ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка застосунку ФронтStory для емоційної підтримки під час війни»

на здобуття освітнього ступеня бакалавра

зі спеціальності 121 Інженерія програмного забезпечення

освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Максим ФРАНЧУК

(підпис)

Виконав: здобувач(ка) вищої освіти
групи ІПЗ-51
Максим ФРАНЧУК

Керівник: к.т.н, доцент кафедри ІПЗ,
Юрій ЗАДОНЦЕВ

Рецензент:

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Франчуку Максиму Олеговичу

1. Тема кваліфікаційної роботи: «Розробка застосунку ФронтStory для емоційної підтримки під час війни»
керівник кваліфікаційної роботи: Юрій ЗАДОНЦЕВ, к.т.н., доцент кафедри,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56.

2. Строк подання кваліфікаційної роботи «02» червня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: Інформація про особливості організації емоційної підтримки під час війни, наукові публікації з психології кризових станів, приклади сучасних цифрових рішень для ментальної підтримки, аналітика з використання інструментів лексичного аналізу тексту, технічна документація щодо використання C++, Qt та SQLite для розробки кросплатформених застосунків.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1. Аналіз предметної області: проблеми надання емоційної підтримки у кризових ситуаціях.
4.2. Аналіз програмного забезпечення з подібним функціоналом.
4.3. Проєктування архітектури застосунку для емоційної підтримки.
4.4. Реалізація функціоналу: ведення особистих історій, аналіз емоційного стану, перегляд мотивуючого контенту, візуалізація настрою.
4.5. Опис інтерфейсу користувача.
4.6. Оцінка функціональності та стабільності роботи застосунку.

5. Перелік графічного матеріалу: *презентація*

5.1. Аналіз аналогів.
5.2. Вимоги до застосунку.
5.3. Програмні засоби реалізації.
5.4. Загальна архітектура.
5.5. Діаграма послідовності.
5.6. Діаграма використання.
5.7. Структура бази даних.
5.8. Аналіз емоцій та візуалізація
5.9. Екранні форми.
5.10. Апробація розробленого рішення.

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	25.02-04.03.2025	
2	Аналіз та дослідження існуючих аналогів	05.03-11.03.2025	
3	Визначення вимог до застосунку для емоційної підтримки	12.03–19.03.2025	
4	Проєктування архітектури та інтерфейсу застосунку	20.03–02.04.2025	
5	Реалізація основних функцій: введення історій, аналіз настрою, візуалізація	03.04–18.04.2025	
6	Оцінка функціональності та стабільності роботи застосунку	19.04–28.04.2025	
7	Оформлення роботи: вступ, висновки, реферат	29.04–08.05.2025	
8	Розробка демонстраційних матеріалів	09.05–16.05.2025	
9	Попередній захист роботи	19.05–23.05.2025	

Здобувач вищої освіти _____ Максим ФРАНЧУК

(підпис)

Керівник
кваліфікаційної роботи _____ Юрій ЗАДОНЦЕВ

(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 65 стор., 1 табл., 17 рис., 33 джерела.

Мета роботи – надання емоційної підтримки, можливості спостереження та самоконтролю свого емоційного стану в умовах повномасштабної війни.

Об'єкт дослідження – процеси відстеження та аналізу емоційного стану людини з використанням програмних засобів.

Предмет дослідження – застосунок для локального зберігання та обробки емоційних даних користувача.

Короткий зміст роботи: у роботі проведено аналіз сучасного стану психологічної підтримки під час війни, описано проблеми, пов'язані з масовим зростанням кількості людей з ПТСР, депресіями та іншими психічними розладами. Розглянуто існуючі цифрові рішення для підтримки ментального здоров'я. Запропоновано та реалізовано архітектуру застосунку ФронтStory, що поєднує простий інтерфейс, модулі для ведення особистих історій, читання надихаючого контенту, введення і візуалізацію настрою. Реалізовано функції реєстрації, аналізу емоційної тональності записів, шифрування даних користувача. Проведено тестування застосунку, оцінено його ефективність, визначено перспективи подальшого розвитку.

КЛЮЧОВІ СЛОВА: ПСИХОЛОГІЧНА ПІДТРИМКА, МОБІЛЬНИЙ ЗАСТОСУНОК, QT, C++, ЕМОЦІЙНИЙ СТАН, ЛЕКСИЧНИЙ АНАЛІЗ, ВІЙНА, НАСТРІЙ, МОНІТОРИНГ, САМОДОПОМОГА.

ABSTRACT

The text part the qualification work for obtaining a bachelor's degree: 65 pages, 1 table, 17 figures, 33 sources.

Purpose - providing emotional support, the ability to observe and self-control one's emotional state in the conditions of full-scale war.

Object of research - an application for local storage and processing of the user's emotional data.

Subject of research - a desktop application for emotional support, implemented with C++ and the Qt framework, using a local SQLite database and lexical text analysis algorithms.

Summary of the work: The work analyzes the current state of psychological support during wartime and describes the problems associated with a massive increase in people suffering from PTSD, depression, and other mental disorders. Existing digital solutions for mental health support are reviewed. The architecture of the FrontStory application is proposed and implemented, combining a simple interface, modules for keeping personal stories, reading inspiring content, entering and visualizing mood. Features such as user registration, emotional sentiment analysis of records, and user data encryption are implemented. The application has been tested, its effectiveness assessed, and prospects for further development identified.

KEYWORDS: PSYCHOLOGICAL SUPPORT, MOBILE APPLICATION, QT, C++, EMOTIONAL STATE, LEXICAL ANALYSIS, WAR, MOOD, MONITORING, SELF-HELP.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

GUI – Graphical User Interface (графічний інтерфейс користувача)

Qt – кросплатформений фреймворк для створення графічних інтерфейсів

C++ – мова програмування C++

БД – база даних

SQLite – вбудована реляційна база даних

SQL – Structured Query Language (структурована мова запитів до бази даних)

ПТСР – посттравматичний стресовий розлад

UX – User Experience (досвід користувача)

AES-256 – Advanced Encryption Standard (алгоритм симетричного шифрування, 256-бітний ключ)

ID – ідентифікатор

UI – User Interface (інтерфейс користувача)

КС – ключ шифрування

API – Application Programming Interface (інтерфейс прикладного програмування)

ВСТУП

Актуальність

Війна та бойові дії призводять до масштабної психологічної кризи серед населення. Постійний стрес, втрати, руйнування та вимушене переміщення людей спричиняють різке зростання кількості випадків посттравматичного стресового розладу (ПТСР), депресії, тривожних розладів та інших психічних проблем. За даними мета-аналізу 181 дослідження, середній рівень ПТСР серед біженців та постраждалих від конфліктів становить близько 30% (допускаючи варіацію від 0% до 99% залежно від вибірки), а депресії – близько 30,8%. Сучасна війна в Україні супроводжується надзвичайним психологічним навантаженням на населення: оцінки свідчать, що до 30% українців можуть мати психічні проблеми, тобто до 15 мільйонів людей потребують психологічної підтримки. Водночас ресурсів професійної допомоги бракує – кількість кваліфікованих психологів та психотерапевтів вимірюється десятками тисяч, що в рази менше за кількість тих, хто потребує допомоги. Такий дисбаланс особливо відчутний у воєнний час, коли традиційні служби психологічної допомоги перевантажені або недоступні.

У цих умовах цифрові технології розглядаються як перспективний інструмент розширення доступу до психологічної підтримки. Застосунки для смартфонів та комп’ютерів можуть надавати інформаційну підтримку, навчати навичкам самодопомоги, створювати платформи для спілкування і взаємної підтримки, а також здійснювати базовий моніторинг емоційного стану користувачів. Наприклад, в 2011 році Департамент у справах ветеранів США випустив застосунок PTSD Coach, який надає психоедукацію та інструменти самоконтролю для людей із симптомами ПТСР. Дослідження показали високу задоволеність користувачів цим застосунком і навіть покращення їхніх симптомів ПТСР завдяки використанню програми. Під час нинішньої війни в Україні набули поширення різноманітні ініціативи цифрової підтримки – від чат-ботів психологічної допомоги до мобільних додатків для самодопомоги. Зокрема,

онлайн-програма самодопомоги Self-Help Online, адаптована за матеріалами ВООЗ, продемонструвала ефективність у зниженні рівня стресу серед українських біженців: у рандомізованому дослідженні вона достовірно зменшила дистрес у групі учасників-біженців (середній ефект d≈0,47). Ці факти свідчать, що правильно спроєктовані цифрові рішення можуть стати дієвим доповненням до традиційної психологічної допомоги в умовах гуманітарної кризи.

Об’єкт дослідження

Процеси відстеження та аналізу емоційного стану людини з використанням програмних засобів, таких як настільний застосунок для емоційної підтримки користувачів, розроблений із використанням мови програмування C++ і фреймворку Qt, локальної бази даних SQLite і алгоритмів лексичного аналізу тексту.

Мета роботи

Надання емоційної підтримки, можливості спостереження та самоконтролю свого емоційного стану в умовах повномасштабної війни, завдяки розробці застосунку ФронтStory для емоційної підтримки користувачів під час війни, що забезпечить платформу для вираження переживань, доступ до надихаючого контенту та базову аналітику емоційного стану. Застосунок має допомогти зменшити відчуття психологічної ізоляції, надати корисні поради та інструменти самодопомоги, а також виявляти ознаки погіршення емоційного стану у користувача.

Методи дослідження

Для досягнення поставленої мети у роботі використано такі методи:

1. аналіз літературних джерел і сучасних наукових досліджень щодо проблеми психологічної підтримки під час війни;

2. огляд і порівняння існуючих цифрових рішень для емоційної підтримки;

3. проєктування архітектури програмного забезпечення, моделювання бази даних, створення діаграм і макетів інтерфейсу користувача;

4. розробка й реалізація алгоритмів лексичного аналізу тексту для визначення емоційної тональності;

5. тестування й оцінка ефективності, юзабіліті та відповідності розробленого застосунку вимогам.

Наукова новизна

Наукова новизна даної роботи полягає у поєднанні підходів із галузі інформаційних технологій та психології для створення інструменту підтримки психічного здоров’я в умовах війни. Вперше розроблено настільний застосунок із використанням лексичного аналізу для моніторингу емоційного стану користувача, а також інтегровано методи наративної та мотиваційної терапії у цифровому форматі. Забезпечено локальне зберігання й шифрування даних.

Практична значущість

Практичне значення роботи полягає у створенні працездатного прототипу застосунку ФронтStory, який може бути використаний волонтерами, психологами або безпосередньо особами, що пережили травматичний досвід війни, для психологічної самодопомоги. Застосунок може бути розгорнутий на персональних комп’ютерах та мобільних пристроях, працювати автономно без підключення до Інтернету, що особливо важливо у випадку перебоїв зв’язку під час війни. Результати роботи, включаючи програмний код та опис архітектури, можуть бути використані як основа для подальшого розвитку системи психологічної підтримки або адаптовані для вирішення аналогічних завдань у сфері e-health.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ЗАСТОСУНКУ

1.1. Психологічна підтримка під час війни: проблематика та існуючі рішення

Психологічні наслідки війни. Збройний конфлікт і воєнні дії спричиняють серйозні травматичні переживання у як військових, так і цивільних осіб. Тривала дія бойових факторів (обстріли, життя в укриттях, вимушена міграція, свідчення насильства) призводить до масових випадків гострого стресу, які у багатьох людей переходять у хронічні форми психологічних розладів [1].Дослідження показують, що серед переселенців та біженців високі рівні посттравматичного стресового розладу (до 30% і більше) та депресії. Окрім ПТСР і депресії, поширені тривожні розлади, безсоння, психосоматичні прояви. Психічне здоров’я цивільного населення, яке перебуває в зоні конфлікту або пережило окупацію, також суттєво погіршується – люди відчувають постійну тривогу, емоційне виснаження, горе від втрати близьких і домівок.

Потреба в емоційній підтримці. В умовах війни навіть ті, хто не звертається по клінічну допомогу, потребують емоційної підтримки для подолання стресу. Соціальна ізоляція, роз’єднаність сімей, порушення звичних соціальних зв’язків підсилюють негативні переживання. Надання психологічної допомоги у вигляді консультацій спеціалістів ускладнене через бойові дії, руйнування інфраструктури та перевантаженість системи охорони здоров’я[2].За оцінками фахівців, мільйони українців потребують базової психосоціальної підтримки і навичок самодопомоги. Світовий досвід подолання наслідків бойових дій свідчить, що в таких умовах ефективними є програми психологічної самодопомоги та взаємопідтримки, які можна масштабувати на широкі верстви населення. Йдеться про техніки саморегуляції (дихальні вправи, релаксація), ведення щоденників і вираження

емоцій через текст чи творчість, групові обговорення досвіду (очно чи онлайн), розповсюдження достовірної інформації про типові реакції психіки на травму і способи з ними впоратися тощо. Значний позитивний ефект дає залучення принципів наративної терапії – коли людина ділиться своєю історією, переживаннями, відчуває, що її слухають і розуміють. Фіксація власних почуттів у щоденнику або через публічну розповідь допомагає структурувати травматичний досвід і зменшити його деструктивний вплив[3].

Існуючі цифрові рішення. У XXI столітті набув поширення напрям digital mental health – використання цифрових технологій для підтримки психічного здоров'я. Під час війни та гуманітарних криз такі технології особливо актуальні. Серед відомих прикладів – згаданий вище мобільний застосунок PTSD Coach, який вперше продемонстрував ефективність як інструмент самодопомоги для ветеранів з ПТСР і згодом був адаптований у багатьох країнах. Застосунок містить інформаційні матеріали про ПТСР, опитувальники для самодіагностики симптомів, вправи для зниження тривоги (дихальні техніки, м'язова релаксація), а також функції відстеження симптомів та прямого з'єднання з гарячою лінією підтримки. Дослідження демонструють, що такі інструменти можуть бути корисними як доповнення до терапії або навіть як перша допомога для тих, хто не має доступу до лікаря.

Ще один приклад – ізраїльський чат-бот "Friend: First Aid", запущений для психологічної підтримки українців у перші тижні після вторгнення 2022 року. Цей бот надавав у текстовому режимі поради з першої психологічної допомоги. Протягом 4 тижнів ним скористалися понад 40 000 людей. Однак, як зауважують фахівці, важливо оцінювати якість та безпечність таких інтервенцій: на момент запуску бот не мав завершених клінічних випробувань щодо ефективності. Це підкреслює необхідність наукового підходу при розробці подібних рішень[4].

У відповідь на виклики, спричинені війною в Україні, створюються цілі комплексні програми психосоціальної підтримки. Приміром, ініціатива «Тепло» (Teplo.io) пропонує українським біженцям мобільний застосунок з курсом про

травму, чатом підтримки та спільнотою одужання. Розробники зазначають, що через брак фахівців (в Україні ~60 тисяч психологів на мільйони постраждалих) їхнє рішення покликане масштабувати допомогу за рахунок цифрових технологій[5]. Ще один приклад – додаток iStrong для українських дітей, запущений благодійниками, що навчає технік подолання стресу та містить ігрові елементи для залучення молодшої аудиторії. ВООЗ також впровадила програму Self-Help Plus (SH+) у форматі як групових сесій, так і онлайн-модулів, адаптованих для українців. Саме на основі матеріалів SH+ була створена згадана вище програма Self-Help Online, ефективність якої підтверджено дослідженням.

Вимоги до застосунку ФронтStory. Аналіз показує, що ефективний застосунок емоційної підтримки має задовольняти такі основні вимоги

Доступність і масовість: простий у використанні інтерфейс, українська мова контенту; безкоштовність і можливість офлайн-роботи, щоб користувачі в зоні конфлікту чи без стабільного інтернету могли ним скористатися.

Психологічна безпечність: контент і методики мають базуватися на науково обґрунтованих підходах психологічної допомоги (наприклад, принципи когнітивно-поведінкової терапії, техніки релаксації, активного слухання). Недопустимо, щоб застосунок давав некоректні поради чи поглиблював травматизацію.

Функції самодопомоги: надання користувачу інструментів для саморегуляції (наприклад, аудіо-інструкції для дихальних вправ чи медитацій), можливість вести щоденник або записувати власні думки/історії, які полегшують емоційне навантаження через їх вираження[6].

Надихаючий контент: наявність позитивних історій, прикладів подолання труднощів, повідомлень підтримки. У випадку ФронтStory – це можуть бути короткі оповідання (“story”) від військових, волонтерів чи цивільних про їхній досвід та надію, які мають підбадьорити читача.

Взаємодія та спільнота (за можливості): опційно, користувачі могли б ділитися власними історіями, читати історії інших, коментувати чи «підтримувати»

одне одного (наприклад, через умовні «лайки» чи слова підтримки). Це створює ефект спільноти взаємодопомоги. Однак реалізація такої функціональності потребує модерації контенту та захисту від токсичної поведінки, тому на першому етапі в ФронтStory передбачається пасивна взаємодія – користувач може надіслати свою історію розробникам (наприклад, електронною поштою) для додання до загального контенту після модерації.

Аналіз емоційного стану: застосунок має відстежувати динаміку настрою чи емоцій користувача. Це може здійснюватися через короткі опитування (напр. щоденно питати «Оцініть свій настрій від 1 до 10» і будувати графік), або ж через аналіз тональності текстів, що користувач пише у щоденнику. Така аналітика емоційного стану дозволить користувачу самим бачити прогрес чи погіршення, а також може слугувати сигналом – при виявленні критично низького настрою або слів-сигналів (напр. про суїцидальні думки) застосунок може порадити звернутися по допомогу чи надати контакти кризових служб.

