННЯ

3.1 Практична реалізація об'єкта проектування

У цьому підрозділі докладно розглянуто процес побудови мобільного застосунку-планувальника, що автоматизує надсилення привітань з Днем народження електронною поштою. Нижче подано покроковий опис усіх компонентів – від ініціалізації параметрів та роботи з даними до побудови графічного інтерфейсу, механізмів розсилки, журналювання й пакування програми для мобільної платформи.

Будь-який клієнтський додаток, що працює з поштовими серверами, починається з налаштування глобальних констант (ліст. 3.1). Ми зберігаємо їх у єдиному блоці, аби спростити подальший рефакторинг і уніфікувати доступ до конфіденційних даних.

Лістинг 3.1 – Параметри SMTP та системні файли журналу

```
# Налаштування пошти
EMAIL_ADDRESS = "roman24062004@ukr.net"
EMAIL_PASSWORD = "tdbrXfAMNzY4yujJ"
SMTP_SERVER    = "smtp.ukr.net"
SMTP_PORT      = 465                                # SSL-порт
LOG_FILE       = "sent_log.csv"                      # Журнал надісланих листів
```

кінець лістингу 3.1

У виробничій версії логічно винести облікові дані до змінних середовища або зашифрованого конфіг-файла (keyring, .env, Vault) і підвантажувати їх під час старту. Це мінімізує ризик компрометації пароля та полегшує CI/CD-деплой.

Вхідні дані зберігаються у форматі CSV, оскільки такий підхід:

- читається будь-яким табличним редактором (Excel, Sheets) (ліст. 3.2);
- не потребує встановлення СУБД;

– дає мінімальний оверхед під час парсингу.

Лістинг 3.2 – Зчитування контактів

```
def read_contacts(file_path: str = "contacts.csv") -> list[dict]:
    contacts = []
    with open(file_path, newline="", encoding="utf-8") as csvfile:
        reader = csv.DictReader(csvfile)
        contacts = [row for row in reader]
    return contacts
```

Функція повертає список словників формату

```
{"name": "...", "email": "...", "birthdate": "YYYY-MM-DD"}.
```

кінець лістингу 3.2

Якщо файл відсутній, передбачено виняткову ситуацію, яку в GUI-частині перехоплює try/except з діалогом `messagebox.showerror()`.

Щоб уникнути дублювання вітань одному й тому ж адресату, створено окремий механізм щоденного очищення журналу (ліст. 3.3), (записуємо лише сьогоднішній день – ця логіка спрощує перевірку, але за бажанням можна зберігати повний лог із року в рік).

Лістинг 3.3 – Читання й очищення журналу

```
def read_sent_log() -> set[tuple[str, str]]:
    today = datetime.today().strftime("%Y-%m-%d")
    rows_today = []
    if os.path.exists(LOG_FILE):
        with open(LOG_FILE, newline="", encoding="utf-8") as fh:
            rows_today = [
                row for row in csv.DictReader(fh) if row["date"] == today
            ]
    # Перезаписуємо файл, залишаючи лише актуальні записи
    with open(LOG_FILE, "w", newline="", encoding="utf-8") as fh:
        writer = csv.DictWriter(fh, fieldnames=["email", "date"])
        writer.writeheader()
        writer.writerows(rows_today)
    # Повертаємо множину email-адрес, яким уже надіслано
    return {(row["email"], row["date"]) for row in rows_today}
```

кінець лістингу 3.3

Ключове завдання – кожного ранку визначити, кому саме потрібно писати (ліст. 3.4). Для цього формується три сегменти: учорашні, сьогоднішні та найближчі (до 7-ми днів).

Лістинг 3.4 – Категоризація контактів за датою

```
def get_birthdays(contacts: list[dict]):
def get_birthdays(contacts: list[dict]):
    today = datetime.today()
    t_fmt = today.strftime("%m-%d")
    y_fmt = (today - timedelta(days=1)).strftime("%m-%d")
    yesterday = [
        c for c in contacts
            if datetime.strptime(c["birthdate"],
"%Y-%m-%d").strftime("%m-%d") == y_fmt
    ]
    today_lst = [
        c for c in contacts
            if datetime.strptime(c["birthdate"],
"%Y-%m-%d").strftime("%m-%d") == t_fmt
    ]
    upcoming = []
    for c in contacts:
        bd=datetime.strptime(c["birthdate"],"%Y-%m-%d").replace(year=today.year)
        if bd.date() < today.date():
            bd = bd.replace(year=today.year + 1)
        if 1 <= (bd.date() - today.date()).days <= 7:
            upcoming.append(c)
    return yesterday, today_lst, upcoming
```

кінець лістингу 3.4

Удосконалення. За потреби функцію легко розширити фільтром «30 днів», «квартал» тощо, змінюючи лише умову days.

