



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 3
з дисципліни «Технології розроблення програмного забезпечення»
Тема: «Основи проектування розгортання»

Виконала:

Студентка групи ІА-31

Соколова Поліна

Мета: Навчитися проєктувати діаграми розгортання та компонентів для системи що проєктується, а також розробляти діаграми взаємодії, а саме діаграми послідовностей, на основі сценаріїв зроблених в попередній лабораторній роботі.

15. E-mail клієнт

E-mail клієнт - це програма для управління електронною поштою, яка повинна забезпечувати:

- Підтримку протоколів POP3, SMTP, IMAP
- Автоналаштування для українських провайдерів (Gmail, ukr.net, i.ua)
- Організацію повідомлень у папки/категорії/важливість
- Роботу з чернетками
- Управління прикріпленнями до повідомлень

Патерни проєктування для використання:

Singleton - для менеджера конфігурації, Builder - для створення складних email повідомлень, Decorator - для додаткових функцій повідомлень, Template Method - для різних протоколів email, Interpreter - для обробки email фільтрів, Client-Server - архітектура взаємодії з поштовими серверами.

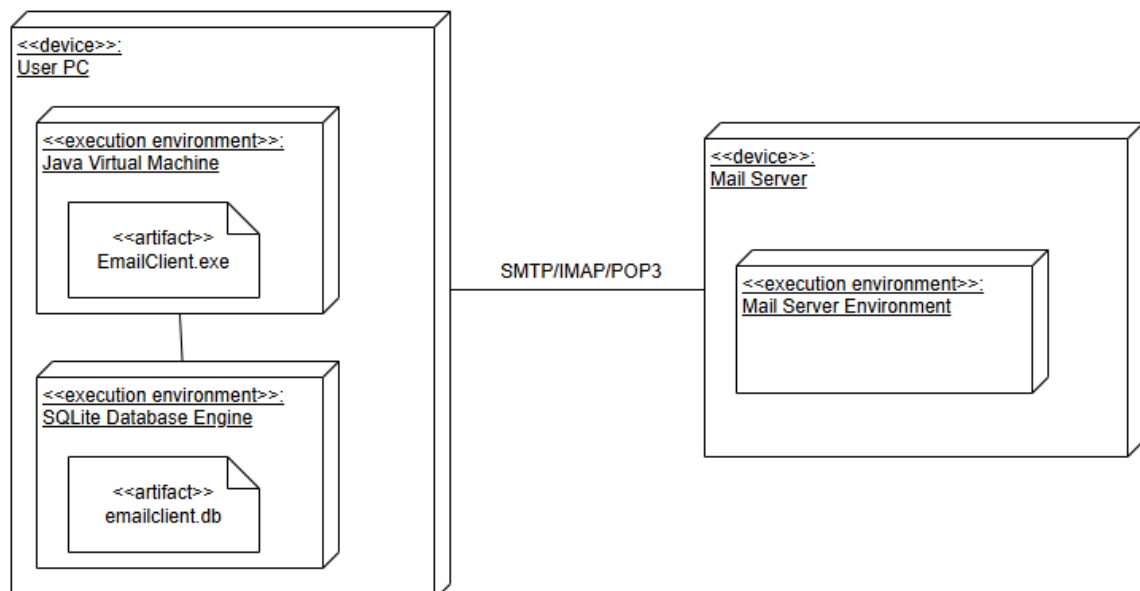


Рис.1 Діаграма розгортання (Deployment Diagram)