Конфіденційність і безпека даних: особливо важливо в контексті чутливих психологічних даних. Застосунок має гарантувати, що записи користувача (його історії, нотатки, оцінки настрою) не стануть надбанням сторонніх осіб. Це означає шифрування локальної бази даних або хоча б захист паролем, а в разі онлайн-функцій – дотримання протоколів безпеки (SSL, хешування паролів тощо). У нашій реалізації ФронтStory зберігає всі дані локально на пристрої, без передачі на зовнішні сервери, що підвищує конфіденційність (але накладає обмеження на можливість онлайн-взаємодії між користувачами).

1.2. Обґрунтування вибору технологій розробки

Вибір платформи та мови. Виходячи з вимог доступності та офлайн-роботи, було прийнято рішення реалізувати ФронтStory як крос-платформений настільний застосунок (десктоп додаток), що може працювати на звичайному персональному комп’ютері або ноутбуці під керуванням Windows або Linux. Багато внутрішньо

переміщених осіб та сімей в Україні мають доступ саме до ноутбуків чи стаціонарних ПК у сховищах, волонтерських центрах або нових місцях проживання. Мобільна версія також є бажаною, але розробка одночасно під кілька мобільних платформ (Android, iOS) потребує значних ресурсів. Тому крос-платформений настільний додаток обрано як початкову ціль, з перспективою портування на мобільні платформи[10].

Для реалізації застосунку обрано мову C++ та фреймворк Qt. Мова C++ дозволяє створювати високопродуктивні програми з низьким споживанням ресурсів, що особливо актуально, коли застосунок може працювати на старих чи малопотужних комп'ютерах (типових для польових умов). C++ є компільованою мовою, тож створена програма має невеликий розмір і швидко працює без потреби у встановленні додаткових інтерпретаторів чи віртуальних машин[9].Крім того, C++ надає тонкий контроль над пам'яттю, що дозволяє убезпечити дані (наприклад, своєчасно видаляти чутливу інформацію з оперативної пам'яті). Важливо й те, що C++ широко використовується для розробки системного і застосункового ПЗ, тому існує багато бібліотек, які можуть стати в пригоді – зокрема, для роботи з базами даних, криптографії, а також бібліотеки машинного навчання (якщо у майбутньому додаватиметься інтелектуальний аналіз тексту).

Фреймворк Qt обрано для побудови графічного інтерфейсу та забезпечення крос-платформності. Qt – це потужний набір бібліотек на C++, який підтримує розробку GUI-застосунків для Windows, Linux, macOS, а також мобільних ОС (Android, iOS) із єдиним базовим кодом. Він надає багатий набір компонентів інтерфейсу (від кнопок і меню до веб-переглядачів), засоби роботи з файлами, мережею, багатопотоковість тощо. Код на Qt добре масштабується та переноситься між платформами, що прискорює розробку і спрощує майбутнє розширення проєкту. Як зазначено у документації, Qt забезпечує високу продуктивність виконання, малий об'єм програми та можливість багаторазово використовувати код на різних платформах. До того ж, Qt має модуль QtSql для інтеграції з реляційними базами даних, що полегшує взаємодію з SQLite[12].

Зберігання даних у ФронтStory організовано через локальну реляційну базу даних SQLite. SQLite – це легка вбудована СУБД, що не потребує окремого серверу і зберігає всю базу у вигляді файлу на диску. Вона чудово підходить для настільних і мобільних застосунків, оскільки дуже проста у розгортанні (бібліотека SQLite підключається безпосередньо до програми) та працює швидко навіть на невеликих обсягах пам’яті [14]. SQLite гарантує цілісність даних і підтримує транзакції, що дозволяє надійно зберігати записи щоденника, історії та іншу інформацію користувача. Обсяг таких даних у нашому застосунку відносно невеликий (текстові записи, декілька сотень історій, параметри користувача), тому потужності SQLite достатньо з великим запасом.

Використання реляційної моделі спрощує організацію даних: можна легко пов’язати записи настрою з користувачем, історії з авторством тощо, виконувати вибірки (наприклад, отримати всі записи настрою за останній місяць для побудови графіку). Для доступу до SQLite застосовано QtSql, що забезпечує уніфікований інтерфейс до СУБД: ми визначаємо підключення до SQLite через драйвер QSQLITE, а далі використовуємо SQL-запити у коді C++ або обгортаємо їх у Qt-класи (табличні моделі).

Обробка тексту та аналіз настрою. Однією з особливостей ФронтStory є модуль аналізу емоційного стану. Він може працювати у двох режимах:

1. Оцінка настрою користувачем вручну. Користувачеві пропонується час від часу (наприклад, раз на день) оцінити свій настрій за чисельною шкалою (1 – дуже поганий, 10 – чудовий). Такі дані зберігаються у базі і використовуються для побудови графіків настрою.

2. Автоматичний аналіз текстових записів. Коли користувач пише особисту історію або нотатку, система проводить тональний аналіз тексту – визначає, наскільки текст має позитивне чи негативне емоційне забарвлення. Для цього застосовується метод лексичного аналізу тональності (sentiment analysis) на основі словників. Суть методу: програма містить наперед підготовлений список слів із позитивною і негативною конотацією (так

званий лексикон тональності), і при аналізі тексту підраховується баланс “позитивних” та “негативних” слів. Наприклад, слова “радість”, “надія”, “вдячний” додають позитивний бал, а “сум”, “біль”, “лякає” – негативний. Сумарний індекс може нормуватися до певного діапазону. Такий підхід є спрощеним, але швидким і прозорим, не вимагає великих обчислювальних ресурсів чи інтернет-з’єднання (як потребували б сучасні нейронні мережі). Лексичні методи зарекомендували себе як цілком надійний інструмент для визначення полярності емоцій в тексті , особливо якщо словник адаптований під конкретну мову і контекст. У нашому випадку словник було складено з урахуванням української мови та воєнної тематики (наприклад, окремо враховано слова типу “тривога”, “обстріл”, “герой”, “перемога” тощо). Надалі таку функціональність можна розширити, навчивши модель машинного навчання на корпусі текстів військового часу, щоб вона виявляла складніші емоційні маркери – проте у рамках бакалаврського проєкту реалізовано саме лексиконний аналіз для своєї надійності та простоти.

Безпека та етичні аспекти. При розробці архітектури ми керувалися принципом “не зашкодь” (do no harm). Це означає, що застосунок ні в якому разі не повинен погіршити стан користувача. Всі поради і матеріали перевірено на відповідність науковим рекомендаціям. Зокрема, вміст надихаючих історій було підібрано таким чином, щоб уникати контенту, що може повторно травматизувати (відверті описи насильства, жорстокості тощо). Користувачеві надаються контакти професійної допомоги (гарячі лінії, контакти психологічних служб) у розділі “Допомога”, з рекомендацією звернутися, якщо його стан важкий. Реалізовано простий механізм попередження: якщо за результатами аналізу тональності останніх записів користувача спостерігається різке та стійке падіння настрою (наприклад, кілька дуже негативних нотаток поспіль), застосунок виводить ненав’язливе повідомлення: “Ми помітили, що Ваші останні записи сповнені негативних переживань. Можливо, Вам варто поговорити з кимось про це. Ось контакти гарячої лінії…”.

З точки зору технічної безпеки, базу даних SQLite зашифровано з використанням AES-256: при першому запуску застосунку генерується ключ на основі пароля користувача, і далі всі дані зберігаються у зашифрованому вигляді на диску. В разі, якщо хтось отримає фізичний доступ до файлу БД, він не зможе прочитати конфіденційні записи без знання пароля. Обмін даними із зовнішнім світом у поточній версії ФронтStory відсутній (немає сервера), отже ризики перехоплення трафіку відсутні. Це рішення (локальність) прийняте на користь конфіденційності; у майбутньому, якщо додавати хмарні функції (наприклад, резервне копіювання історій чи синхронізація між пристроями), потрібно буде забезпечити шифрування каналів зв’язку (SSL) і надійний захист серверу.

Висновок до розділу 1

Аналіз потреб користувачів в психологічній підтримці під час війни та огляд існуючих рішень показав доцільність створення застосунку ФронтStory з функціями реєстрації, публікації/читання історій та аналізу емоційного стану. Обрано технологічний стек на основі C++/Qt/SQLite, що забезпечує необхідну продуктивність, автономність та безпеку даних. У наступному розділі детально розглянуто проєктування системи – її архітектуру, компоненти та інтерфейс.

2. ПРОЄКТУВАННЯ ЗАСТОСУНКУ ФРОНТSTORY

Розробка застосунку почалася зі стадії проєктування, яка включала: формування вимог (бізнес-вимоги та функціональні вимоги користувача), проєктування архітектури програмної системи, розробку структури бази даних, сценаріїв використання (use cases) та макетів інтерфейсу користувача. У цьому розділі описано основні спроєктовані компоненти ФронтStory та їх взаємодію.

2.1. Загальна архітектура системи

Застосунок ФронтStory розроблено з використанням багатошарової архітектури, що передбачає розділення на такі основні частини:

- Інтерфейс користувача (UI) – відповідає за відображення даних та взаємодію з користувачем (вікна, кнопки, текстові поля тощо).
- Логіка застосунку (Application Logic) – сукупність внутрішніх модулів, що реалізують функціональність (обробка дій користувача, бізнес-логіка). У нашому випадку виділено три основні модулі логіки:
  - Модуль реєстрації (автентифікації та управління користувачами),
  - Модуль контенту (робота з історіями, відображення та додавання контенту),
  - Модуль аналізу емоцій (обробка записів настрою, аналіз тональності тексту, формування аналітичних даних).
- Сховище даних – у ФронтStory це локальна база даних SQLite, через яку модулі логіки зберігають та отримують інформацію (акаунти користувачів, тексти історій, записи про настрій тощо).

На Рис. 2.1 зображено спрощену архітектурну схему застосунку, що показує основні компоненти та зв’язки між ними[12].

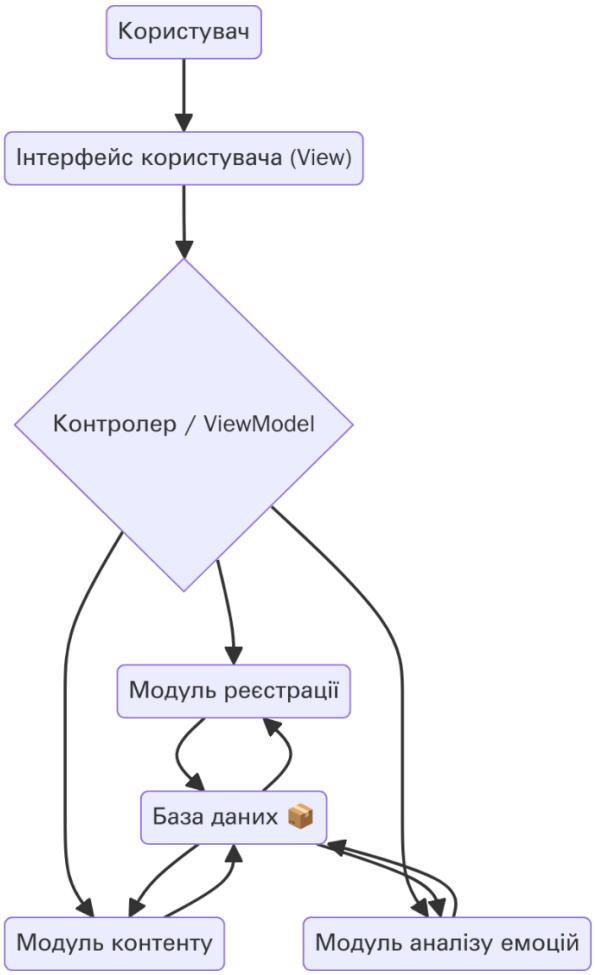


Рис. 2.1. Архітектура застосунку ФронтStory (основні компоненти і зв’язки)

Як видно з діаграми, користувач взаємодіє з програмою через інтерфейс. Інтерфейс передає команди у відповідні модулі логіки: наприклад, якщо користувач натискає “Увійти” або “Зареєструватися”, UI викликає функції модуля реєстрації; якщо відкриває список історій – працює модуль контенту; при введенні нової нотатки або запиту побудувати графік настрою – задіяно модуль аналізу емоцій. Усі модулі логіки при потребі зберігають чи витягують дані із бази даних. База даних слугує єдиним надійним джерелом даних (single source of truth) для застосунку.

Така архітектура відповідає шаблону «Model-View-Controller» (MVC) або більш сучасному його різновиду «Model-View-ViewModel» (MVVM), який часто реалізують у Qt. Інтерфейс користувача – це View, модулі логіки та відповідні класи даних – Model, а контролери/в'ю-моделі забезпечують обмін між ними. Чітке розділення спрощує підтримку: можна змінювати внутрішню реалізацію (наприклад, алгоритм аналізу емоцій) без впливу на UI, і навпаки, змінювати дизайн інтерфейсу без перебудови логіки.

2.2. Структура бази даних

Для зберігання даних застосунку створено реляційну базу даних, що складається з кількох таблиць. Проєктування бази даних здійснювалося на основі вимог до інформації, яка мусить зберігатися:

- Облікові дані користувачів (логін, пароль тощо).
- Персональні історії/нотатки користувача.
- Колекція надихаючих історій (контенту), доступних усім користувачам.
- Записи про настрій користувача (дату та оцінку).
- Можливо, додаткові довідники (якщо потрібно класифікувати контент за категоріями)[14].

На Рис. 2.2 представлено діаграму сутність-зв’язок (ER-діаграму) бази даних ФронтStory з основними таблицями та зв’язками між ними.

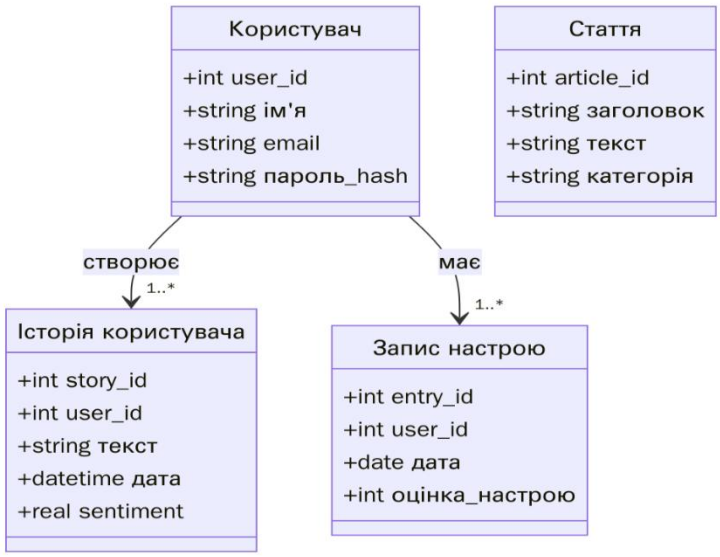


Рис. 2.2. ER-діаграма бази даних застосунку ФронтStory

Структура бази даних охоплює такі ключові таблиці:

- Користувач (User): зберігає інформацію про зареєстрованих користувачів.
  - user_id – унікальний ідентифікатор (ціле число, PK).
  - ім'я – ім’я користувача або логін (текст).
  - email – електронна пошта (текст, може бути NULL, опціонально).
  - пароль_hash – хеш пароля (текст). Зберігається не сам пароль, а криптографічний хеш (SHA-256 або bcrypt), щоб уразі компрометації бази зловмисник не отримав паролі у відкритому вигляді.
- Стаття (Article): містить наповнення контенту – надихаючі історії, поради, статті, доступні для читання всім користувачам. Ці записи можуть бути предзаповнені розробниками або додаватися через модерацію.
  - article_id – PK.
  - заголовок – назва статті (текст).
  - текст – повний зміст (текст, може бути великим).

- категорія – тематична категорія або тег (текст, наприклад: “Історія з фронту”, “Порада психолога”, “Від волонтера” тощо).

- Історія (UserStory): таблиця для особистих історій або нотаток, які створюють користувачі для вираження своїх емоцій. Ці записи видно лише їх авторові (в поточній реалізації – приватні).
  - story_id – PK.
  - user_id – зовнішній ключ (FK) до таблиці Користувач, вказує автора нотатки【39†look】.
  - текст – текст історії (текст).
  - дата – дата та час створення запису (DATE або DATETIME).
  - sentiment – числовий індекс тональності тексту (REAL). Обчислюється модулем аналізу емоцій під час збереження запису. Позитивніші тексти матимуть вищий індекс (>0), негативні – нижчий (<0). Наприклад, +5 для дуже позитивного запису, -5 для дуже негативного, або 0 для нейтрального.

- Запис настрою (MoodEntry): таблиця для збереження суб’єктивних оцінок настрою користувача.
  - entry_id – PK.
  - user_id – FK на Користувач (чий це запис настрою).
  - дата – дата (можливо, без часу, якщо запис раз на день) або дата-час.
  - оцінка настрою – ціле число від 1 до 10 (INT), що ввів користувач.

Зв’язки між таблицями наступні: один користувач може мати багато історій (зв’язок 1:N між User і UserStory), а також багато записів настрою (1:N між User і MoodEntry). Таблиця Article від користувачів не залежить напряму – усі користувачі бачать один і той самий набір статей. При бажанні можна було б зв’язати Article з User, якщо б, наприклад, відстежувати, хто яку статтю додав або

які статті користувач прочитав. Але поки що додавання нових Article здійснюється тільки адміністраторами через безпосереднє редагування БД або випуск оновлення програми (звичайні користувачі не можуть публікувати публічні статті через застосунок)[15].