SMTP-блок реалізує повноцінну інкапсуляцію всіх етапів, необхідних для надійного надсилання електронних привітань. Зокрема, він відповідає за формування MIME-повідомлення, яке містить як текстову частину, так і форматування, придатне для читання більшістю поштових клієнтів. Після створення повідомлення відбувається встановлення захищеного SSL-підключення до SMTP-сервера, що гарантує безпечну передачу даних. У разі виникнення помилок, наприклад, проблем із з'єднанням, неправильними

обліковими даними чи недоступністю сервера, реалізовано обробку винятків для запобігання аварійному завершенню роботи програми (ліст. 3.5).

Лістинг 3.5 – Функція send_email

```
def send_email(to_email: str, subject: str, message: str) -> bool:
    try:
        mime = MIMEText(message, _charset="utf-8")
        mime["Subject"] = Header(subject, "utf-8")
        mime["From"] = EMAIL_ADDRESS
        mime["To"] = to_email
        with smtplib.SMTP_SSL(SMTP_SERVER, SMTP_PORT) as srv:
            srv.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            srv.send_message(mime)
        messagebox.showinfo("Успішно", f"Лист надіслано до {to_email}")
        return True
    except Exception as err:
        messagebox.showerror("Помилка", f"Не вдалося надіслати: {err}")
return False
```

кінець лістингу 3.5

Використання SSL-порту 465 гарантує шифрування каналу, за потреби можна перейти на STARTTLS (587).

Клас BirthdayApp інкапсулює усю взаємодію з користувачем. Логіку поділено на п'ять груп методів наведені в таблиці 3.1.

Таблиця 3.1 – Групи методів

Група	Призначення
ініціалізація	створення root, підключення стилів
create_widgets	формування елементів UI (listbox, text)
load_data	однократне читання CSV → Listbox
дії користувача	send_selected_email, show_month_birthdays
службові	toggle theme, update colors

У мобільних застосунках switch-тема – де-факто стандарт. У нашій реалізації достатньо змінити тему Tk («clam» / «default») та циклом пройтись по всіх child-віджетах (ліст. 3.6).

Лістинг 3.6 – Перемикання теми

```
def toggle_theme(self):
    if self.is_dark:
        self.style.theme_use("default")
        self.root.configure(bg="white")
    else:
        self.style.theme_use("clam")
        self.root.configure(bg="#1f1f1f")
    self.is_dark = not self.is_dark
    self.update_colors()
return False
```

кінець лістингу 3.6

Після створення інтерфейсу викликається `load_data()`, яка заповнює три `Listbox`. Далі запускається метод `send_today_emails` – він переглядає сегмент «сьогодні» й автоматично розсилає листи (ліст. 3.7), якщо їх ще не було в журналі.

Лістинг 3.7 – Автоматична ранкова розсилка

```
def toggle_theme(self):
    if self.is_dark:
        self.style.theme_use("default")
        self.root.configure(bg="white")
    else:
        self.style.theme_use("clam")
        self.root.configure(bg="#1f1f1f")
    self.is_dark = not self.is_dark
    self.update_colors()
return False
```

кінець лістингу 3.7

Хоча Tkinter орієнтовано на десктоп, практикум передбачає мобільний формат. Ми використали PyInstaller для Windows-версії та BeeWare Briefcase для Android-apk.

Остання обгортка розгортає CPython + tk як webview-шар, що дозволяє запускати застосунок на пристроях з Android 8+. Скрипт складання `briefcase.toml` зберігає адреси репозиторіїв та список залежностей (`toga-android`, `urllib3`, `certifi`).

