

УДК 004:316.444]:316.77

DOI: 10.31866/2617-796X.6.2.2023.293596

**Олександр Ткаченко,**

кандидат фізико-математичних наук, доцент,  
доцент кафедри інформаційних технологій,  
Державний університет інфраструктури  
та технологій,  
Київ, Україна  
aatokg@gmail.com  
<https://orcid.org/0000-0001-6911-2770>

**Денис Гончаренко,**

магістрант, кафедра інформаційних технологій,  
Державний університет інфраструктури  
та технологій,  
Київ, Україна  
rogalikck@gmail.com  
<https://orcid.org/0009-0002-3648-5882>

## ДЕЯКІ АСПЕКТИ РОЗРОБКИ МЕСЕНДЖЕРІВ – ЗАСОБІВ СОЦІАЛЬНОЇ КОМУНІКАЦІЇ

**Мета статті** – дослідити та проаналізувати різні desktop-месенджери, виявити наявні проблеми й шляхи їх вирішення, застосувати інноваційні підходи до реалізації desktop-месенджера, який поєднує в собі гарантування високого рівня безпеки та відмінної продуктивності з широким набором функціональних можливостей, що відповідають сучасним вимогам користувачів.

Desktop-месенджери стали важливим складником повсякденної комунікації багатьох категорій користувачів, забезпечуючи зручний і швидкий спосіб обміну різноманітною інформацією. У статті розглянуто, зокрема, як інтеграція сучасних технологій в комунікаційний процес сприяє зміні підходу до комунікацій, збільшенню обсягу інформації, що передається, зменшенню часу на отримання відповіді, надання користувачеві відомостей про отримання переданої інформації його «співрозмовником» (чи «співрозмовниками»).

**Методами дослідження** є основні методологічні підходи та технологічні засоби для створення високопродуктивних і зручних у користуванні месенджерів.

**Новизною проведеного дослідження** є аналіз сучасних технологій розробки платформ підтримки онлайн-процесу комунікацій і спілкування, результати якого можуть застосовуватися під час розробки власного програмного продукту – месенджера.

У статті обґрунтовано актуальність розробки desktop-месенджера, яка зумовлена тим, що швидкий і безпечний обмін повідомленнями має на сьогодні велике значення, бо сучасні месенджери стали одним з найважливіших засобів спілкування, що сприяє віддаленій роботі й так званому «соціальному» спілкуванню.

**Висновки.** У роботі досліджено відомі погляди на сучасний підхід до розробки desktop-месенджерів. Проведений аналіз обумовив вибір концепції розробки сучасного

авторського месенджера на основі використання C++ та Qt і, як результат, розробленого на її основі прототипу авторського месенджера.

Розробка месенджера з використанням C++ і Qt сприяє вирішенню проблем безпеки та приватності комунікацій (зокрема, захищаючи особисту інформацію, що поширюється в месенджері). Підтримка шифрування, захист даних й управління правами доступу реалізуються завдяки можливостям C++ та Qt.

Запропонований підхід до розробки месенджера забезпечує одне з найкращих рішень, зменшуючи розподіл між frontend- і backend-частинами та полегшуючи гнучкість розробки. Це дає змогу створювати масштабовані та високопродуктивні месенджери. Крім того, цей підхід гарантує безпеку й оптимізацію продуктивності desktop-месенджерів і підвищення рівня їх функціональності та зручності використання.

**Ключові слова:** комунікації; спілкування; інтерфейс користувача; desktop-месенджер; безпека даних; frontend; backend; SSL/TLS.

**Вступ.** Сучасні різноманітні desktop-месенджери стали невід’ємним і важливим складником повсякденної комунікації майже всіх категорій користувачів (різних верств населення), забезпечуючи зручний і швидкий спосіб обміну інформацією (файловою інформацією, звуковими та відеофайлами, текстовими повідомленнями, фотографіями тощо).

Проте розробка ефективного та безпечного desktop-месенджера потребує вирішення цілої низки складних наукових і практичних проблем, пов’язаних з необхідністю, зокрема:

- збільшення функціональних можливостей месенджера;
- гарантування безпеки даних користувачів;
- підтримки актуальних і новітніх тенденцій у сфері комунікаційних технологій.

Саме тому інтеграція сучасних технологій у комунікаційний процес (процес швидкого та комфортного спілкування між різними користувачами месенджерів) сприяє зміні підходу до комунікацій, збільшенню обсягу інформації, що передається, зменшенню часу на отримання відповіді, надання користувачеві відомостей про отримання переданої ним інформації його «співрозмовником» (чи «співрозмовниками»), використання нових інформаційних технологій для реалізації відповідних комунікаційних послуг.

З появою дистанційного навчання можливості desktop-месенджерів почали використовувати частіше й більше, підвищуючи ефективність освітнього процесу, рівні комфортності швидкого отримання знань і компетенцій.