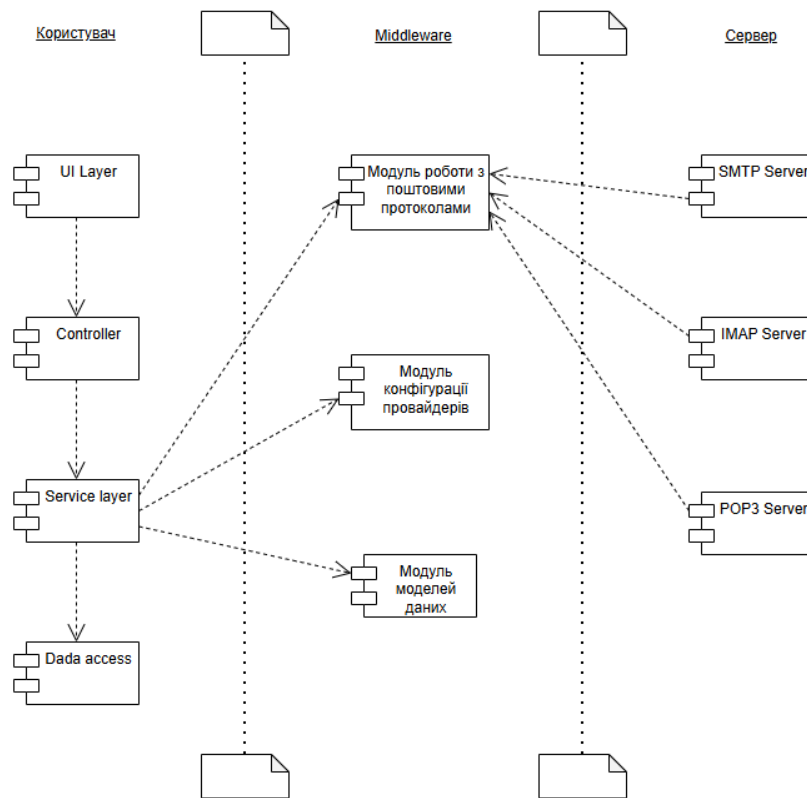


Рис. 2 Діаграма компонентів

Діаграми послідовностей

Сценарій 1: Налаштування облікового запису

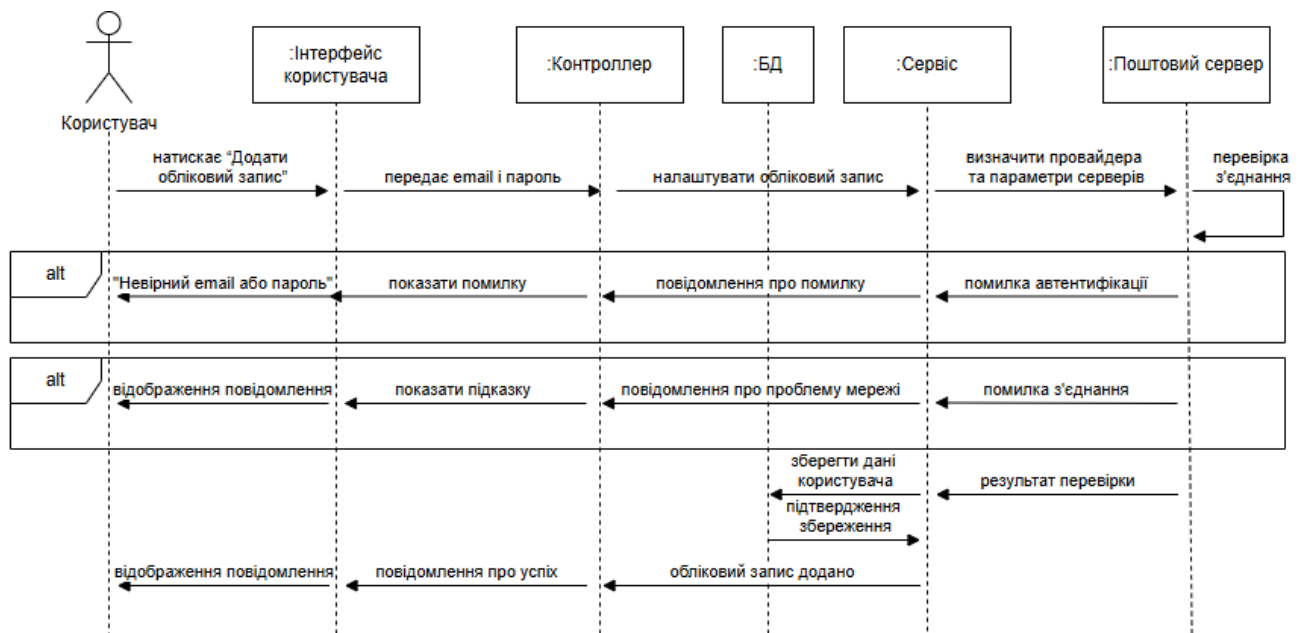


Рис. 3 Діаграми послідовностей.