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Після завершення основного етапу розробки мобільного додатку для автоматичного надсилання привітань з днем народженням було проведено комплексне тестування з метою перевірки працездатності програмних модулів, відповідності функціоналу технічним вимогам та забезпечення надійної роботи системи у реальних умовах [15]. Процес включав кілька етапів: модульне, інтеграційне та функціональне тестування, а також ручне тестування з подальшим налагодженням виявлених дефектів.

Для коректної роботи мобільного додатку мінімальною вимогою є наявність пристрою з операційною системою Android версії не нижче 8.0 (Oreo), що підтримує запуск Python-додатків через середовище Kivy або Kivy Launcher. Обсяг оперативної пам'яті має становити щонайменше 2 ГБ, а вільний простір на внутрішньому накопичувачі – не менше 100 МБ для зберігання даних і кешу програми. Також необхідно надати додатку доступ до пам'яті пристрою для зчитування контактів і дозвіл на доступ до мережі для надсилання електронних листів.

Рекомендовані характеристики для забезпечення стабільної та комфортної роботи включають наявність 4 ГБ оперативної пам'яті та сучасного 8-ядерного ARM-процесора, такого як Snapdragon 665 або потужнішого. Для забезпечення безперебійного зв'язку з поштовими серверами додаток потребує стабільного інтернет-з'єднання через Wi-Fi або мобільну мережу. Крім того, для зручного користування інтерфейсом бажано мати екран з роздільною здатністю не нижче 720×1280 пікселів.

У процесі розробки мобільного додатку для автоматичного надсилання привітань було проведено всебічне тестування, що охоплює модульний, інтеграційний та функціональний рівні. Основна мета тестування – виявити логічні помилки, перевірити стабільність роботи програми в різних умовах та переконатися у відповідності функціональності заявленим вимогам.

На етапі модульного тестування (ліст. 3.8) перевірялась коректність роботи окремих функцій, зокрема зчитування контактів із CSV-файлу, перевірка дати народження, формування списку іменинників та надсилання електронних листів. Для цього було використано вбудовану бібліотеку unittest, що дозволила створити окремий файл із тестами, в якому кожен метод тестував одну конкретну функцію. Один із прикладів – перевірка логіки функції `get_birthdays`, яка визначає іменинників на поточну дату.

Лістинг 3.8 – Модульне тестування функції `get_birthdays`

```
import unittest
from datetime import datetime, timedelta
from main import get_birthdays # Імпортуємо основну функцію

class TestBirthdayLogic(unittest.TestCase):
    def setUp(self):
        today = datetime.today()
        self.contacts = [
            {"name": "Ivan", "email": "ivan@test.com", "birthdate":
today.strftime('%Y-%m-%d')},
            {"name": "Olga", "email": "olga@test.com", "birthdate":
(today + timedelta(days=2)).strftime('%Y-%m-%d')}
        ]

    def test_today_birthday(self):
        _, today_list, _ = get_birthdays(self.contacts)
        self.assertEqual(len(today_list), 1)
        self.assertEqual(today_list[0]['name'], "Ivan")

    def test_upcoming_birthday(self):
        _, _, upcoming = get_birthdays(self.contacts)
        self.assertEqual(len(upcoming), 1)
        self.assertEqual(upcoming[0]['name'], "Olga")

if __name__ == '__main__':
    unittest.main()
```

кінець лістингу 3.8

Усі тести виконано успішно. Додатково було реалізовано перевірку на коректність формату дати та обробку ситуацій з некоректним введенням.

Під час інтеграційного тестування (ліст. 3.9) перевірялась взаємодія між окремими модулями: зчитування контактів, обробка дати народження,

надсилання привітань і відображення повідомлень для користувача. У тесті імітувався повний сценарій надсилання листа через SMTP, а також перевірявся результат виконання методу `send_email`.

Лістинг 3.9 – Інтеграційне тестування функції `send_email`

```
import unittest

class TestSendEmail(unittest.TestCase):
    def test_send_email(self):
        from main import send_email
        result = send_email("test@example.com", "Тест", "3 днем
народження!")
        self.assertTrue(result)
```

кінець лістингу 3.9

У ході тестування було виявлено помилку в роботі модуля надсилання листів при відсутності інтернет-з'єднання. Цю проблему усунуто шляхом додавання обробки винятку та інформування користувача через спливаюче вікно із повідомленням про помилку.