На сучасному етапі розвитку інформаційного суспільства роль desktop-месенджерів набуває все більшого значення в організації особистих і професійних зв’язків і комунікацій. Проте є деякі проблеми, що потребують ретельного вивчення та вирішення. Серед цих проблем слід виділити такі:

1. *Забезпечення безпеки даних.* Збільшення обсягів інформації, якою обмінуються користувачі через desktop-месенджери, обумовило важливість проблеми захисту особистих даних. Потенційні загрози, такі як злам облікових записів, перехоплення повідомлень і розповсюдження шкідливих програм, вимагають розробки ефективних заходів безпеки.

2. *Оптимізація продуктивності.* Під час розробки desktop-месенджера слід забезпечити його оптимальну продуктивність, зменшуючи споживання ресурсів комп'ютера, що дасть змогу використовувати його з високим рівнем комфортності та зручності на різних пристроях.

3. *Розширення функціональності.* Сучасні користувачі очікують від desktop-месенджерів надання широкого спектра можливостей. Вони вимагають розширення функціональних можливостей, таких як аудіо- та відеодзвінки, файловий обмін, інтеграцію з іншими сервісами та платформами.

Отже, проблеми розробки програмного забезпечення для якісного комфортного та зручного desktop-месенджера, у якому вирішуються вказані вище проблеми, є актуальними.

Актуальність розробки desktop-месенджера зумовлена тим, що швидкий і безпечний обмін повідомленнями має вирішальне значення для сучасного суспільства.

На сьогодні месенджери стали одним з найважливіших засобів спілкування, що сприяє віддаленій роботі й так званому «соціальному» спілкуванню.

Ураховуючи сучасну ситуацію в Україні, зумовлену пандемією та воєнним станом, а також загальну тенденцію до інтернет-спілкування, проблема розробки засобів спілкування вітчизняного виробництва є актуальною.

Аналіз досліджень і публікацій у сфері розробки та використання desktop-месенджерів виокремив проблеми, пов'язані з важливими аспектами, такими як безпека, продуктивність і розширення функціональності.

Основною метою статті є дослідження й аналіз різних desktop-месенджерів, виявлення наявних проблем і шляхів їх вирішення, застосування інноваційного підходу до реалізації desktop-месенджера, який поєднує в собі гарантування високого рівня безпеки та відмінної продуктивності з широким набором функціональних можливостей, що відповідають сучасним вимогам користувачів.

Досягнення мети передбачає виконання низки завдань, таких як:

- розробка desktop-месенджера, що інтегрує передові технології шифрування та безпеки, гарантуючи надійний захист особистих даних користувачів під час обміну повідомленнями;
- застосування інноваційного підходу до оптимізації продуктивності, що надає можливість забезпечити високу швидкодію та мінімізувати використання ресурсів комп'ютера;
- розширення функціональних можливостей месенджера, додаючи інноваційні інструменти поліпшення комунікації та зручного (комфортного) використання месенджера.

**Результати дослідження.** Desktop-месенджери є популярними інструментами для комунікації між користувачами на комп'ютерах. На сьогодні вони стали необхідними як для бізнесу, так і для особистого використання, бо забезпечують швидкий, зручний і безпечний обмін повідомленнями.

Під час розробки сучасних desktop-месенджерів основні зусилля спрямовують на різні аспекти їх подальшого функціонування, такі як безпека, продуктивність, інтерфейс користувача, функціональність і дизайн програмного продукту, візуалізація інформації, що передається.

*Безпека.* Проблеми безпеки в desktop-месенджерах є дуже важливими, бо вони містять конфіденційну інформацію користувачів. У процесі розробки desktop-месенджерів наголошують на їхньому захисті від хакерських атак, перехоплення повідомлень і зловмисних дій, зокрема таких, як фішинг.

Для гарантування безпеки даних під час комунікації користувачів (з відповідним передаванням різноманітної інформації) розробляють та використовують нові алгоритми шифрування та автентифікації.

Деякі проблеми з безпекою, наприклад, такі як неспроможність відбити зовнішні атаки, можуть виникнути через недостатньо надійні алгоритми шифрування чи некоректну автентифікацію користувачів. Тому для забезпечення високого рівня безпеки слід удосконалювати шифрувальні методи та застосовувати багатофакторну (зокрема двофакторну) автентифікацію.

*Продуктивність.* Сучасні desktop-месенджери мають бути ефективними під час використання ресурсів (інформаційних, апаратних, програмних тощо) і мінімізувати час відправлення та отримання повідомлень. Більш удосконалені алгоритми оптимізації та зменшення навантаження на процесор і пам'ять комп'ютера мають забезпечувати продуктивність.

*Інтерфейс користувача та функціональність.* Розробники сучасних desktop-месенджерів намагаються поліпшити інтерфейс користувача, зробивши його більш інтуїтивним і зручним, додаючи нові функції, такі як:

- підтримання голосових і відеовикликів;
- підтримання групових чатів;
- інтеграція із соціальними мережами й іншими застосунками.