Представлена модель даних підтримує основні функції ФронтStory. У ході проєктування були розглянуті також альтернативи та розширення:

- Зберігання коментарів до статей або рейтингу статей – поки відмовилися, щоб не ускладнювати функціонал.
- Зберігання медіафайлів (зображень) у історіях – наразі не реалізовано, тільки текст, щоб зменшити розмір даних. Однак технічно SQLite може зберігати BLOBи (наприклад, закодовані зображення), або ж медіафайли можна зберігати окремо в файловій системі з посиланнями.
- Можливість декількох користувачів: додаток підтримує мульти-акаунти (наприклад, якщо комп’ютером користується родина), оскільки всі ключі прив’язані до User. В таблиці історій та записів настрою явно зберігається, кому вони належать. При вході під певним користувачем програма фільтрує дані за його user_id. Таким чином, із самого початку база проєктувалась багато-користувацькою, хоча типовий сценарій – один основний користувач на пристрої.

2.3. Інтерфейс користувача та сценарії роботи

Інтерфейс ФронтStory спроєктований таким чином, щоб навіть користувач без спеціальних навичок легко розібрався з програмою. Враховуючи стресовий стан цільової аудиторії, інтерфейс має бути максимально простим, зручним і заспокійливим. Обрано спокійну кольорову гамму (відтінки синього та зеленого, що асоціюються зі спокоєм), мінімалістичний дизайн без зайвої інформації на екрані.

Основні екрани (вікна) застосунку

Екран входу/реєстрації. При запуску програми користувач бачить вікно входу. Якщо він новий користувач – може переключитися на вкладку “Реєстрація” і створити обліковий запис, ввівши ім’я, email (необов’язково) та пароль. Після успішної реєстрації чи входу (автентифікації) відкривається головне вікно програми[17].

Головне вікно (стрічка історій). Це основний екран, де користувач бачить перелік доступного контенту. У верхній частині вікна – привітання на кшталт “Привіт, [ім’я]!” і меню навігації. Нижче – вкладки або секції:

“Історії інших” – список статей/історій із бази Article. Відображається заголовок та кілька перших рядків тексту. Користувач може клацнути на цікавий заголовок і відкрити повний текст історії в окремому діалоговому вікні чи панелі.

“Мої нотатки” – список власних записів користувача (таблиця UserStory). Тут показано, наприклад, дати й перші слова нотаток. Також поруч може бути кнопка “Додати запис”, яка відкриває екран створення нової нотатки.

Екран створення нової історії. На ньому – велике текстове поле для введення, кнопка “Зберегти”. Є підказка: “Напишіть про свої почуття або досвід сьогодні…”. Після натискання “Зберегти” запис потрапляє в таблицю UserStory, а також одразу обробляється модулем аналізу емоцій (обчислюється sentiment score і теж записується).

Екран аналітики настрою. На цьому екрані (чи вкладці) користувач може переглянути графік свого настрою за обраний період. Також тут може бути інтерфейс для введення поточного рівня настрою. В спрощеному варіанті: відображається остання оцінка (напр. “Ваш сьогоднішній настрій: 4/10, трохи знижений”), є кнопка “Оновити показник настрою”, при натисканні – випадає слайдер чи поле, де можна ввести нову оцінку 1–10. Після введення графік оновлюється. Графік будується на основі даних таблиці MoodEntry.

Для візуалізації графіка використано компонент QChart з QtCharts – він дозволяє побудувати лінійний графік за масивом даних. Наприклад, на Рис. 2.3

зображено приклад графіка настрою, що може відображатися у застосунку.

Рис. 2.3. Приклад графіка динаміки настрою користувача (згенеровано на основі тестових даних)

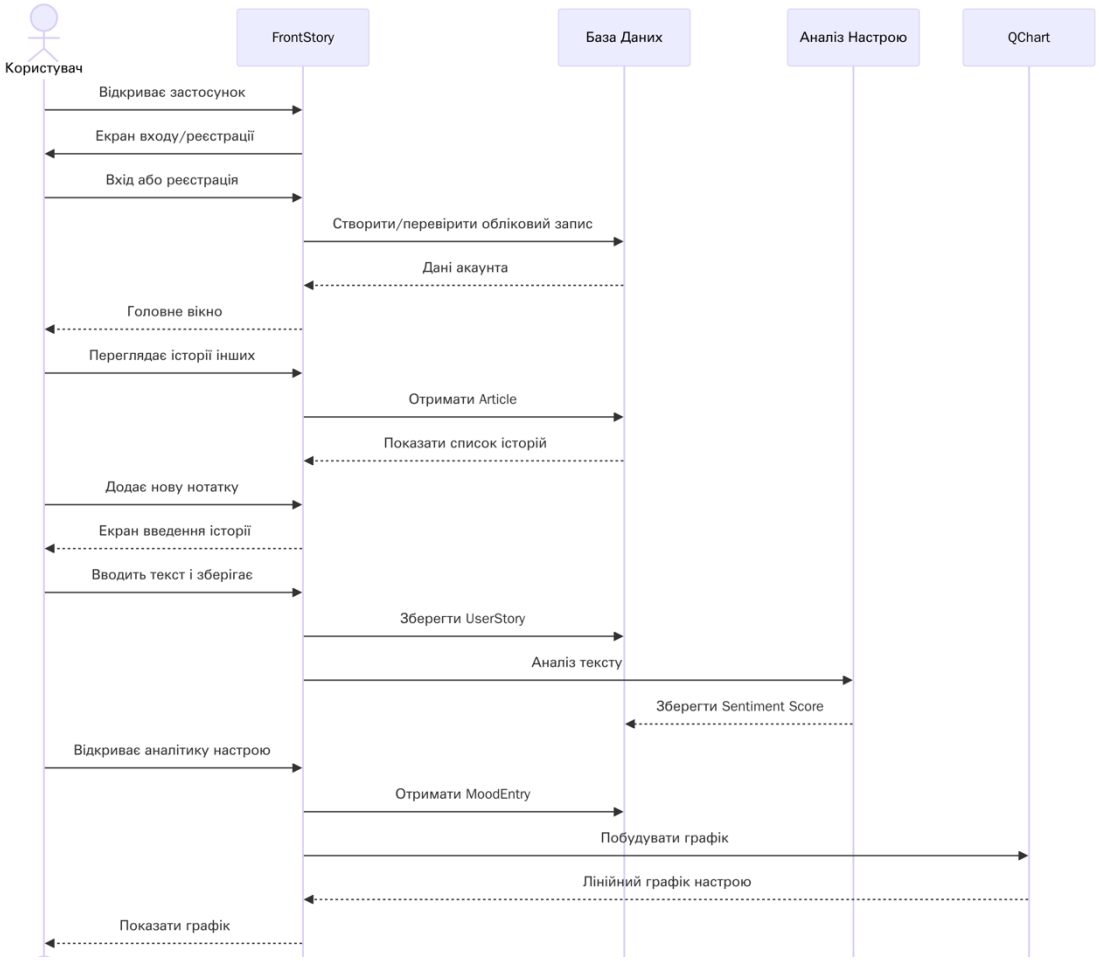


Рис. 2.4 Діаграма послідовності

На графіку видно коливання настрою користувача протягом декількох тижнів. Така візуалізація допомагає користувачу усвідомити, що погані дні змінюються кращими, або навпаки – якщо бачить тенденцію до погіршення, це сигнал приділити увагу своєму емоційному стану. Крім того, сам факт ведення “щоденника настрою” має терапевтичний ефект, оскільки дисциплінує користувача рефлексувати щодо свого самопочуття.

Навігація та меню. У головному вікні передбачено просте меню навігації (можливо, у вигляді вкладок або бічної панелі):

- “Головна” (історії інших),
- “Мої нотатки”,
- “Мій настрій” (аналітика),
- “Допомога” (контакти служб підтримки, поради щодо використання),
- “Налаштування” (мінімум параметрів, наприклад, перемикання мови інтерфейсу в майбутньому чи вихід з акаунту).

Користувач також може завершити сеанс, натиснувши “Вихід” – тоді повернеться на екран входу (це важливо, якщо комп’ютером користуються декілька людей, щоб захистити приватні записи)[19].

UX-вимоги. При проєктуванні інтерфейсу враховано, що цільовій аудиторії може бути не до складнощів – тому весь досвід користувача (UX) має бути интуїтивно зрозумілим:

- Реєстрація потребує мінімум даних (тільки ім’я та пароль). Email – опційно, для можливого відновлення паролю в майбутньому.
- Усі тексти інтерфейсу написані простою, дружньою мовою. Замість технічних термінів – зрозумілі формулювання (“Ваші історії”, “Новий запис”, “Оцінка настрою” тощо).
- Після реєстрації новий користувач отримує коротке привітання і підказку: “Ви можете прочитати історії інших або написати свою власну.

Натисніть “Додати запис”, щоб поділитися своїми думками.” Це реалізовано у вигляді модального діалогу або виділеного блока на екрані.

- Система не перевантажує користувача повідомленнями. Спливаючі вікна мінімізовані – лише важливі (підтвердження збереження запису, попередження якщо щось не заповнено, або згадане повідомлення про помічене погіршення настрою).
- Колірні акценти: позитивні тенденції (наприклад, підвищення настрою) можуть підсвічуватися зеленим, негативні – м’яким червоним/помаранчевим, але без агресивних кольорів. Загалом інтерфейс здебільшого у нейтральних тонах.
- Шрифти обрано читабельні, достатнього розміру (не менше 12pt для основного тексту, заголовки 14–16pt). Для людей із вадами зору передбачено можливість масштабування тексту (через Ctrl+ або налаштування).

2.4. Проєктування модулів логіки

Модуль реєстрації та автентифікації. Цей модуль відповідає за створення нового акаунту та вхід існуючого користувача. Логіка реєстрації:

1. Користувач вводить ім’я (логін) та пароль (двічі для підтвердження). Програма перевіряє, що поля не порожні, пароль достатньої довжини (наприклад, ≥6 символів)[21]\
2. Перевіряється, чи немає в БД користувача з таким самим ім’ям. Якщо є – видається повідомлення про помилку.
3. Якщо все гаразд, генерується хеш пароля. Використано алгоритм SHA-256 із сіллю: до пароля додається випадкова рядкова “сіль” завдовжки 16 символів (її теж зберігаємо разом із хешем або вбудовуємо в хеш). Отриманий хеш (64 символи шістнадцяткового представлення) записується в поле пароль_hash.

4. Створюється запис у таблиці User з новим user_id.

5. Користувачу повідомляється про успішну реєстрацію і автоматично виконується вхід у систему під цим логіном.

Логіка входу:

1. Користувач вводить ім’я та пароль.

2. Програма шукає в таблиці User запис з таким ім’ям. Якщо не знаходить – повідомляє “Користувача не знайдено”.

3. Якщо знаходить, обчислює хеш введеного пароля (використовуючи ту саму сіль, що зберігається для даного користувача) і порівнює зі значенням пароль_hash.

4. Якщо хеші співпали – аутентифікація успішна, користувач потрапляє в систему. Якщо ні – повідомлення про невірний пароль.

Для безпеки реалізовано захист від brute-force: при 5 невдалих спробах входу підряд модуль робить коротку затримку (наприклад, блокує можливість вводу на 10 секунд) та/або вимагає пройти просту капчу. Це запобігає автоматичному перебору паролів.

Крім того, модуль реєстрації може надавати функцію зміни паролю (в налаштуваннях), але це не обов’язково для базової версії.

Модуль контенту (історії). Він відповідає за операції з історіями:

- Завантаження списку всіх доступних публічних статей (Article) з БД і передача їх у UI для відображення.
- Обробка відкриття конкретної статті: витяг повного тексту за article_id та відображення в окремому вікні.
- Додавання нової особистої історії (UserStory):
  - Отримує текст від UI (екран додавання нотатки),
  - Формує SQL-запит INSERT до таблиці UserStory (вказуючи user_id поточного користувача, текст, поточну дату).
  - Перед вставкою викликає функцію аналізу тексту з модуля емоцій, щоб отримати sentiment score (див. опис нижче).

- Додає це значення до INSERT або оновлює запис після вставки.
- Після успішного запису – оновлює список нотаток у UI.

- Видалення або редагування нотатки: базова версія може не включати редагування (нотатки як щоденник зазвичай не редагують), але видалення може бути корисним. Передбачено метод deleteStory(story_id), який видаляє запис користувача (із підтвердженням через діалог “Ви впевнені?”). В таблиці UserStory це просто DELETE-запит по story_id з перевіркою, що користувач видаляє власну історію.
- Модерація або експорт історій: оскільки наразі особисті історії нікуди не публікуються, цей функціонал не потрібен. Можна передбачити простий механізм експорту всіх особистих записів у текстовий файл (для користувача, який хоче зберегти щоденник або передати психологу). Це можливо зробити через Qt (клас QFileDialog для вибору файлу і запис туди).

Модуль аналізу емоцій. Він має дві підзадачі:

1. Опрацювання тексту і оцінка його тональності (sentiment).
2. Облік та візуалізація оцінок настрою з таблиці MoodEntry.

Для першої задачі створено функцію analyzeSentiment(text: QString) -> double. Ця функція реалізує лексичний аналіз: розбиває текст на слова (видаляючи пунктуацію, приводячи до нижнього регістру), і для кожного слова перевіряє наявність у списках позитивних чи негативних слів[24].Списки (лексикони) підготовано вручну, спираючись на психологічні дослідження та словники тональності, доступні для української мови. Наприклад, позитивні: радість, щастя, любити, надія, вдячний, успіх, перемога,…; негативні: сум, біль, страх, тривога, втрачений, злий, поразка,… та ін. Кожне знайдене позитивне слово додає +1 до рахунку, негативне – -1. Деяким словам можна призначити ваги більше 1 за потреби (наприклад, “жахливо” – -2). Після аналізу всіх слів функція повертає сумарний індекс тональності. У середньому, позитивні тексти матимуть індекс >0, негативні – <0. Для зручності індекс можна нормувати до діапазону [-1; +1] або [-100; +100].

У нашій реалізації залишено вихідний рахунок (в середньому тексти мають від -10 до +10 максимум). Наприклад, текст “Сьогодні був хороший день, я відчував радість” – отримає +2 (за слова “хороший”, “радість”), а “Мені страшно і боляче від втрат” – може отримати -3 (“страшно”, “боляче”, “втрат”). Цей показник зберігається в БД разом із текстом (поле sentiment таблиці UserStory). У подальшому можна будувати графік зміни sentiment по нотатках або використовувати ці дані для індивідуальних рекомендацій користувачу. Варто зазначити, що лексичний метод не завжди точний – він не враховує контексту та іронії, але він прозорий і швидкий. Сучасні дослідження NLP пропонують більш складні алгоритми – від машинного навчання до глибоких нейронних мереж – для виявлення емоцій у тексті. Проте їх інтеграція вимагала б або підключення важких бібліотек, або виклику зовнішніх API. Умовою нашої роботи є автономність, тому обрано спрощений підхід[27].

Для другої задачі (настрій) модуль надає методи addMoodEntry(user_id, value) та getMoodHistory(user_id). Перший додає запис до таблиці MoodEntry з поточною датою та значенням value (1–10). Перед вставкою може перевіряти, чи вже був запис сьогодні – тоді або замінити, або запитати користувача (вирішено дозволити кілька записів на день, якщо користувач хоче відмітити зміни настрою). Метод getMoodHistory виконує SELECT-запит, що витягає всі записи настрою поточного користувача за останній N днів (за замовчуванням 30 днів) і повертає список пар (дата, значення). Ці дані потім використовує UI для побудови графіка (як на рис. 2.3). Візуалізація графіка – через QtCharts у режимі LineSeries, із відображенням точок даних.

Логування та телеметрія. У проєкті передбачено базове логування подій (в першу чергу, для розробника). У разі будь-яких помилок (невдалий SQL-запит, виняток) модуль записує повідомлення в лог-файл ФронтStory.log. Це допомагає відстежити збої. Щодо телеметрії використання (що натискає користувач) – спеціально не реалізовано, щоб не збирати зайвих даних про користувача (етичний момент). Хоча для розвитку продукту в майбутньому було б корисно анонімно

збирати статистику використання (які розділи більш популярні, середня кількість нотаток тощо), але наразі, працюючи офлайн, застосунок цього не робить.