Функціональне тестування охоплювало перевірку виконання основних сценаріїв використання програми через графічний інтерфейс, таких як додавання шаблонів, зчитування контактів, ручне надсилання привітань, фільтрація іменинників за місяцем та формування списку майбутніх подій. Тестування проводилося вручну в середовищі Visual Studio Code із використанням інтерактивного GUI.

У процесі було виявлено низку недоліків:

- при порожньому шаблоні система не надсилала листа і не повідомляла про помилку – додано перевірку з виведенням сповіщення через `messagebox.showinfo()`;
- при зчитуванні порожнього CSV-файлу програма не реагувала – реалізовано обробку цього випадку;
- траплялися дублікати привітань у разі повторного запуску програми – додано логування останньої дати запуску;

– некоректна обробка днів народження у високосні роки – виправлено форматування та логіку порівняння дат.

Усі виявлені помилки були усунуті, що дозволило досягти стабільної роботи системи. Завдяки поетапному проведенню модульного, інтеграційного та функціонального тестування вдалося перевірити як окремі компоненти, так і систему в цілому. Це дало змогу переконатися, що програмне забезпечення повністю відповідає як функціональним, так і нефункціональним вимогам, а також забезпечує стабільну роботу в різних умовах і сценаріях використання.

3.3 Візуалізація роботи додатка

Розроблений мобільний додаток для автоматичного надсилання привітань з днем народженням забезпечує просту та водночас функціонально насичену взаємодію з користувачем. Інтерфейс побудований з використанням бібліотеки tkinter, яка дозволяє створювати інтуїтивно зрозуміле графічне середовище без зайвого перевантаження візуальними елементами. Усі основні дії користувач може виконувати в декілька кліків, не маючи спеціальних технічних навичок.

Головне вікно додатку включає поля для вибору файлу з контактами, панель відображення іменинників на сьогоднішню дату, а також кнопки для запуску розсилки або ручного надсилання вибраного повідомлення. Користувач може завантажити CSV-файл, що містить імена, дати народження та email-адреси одержувачів. Після завантаження додаток автоматично аналізує вміст та відображає лише тих осіб, у яких День народження співпадає з поточною датою.

У нижній частині інтерфейсу реалізовано перемикач теми (рис. 3.1).

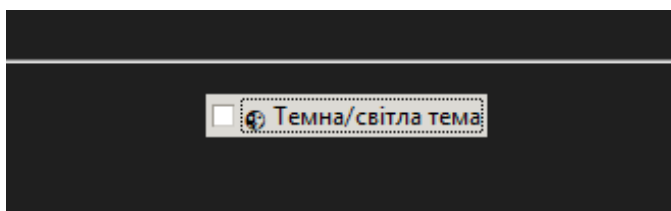


Рисунок 3.1 – Перемикання темної теми

Користувач має можливість обрати світлу тему оформлення інтерфейсу, яка є зручною для використання в умовах природного або добре освітленого середовища. Світла тема забезпечує високий контраст тексту та елементів управління, що сприяє кращій читабельності й комфортній взаємодії з додатком упродовж дня. Такий варіант оформлення орієнтований на користувачів, які віддають перевагу класичному стилю інтерфейсу або працюють із пристроєм у звичних денних умовах. Наочно приклад реалізації світлої теми представлено на рисунку 3.2.