*Дизайн та візуалізація.* Розробка сучасного дизайну desktop-месенджерів та візуалізація інформації, що передається в них, передбачає розробку відповідних методів і способів покращення як безпосередньо самого дизайну, так і використання анімацій та графічних ефектів для забезпечення більш привабливого інтерфейсу для користувачів месенджерів.

*Взаємодія з різними платформами* є одним з ключових аспектів, який слід урахувати під час розробки desktop-месенджерів. Ця взаємодія полягає в забезпеченні сумісності між desktop-месенджером і різними платформами, такими як Windows, macOS, Linux, а також мобільними пристроями.

Важливо, щоб месенджер працював без збоїв і надавав функціонал, який не залежав би від обраної платформи (Deep Knowledge Analytics, n.d.).

*Підтримка розширень та API* (Reddy, 2011). Розширення та API дають змогу збільшити функціональність desktop-месенджера, додавши додаткові можливості через сторонні розробки. До того ж слід забезпечити відкритий API та відповідну документацію для зручного створення розширень, що полегшить процес розробки й надасть досвід користувачам (Dzoghbenuki, n.d.).

Серед сучасних месенджерів є кілька лідерів, які відрізняються зручністю, функціональністю та безпекою. До них зараховують WhatsApp (<https://www.whatsapp.com/?lang=en>), Viber (<https://www.viber.com/en/>), Telegram (<https://telegram.org/?setln=en>) і Slack (<https://slack.com/>).

WhatsApp є одним з найпопулярніших месенджерів, який пропонує зручні можливості для текстового спілкування (наприклад, у вигляді текстових повідо-

млень), голосових і відеодзвінків (відеовикликів), обмін аудіо-, відео- та текстовими файлами.

Viber – популярний месенджер, який пропонує голосові та відеодзвінки, обмін текстовими повідомленнями, надає можливість міняти стікери та розширення.

Slack – корпоративний месенджер, який забезпечує ефективне спілкування в команді (за напрямками інтересів, керівними особами, проектами, що розробляються, тощо), надаючи можливість:

- створення каналів і приватних груп;
- інтегрування з іншими сервісами та розширеннями;
- пошуку відповідної інформації в повідомленнях.

Усі месенджери мають власний відповідний інтерфейс. Зокрема Viber та WhatsApp (рис. 1) мають схожий дизайн, де основні вкладки розташовані зверху екрана, а конверсії відображаються в окремих вікнах.

Telegram та Slack мають більш сучасний інтерфейс з боковою панеллю навігації та великим полем для введення тексту.

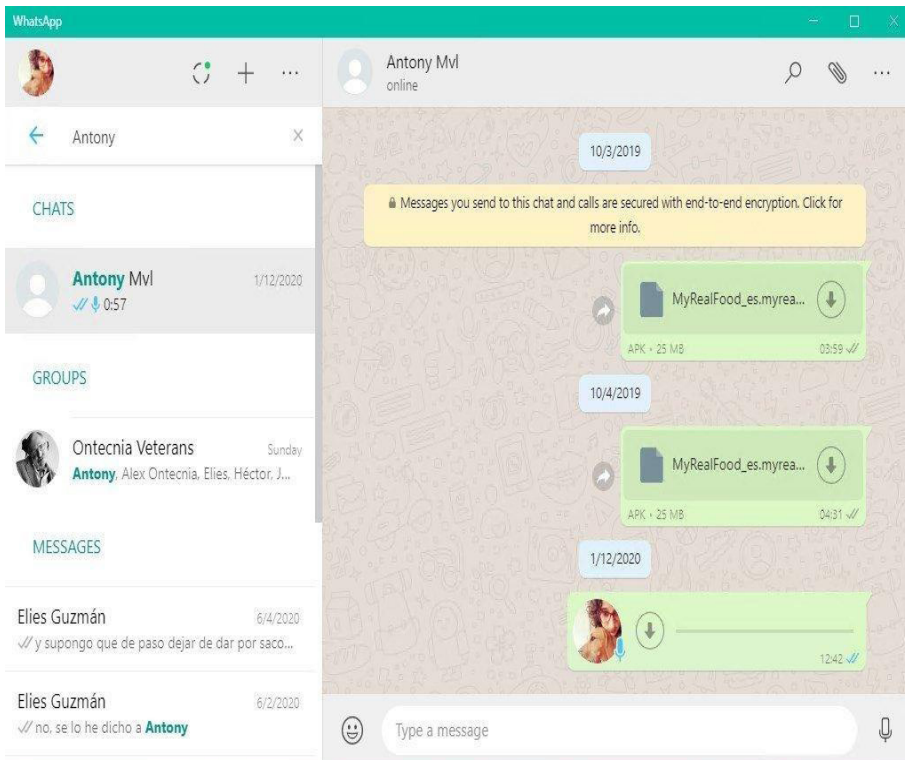


Рис. 1. Вигляд спілкування у WhatsApp

У Viber та WhatsApp, так само як і на мобільних пристроях, є клавіатура для введення тексту повідомлення, тоді як у Slack (рис. 2) і Telegram поля для введення тексту розміщені знизу екрана.