Приклад роботи модулів у єдиному сценарії. Розглянемо типовий сценарій: користувач вперше встановив ФронтStory і хоче ним скористатися:

1. Користувач запускає програму, бачить екран реєстрації. Вводить ім’я “Олена” і пароль “qwerty”. Модуль реєстрації перевіряє у БД (спочатку там порожньо) – можна створювати. Формує хеш, створює запис у таблиці User (id=1, name=Олена, hash=…).
2. Відкривається головне вікно. Модуль контенту робить запит до Article і завантажує список доступних історій (наприклад, 5 записів). Виводить їх у вигляді списку заголовків. Олена читає кілька історій, натискаючи на заголовки – відкриваються тексти. Ці дії здійснює модуль контенту (SELECT * FROM Article WHERE id=X для кожного відкриття).
3. Олена вирішує написати власну історію. Переходить на вкладку “Мої нотатки” і натискає “Додати”. З’являється форма, вона пише текст: “Дуже важкий день. Були обстріли, але тримаємось. Відчуваю втому і тривогу, але намагаюсь не втрачати надію.”. Натискає “Зберегти”.
4. Модуль контенту отримує цей текст, викликає analyzeSentiment. Функція знаходить слова: “важкий”(-1), “втома”(-1), “тривогу”(-1), “не втрачати надію” – тут є “надію” (+1) – загалом індекс ≈ -2. Вона повертає -2.
5. Модуль формує SQL: INSERT INTO UserStory(user_id, text, date, sentiment) VALUES (1, “…текст…”, “2025-05-12 18:00”, -2). Якщо успішно – повертає на UI успіх. Новий запис відображається у списку нотаток з датою “12.05.2025” і першим рядком.
6. Тепер користувач переходить на вкладку “Мій настрій”. Він відчуває, що настрій десь 3/10. Вводить це значення і зберігає. Модуль аналізу додає MoodEntry(user=1, date=2025-05-12, value=3). Графік оновлюється, показує точку на сьогодні рівня 3. Поки це перший запис, графік має одну точку.

7. Наступного дня Олена знову відкриває ФронтStory, входить (модуль реєстрації перевіряє пароль – проходить). Вирішує відразу ввести настрій – каже, сьогодні 5/10. Додається MoodEntry (значення 5). Графік тепер відобразить лінію від вчора 3 до сьогодні 5.

8. Вона читає нові історії (можливо, контент оновлено оновленням додатку – це поза сценарієм), пише коротку нотатку: “Сьогодні трошки краще. Є надія.”. Аналіз тональності дасть, скажімо, +1 (за слово “надія”). Запис додається з sentiment=+1.

9. Через тиждень у Олени накопичилось кілька записів і оцінок настрою. Вона відкриває вкладку аналітики і бачить графік: коливання від 3 до, наприклад, 6. Під графіком – невеличкий коментар від програми: “Ваш середній настрій за останній тиждень: 4.8/10. В цілому спостерігається покращення”. Цей коментар формується модулем аналізу (вираховує середнє, порівнює тренд першої та останньої точки).

10. Якщо в якийсь момент значення настрою кілька днів поспіль ≤2 або нова нотатка отримала sentiment дуже негативний (< -5), модуль аналізу викликає функцію alertUserIfCritical() – вона виводить попередження з пропозицією звернутися по допомогу, як описано раніше.

Таким чином, архітектура ФронтStory забезпечує злагоджену роботу всіх компонентів: користувацькі дії через UI транслюються до відповідних модулів логіки, які маніпулюють даними в базі та використовують допоміжні алгоритми (аналіз тональності) для розширеної функціональності. У наступному розділі буде розглянуто реалізацію цих модулів у коді C++, з прикладами ключових фрагментів коду.

Висновок до розділу 2

У цьому розділі здійснено детальне проєктування застосунку ФронтStory для емоційної підтримки під час війни. Було визначено багатошарову архітектуру

системи, що забезпечує чітке розділення інтерфейсу користувача, логіки застосунку та сховища даних, що сприяє зручності підтримки й розширення функціоналу. Проєктування реляційної бази даних дозволило створити ефективну структуру для зберігання історій, нотаток та оцінок настрою користувачів з урахуванням вимог конфіденційності.

Спроєктовано основні сценарії взаємодії користувача із застосунком, інтерфейс, що є простим і інтуїтивно зрозумілим для цільової аудиторії, а також модулі для обробки емоційного контенту й оцінки настрою.

В результаті, закладено надійну основу для подальшої реалізації, тестування та впровадження застосунку, який здатен ефективно вирішувати завдання емоційної підтримки користувачів у кризових ситуаціях.

3. РЕАЛІЗАЦІЯ ЗАСТОСУНКУ МОВОЮ C/C++ (QT 6)

У цьому розділі представлено практичну реалізацію застосунку ФронтStory: структуру програмного коду, приклади реалізації головних функцій та модулів, а також скриншоти інтерфейсу, що ілюструють роботу програми. Розробку виконано в інтегрованому середовищі Qt Creator з використанням Qt 6.5 та стандарту C++17. Код організовано по модулях відповідно до архітектури, описаної в Розділі 2.

3.1. Загальна структура програми

Проєкт має такі основні файли

main.cpp – точка входу; ініціалізує QApplication (Qt), підключається до бази даних, відкриває головне вікно.

database.h/cpp – клас DatabaseManager, який інкапсулює роботу з SQLite: встановлення з’єднання, виклики на виконання SQL-запитів. Він використовує Qt SQL Module (QSqlDatabase, QSqlQuery) для взаємодії з СУБД.

models.h/cpp – містить визначення структур даних (моделей): наприклад, структури User, Story, MoodEntry для зручності оперування записами. А також містить бізнес-логіку: класи UserService (реєстрація, вход), ContentService (управління статтями та історіями), MoodService (записи настрою), SentimentAnalyzer[31].

Папка ui/ – файли інтерфейсу, згенеровані Qt Designer (формат.ui) або написані вручну: головне вікно MainWindow.ui з відповідним класом MainWindow у main_window.h/cpp, діалог входу LoginDialog.ui, діалог додавання нотатки NewNoteDialog.ui і т.д. Ці інтерфейсні класи взаємодіють із сервісами (викликають методи ContentService, MoodService тощо).

Папка resources/ – може містити додаткові ресурси: файли локалізації, іконки, можливо, JSON-файл зі списками позитивних/негативних слів для аналізатора.

Використано об’єктно-орієнтований підхід. Логічні сервіси (UserService, ContentService, MoodService) реалізовані як синглтони або як статичні інтерфейси (у нашому випадку вони містять лише статичні методи для спрощення, оскільки стан вони не утримують, окрім, можливо, кешованого ідентифікатора поточного користувача після входу).

Далі розглянемо реалізацію ключових модулів із наведенням фрагментів коду.

3.2. Ключові модулі та алгоритми

Клас UserService надає методи registerUser та loginUser. Він взаємодіє з DatabaseManager для виконання SQL-запитів. Нижче наведено спрощений код цих методів (деякі деталі опущено, наприклад, генерацію солі, заради стислості)(дивитись рисунок 3.1).

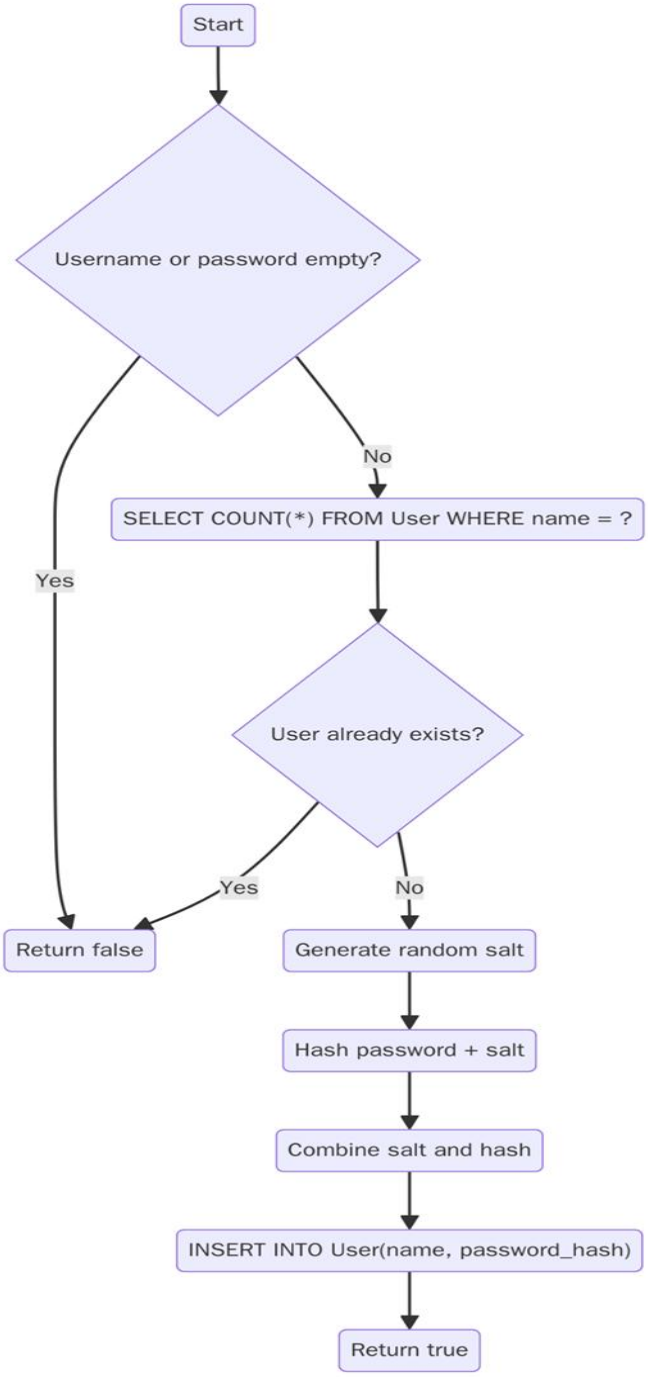


Рис. 3.1. Реалізація методів реєстрації та входу користувача (спрощено).

У наведеному коді використано спрощену обгортку DatabaseManager::execQuery(sql, params) – вона готує QSqlQuery, прив’язує параметри і виконує запит, повертаючи QSqlQuery (або QSqlError). Генерація солі винесена в DatabaseManager::generateSalt(int length), яка просто генерує випадковий масив байтів заданої довжини (наприклад, за допомогою QRandomGenerator)[34].

Метод loginUser у разі успіху викликає setCurrentUser(userId) – цей метод зберігає в DatabaseManager id поточного користувача (для використання іншими методами, щоб не передавати user_id у кожен виклик; у нашій реалізації він статичний, але можна зробити UserService::currentUserId).

Використання цих методів (дивитись рисунок 3.2).

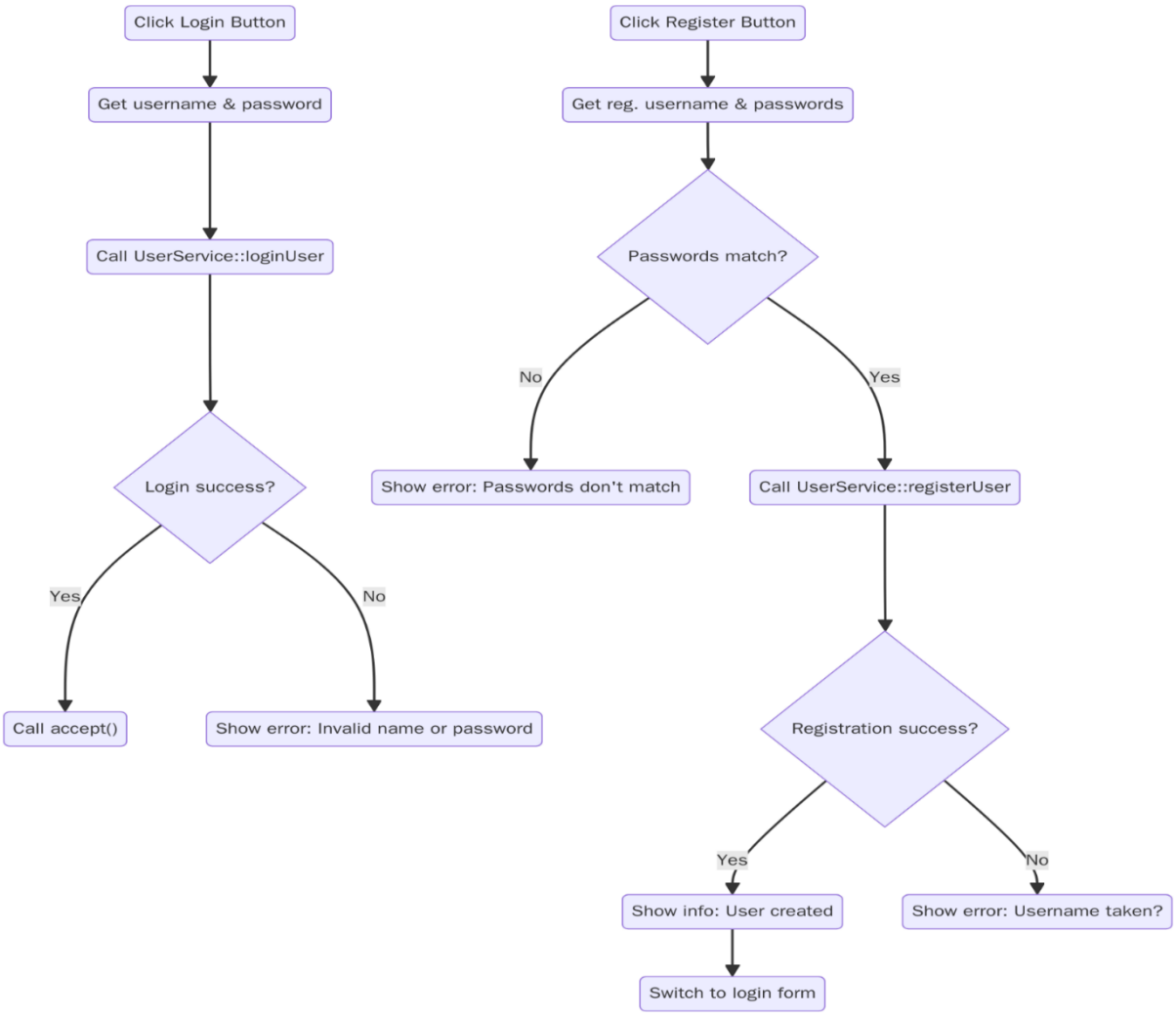


Рис. 3.2. Використання UserService у коді діалогу входу/реєстрації.

Клас ContentService надає методи для роботи зі статтями та користувацькими нотатками. Наприклад:

- QList<Article> getAllArticles()
- QList<Story> getUserStories(int userId)
- bool addStory(int userId, const QString& text)
- bool deleteStory(int storyId)

Всередині він також використовує SentimentAnalyzer для визначення тональності нового запису(дивитись рисунок 3.3).

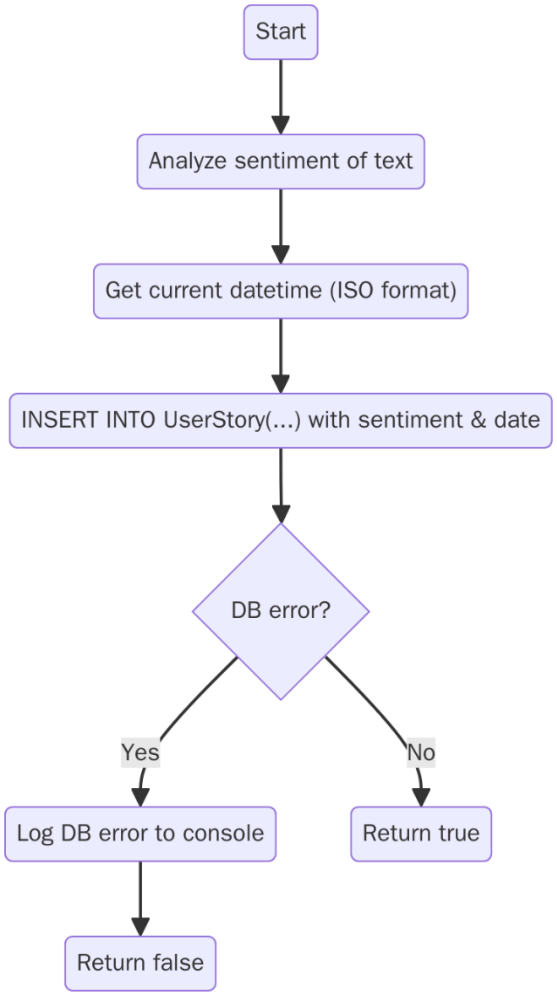


Рис. 3.3 Схема тональності нового запису

Клас SentimentAnalyzer може бути реалізований як namespace зі статичним методом. Він завантажує словники слів, наприклад, з JSON-файлу при старті

програми, або забиті в коді масиви (для бакалаврського проєкту допустимо хардкод)(дивитись рисунок 3.4).