Сценарій 1: Налаштування облікового запису

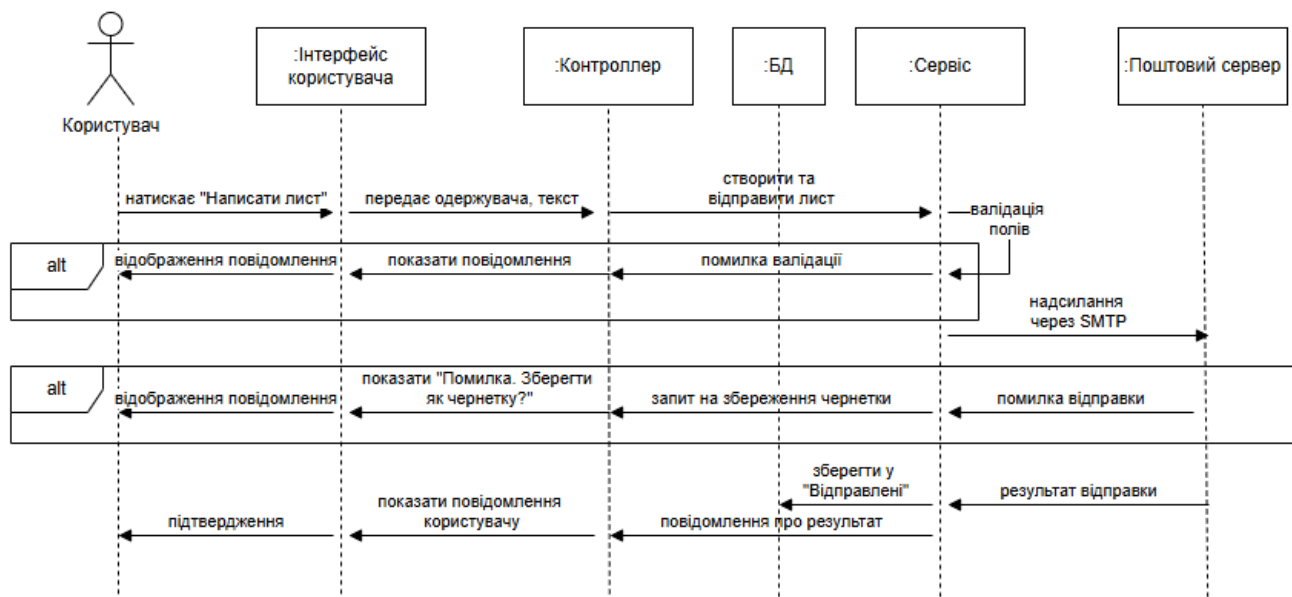


Рис. 4 Діаграми послідовностей.

Сценарій 2: Створення та відправка повідомлення

Email Client - Users

Додати користувача

Ім'я користувача

Email

Пароль

Зберегти

Переглянути деталі

ID	Ім'я	Email	
1	gggppp	test@gmail.com	
2	deeeett	reeeefff@gmail.com	
3	eryyyu	aassd@gmail.com	

WHERE

ORDER BY

	id	name	email	password
1	1	gggppp	test@gmail.com	12345
2	2	deeeett	reeeefff@gmail.com	567778
3	3	eryyyu	aassd@gmail.com	12345

Рис. 5 Користувацьке відображення сценарію 1

Відповіді на питання:

1. Що собою становить діаграма розгортання?

Діаграма розгортання представляє фізичне розташування системи, показуючи, на якому фізичному обладнанні запускається та чи інша складова програмного забезпечення. Вона відображає архітектуру системи з точки зору апаратного та програмного забезпечення.

2. Які бувають види вузлів на діаграмі розгортання?

Вузли бувають двох типів: пристрій (фізичне обладнання) і середовище виконання (програмне забезпечення).