Slack також надає розширені можливості обміну файлами й інтеграції з іншими інструментами для роботи з файлами.

Усі месенджери підтримують передачу повідомлень у вигляді файлів, проте в Telegram є додаткові можливості для обміну великими файлами, включаючи можливість передавати файли обсягом до 2 Гб.

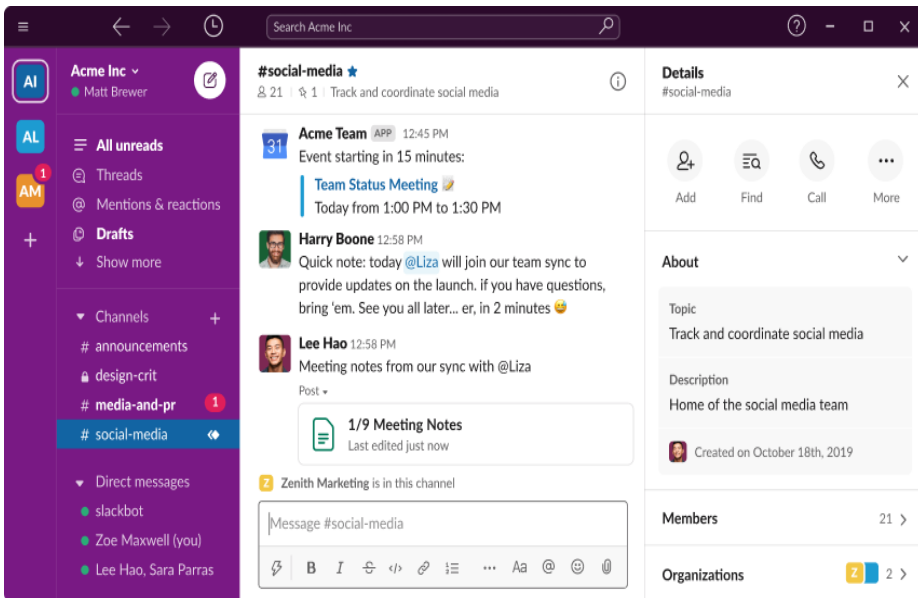


Рис. 2. Вигляд спілкування у Slack

Усі месенджери підтримують використання емодзі та наліпок. У всіх сучасних месенджерах є можливість створювати групові чати.

Однак у Telegram (рис. 3) можна створити набагато більші групи, ніж у WhatsApp та Viber.

Telegram та Slack мають розширені можливості щодо безпеки та захисту даних, зокрема, таких як шифрування повідомлень і двофакторна автентифікація. У Viber та WhatsApp також є можливості для захисту приватності користувача, але вони менш ефективні, ніж у Telegram та Slack.

У сучасних месенджерах часто використовується технологія handshake SSL/TLS (Khare and Lawrence, 2000) – спілкування між двома сторонами в мережі, такими як браузер і вебсервер, для встановлення параметрів, характеристик й умов такого спілкування. При цьому визначається:

- версія SSL/TLS, яка буде використовуватися в комунікаційному сеансі;
- набір шифрів для шифрування інформації, яка підтримує зв'язок та є повідомленнями між браузером і вебсервером;
- спосіб перевірки сервера (а іноді й клієнта);
- наявність умов гарантування безпечного з'єднання перед передаванням (чи обміном) даних.

На рис. 4 продемонстровано принцип роботи SSL/TLS.

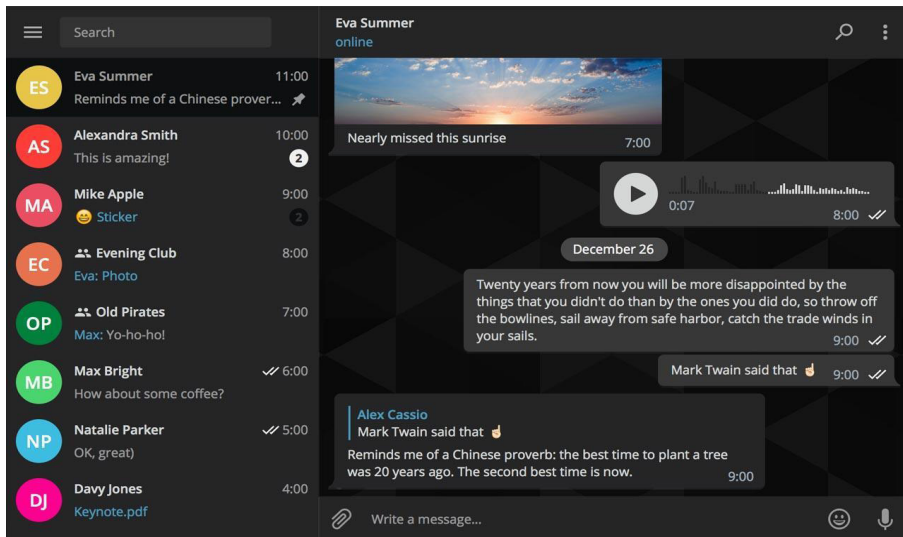


Рис. 3. Вигляд спілкування у Telegram



Рис. 4. Приклад роботи TLS Handshake (SSL/TLS in Detail, 2009)

Для досягнення поставленої мети використовуємо комплексний підхід, який базується на методах криптографії (І. Горбенко та Ю. Горбенко, 2012), алгоритмах оптимізації, інноваційних технологіях штучного інтелекту та машинного навчання.