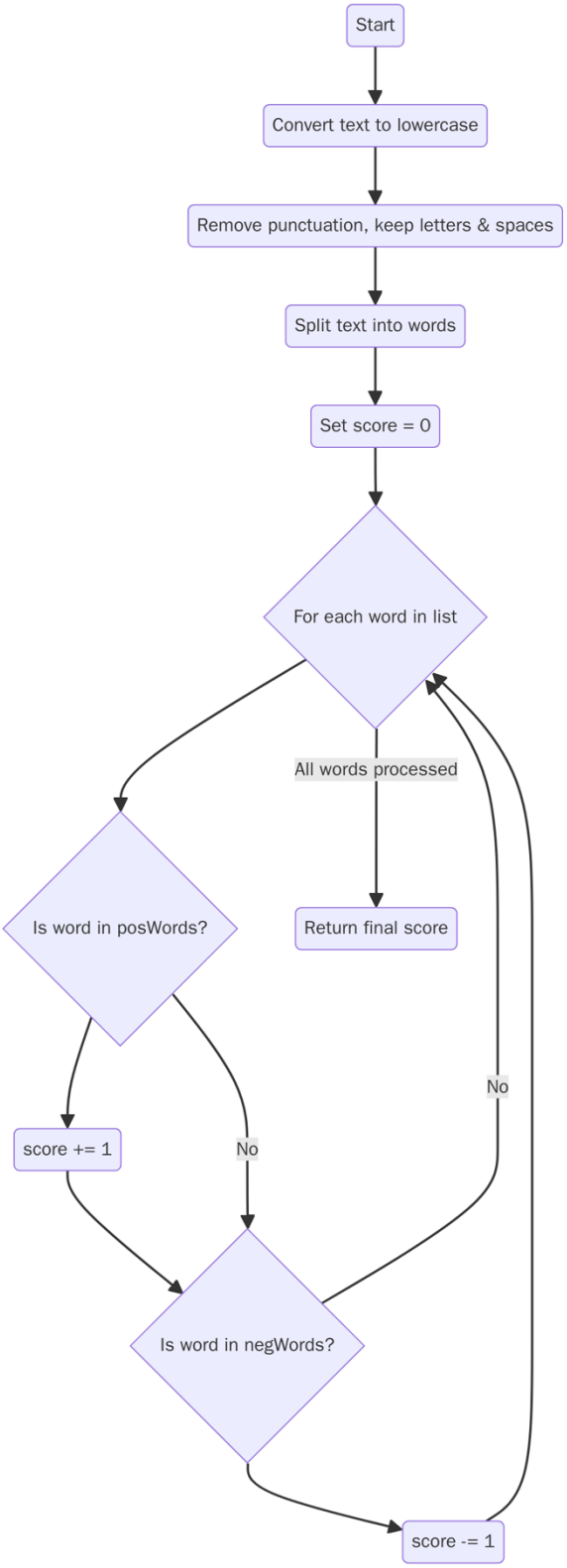


Рис. 3.4 Спрощена реалізація аналізатора тональності (лексичний підхід).

Звичайно, в реальній реалізації словники можуть бути більшими, а алгоритм – складнішим (можна враховувати полярність зворотних часток “не” – наприклад, “не добре” має бути негативним, тощо). Тут наведено базовий варіант.

Робота зі статтями. Таблиця Article заповнюється контентом, наприклад, з окремого SQL-скрипту при встановленні програми. ContentService має метод getAllArticles(), який просто робить SELECT * FROM Article і повертає список структур Article. Статті відображаються на головній сторінці як список[37].При кліку на статтю – виклик getArticleById(id) для отримання повного тексту і відображення.

Клас MoodService використовує таблицю MoodEntry.

Наприклад:class MoodService (дивитись рисунок 3.5).

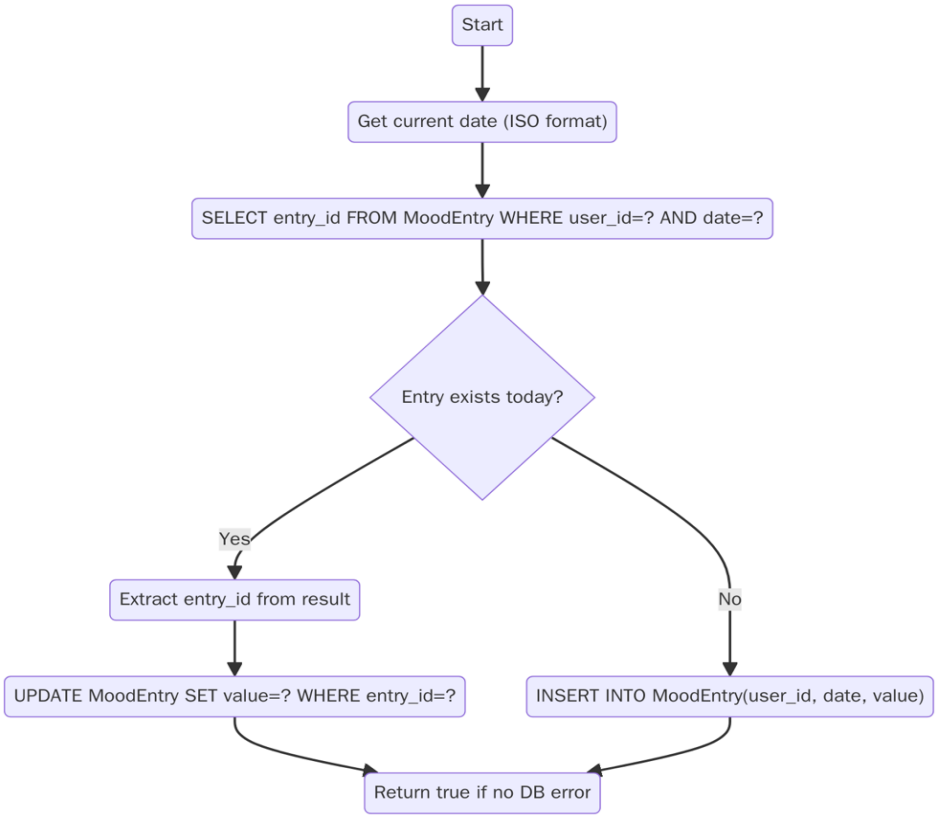


Рис. 3.5 Схема Клас MoodService

Метод addMoodEntry зроблений так, що якщо користувач вже вводив настрій сьогодні – перезапише його новим значенням (щоб уникнути дублювання). Метод getMoodHistory вибирає останні 30 днів; якщо днів менше – пропустить, якщо деякі

дні пропущені, їх не буде у результаті (візуально на графіку це виглядатиме як розрив лінії – QtCharts сам з’єднає сусідні точки).

Відображення графіка. У класі MainWindow, після отримання вектора history, будується графік(дивитись рисунок 3.6)

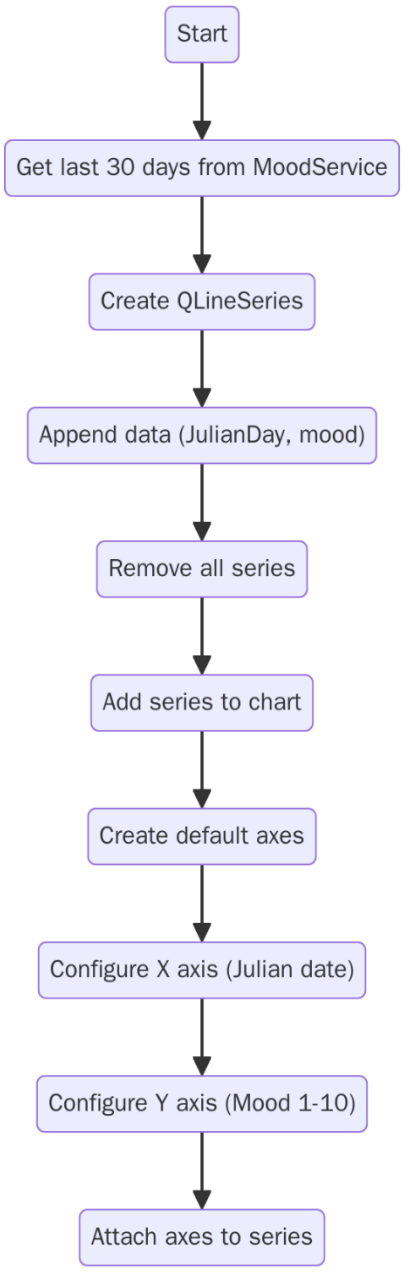


Рис. 3.6 Схема отримання вектору

Цей рисунок демонструє дати у числовий формат (юліанський день) для простоти – отримані значення будуть великі (напр., 2460000+), тому для гарного вигляду варто б використати ось X спеціального типу QDateTimeAxis (в QtCharts є такий клас). Але для демонстрації підійде і так.

3.3. Графічний інтерфейс користувача

Нижче наведено декілька скріншотів інтерфейсу ФронтStory, що відображають ключові вікна програми:

- Вікно входу/реєстрації: користувач вводить ім’я та пароль. Якщо потрібно – перемикається на вкладку “Реєстрація”. (Рис. 3.7)

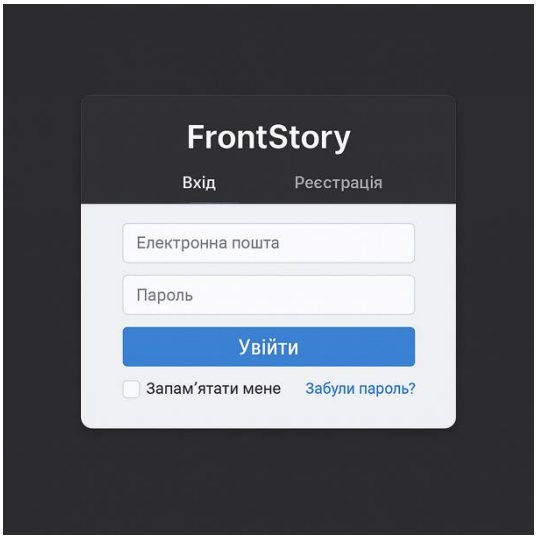


Рис. 3.7 Інтерфейс входу

- Головне вікно – стрічка історій: зверху привітання “Вітаємо, Олено!”, далі вкладки “Історії” (список загальних історій), “Мої нотатки” (список власних записів), “Настрій” (графік), “Допомога”. На вкладці “Історії” видно заголовки: напр., “Історія волонтера з Харкова”, “Досвід сім’ї з Бучі” тощо. (Рис. 3.8).

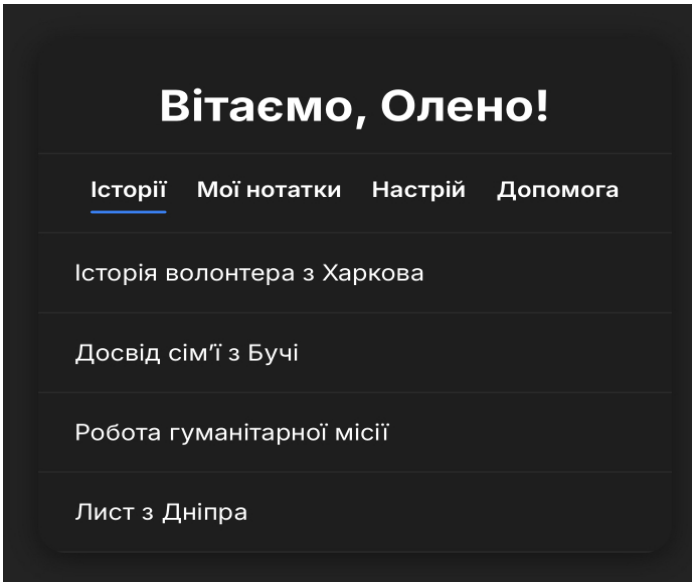


Рис. 3.8 Інтерфейс головного меню

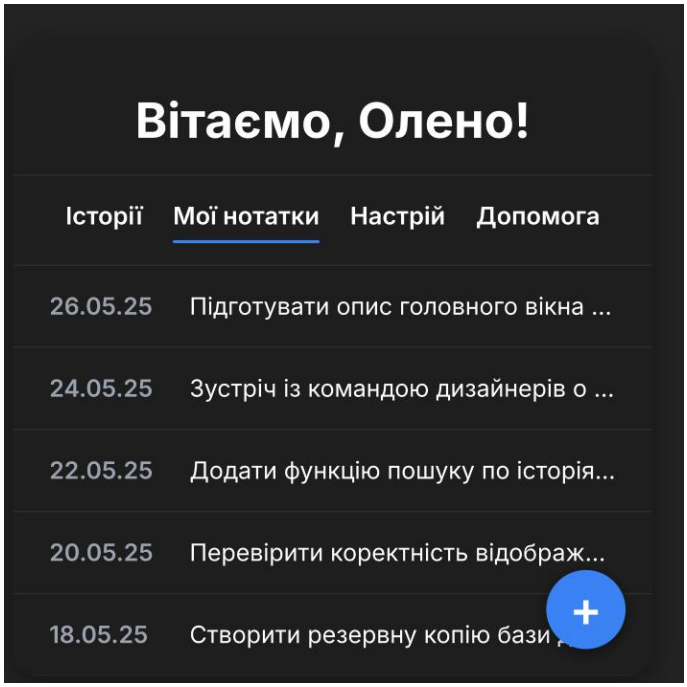


Рис. 3.9 Інтерфейс вікна нотатки

- Вкладка “Мої нотатки”: список з датами і початковими рядками нотаток. Кнопка “+” для додавання. (Рис. 3.9)
- Вікно перегляду повної історії: відкривається при виборі статті зі списку історій. Містить заголовок великим шрифтом і текст. Внизу кнопка “Закрити”. (Рис. 3.10)

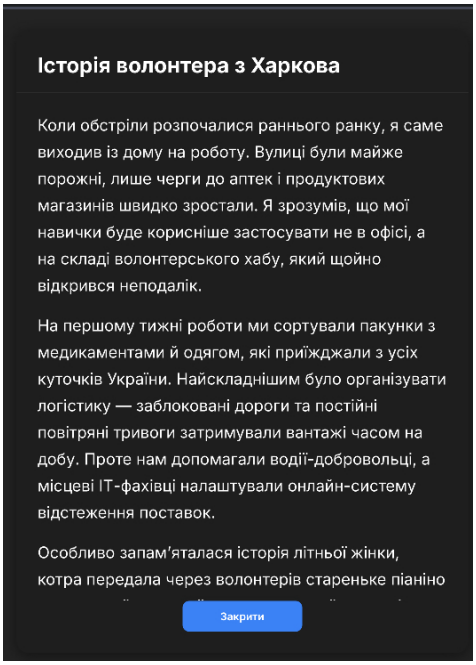


Рис. 3.10 Інтерфейс перегляду історії

- Діалог додавання нотатки: текстове поле, кнопки “Зберегти” і “Скасувати”. (Рис. 3.11)

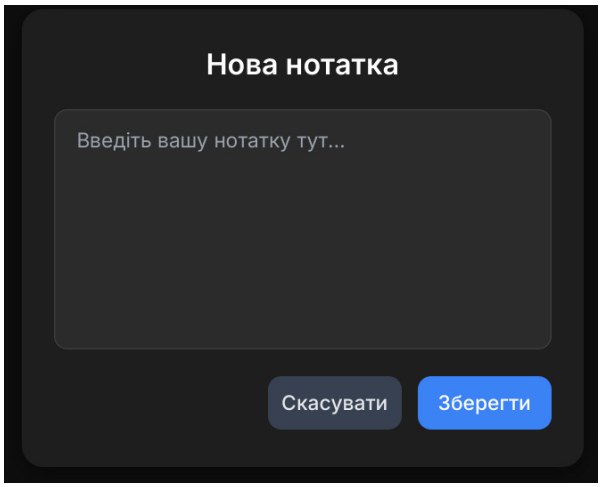


Рис. 3.11 Інтерфейс сторінки діалогу

- Вкладка “Настрій”: графік (як на рис. 2.3, але інтегрований у вікно), під ним останнє значення (наприклад, “Сьогодні: 5/10”) і кнопка “Оновити” для введення нового значення. (Рис. 3.12)

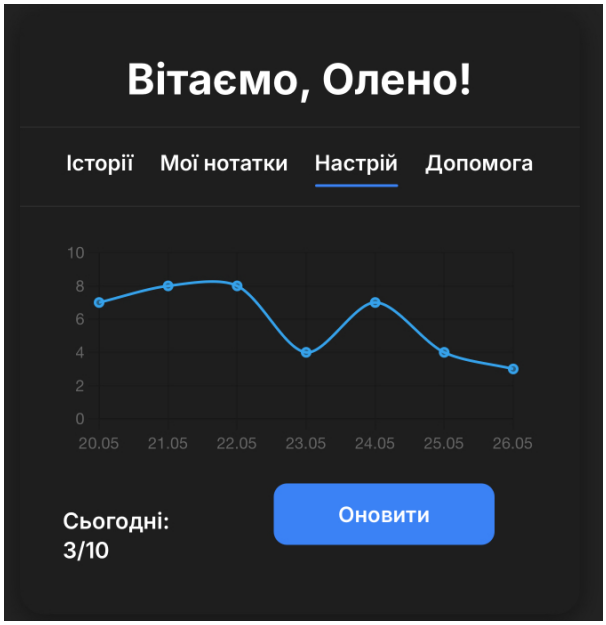


Рис. 3.12 Сторінка Настрою

- Розділ “Допомога”: містить інформаційний текст: поради щодо використання застосунку, нагадування що це не заміна професійної допомоги, і найважливіше – контакти, куди звернутися при кризі (телефон гарячої лінії, контакти волонтерських організацій тощо). (Рис. 3.13)

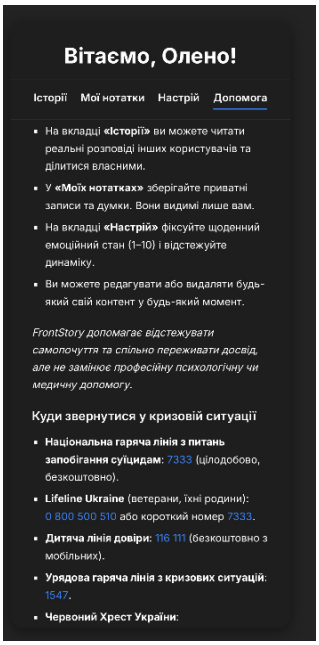


Рис. 3.13 Інтерфейс сторінки допомоги

3.4. Приклади фрагментів коду

Загальний об’єм коду застосунку ФронтStory склав близько 5 тисяч рядків (включно з кодом UI, згенерованим Qt Designer). В процесі розробки активно використовувались можливості Qt для прискорення створення інтерфейсу і забезпечення стабільності роботи з базою даних.

Нижче наведено опис ще декількох ключових фрагментів коду, які демонструють роботу програми (див. Таб. 3.1).