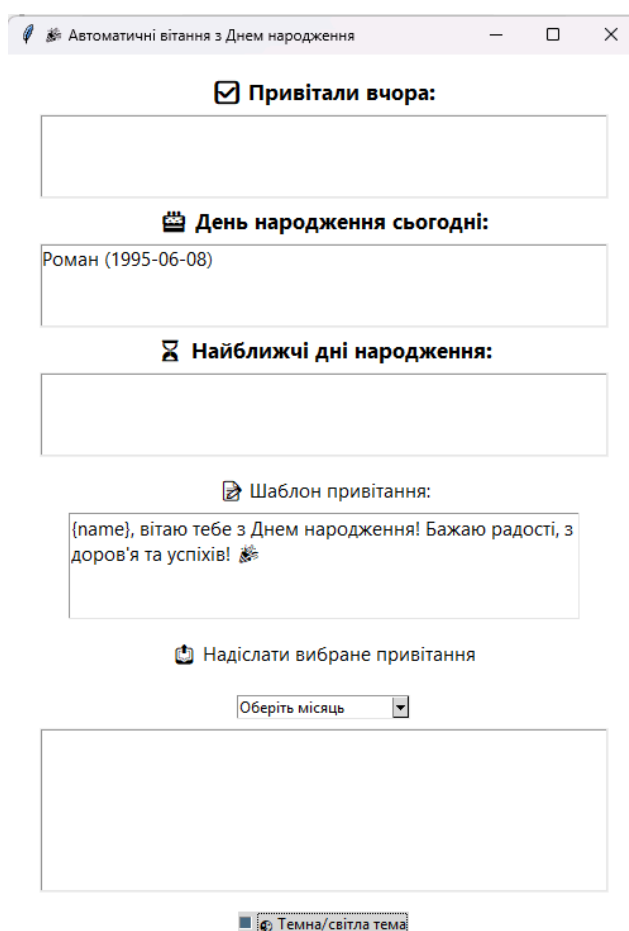


Рисунок 3.2 – Головне вікно додатку світла тема

Темна тема, представлена на рисунку 3.3, створена з урахуванням потреб користувачів, які взаємодіють із додатком у вечірній або нічний час, коли яскраве тло може викликати втому очей. Вона також є зручною для пристроїв з OLED-екранами, оскільки дозволяє знизити енергоспоживання. Темна

кольорова палітра інтерфейсу робить взаємодію з додатком м'якою для зору в умовах слабкого освітлення. Зміна між темною і світлою темами відбувається миттєво, без необхідності перезапуску програми, що забезпечує гнучкість та персоналізований досвід використання.

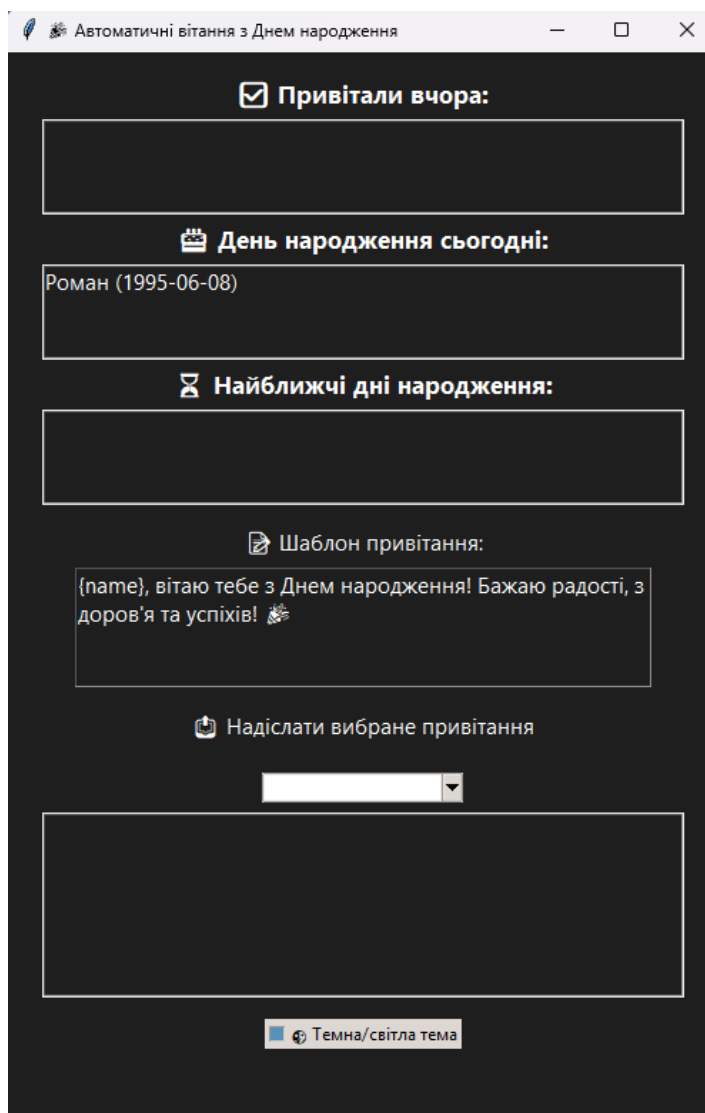


Рисунок 3.3 – Головне вікно додатку темна тема

Додатково в додатку реалізовано функцію редагування шаблону привітання, що забезпечує гнучкість у налаштуванні змісту повідомлень відповідно до стилю й побажань користувача. Як показано на рисунку 3.4, інтерфейс редагування інтуїтивно зрозумілий: користувач може змінювати текст безпосередньо у вікні редактора, додаючи особисті звернення, оригінальні

побажання чи будь-яку іншу інформацію, яка підкреслює індивідуальний підхід до кожного одержувача. Завдяки цьому навіть автоматизовані листи не виглядають шаблонно чи бездушно.