Пропонований комплексний підхід ґрунтується на результатах аналізу наявних технологій і рішень, що застосовуються у сфері розробки та використання месенджерів, і передбачає нові розширення, гарантуючи покращений рівень безпеки, продуктивності та функціональних можливостей.

Розглянемо розробку авторського месенджера, починаючи з проектування інтерфейсу користувача, яке передбачає backend-розробку та frontend-розробку.

Розробляючи backend-частину месенджера на платформі C++ (C++ Programming Language, n.d.; Mahapatra, 2023) з використанням Qt (The future is written with Qt, n.d.; Qt Designer, n.d.), користувач стикається з цілою низкою проблем і характерних цьому процесу труднощів, наприклад:

1. *Багатопотоковість.* Месенджери зазвичай обслуговують багато користувачів одночасно, тому важливо забезпечити правильну багатопотоковість, яка в backend-частині часто реалізовується за допомогою відповідних шаблонів проектування (Saha and Praharaj, 2023), наприклад, таких як Immutable object (Amarasinghe, et al., 2015). Якщо потоки використовуються некоректно, то це може призвести до гонитви за ресурсами, збоїв і неефективного використання чи витрати ресурсів.

2. *Безпека.* Гарантування безпеки є важливою характеристикою будь-якого месенджера. У часи просування ІТ-продуктів у різні сфери діяльності людини та суспільства стає дуже важливим захист персональних даних користувачів під час комунікації за допомогою месенджера. Щоб зберегти персональні дані й убезпечити їх захист, будемо використовувати сучасні методи шифрування, перевіряти достовірність інформації, зокрема запитів щодо отримання інформації, та здійснювати безпосередній захист інформації, яка є об'єктом комунікацій між користувачами месенджера, від атак з боку потенційних кіберзловмисників.

3. *Робота з мережами.* У процесі свого функціонування сучасні месенджери зазвичай потребують (під час комунікаційного спілкування) обміну інформацією між їхніми користувачами (клієнтами) та сервером. Це обумовлює необхідність «правильної» реалізації мережевого коду, щоб забезпечити стійке й ефективне передавання як простих коротких повідомлень (Client-Server Architecture, n.d.), так і даних (зокрема, відео, аудіо, текстових файлів, фото тощо).

4. *Синхронізація даних.* Сучасний месенджер має забезпечувати можливість його використання на кількох пристроях одночасно (наприклад, здійснювати аудіо чи відеозв'язок на мобільному телефоні, а на комп'ютері обмінюватися повідомленнями, спілкуватися в чаті тощо). В авторському месенджері вирішуються проблеми синхронізації даних між різними пристроями, щоб уникнути конфліктів, зберігаючи узгодженість даних, якими обмінюються користувачі під час комунікаційних сеансів.

5. *Масштабованість.* Сучасний месенджер має постійно працювати над тим, щоб кількість його користувачів збільшувалася (Sherman and Schiela, 2018), тому backend-частина авторського месенджера повинна бути готовою до масштабування для забезпечення комунікацій показників кількості користувачів, що зростають, і обсягів інформації, що передається під час спілкування за допомогою месенджера.



6. *Управління базою даних.* Сучасний месенджер має підтримувати збереження повідомлень, інформації про користувачів та інші дані. Усе це неможливо без ефективного управління відповідною базою даних.

7. *Тестування.* Backend-частина будь-якого (і авторського в тому числі) месенджера після розробки проходить ретельне тестування для виявлення та виправлення можливих помилок, а також перевірки правильності роботи всіх функцій і можливих сценаріїв використання месенджера в цілому та його окремих компонентів зокрема.

Для зберігання даних у базі даних авторського месенджера будемо використовувати QSqlDatabase Class (n.d), де інформація зберігається в зашифрованому вигляді. Для кожного підключеного користувача виділяється окремий тред (Mahapatra, 2023) та сокет, які чекають на нові повідомлення та обробляють їх.

Щоб повідомлення користувачів не губилися, в авторському месенджері реалізована система handshake (Davies, 2019), яка дає змогу бути впевненим, що дані, які розбиті на декілька пакетів, прийшли повністю.

Frontend-розробка авторського месенджера передбачає, зокрема, використання та забезпечення таких аспектів, як:

1. Використання QML та C++. Однією з ключових особливостей розробки frontend-частини месенджера на C++ з Qt (The future is written with Qt, n.d.) є використання QML (Qt Meta-Object Language) (The Meta-Object System, n.d.) разом з C++. QML – це декларативна мова створення користувацьких інтерфейсів, яка дає змогу створювати динамічні й анімовані ефекти. C++ використовується для бізнес-логіки та взаємодії з backend-частиною месенджера. Тому, розробляючи авторський месенджер, використано як QML, так і C++, ефективно поєднуючи їх для досягнення поставлених цілей розробки.