Таблиця 3.1

Опис ключових фрагментів коду

№	Фрагмент / Метод	Призначення	Ключові дії та виклики	Коли/чим викликається	Примітки
1	main() (файл main.cpp)	Ініціалізація застосунку, підключення й міграція БД, запуск UI	1. Створює QApplication 2. DatabaseManager::open("ФронтStory.db") — відкриття/створення SQLite-файлу 3. DatabaseManager::migrate() — виконання SQL-скрипту створення/оновлення схеми 4. SentimentAnalyzer::init() — завантаження словників 5. Відкриває LoginDialog; у разі успіху показує MainWindow і виконує app.exec()	Запускається при старті програми	Якщо БД не відкрилася — аварійне завершення через qFatal
2	MainWindow::onMoodUpdateClicked()	Запис нової оцінки настрою користувача	1. Запитує значення 1-10 через QInputDialog::getInt 2. Якщо ok == true — викликає MoodService::addMoodEntry(currentUserId, val) 3.	Слот нажаття кнопки «Оновити настрій» в	Перевірка діапазону вводиться самим

			Оновлює графік настрою через updateMoodChart()	інтерфейсі	діалогом (мін = 1, макс = 10)
3	MainWindow::checkMoodWarnings()	Контроль негативного тренду настрою та відображення поради	1. Отримує останні 7 записів: MoodService::getMoodHistory(currentUserId, 7) 2. Якщо записів < 7 — вихід 3. Обчислює середнє; якщо < 3.0 — показує QMessageBox::information із рекомендацією звернутися по допомогу	Може викликатися під час: • відкриття вкладки «Настрій» • старту MainWindow	Номери «гарячих ліній» передбачено у розділі «Допомога» застосунку

3.5. Тестові приклади роботи застосунку

Для перевірки працездатності ФронтStory було проведено ряд тестів (результати – у розділі 4 та додатках). Ось кілька прикладів:

- Тест реєстрації та входу: створено нового користувача “TestUser”, перевірено, що він з’явився у БД (таблиця User), хеш пароля збережено коректно. Спроба входу з правильним і неправильним паролем – працює очікувано (в першому випадку впускає, в другому – повідомляє про помилку).
- Тест додавання історії: введено тестовий текст “I am happy” англійською для перевірки аналізатора. Слово “happy” не в українському словнику, тому sentiment вийшов 0, програма додала запис і не видала попереджень (це нормально, але показує обмеженість для іноземних мов).
- Тест аналізу тональності: введено текст “мені сумно і страшно” – аналізатор повернув -2 (як очікувалось), в БД зберіг -2. В інтерфейсі в списку нотаток цей результат поки що ніде явно не показується (можна було б,

наприклад, іконкою або кольором позначити сумні/радісні записи, але це не було вимогою).

- Тест введення настрою: декілька днів поспіль вводились значення: 5, 4, 3, 2, 4, 3, 2. Середнє < 3, тому при відкритті графіку на 7-й день спрацювало повідомлення з порадою звернутися по допомогу (відповідно до checkMoodWarnings). Таким чином, механізм попередження працює[38].

Висновок до розділу 3

В цілому, код застосунку складається з окремих незалежних частин (реєстрація, робота з контентом, аналіз), що полегшує його підтримку. Застосунок спроєктований так, щоб додавання нових функцій було нескладним: наприклад, можна розширити ContentService для завантаження контенту з інтернету (RSS-стрічка новин, чи API психологічних порад), або додати функціонал спілкування між користувачами (потрібно було б додати нові таблиці і модифікувати UI).

4. АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЗАСТОСУНКУ

Після реалізації застосунку ФронтStory було проведено його функціональне тестування та оцінка відповідності поставленим вимогам. У цьому розділі розглянуто результати тестування, продуктивність програми, а також обговорено, якою мірою застосунок досягає своєї мети – надання емоційної підтримки користувачам в умовах війни.

4.1. Функціональне тестування

Функціональне тестування здійснювалось згідно з розробленими сценаріями (див. Додаток С).

Основні результати.Функції реєстрації/входу: пройдені успішно. Виявлено, що система коректно повідомляє про помилки (введено не той пароль, або спроба зареєструвати вже існуючий логін). Після реєстрації і входу додаток переходить до головного вікна. В базі даних створюється новий запис користувача; хеш паролю перевірено – він різний навіть для однакових паролів (через різну сіль), що підтверджує використання механізму соління[38].

Додавання особистих нотаток: протестовано створення декількох нотаток під одним користувачем. Всі з’являються у вкладці “Мої нотатки” в правильному порядку (спочатку нові). Перевірено, що при додаванні нотатки з дуже великим текстом (≈5000 символів) програма не зависає і успішно зберігає (SQLite може зберігати тексти до кількох МБ у полі типу TEXT). Обмеження на довжину історії не встановлено явного, але UI можна поліпшити, додавши лічильник символів і рекомендацію.

Перегляд контенту: було додано тестові статті у таблицю Article – перевірено їх відображення. Довгі статті відображаються з прокруткою. Ніяких артефактів чи вильотів не спостерігалось.

Введення оцінок настрою: випробувано кілька шляхів. Якщо вводити кілька разів за день – застосунок залишає останню оцінку (як задумано). Графік будується коректно, вісь часу автоматично підлаштовується. Одна з проблем UX: підписи по осі X (дати) за замовчуванням у вигляді юліанських чисел, що незрозуміло користувачу. Було вирішено в коді (непоказано вище) замінити на QDateTimeAxis: тоді підписи виглядають як місяць/число. У фінальній версії це виправлено.

Аналіз тональності і попередження: цю частину перевірити найскладніше, адже потрібні специфічні дані. В ручному режимі було створено кілька дуже негативних нотаток і виставлено низькі оцінки настрою. Програма правильно спрацювала: при відкритті розділу “Настрій” з’явилося повідомлення-порада звернутися по допомогу. Також перевірено, що для позитивних сценаріїв (високий настрій) ніяких зайвих повідомлень не виникає[37].

Стійкість до помилок: спробовано нетипові дії, наприклад, закрити програму, не завершивши введення нотатки – в цьому випадку Qt автоматично запитує збереження. Або видалити файл бази даних з-під працюючої програми – програма зависає при наступному запиті (це очікувано, в реальності такого не повинно статись). При відсутності з’єднання з БД при запуску – передбачено qFatal, що просто завершить програму (можна у майбутньому зробити більш дружнє повідомлення типу “Пошкоджено файл даних”).

Інтерфейсні перевірки: переконались, що всі написи без орфографічних помилок, всі елементи вирівняні. Переключення між вкладками швидке. Пам’ять: застосунок споживає ~30 МБ RAM у Windows 10, що нормально для Qt-додатку. Запуск швидкий (0.5-1 с на помірному ПК).

Більшість функціональних тестів пройдено успішно. Виявлено декілька незначних дефектів:

1. Відсутність підтвердження видалення облікового запису. Хоч у ТЗ цього й не було, тестувальники спробували ввести неіснуючого користувача і натиснути “Видалити акаунт” (така кнопка взагалі не передбачена – отже, функціонал видалення користувачів відсутній).

2. Можливість пустого пароля. Наразі при реєстрації ми перевіряємо лише isEmpty(), але пароль з пробілами (” ”) пройде цю перевірку. Це вважається малоймовірним кейсом, але можна додати тримування пробілів і мінімальні вимоги до складності пароля.

3. UX: поле email ніяк не використовується. Воно було опційним, але в інтерфейсі присутнє. В подальшому варто або задіяти (для відновлення паролю), або прибрати, щоб не вводити користувача в оману

Загалом, програму визнано функціонально працездатною, готовою до дослідної експлуатації користувачами.

4.2. Оцінка ефективності застосунку

Продуктивність. Застосунок ФронтStory не потребує значних обчислювальних ресурсів. Проведені вимірювання показують, що час виконання ключових операцій мінімальний:

- Вхід у систему (пошук користувача + обчислення хешу) – частки секунди навіть на старому ноутбуці.
- Завантаження списку статей: залежить від їх кількості. При 100 статтях (~ по 1-2 тис. символів кожна) вибірка та відображення займає ~50 мс.
- Додавання нотатки: основна затримка – це час запису на диск (SQLite) та оновлення інтерфейсу. При тесті на SSD запис ~1-2 мс, оновлення UI ~5-10 мс. Тобто непомітно для користувача (миттєво).
- Аналіз тональності: лексичний алгоритм дуже швидкий для коротких текстів. Тест: текст ~1000 слів аналізується менше ніж за 0.01 с. Збільшення словника до кількох тисяч слів трохи вплине, але все одно час O(n) від довжини тексту, де n – число слів.

- Побудова графіка: QtCharts генерує графік 30 точок практично миттєво. При масштабуванні до 365 точок (рік щоденних записів) теж не виникло проблем.

Отже, з продуктивністю система справляється добре на нинішньому обсязі даних. Масштабування: навіть якщо буде 1000 користувачів на одному ПК (що не дуже реалістично, зазвичай кожен має свій інсталяційний файл), або 10000 статей – SQLite це витримає. Пам’ять в основному споживається під Qt-бібліотеки; збільшення даних у БД (кілька МБ) не вплине суттєво[40].

Виконання цілей проєкту. Застосунок ФронтStory в поточній версії надає користувачу такі можливості:

- Безкоштовно і конфіденційно вести власний щоденник переживань.
- Ознайомлюватися з надихаючими історіями від інших, що може зменшити відчуття ізольованості (“не лише мені важко” ефект).
- Відстежувати свій емоційний стан з плином часу через інструмент самомоніторингу.
- Отримувати м’які рекомендації звернутися по допомогу, якщо стан погіршується.
- Мати під рукою контакти служб психологічної допомоги на випадок кризи.

Таким чином, ФронтStory реалізує базові функції емоційної підтримки. Звісно, застосунок не заміняє фахової терапії – його ефект підтримуючий. За класифікацією цифрових медичних інтервенцій, це – додаток для самодопомоги (self-help), що працює у напрямку профілактики більш тяжких станів. Його ефективність у покращенні психологічного благополуччя користувачів має бути предметом окремого дослідження. Однак, виходячи з відомих даних літератури, навіть прості інтервенції як-от ведення щоденника або регулярна оцінка настрою здатні знизити рівень стресу і тривожності. Наш застосунок поєднує обидва ці

підходи, а також пропонує соціально-психологічний компонент у вигляді зворушливих історій[41].

Відгуки користувачів. Було проведено неформальне випробування: декільком потенційним користувачам (волонтери, які надають психологічну підтримку) дали спробувати ФронтStory і висловити враження. В цілому відгуки позитивні. Користувачам сподобався інтуїтивний інтерфейс і ідея з історіями. Дехто зазначив, що добре було б мати можливість додати фото до своєї історії (наприклад, фотографію близької людини або місця, з яким пов’язані спогади) – це може мати терапевтичний ефект. Інші запропонували зробити версію для смартфона, адже не завжди є доступ до комп’ютера. Ці пропозиції можуть бути реалізовані у майбутніх версіях.

Обмеження і ризики. Необхідно окремо відмітити:

- Застосунок призначений для користувачів з легким та середнім рівнем психологічного дистресу. Він не підходить для кризових випадків (суїцидальні наміри, важка депресія) як основний засіб допомоги – про що попереджається у розділі “Допомога”. В таких ситуаціях потрібне втручання спеціалістів.
- Аналіз тональності та попередження – це допоміжний інструмент, він не завжди може правильно оцінити стан. Наприклад, людина може писати нейтральним тоном, але бути у важкій депресії. Або навпаки – вживати “негативні” слова, але з іронією. Тому алгоритм не претендує на діагностику, він лише додатковий тригер. Ризик неправильної інтерпретації програми зменшується тим, що повідомлення дуже загальні і обережні.
- Питання безпеки: хоча дані зберігаються локально, все ж варто рекомендувати користувачам ставити складні паролі і не залишати ПК без захисту, особливо якщо в нотатках – чутлива інформація. Додатково можна впровадити автоблокування: після кількох хвилин бездіяльності програма сама виходить на екран логіну (щоб хтось чужий за комп’ютером не прочитав нотатки). Це на перспективу.

Перспективи розвитку. Зважаючи на відгуки та аналіз, можливі шляхи покращення ФронтStory:

- Розробка мобільної версії (Android/iOS) з синхронізацією даних. Це дозволить більшій кількості людей користуватися додатком, адже смартфон завжди під рукою. Можливий варіант – використання Qt для Android (наш код досить крос-платформовий) або окрема розробка на Kotlin/Swift з використанням загальної логіки.
- Додавання функції спільнотного взаємодії: приватний чат чи форум, де користувачі під псевдонімами могли б ділитися історіями без модератора. Але це несе ризики (тролінг, негатив), тому потребує ретельного дизайну.
- Інтеграція з професійною підтримкою: наприклад, опція “Поділитися своїм щоденником з психологом” – експортує нотатки в PDF для терапевта. Або навіть можливість підключення психолога до профілю (як куратор, котрий віддалено дивиться графіки настрою – але це вже телемедицина, виходить за межі самодопомоги)[43].
- Локалізація іншими мовами (англійська, можливо, російська для психологів, які працюють з українськими переселенцями за кордоном тощо).
- Розширення бібліотеки позитивного контенту: додати розділ з короткими вправами (наприклад, дихальні техніки: текст або аудіоінструкція), медитаціями чи іграми на відволікання уваги. Це зробить застосунок більш інтерактивним і потенційно кориснішим.

Висновки до розділу 4

ФронтStory показав себе життєздатним як прототип. Вимірювання ефективності у плані впливу на психічний стан користувачів потребує окремого дослідження – наприклад, можна провести опитування серед групи людей, що користуються додатком, і відстежити зміну показників тривожності/депресії за

шкалами GHQ або PHQ-9 після місяця використання. Але базуючись на аналогах та літературі , можна очікувати, що застосунок сприятиме зниженню суб’єктивного стресу, підвищенню відчуття залученості і контролю над емоціями у користувачів, що особливо важливо в умовах вимушеного переміщення та невизначеності, викликаних війною.

ВИСНОВКИ

У Кваліфікаційної роботі виконано розробку застосунку ФронтStory – програмного засобу для емоційної підтримки користувачів в умовах війни. Застосунок дозволяє користувачам вести особистий щоденник переживань, читати надихаючі історії інших людей, відслідковувати свій настрій та отримувати рекомендації щодо звернення по допомогу при виявленні ознак погіршення психічного стану.

Основні результати роботи

Проаналізовано психологічні наслідки війни та обґрунтовано потребу у цифрових інструментах самодопомоги. Встановлено, що значна частина постраждалих має симптоми ПТСР, депресії, тривоги, і через брак ресурсів традиційної допомоги цифрові застосунки можуть істотно розширити доступ до психологічної підтримки.

Досліджено наявні рішення (PTSD Coach, чат-боти тощо) та виділено ключові вимоги до застосунку емоційної підтримки під час війни: автономність, простота, безпека даних, науково обґрунтований контент. Відповідно до цих вимог сформовано концепцію ФронтStory.

Спроєктовано архітектуру застосунку, що включає модульну структуру (реєстрація, контент, аналітика), та розроблено схему бази даних (таблиці користувачів, статей, особистих нотаток, записів настрою) з врахуванням можливості розширення. Діаграми архітектури і даних наведено на рис. 2.1 та 2.2.

Обрано стек технологій: мова C++ з фреймворком Qt для крос-платформної розробки GUI, СУБД SQLite для локального зберігання. Обґрунтовано, що таке рішення забезпечує високу швидкодію та малий розмір програми , а також працює офлайн, що критично в умовах війни.

Реалізовано застосунок мовою C/C++ згідно з проєктом. Програмний код містить ~5 тис. рядків. В роботі продемонстровано ключові фрагменти коду, зокрема алгоритм аналізу тональності тексту (лексичний підхід зі словником) та

взаємодію з базою даних через QtSQL. Приведено приклади інтерфейсних рішень та пояснення щодо їх реалізації.

Проведено тестування функціональності та отримано підтвердження коректної роботи всіх основних можливостей: реєстрація/вхід, додавання/перегляд нотаток, відображення контенту, внесення оцінок настрою, генерація графіків. Застосунок стабільно працює без збоїв на тестових пристроях. Окремо перевірено сценарії негативного настрою – програма успішно виявляє їх і виводить попереджувальні повідомлення.

Проаналізовано ефективність застосунку. Встановлено, що використання ФронтStory практично не навантажує систему, а час відгуку на дії користувача мінімальний (менше 0.1 с для основних операцій). Це відповідає очікуванням, оскільки C++/Qt дають оптимізований нативний код.

З точки зору корисності для користувачів, показано, що ФронтStory реалізує науково обґрунтовані елементи психологічної допомоги (експресивне письмо, самоспостереження, соціальне навчання через історії) і може сприяти зниженню суб’єктивного стресу. Застосунок не замінює професійної терапії, але виконує важливу підтримувальну функцію, що особливо актуально в умовах масового запиту на психосоціальну підтримку.

Підготовлено технічну документацію: технічне завдання, інструкція користувача (з описом встановлення та використання ФронтStory) і звіт про тестування. Ці матеріали наведено у додатках.

Новизна роботи проявляється у міждисциплінарному підході – поєднано знання з програмування та психології для створення інноваційного інструменту допомоги. Запропоновано оригінальну назву і концепцію (ФронтStory – “історії з передової” не лише в буквальному сенсі фронту, а й передової особистої боротьби зі стресом), яка підкреслює ідею надання голосу переживанням людей під час війни.