3. Які бувають зв'язки на діаграмі розгортання?

Зв'язки між вузлами зображують у вигляді прямої лінії. Зв'язок може бути множинним для демонстрації підключення 2-х і більше вузлів до іншого.

Назва зв'язку містить спосіб зв'язку між двома вузлами (назва протоколу/технології)

4. Які елементи присутні на діаграмі компонентів?

Компоненти – окремі самостійні модулі системи (можуть бути представлені як логічні, фізичні або виконувані файли).

5. Що становлять собою зв'язки на діаграмі компонентів?

Зв'язки на діаграмі компонентів показують залежності між компонентами.

Залежності показують, що класи з одного компонента використовують класи з іншого компонента. Це допомагає зрозуміти, які компоненти повинні бути зібрані в інсталяційний пакет, і як зміни в одному компоненті вплинуть на інші компоненти.

6. Які бувають види діаграм взаємодії?

Можуть бути трьох видів залежно від способу поділу на модулі:

Логічні – система віртуально уявляється як набір самостійних, автономних модулів

Фізичні – показують компоненти та залежності між ними (фізичні файли)

Виконувані – кожен компонент являє собою деякий файл (виконувані файли, файли вихідних кодів, сторінки html, бази даних тощо)

7. Для чого призначена діаграма послідовностей?

Діаграма послідовностей призначена для моделювання взаємодії між об'єктами системи у певній послідовності часу. Вона відображає, як об'єкти обмінюються повідомленнями, показуючи порядок і логіку виконання операцій. Діаграми є корисними для моделювання бізнес-процесів, проєктування архітектури систем і тестування.

8. Які ключові елементи можуть бути на діаграмі послідовностей?

Актори – користувачі чи інші системи, які взаємодіють із системою.

Об'єкти або класи – розміщуються горизонтально на діаграмі.

Лінії життя – вертикальні пунктирні лінії, що представляють життєвий цикл об'єкта.

Повідомлення – лінії зі стрілками, які показують передачу повідомлень чи виклик методів (синхронні або асинхронні).

Активності – прямокутники на лінії життя, що вказують періоди виконання певної дії.

Контрольні структури – блоки "alt" (альтернатива) або "loop" (цикл) для відображення умов і циклів.

9. Як діаграми послідовностей пов'язані з діаграмами варіантів використання?

Діаграми послідовностей деталізують сценарії, описані у варіантах використання. Кожен варіант використання може бути представлений однією або кількома діаграмами послідовностей, які показують конкретну взаємодію об'єктів для реалізації цього варіанту використання.

10. Як діаграми послідовностей пов'язані з діаграмами класів?

Діаграми послідовностей використовують об'єкти класів, визначених на діаграмі класів. Методи, які викликаються у повідомленнях на діаграмі послідовностей, повинні бути визначені у відповідних класах на діаграмі класів. Діаграма класів показує статичну структуру, а діаграма послідовностей – динамічну поведінку цих класів.

Посилання на репозиторій:

<https://github.com/sokolovapolina230/labEmailClient/tree/13>

Висновок: у даній лабораторній роботі побудовано діаграми компонентів і розгортання для системи для обраної теми – Email клієнт, створено діаграми послідовностей за сценаріями варіантів використання. Продовжено розробку застосунку E-mail клієнт на базі мови програмування Java, використано SQLite як локальну базу даних. Було розроблено логіку завантаження FXML-файлів у класі Main.java та реалізовано ініціалізацію таблиці користувачів у UserController.java, що відображає дані з таблиці users бази даних.

Вдосконалено структуру проєкту, яка тепер має чітке розділення на пакети:

emailclient.model – модель даних (класи User, Account, Message тощо);

emailclient.repository – репозиторії для доступу до бази;

emailclient.controller – контролери інтерфейсу;

emailclient.ui.view – FXML-файли;

emailclient.database – SQL-скрипти для створення таблиць.

У результаті роботи створено основу GUI-застосунку, реалізовано зв'язок між інтерфейсом користувача та базою даних, а також закладено архітектурну структуру проєкту, необхідну для подальшої реалізації функціоналу.