2. *Асинхронна взаємодія.* Для обміну повідомленнями сучасні месенджери часто використовують асинхронну взаємодію. Тому під час розробки авторського месенджера слід було ефективно використовувати механізми асинхронності в Qt (наприклад, сигнали та слоти), щоб забезпечити плавний та ефективний обмін даними між frontend- та backend-частинами.

3. *Мультимедіа та графіка.* Сучасні месенджери часто використовуються для передачі мультимедійних даних, таких як зображення, відео, аудіофайл тощо. Тому під час розробки авторського месенджера багато уваги було приділено забезпеченню ефективної обробки таких даних (мультимедіа та графіки) й оптимізації їх візуалізації (відображення). Використання графічних бібліотек Qt сприяло створенню привабливого та дружнього (до користувача) інтерфейсу для обробки мультимедіа.

4. *Підтримка різних платформ.* Розробка frontend-частини месенджера на C++ разом з Qt часто потребує підтримки різних платформ, таких як Windows, macOS, Linux чи платформ, реалізованих на мобільних пристроях. Тому під час розробки авторського месенджера забезпечено кросплатформність, щоб користувач міг працювати на різних платформах, маючи адаптивний дизайн.

5. *Тестування інтерфейсу.* Тестування frontend-частини месенджера є важливим етапом розробки. Під час розробки авторського месенджера використовувалося тестування на одиницях (unit testing) (Jorgensen, 2008) та інші методи тестування для забезпечення надійності й стабільності розробленого програмного продукту.

Дизайн месенджера розроблено в Qt Designer (n.d.) з використанням віджетів (табл. 1). Початковий варіант дизайну показано на рис. 5.

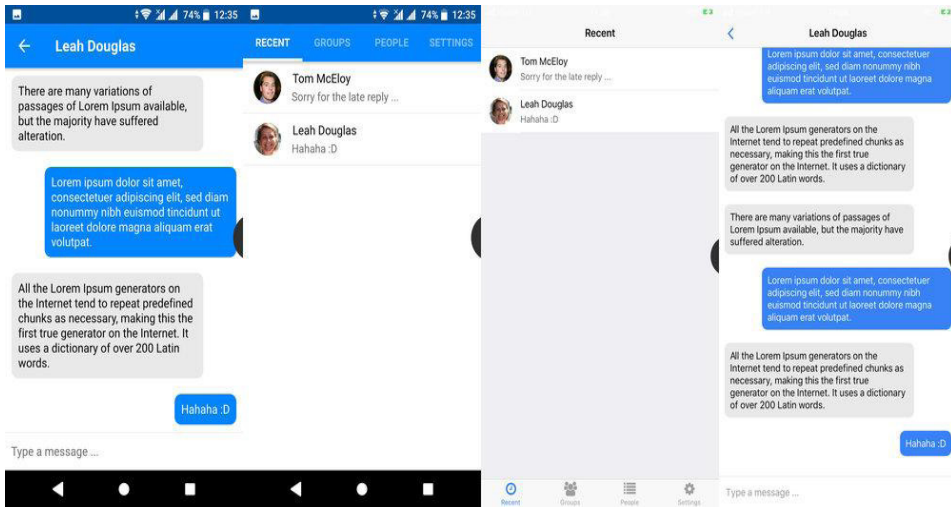


Рис. 5. Початковий варіант дизайну месенджера

Таблиця 1

### Віджети та їхнє призначення

Віджет	Призначення
QComboBox	Комбінована кнопка та вигульковий список
QDateEdit	Віджет для редагування дат на основі віджета QDateTimeEdit
QDateTimeEdit	Віджет для редагування дат і часу
QFocusFrame	Фокусна рамка, яка може бути за межами звичайної зони малювання віджета
QLabel	Відображення тексту або зображення
QLineEdit	Однорядковий текстовий редактор
QMenu	Віджет «меню» для використання в панелях меню, контекстних меню та інших вигулькових меню
QProgressBar	Горизонтальна або вертикальна панель прогресу
QScrollBar	Вертикальна або горизонтальна смуга прокрутки
QSlider	Вертикальний або горизонтальний слайдер
QTabBar	Панель вкладок, наприклад для використання в діалогах із вкладками
QTabWidget	Стосується віджетів із вкладками
QTimeEdit	Віджет для редагування часу на основі віджета QDateTimeEdit
QToolBox	Стовпець елементів віджета з вкладками
QToolButton	Кнопка швидкого доступу до команд або параметрів, зазвичай використовується в QToolBar
QWidget	Базовий клас усіх об'єктів інтерфейсу користувача

*Безпека та конфіденційність.* Особливості реалізації безпеки та конфіденційності в месенджерах, реалізованих на платформі C++ з використанням Qt є надзвичайно важливими, оскільки месенджери зберігають та передають особисті повідомлення користувачів, які містять «чутливу» для останніх інформацію.