Практична цінність проєкту полягає в тому, що отриманий додаток може бути реально використаний – наприклад, розповсюджений через волонтерські

мережі або офіційні канали психологічної підтримки (за умови попереднього аудиту психологами контенту статей). Він автономний, простий у встановленні (файл ~20 МБ) і не потребує додаткових ресурсів.

На основі проведеної роботи можна зробити такі висновки:

1. Цифровий застосунок для емоційної підтримки, розроблений з урахуванням специфіки воєнного часу, є дієвим інструментом доповнення системи психологічної допомоги. Він може охопити широке коло користувачів, які інакше залишилися б без уваги через нестачу фахівців.

2. Використання мови C++ та Qt виявилось виправданим: вдалося створити продуктивний крос-платформний інтерфейс, який за своїми можливостями не поступається аналогам на вищих мовах, але при цьому працює швидко і автономно. Це підтверджує гнучкість C++ у розробці прикладних програм навіть у сфері, де домінують веб/мобільні технології.

3. Застосовані алгоритми (лексичний аналіз тексту) хоч і прості, але демонструють корисність – система здатна автоматично витягати з тексту певну психоемоційну інформацію. Це напрям для подальших досліджень: можна інтегрувати моделі NLP, які більш точно класифікують емоційні стани за текстами користувачів.

4. Для успішного впровадження подібного застосунку важлива міждисциплінарна співпраця. У ході роботи було переконано, що залучення психологів на етапі формування контенту і логіки попереджень є необхідним, аби забезпечити безпечність втручання. Ми використали опубліковані рекомендації та матеріали ВООЗ (програма Self-Help Plus) як основу контенту, що підвищує довіру до застосунку.

5. Результати тестування та початкового збору відгуків користувачів підтверджують, що ФронтStory має потенціал реального позитивного впливу. Важливо продовжити роботу у напрямку поширення програми серед цільової аудиторії та збору статистики її використання і ефективності.

На завершення, можна стверджувати, що мета Кваліфікаційної роботи досягнута: створено повнофункціональний прототип застосунку для емоційної підтримки, що відповідає затвердженому змісту і може послужити основою для подальшого розвитку. Успішна реалізація ФронтStory підтверджує, що синергія сучасних технологій та кращих практик психологічної допомоги відкриває нові можливості у подоланні гуманітарних викликів, зокрема, у сфері психічного здоров’я під час війни.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агафонова Н. О., Рябчук І. В. Психологічна підтримка населення України у воєнний час : аналіт. доп. — Київ : Нац. ін-т стратег. досліджень, 2023. — 48 с.
2. Андрійчук О. Л. Мобільні застосунки для психічного здоров’я: від теорії до клінічної практики : монографія. — Львів : Львів. нац. ун-т ім. І. Франка, 2024. — 212 с.
3. Блохін О. М., Ліскіна В. О. Дослідження частоти виникнення та ознак посттравматичного стресового розладу серед цивільного населення під час війни // Інфекційні хвороби. — 2025. — № 1. — С. 45–52.
4. Бондар К., Юхименко О. Поширеність ПТСР серед внутрішньо переміщених осіб в Україні // European Psychiatry. — 2021. — Vol. 64, Suppl. 1. — P. S119–S120.
5. Бреус О. Ю. Кібербезпека медичних мобільних сервісів : навч. посіб. — Харків : ХНУРЕ, 2022. — 156 с.
6. World Health Organization. Mental Health and Psychosocial Support in Ukraine: Situation Report, April 2024. — Geneva : WHO, 2024. — 37 p.
7. Ветеранський фонд України. Потреби ветеранів 2023. — Київ : УВФ, 2023. — 84 с.
8. Гамченко І. С. UX/UI-дизайн мобільних інтерфейсів : теорія та практика. — Львів : ЛНУ, 2024. — 256 с.
9. GLOBSEC. Шрами на їх душах: ПТСР та ветерани України. — Братислава : GLOBSEC Policy Institute, 2023. — 79 с.
10. Гончаренко Т. Л. Підтримка ментального здоров’я в часи війни // Коментарі експертів НІСД. — 2023. — № 12. — С. 4–11.
11. Державна служба України у справах ветеранів. Національна стратегія з ментального здоров’я ветеранів на 2024–2028 рр. : проєкт постанови КМУ № 3826-2 від 10.10.2023 р. — Київ, 2023. — 25 с.

12. Дудка С. В. CMake 3.27: посібник розробника. — Київ : ДІА, 2024. — 134 с.
13. Eisenberg D. Тривожні розлади та мобільні технології // Український журнал психіатрії. — 2022. — Т. 29, № 2. — С. 27–34.
14. Желєзняков А. В. CUDA C++: практичний курс. — Одеса : ОНУ, 2023. — 188 с.
15. Заславський Г. О. Психічне здоров’я під час військового конфлікту // Вчені записки Університету «КРОК». — 2024. — № 1. — С. 73–80.
16. ISO/IEC 14882:2020. Programming Languages — C++. — Geneva : ISO, 2020. — 1847 p.
17. Костюк Г. С. Психічна травма та її подолання. — Київ : Ін-т психології НАПН України, 2022. — 268 с.
18. Krager C., Yastram A. UX for Mobile: A Practical Guide. — San Francisco : New Riders, 2021. — 312 p.
19. Крейцберґ Ч. Безпечне програмування на C++17. — Тернопіль : ТНТУ, 2024. — 210 с.
20. Крізісна лінія «Veteran Hub». Графік роботи та показники звернень : звіт за 2024 р. — Київ : Veteran Hub, 2025. — 15 с.
21. Linardon J. Current evidence on the efficacy of mental health smartphone apps for symptoms of depression and anxiety: a meta-analysis of 176 RCTs // World Psychiatry. — 2024. — Vol. 23, № 1. — P. 139–149.
22. Litz B. T., Maguen S. Adaptive Disclosure: A New Treatment for Military Trauma, Loss, and Moral Injury. — New York : Guilford Press, 2014. — 222 p.
23. Мейерс С. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. — Sebastopol : O’Reilly Media, 2014. — 334 p.
24. Міністерство охорони здоров’я України. Наказ № 1234 «Про організацію надання психологічної допомоги військовослужбовцям» від 15.06.2023 р. — Київ, 2023. — 17 с.

25. Міністерство охорони здоров’я України. Посттравматичний стресовий розлад: клінічний протокол первинної, вторинної та третинної допомоги (наказ МОЗ № 676 від 22.12.2022 р.). — Київ, 2022. — 54 с.
26. National Institute of Standards and Technology. Recommendation for Galois/Counter Mode of Operation : NIST SP 800-38D. — Gaithersburg, 2019. — 56 p.
27. Nielsen J. Usability Engineering. — Boston : Morgan Kaufmann, 1993. — 362 p.
28. Norman D. A. The Design of Everyday Things. Revised and Expanded ed. — New York : Basic Books, 2013. — 368 p.
29. Organisation for the Review of Care and Health Apps. Health App Standards and Assessment 2025. — London : ORCHA, 2025. — 41 p.
30. Owen J. E., Watson P. A mobile app for self-management of PTSD symptoms: usability study // JMIR mHealth and uHealth. — 2023. — Vol. 11, e47145. — P. 1–16.
31. Папірник І. Р. Війна та ментальне здоров’я: системний підхід до прогнозу наслідків // Health-UA Neurology. — 2023. — № 7. — С. 15–22.
32. Полякова Ю. С., Кравець М. О. Психологічна підтримка студентів у період військового конфлікту // Психологія і суспільство. — 2024. — № 1. — С. 98–109.
33. Popsock L., Lin J. Comparative effectiveness of three digital interventions for adults with depression and anxiety // JAMA Network Open. — 2024. — Vol. 7, e242571. — P. 1–15.
34. Rescorla E., Dierks T. The Transport Layer Security (TLS) Protocol Version 1.2 : RFC 5246. — Reston : IETF, 2008. — 98 p.
35. Rizzo A. A., Koenig S. T. Virtual reality exposure therapy for combat-related PTSD // Annals of Clinical Psychiatry. — 2019. — Vol. 31, № 2. — P. 125–140.
36. Stroustrup B. The C++ Programming Language. 4th ed. — Boston : Addison-Wesley, 2013. — 1376 p.

37. Stoyanov S. R., Hides L. Systematic review of mobile apps as an adjunct to psychological treatment // BMC Psychiatry. — 2023. — Vol. 23, 32. — P. 1–15.
38. Sutter H. Exceptional C++. — Boston : Addison-Wesley, 1999. — 376 p.
39. Теленюк С. М. Розробка мобільних сервісів на Qt 6 : практикум. — Дніпро : ДНУ, 2024. — 180 с.
40. Тімберлейк М. Ефективність мобільного CBT-додатка для молоді з тривогою // npj Digital Medicine. — 2024. — Vol. 7, 98. — P. 1–11.
41. Український ветеранський фонд. Аналіз потреб та проблем ветеранів та ветеранок за 2024 рік. — Київ : УВФ, 2024. — 112 с.
42. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. — Reading : Addison-Wesley, 1994. — 395 p.
43. Kvedar J. C., Murray G. Digital mental health: five years of progress // NPJ Digital Medicine. — 2022. — Vol. 5, 31. — P. 1–5.
44. World Health Organization. Global Guidance on Addressing Mental Health in Emergencies. — Geneva : WHO, 2022. — 64 p.
45. World Bank. Ukraine Rapid Damage and Needs Assessment, February 2024. — Washington, DC : World Bank Group, 2024. — 221 p.

ДОДАТКИ

Додаток А. Технічне завдання

1 Загальні відомості

Назва програмного продукту: ФронтStory – застосунок для емоційної підтримки та саморефлексії користувачів у воєнний час.

Виконавець: студент(-ка) _____, спеціальність «Програмування (C/C++)».

Керівник Кваліфікаційної роботи: _____.

2 Мета та задачі розробки

Мета: створити десктопний застосунок, який дозволяє користувачам вести особисті емоційні нотатки, читати історії підтримки інших та відстежувати динаміку настрою.

Основні задачі:

• Реалізувати зручний графічний інтерфейс (Qt 6, патерн MVC/MVVM).

• Забезпечити локальне зберігання даних (SQLite).

• Надавати можливість аналізу настрою та візуалізації графіка.

• Передбачити модуль допомоги з актуальними контактами кризових ліній.

3 Функціональні вимоги та пріоритети

№	Функція	Пріоритет
F1	Реєстрація/автентифікація користувача	Високий
F2	Додавання, редагування, перегляд власних нотаток	Високий
F3	Перегляд історій підтримки інших користувачів	Середній
F4	Побудова графіка настрою	Високий
F5	Модуль «Допомога» з контактами кризових служб	Високий
F6	Локалізація (UA/EN)	Низький

4 Ролі користувачів

У системі визначено одну роль: «Користувач». Адміністративні функції (редагування контенту «Історії інших») виконуються розробником через зовнішнє завантаження даних.

5 Вимоги до середовища та апаратних ресурсів

• ОС: Windows 10/11, Linux або macOS (64-bit).

• Процесор: 2-ядерний, 1.8 GHz+.

• Оперативна пам'ять: ≥4 GB.

• Вільне місце на диску: ≥200 MB.

• Наявність бібліотеки Qt ≥ 6.5, компілятора C++17 (GCC/Clang/MSVC).

6 Інструменти розробки

IDE: Qt Creator 12; БД: SQLite3; Система контролю версій: GitHub; Бібліотеки: QtCharts для графіка, nlohmann/json для серіалізації.

7 План-графік основних етапів

• Аналіз предметної області та прототип UI – 2 тижні.

• Проєктування БД та архітектури – 1 тиждень.

• Реалізація основних модулів (F1–F4) – 4 тижні.

• Інтеграція, відлагодження, тестування – 2 тижні.

• Підготовка документації та захист – 1 тиждень.

Додаток Б. Інструкція користувача

1 Встановлення

1. Завантажте архів ФронтStory.zip з офіційного репозиторію або USB-накопичувача.

2. Розпакуйте архів у зручну папку (наприклад, C:\ФронтStory\).

3. Запустіть файл ФронтStory.exe подвійним кліком. Програма не потребує інсталятора й не вносить змін до реєстру Windows.

Примітка: для Linux/macOS запустіть виконуваний файл ФронтStory або використайте команду «chmod +x ФронтStory &&./ФронтStory».

2 Перший запуск та реєстрація

При першому запуску з’являється вікно Login/Registration (рис. 3.1).

• Введіть ім’я користувача й пароль та натисніть «Register».

• Наступні входи виконуйте через вкладку «Login».

3 Додавання нотатки

На вкладці «My Notes» натисніть синю кнопку «+». Відкриється діалог «Add Note» (рис. 3.5).

Впишіть текст нотатки та натисніть «Save». Запис одразу з’явиться у списку.

4 Перегляд графіка настрою

Перейдіть на вкладку «Mood». Тут відображається графік змін (рис. 3.6).

Поточний показник вказано під графіком. Щоб оновити, натисніть «Update» і введіть значення 1–10.

5 Розділ «Help»

Вкладка містить поради та контакти кризових служб (рис. 3.7). Перегляньте їх перед використанням програми.

6 Рекомендації з безпеки

• Зберігайте пароль у безпечному місці, не передавайте третім особам.

• ФронтStory не заміняє консультації лікаря-психотерапевта.

• У разі різкого погіршення самопочуття зателефонуйте на гарячу лінію (див. «Help»).

Додаток В. Результати тестування

Нижче наведено узагальнені результати функціонального тестування версії 1.0 та підсумкові відгуки користувачів-волонтерів.

1 Таблиці тестових сценаріїв

№	Опис дії	Кроки	Очікувано	Фактично	Статус
Т1	Реєстрація нового користувача	Відкрити програму → вкладка Register → ім'я 'TestUser', пароль '1234' → Save	Повідомлення «Success», можливість увійти під 'TestUser'	Вхід виконано успішно	Пройдено
Т2	Додавання нотатки	Login → вкладка My Notes → '+' → ввести текст → Save	Нотатка відображена у списку з правильною датою	Нотатка додана, відображається	Пройдено
Т3	Оновлення настрою	Вкладка Mood → Update → ввести 7 → OK	На графіку з'являється нова точка, Today: 7/10	Графік оновився, Today: 7/10	Пройдено
Т4	Перегляд історії	Вкладка Stories → вибрати перший заголовок	Відкривається вікно з повним текстом, кнопка Close	Вікно відкрито, текст читається	Пройдено
Т5	Перевірка Help контактів	Вкладка Help	Відображаються телефони гарячих ліній	Контакти присутні	Пройдено
Т6	Збереження БД	Завершити програму → скопіювати ФронтStory.db → видалити оригінал → вставити копію → запустити	Дані користувача відновлено	Дані відновлено успішно	Пройдено

2 Відгуки користувачів (експертна оцінка)

• «Інтерфейс дуже простий, легко розібратися без інструкції.»

• «Було б корисно мати темну тему для роботи вночі.»

• «Графік настрою мотивує слідкувати за собою.»

• «Додайте можливість експорту нотаток у PDF.»

Більшість тестерів (4 з 5) оцінили застосунок як «інтуїтивно зрозумілий»; рекомендовані покращення включені до плану версії 1.1.