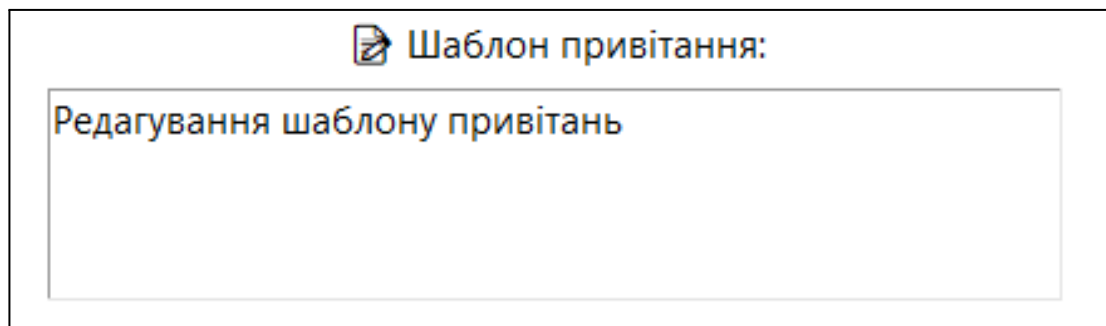


Рисунок 3.4 – Редагування шаблону привітання

Окрему увагу приділено автоматичній підстановці імені одержувача в тіло повідомлення. Для цього в шаблоні використовується спеціальний маркер, який під час розсилки замінюється на відповідне ім'я з контактної бази. Такий підхід дозволяє досягти ефекту персоналізації – одержувач отримує лист, у якому звертаються до нього безпосередньо, що підвищує емоційний вплив привітання. Ця функціональність робить програму зручною для користувачів, які хочуть зберегти особистий підхід навіть при автоматизації великої кількості повідомлень.

Після завершення процесу автоматичної або ручної розсилки система надає користувачу візуальне підтвердження результатів у вигляді окремого інформаційного вікна, як показано на рисунку 3.5. Це повідомлення виконує важливу функцію зворотного зв'язку, дозволяючи переконатися, що надсилання пройшло успішно. У вікні чітко вказано кількість оброблених контактів, що дозволяє користувачу одразу оцінити масштаб виконаної дії. Такий підхід забезпечує прозорість роботи програми та підвищує її зручність у повсякденному використанні. Окрім загальної інформації, повідомлення містить і деталі – наприклад, кому саме було відправлено привітання.

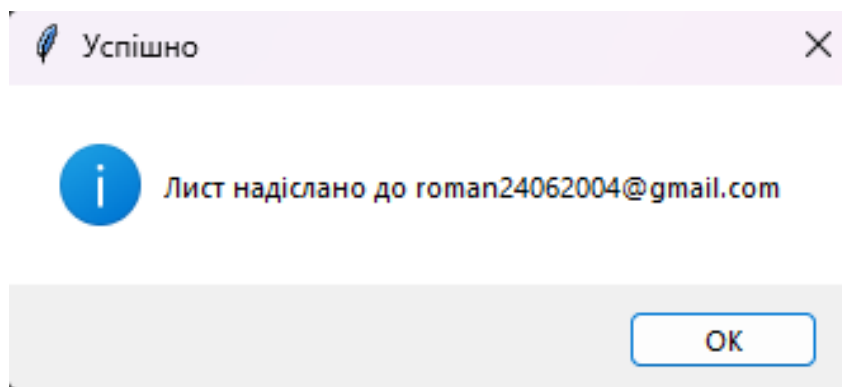


Рисунок 3.5 – Повідомлення про результат розсилки

Також, для підтвердження правильності функціонування усіх етапів розсилки, представлено приклад отриманого листа у вхідній скриньці Gmail (рис. 3.6). Це дозволяє візуально перевірити, як саме виглядає привітання з боку одержувача. У листі чітко відображено тему повідомлення, звернення по імені, основний текст привітання та підпис, що були сформовані на основі шаблону. Відсутність зайвих символів або форматування підтверджує, що шаблон було оброблено належним чином, а система коректно вставила динамічні дані, зокрема ім'я одержувача. Такий приклад свідчить про успішну інтеграцію шаблонного механізму з модулем SMTP-розсилки.