Розглянемо деякі проблеми та шляхи їх вирішення при розробці авторського месенджера, які були пов'язані із гарантуванням безпеки та конфіденційності даних:

1. *Шифрування даних.* Вирішення цієї проблеми здійснено через використання протоколів шифрування, таких як TLS/SSL (Кузнецов та Семенов, 2009), що надало можливість забезпечити захищене з'єднання між клієнтом і сервером, уникнути перехоплення та читання повідомлень з боку кіберзловмисників.

2. *Безпека автентифікації.* В авторському месенджері забезпечено надійну автентифікацію користувачів, щоб переконатися в їх ідентичності перед наданням доступу до особистої інформації та повідомлень. Безпека автентифікації досягається, зокрема, через використання безпечних методів автентифікації, таких як хешування паролів і використання токенів.

3. *Захист від SQL Injection та Cross-Site Scripting* (Sharma, 2022). Використання параметризованих запитів до бази даних і валідація введених даних сприяє уникненню потенційних загроз (атак SQL Injection та Cross-Site Scripting).

4. *Захист даних у пам'яті.* Під час обробки та зберігання даних у пам'яті слід дотримуватися найкращих практик безпеки, таких як очищення та надійне видалення чутливих даних з оперативної пам'яті після використання.

5. *Аудит подій.* В авторському месенджері ведеться аудит усіх подій, що відбуваються в системі, для виявлення можливих небезпечних інцидентів та реагування на них з метою їх усунення чи зменшення ризиків і загроз, що виникають, зокрема допомагає виявляти неправомірні дії чи спроби злому.

6. *Захист від DoS атак.* Для вирішення цієї проблеми в авторському месенджері використовуються, зокрема, обмеження на кількість запитів від одного користувача, контроль швидкості.

Тестування та налагодження месенджера передбачає, зокрема:

1. Функціональне тестування, під час якого перевіряється чи працюють основні функції месенджера згідно з очікуваною поведінкою. Це може охоплювати перевірку відправлення та отримання повідомлень, створення та керування контактами, управління групами тощо (ДСТУ ISO/IEC 25012:2016, 2016).

2. Тестування на помилки допомагає виявити та виправити дефекти в програмному коді, перевіряючи різні можливі сценарії використання програми, щоб переконатися, що месенджер правильно обробляє нештатні (непередбачувані, форс-мажорні) ситуації.

3. Тестування з низьким навантаженням сприяє перевірці базового функціоналу месенджера та дає змогу переконатися, що програмний продукт працює стабільно при невеликій кількості користувачів. А тестування з високим навантаженням виконується для перевірки продуктивності та масштабованості месенджера за високого показника кількості користувачів і повідомлень.

4. Тестування безпеки виявляє потенційні загрози месенджеру, такі як уразливість автентифікації, можливість атак з боку кіберзловмисників тощо.

5. Тестування дизайну сприяє впевненості розробників у тому, що інтерфейс коректно відображається та працює на різних пристроях, забезпечуючи користувачу високий рівень зручності й комфортності спілкування (комунікацій).

**Висновки.** У роботі було проаналізовано переваги та недоліки наявних desktop-месенджерів. Проведений аналіз обумовив вибір концепції розробки сучасного авторського месенджера на основі використання C++ та Qt і розробленого на її основі прототипу авторського месенджера.

Можливості C++ (низькорівневий доступ до пам'яті, висока продуктивність і широкі можливості оптимізації) сприяли забезпеченню оптимальної та ефективної роботи месенджера. Використання Qt та його можливостей під час розробки дало змогу швидко створювати зручний і привабливий користувацький інтерфейс.

Розробка месенджера з використанням C++ і Qt дала змогу вирішувати проблеми безпеки та приватності комунікацій (зокрема, захищаючи особисту інформацію, що поширюється в месенджері). Вбудована підтримка шифрування, захист даних користувачів і управління правами доступу легко реалізуються завдяки можливостям C++ та Qt.

Запропонований підхід до розробки месенджера забезпечує одне з найкращих рішень, зменшуючи розподіл між frontend- і backend-частинами та полегшуючи гнучкість розробки. Крім того, цей підхід гарантує безпеку й оптимізацію продуктивності desktop-месенджерів та підвищення рівня їх функціональності й зручності використання.

## СПИСОК ПОСИЛАНЬ

---

Горбенко, І. Д. та Горбенко, Ю. І., 2012. *Криптологія. Теорія. Практика. Застосування*. Харків: Форт.

ДСТУ ISO/IEC 25012:2016, 2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Модель якості даних (ISO/IEC 25012:2008, IDT). *БУДСТАНДАРТ Online* [online] 27 грудня. Доступно: <[https://online.budstandart.com/ua/catalog/doc-page?id\\_doc=69135](https://online.budstandart.com/ua/catalog/doc-page?id_doc=69135)> [Дата звернення 01 серпня 2023].