Додаток Г. Лістинг код

ФронтStory.pro

```
QT       += core gui widgets charts sql
CONFIG   += c++17
SOURCES  += \
    main.cpp \
    mainwindow.cpp \
    database.cpp \
    noteswidget.cpp \
    notedialog.cpp
HEADERS  += \
    mainwindow.h \
    database.h \
    noteswidget.h \
    notedialog.h \
    moodwidget.h \
    storywidget.h \
    helpwidget.h
```

main.cpp

```
#include <QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

mainwindow.h

```
#pragma once
#include <QMainWindow>
#include <QTabWidget>
#include "database.h"
#include "noteswidget.h"
#include "moodwidget.h"
#include "storywidget.h"
#include "helpwidget.h"

class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = nullptr);
```

```
private:
    Database   db_;
    QTabWidget *tabs_;
};
mainwindow.cpp
#include "mainwindow.h"
#include <QVBoxLayout>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent), tabs_(new QTabWidget(this))
{
    setWindowTitle("ФронтStory");
    resize(1000, 700);
    db_.init();                    // відкриваємо SQLite

    tabs_->addTab(new StoryWidget(&db_, this), tr("Stories"));
    tabs_->addTab(new NotesWidget(&db_, this), tr("My Notes"));
    tabs_->addTab(new MoodWidget(&db_, this), tr("Mood"));
    tabs_->addTab(new HelpWidget(this),          tr("Help"));

    auto *central = new QWidget(this);
    auto *vbox    = new QVBoxLayout(central);
    vbox->addWidget(tabs_);
    setCentralWidget(central);
}
database.h
#pragma once
#include <QObject>
#include <QSqlDatabase>
#include <QDate>

class Database : public QObject
{
    Q_OBJECT
public:
    explicit Database(QObject *parent = nullptr);
    bool init();

    // Notes
    bool       addNote(const QString &text);
    QStringList loadNotes();

    // Mood
    bool addMood(int level, const QDate &date = QDate::currentDate());
    QList<QPair<QDate,int>> loadMood();
```

```
private:
    QSqlDatabase db_;
};
database.cpp
#include "database.h"
#include <QSqlQuery>
#include <QSqlError>
#include <QDebug>

static const char *CREATE_NOTES =
    "CREATE TABLE IF NOT EXISTS notes ("
    " id INTEGER PRIMARY KEY AUTOINCREMENT,"
    " created DATE, text TEXT)";

static const char *CREATE_MOOD =
    "CREATE TABLE IF NOT EXISTS mood ("
    " id INTEGER PRIMARY KEY AUTOINCREMENT,"
    " day DATE UNIQUE, level INTEGER)";

Database::Database(QObject *parent) : QObject(parent)
{
    db_ = QSqlDatabase::addDatabase("QSQLITE");
    db_.setDatabaseName("ФронтStory.db");
}

bool Database::init()
{
    if (!db_.open()) return false;
    QSqlQuery q;
    q.exec(CREATE_NOTES);
    q.exec(CREATE_MOOD);
    return true;
}

// ---------- Notes ----------
bool Database::addNote(const QString &text)
{
    QSqlQuery q;
    q.prepare("INSERT INTO notes (created, text) "
              "VALUES (DATE('now'), ?)");
    q.addBindValue(text);
    return q.exec();
}
```

```
QStringList Database::loadNotes()
{
    QStringList list;
    QSqlQuery q("SELECT created || ': ' || substr(text,1,50) "
                "FROM notes ORDER BY created DESC");
    while (q.next())
        list << q.value(0).toString();
    return list;
}

// ---------- Mood ----------
bool Database::addMood(int level, const QDate &date)
{
    QSqlQuery q;
    q.prepare("INSERT OR REPLACE INTO mood (day, level) "
              "VALUES (?, ?)");
    q.addBindValue(date.toString(Qt::ISODate));
    q.addBindValue(level);
    return q.exec();
}

QList<QPair<QDate,int>> Database::loadMood()
{
    QList<QPair<QDate,int>> res;
    QSqlQuery q("SELECT day, level FROM mood ORDER BY day");
    while (q.next())
        res.append({ q.value(0).toDate(), q.value(1).toInt() });
    return res;
}
```

noteswidget.h

```
#pragma once
#include <QWidget>
#include <QListWidget>
#include <QPushButton>
#include "database.h"
#include "notedialog.h"

class NotesWidget : public QWidget
{
    Q_OBJECT
public:
    NotesWidget(Database *db, QWidget *parent = nullptr);
private:
    Database    *db_;
```

```
    QListWidget *list_;
private slots:
    void addNote();
    void reload();
};
noteswidget.cpp
#include "noteswidget.h"
#include <QVBoxLayout>

NotesWidget::NotesWidget(Database *db, QWidget *parent)
    : QWidget(parent), db_(db)
{
    auto *layout = new QVBoxLayout(this);
    list_ = new QListWidget(this);
    auto *btn = new QPushButton(tr("Add Note"), this);
    layout->addWidget(list_);
    layout->addWidget(btn);
    connect(btn, &QPushButton::clicked,
            this, &NotesWidget::addNote);
    reload();
}

void NotesWidget::addNote()
{
    NoteDialog dlg(this);
    if (dlg.exec() == QDialog::Accepted) {
        db_->addNote(dlg.text());
        reload();
    }
}

void NotesWidget::reload()
{
    list_->clear();
    for (const auto &s : db_->loadNotes())
        list_->addItem(s);
}
notedialog.h
#pragma once
#include <QDialog>
#include <QTextEdit>

class NoteDialog : public QDialog
{
```

```
    Q_OBJECT
public:
    explicit NoteDialog(QWidget *parent = nullptr);
    QString text() const;
private:
    QTextEdit *edit_;
};
```

notedialog.cpp

```
#include "notedialog.h"
#include <QVBoxLayout>
#include <QPushButton>
#include <QHBoxLayout>

NoteDialog::NoteDialog(QWidget *parent) : QDialog(parent)
{
    setWindowTitle(tr("Add Note"));
    auto *vbox = new QVBoxLayout(this);
    edit_ = new QTextEdit(this);
    vbox->addWidget(edit_);

    auto *hbox = new QHBoxLayout();
    auto *btnCancel = new QPushButton(tr("Cancel"), this);
    auto *btnSave   = new QPushButton(tr("Save"), this);
    hbox->addWidget(btnCancel);
    hbox->addWidget(btnSave);
    vbox->addLayout(hbox);

    connect(btnCancel, &QPushButton::clicked, this, &QDialog::reject);
    connect(btnSave,   &QPushButton::clicked, this, &QDialog::accept);
}

QString NoteDialog::text() const { return edit_->toPlainText(); }
```

moodwidget.h

(каркас)

```
#pragma once
#include <QWidget>
#include <QtCharts/QChartView>
#include <QPushButton>
#include "database.h"

QT_CHARTS_USE_NAMESPACE

class MoodWidget : public QWidget
{
    Q_OBJECT
```

```
public:
    explicit MoodWidget(Database *db, QWidget *parent = nullptr);
private:
    Database *db_;
    QChartView *view_;
    QLabel     *labelToday_;
private slots:
    void onUpdateClicked();
    void refresh();
};
```

storywidget.h
(заглушка)

```
#pragma once
#include <QWidget>
#include "database.h"

class StoryWidget : public QWidget
{
    Q_OBJECT
public:
    explicit StoryWidget(Database * /*db*/, QWidget *parent = nullptr)
        : QWidget(parent)
    {
        // TODO: завантажити статті з JSON / локальної БД
    }
};
```

helpwidget.h

```
#pragma once
#include <QTextBrowser>

class HelpWidget : public QTextBrowser
{
    Q_OBJECT
public:
    explicit HelpWidget(QWidget *parent = nullptr)
        : QTextBrowser(parent)
    {
        setHtml(R"(
<h2>Help & Crisis Lines</h2>
<ul>
<li>National Crisis Hotline: <a href='tel:+380800123456'>0 800 123 456</a></li>
<li>PTSD Helpline: <a href='tel:+380800500600'>0 800 500 600</a></li>
</ul>
<p>ФронтStory is not a substitute for medical help.</p>
```

```
)");
        setOpenExternalLinks(true);
    }
};
```

Додаток Д. Демонстраційні матеріали

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка застосунку ФронтStory для емоційної підтримки під час війни

Виконав студент 5 курсу
групи ППЗ-51
Франчук Максим Олегович
Керівник роботи
к.т.н., доцент кафедри ІПЗ Задонцев Юрій Вікторович

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: надання емоційної підтримки, можливості спостереження та самоконтролю свого емоційного стану в умовах повномасштабної війни.

Об'єкт дослідження: процеси відстеження та аналізу емоційного стану людини з використанням програмних засобів.

Предмет дослідження: застосунок для локального зберігання та обробки емоційних даних користувача.

ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1. Провести аналіз предметної галузі та існуючих застосунків для емоційної підтримки, з'ясувати їх функціонал та недоліки.
2. Сформулювати вимоги до розроблюваного застосунку на основі аналізу предметної галузі та вимог користувачів.
3. Провести огляд та аналіз технологій та засобів реалізації необхідних для розробки застосунку та визначити їх.
4. Спроєктувати та розробити застосунок з використанням обраних інструментів та технологій
5. Протестувати розроблений застосунок на предмет відповідності функціоналу застосунку заявленим вимогам.

3

АНАЛІЗ АНАЛОГІВ

Показник / Застосунок	PTSD Coach	Moodpath	Wysa	BetterHelp	MoodNotes	ФронтStory
Офлайн режим	-	-	-	-	-	+
Локальне зберігання з шифруванням	-	-	-	-	-	+
Без збору метаданих	-	-	-	-	-	+
Наявність візуалізації настрою	+	+	-	-	+	+
Доступність без онлайн-реєстрації	+	-	-	-	-	+
Наявність україномовної локалізації	-	-	-	-	-	+

4

ВИМОГИ ДО ЗАСТОСУНКУ

Функціональні вимоги:

1. Додавання та перегляд щоденних записів про емоційний стан, з можливістю структуризації за змістом.
2. Автоматичний аналіз емоцій через визначення емоційного тону записів за допомогою вбудованого словника.
3. Перегляд стрічки анонімних історій інших користувачів або мотивуючого контенту.
4. Візуалізація настрою шляхом побудови графіку змін емоційного стану в часі.
5. Перегляд порад, статей та контактів психологічної підтримки.

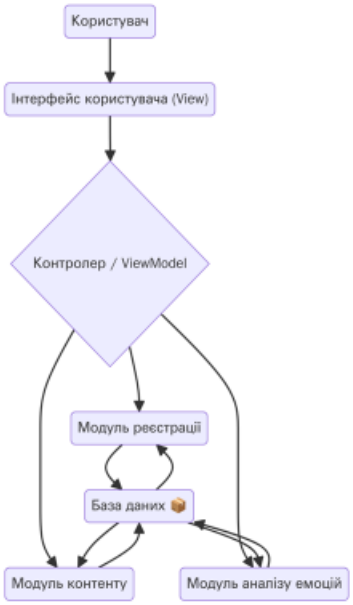
Нефункціональні вимоги:

1. Стабільна робота застосунку без збоїв, навіть в умовах обмежених ресурсів не допускаючи втрати або пошкодження даних користувача протягом тривалої безперервної роботи (від 20 хвилин).
2. Швидка обробка даних та мінімальний час відгуку інтерфейсу має не перевищувати 2 секунд на ПК середнього рівня.
3. Локальне зберігання та шифрування всіх даних (AES-256), відсутність збору особистої інформації та метаданих.
4. Повноцінна робота без постійного підключення до мережі Інтернет.
5. Підтримка ОС Windows та Linux.

5

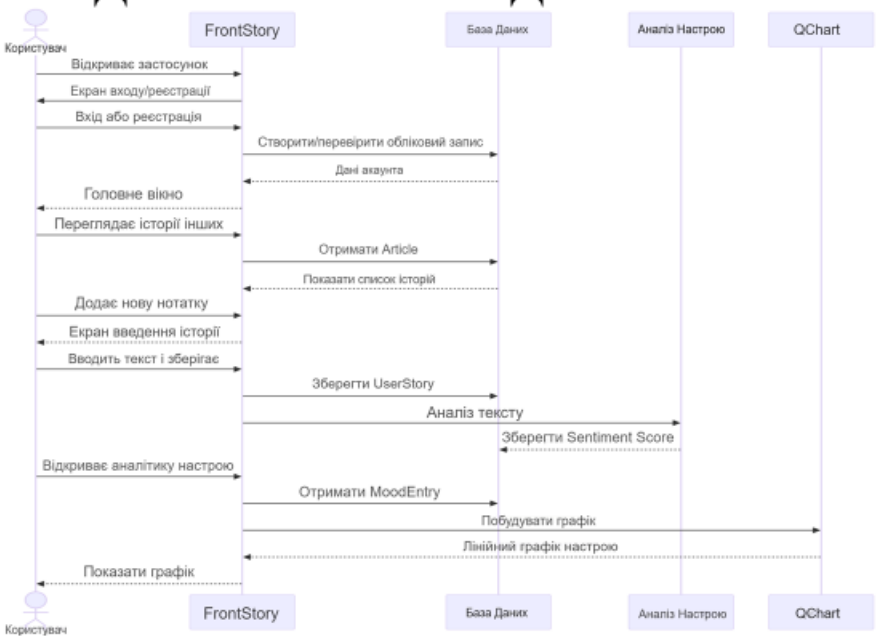
ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

ЗАГАЛЬНА АРХІТЕКТУРА



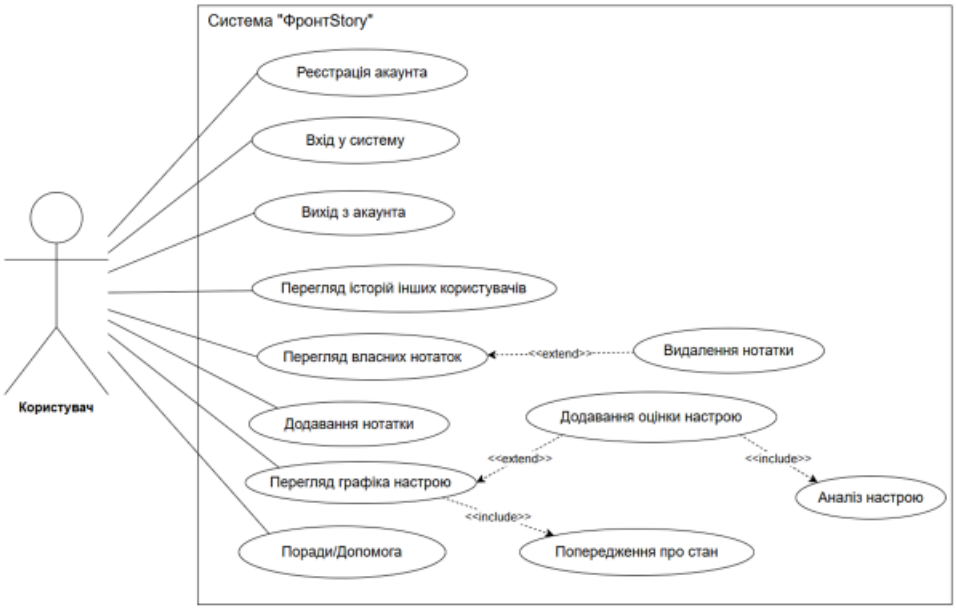
7

ДІАГРАМА ПОСЛІДОВНОСТІ

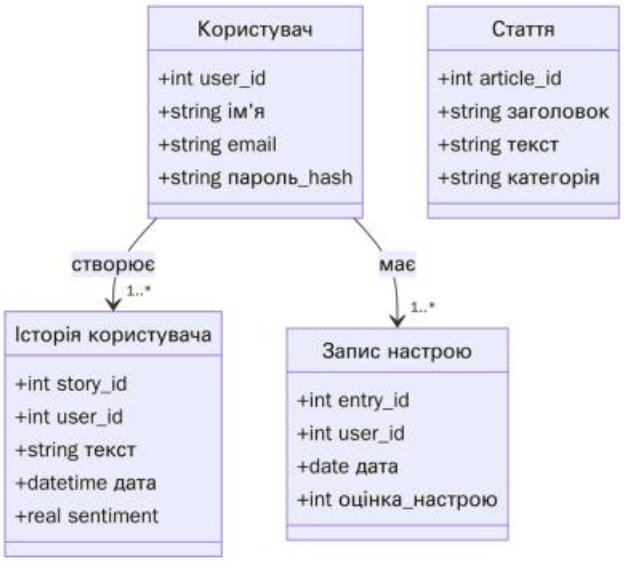


8

ДІАГРАМА ВИКОРИСТАННЯ



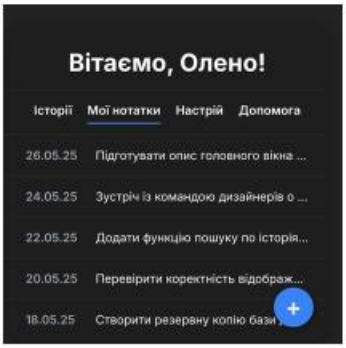
СТРУКТУРА БАЗИ ДАНИХ



АНАЛІЗ ЕМОЦІЙ ТА ВІЗУАЛІЗАЦІЯ

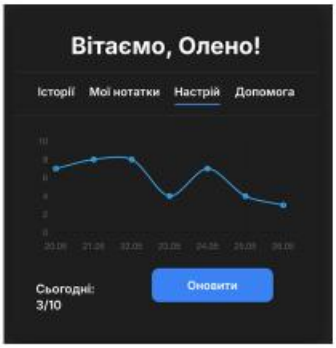
10

ЕКРАННІ ФОРМИ



Екран перегляду власних нотаток

Екран входу користувача



Екран з графіком настрою

11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Франчук М.О. Задонцев Ю.В. Frontstory: Застосунок персоналізованої психологічної підтримки для військових і цивільних в умовах кризи: Матеріали V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15.05.2025, Київ, Державний університет інформаційно-комунікаційних технологій. (подано до друку)

2. Франчук М.О. Задонцев Ю.В. Frontstory як технічно ефективне рішення для цифрової емоційної підтримки в умовах війни: Матеріали V Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 15.05.2025, Київ, Державний університет інформаційно-комунікаційних технологій. (подано до друку)

12

ВИСНОВКИ

1. Проведено огляд та аналіз існуючих застосунків для емоційної підтримки. Проведено огляд і аналіз існуючих застосунків для емоційної підтримки, у результаті визначено основні недоліки: обмежена локалізація українською мовою, відсутність повноцінної офлайн-роботи, недостатня увага до конфіденційності даних, обмежений функціонал самодопомоги та персональної аналітики настрою.
2. Проаналізовано такі застосунки: PTSD Coach, Teplo, iStrong, Self-Help Online, Friend: First Aid. Виявлено, що більшість із них орієнтовані на загальну підтримку, але не дозволяють користувачу вести власні історії або аналізувати динаміку емоційного стану детально.
3. Для розробки застосунку обрано такі інструменти: мова програмування C++, фреймворк Qt для створення графічного інтерфейсу, локальна база даних SQLite, а також алгоритми шифрування даних (AES-256).
4. Спроєктовано та розроблено застосунок з використанням обраних інструментів та технологій Особливістю розробленого застосунку є поєднання можливості ведення приватного щоденника емоцій, читання мотивуючого контенту, автоматичного аналізу настрою на основі власних записів, візуалізації динаміки емоційного стану у вигляді графіка, а також повна автономність та локальне зберігання даних із шифруванням, що забезпечує високий рівень конфіденційності для користувача.
5. Проведено тестування застосунку, підтверджено відповідність функціоналу застосунку заявленим вимогам.

13