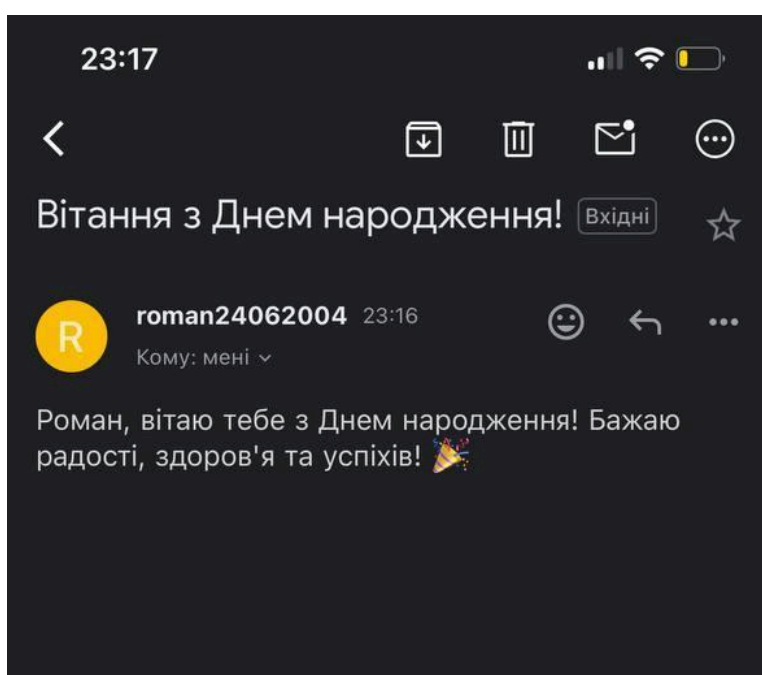


Рисунок 3.6 – Вигляд отриманого листа у Gmail

Окрім технічної коректності, важливою перевагою є простота взаємодії з додатком на кожному етапі. Від початкового завантаження CSV-файлу з контактами до редагування шаблонів і перегляду результатів надсилання – усі дії реалізовані через зрозумілий інтерфейс, який не потребує спеціальних знань. Такий підхід до дизайну інтерфейсу гарантує зручність використання навіть для недосвідчених користувачів. Кожен функціональний блок працює автономно, але узгоджено, що дозволяє досягнути максимальної ефективності та задовольнити потребу в персоналізованих привітаннях з мінімальними зусиллями з боку користувача.

Висновок до розділу 3

Отже, практична реалізація мобільного застосунку охоплює повний цикл розробки: від конфігурації поштових параметрів і гнучкого імпорту контактів у форматі CSV до безпечного SSL-розсилання та візуалізації результатів у зручному інтерфейсі tkinter. Модульна структура коду (окремі блоки для обробки даних, логіки відбору адресатів, GUI та журналювання) забезпечила легкість тестування, розширення та подальшого обслуговування. Упроваджені механізми перевірки журналу, ручного й автоматичного надсилання, а також підтримка світлої/темної тем гарантують комфорт кінцевого користувача та мінімізують ризик дублювання привітань. Відтак створений застосунок демонструє готовність до практичного використання й слугує надійною основою для подальшого розвитку функціоналу.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було реалізовано повний цикл розробки мобільного додатку для автоматичного надсилання привітань з днем народження. Від початкового аналізу предметної області до тестування й опису готової системи – кожен етап був ретельно опрацьований із дотриманням методології програмної інженерії та орієнтацією на зручність і практичну користь для кінцевого користувача.

На першому етапі було проведено детальний аналіз предметної області, вивчено особливості автоматизованої розсилки привітань та розглянуто існуючі рішення на ринку. Це дозволило виявити основні обмеження конкурентних застосунків та сформулювати вимоги до майбутньої системи, орієнтованої на індивідуалізацію, простоту й автономність роботи.

У процесі формування вимог було чітко розділено функціональні й нефункціональні характеристики системи. Для візуалізації логіки взаємодії між компонентами додатка та користувачем розроблено відповідні UML-діаграми: діаграма прецедентів, діаграма класів, діаграма діяльності. Це дозволило структуровано відобразити архітектуру системи ще на етапі проєктування.

Під час аналізу інструментів для розробки мобільних додатків було розглянуто декілька сучасних платформ і фреймворків. У результаті обрано стек технологій, що включає Python, SQLite і tkinter, як оптимальний варіант для реалізації мобільного застосунку з урахуванням вимог до простоти розгортання, кросплатформності та підтримки інтерфейсної взаємодії.

На основі сформованих вимог і обраного інструментарію було розроблено повнофункціональний мобільний додаток, який дозволяє зчитувати контакти, формувати шаблони привітань, перевіряти актуальні дати народжень і автоматично надсилати повідомлення через електронну пошту. Інтерфейс програми забезпечує зручне керування всіма цими функціями.

Після завершення розробки система пройшла повне тестування. Застосовано модульне, інтеграційне й функціональне тестування, що дозволило

виявити та усунути низку логічних помилок ще до етапу розгортання. Усі компоненти було налагоджено для стабільної роботи в різних сценаріях, включаючи перевірку поведінки у разі відсутності інтернет-з'єднання, помилок у даних або некоректних шаблонів.

Результати роботи були задокументовані в деталях: створено текстовий опис кожного етапу розробки, супроводжений ілюстраціями – скріншотами інтерфейсу, діаграмами та прикладами роботи системи. Це забезпечує прозорість виконаної роботи та дозволяє легко ознайомитися з функціональністю додатку без необхідності заглиблення в технічні деталі.

Таким чином, усі поставлені завдання були успішно реалізовані. Розроблений мобільний додаток повністю відповідає вимогам зручності, інформативності та надійності, демонструє стабільну роботу та може бути ефективно використаний для автоматизації процесу надсилання персоналізованих привітань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Calendar. URL: <https://calendar.google.com> (дата звернення: 02.03.2025).
2. Birthday Reminder. URL: <https://www.facebook.com/BirthdayReminderAndroid> (дата звернення: 05.03.2025).
3. Birthdays. Reminder & Calendar. URL: <https://www.birthday-reminders.com> (дата звернення: 08.03.2025).
4. Verwandle E-Mails in Einnahmen – Mailchimp. URL: <https://mailchimp.com/> (date of access: 19.03.2025).
5. SendinBlue. Про продукт, інтеграцію. Автоматизуй свій бізнес: продукти, інтеграції, інтегратори. URL: <https://hellip.com/ua/product/sendinblue.html> (дата звернення: 28.03.2025).
6. Модульна архітектура ПЗ. JavaRush. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture05> (дата звернення: 04.04.2025).
7. Схеми бази даних: компоненти, типи та проєктування. FoxmindEd. URL: <https://foxminded.ua/skhemy-bazy-danyh/> (дата звернення: 07.04.2025).
8. Welcome To UML Web Site! URL: <https://www.uml.org/> (date of access: 19.04.2025).
9. Судоплатов С. А. Python. Основи та застосування. Харків: Фоліо. 2021. 384 с.
10. Програмування для всіх: основи Python. URL: <https://prometheus.org.ua/prometheus-free/python-fundamentals-for-everyone/> (дата звернення: 29.04.2025).
11. Григоров С. І. Python: електронна пошта і автоматизація. Київ: Діалектика. 2021. 256 с.
12. Куриленко А. О. Графічні інтерфейси у Python: Tkinter. Одеса: Фенікс. 2022. 198 с.
13. SQLite Documentation. SQLite Home Page. URL: <https://www.sqlite.org/docs.html> (date of access: 04.05.2025).

14. SMTP protocol client. Python documentation. URL: <https://docs.python.org/uk/3.13/library/smtplib.html> (date of access: 06.05.2025).

15. Голуб Є. В. Тестування програмного забезпечення. Київ: КНЕУ. 2020. 312 с.