Кузнецов, О.О. та Семенов, С.Г., 2009. *Протоколи захисту інформації у комп'ютерних системах та мережах*. Харків: Харківський національний університет радіоелектроніки.

Amarasinghe, S., Chlipala, A., Devadas, S., Ernst, M., Goldman, M., Guttag, J., Jackson, D., Miller, R., Rinard, M. and Solar-Lezama, A., 2015. Reading 9: Mutability & Immutability. In: *6.005: Software Construction*, [online] pp.32-123 Available at: <<https://web.mit.edu/6.005/www/fa16/classes/09-immutability/>> [Accessed 01 August 2023].

C++ Programming Language, n.d. [online] Available at: <<https://devdocs.io/cpp/>> [Accessed 01 August 2023].

Client-Server Architecture, n.d. [online] Available at: <[https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch\\_Design\\_Activity/ClientServer.pdf](https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/ClientServer.pdf)> [Accessed 01 August 2023].

- Davies, J., 2019. A walkthrough of a TLS 1.3 handshake. *Command Line Fanatic*, [online] 25 June. Available at: <<https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art080>> [Accessed 01 August 2023].
- Deep Knowledge Analytics, n.d. Messaging Apps & Communication Platforms Comparative Analysis. *Methodology*. [online] Available at: <<https://analytics.dkv.global/messengers-methodology.pdf>> [Accessed 01 August 2023].
- Dzogbenuku, S.A., n.d. Strategy, Analytics and M&A. *Deloitte*. [online] Available at: <[https://www.deloitte.com/gh/en/services/consulting/services/strategy-analytics-manda.html?icid=top\\_strategy-analytics-manda](https://www.deloitte.com/gh/en/services/consulting/services/strategy-analytics-manda.html?icid=top_strategy-analytics-manda)> [Accessed 01 August 2023].
- Jorgensen, P.C., 2008. *Software Testing: A Craftsman's Approach*. 3rd ed. [e-Book] New York: Auerbach Publications. <https://doi.org/10.1201/9781439889503>.
- Khare, R. and Lawrence, S., 2000. Upgrading to TLS Within HTTP/1.1. *RFC Editor*, [e-journal] *RFC 2817*. <https://doi.org/10.17487/RFC2817>
- Mahapatra, S., 2023. Multithreading in C++. *GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/multithreading-in-cpp/>> [Accessed 01 August 2023].
- QSqlDatabase Class, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/qt-6/qsqldatabase.html>> [Accessed 01 August 2023].
- Qt Designer, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/qt-5/qt designer-index.html>> [Accessed 01 August 2023].
- Reddy, M., 2011. API Design for C++. [e-Book] *Elsevier*. Available at: <[https://books.google.com.ua/books?id=IY29LYt85wC&redir\\_esc=y](https://books.google.com.ua/books?id=IY29LYt85wC&redir_esc=y)> [Accessed 01 August 2023].
- Saha, A. and Praharaj, T., 2023. Introduction to Pattern Designing. *GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/design-patterns-set-1-introduction/>> [Accessed 01 August 2023].
- Sharma, K., 2022. SQL injection and cross-site scripting: The differences and attack anatomy. *ManageEngine Log360*, [online] 15 November. Available at: <<https://www.manageengine.com/log-management/cyber-security/sql-injection-web-application-attack-and-cross-site-scripting-the-differences-and-attack-anatomy.html>> [Accessed 01 August 2023].
- Sherman, M. and Schiela, R., 2018. Building Secure Software for Mission Critical Systems. *Carnegie Mellon University*. [online] Available at: <<https://apps.dtic.mil/sti/trecms/pdf/AD1087868.pdf>> [Accessed 01 August 2023].
- SSL/TLS in Detail, 2009. *Microsoft*. [online] Available at: <[https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785811\(v=ws.10\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785811(v=ws.10))> [Accessed 01 August 2023].
- The future is written with Qt, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/>> [Accessed 01 August 2023].
- The Meta-Object System, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/qt-6/metaobjects.html>> [Accessed 05 August 2023].

---

## REFERENCES

---

- Amarasinghe, S., Chlipala, A., Devadas, S., Ernst, M., Goldman, M., Guttag, J., Jackson, D., Miller, R., Rinard, M. and Solar-Lezama, A., 2015. Reading 9: Mutability & Immutability. In: *6.005: Software Construction*, [online] pp.32-123. Available at: <<https://web.mit.edu/6.005/www/fa16/classes/09-immutability/>> [Accessed 01 August 2023].

- C++ Programming Language, n.d. [online] Available at: <<https://devdocs.io/cpp/>> [Accessed 01 August 2023].
- Client-Server Architecture, n.d. [online] Available at: <[https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch\\_Design\\_Activity/ClientServer.pdf](https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/ClientServer.pdf)> [Accessed 01 August 2023].
- Davies, J., 2019. A walkthrough of a TLS 1.3 handshake. *Command Line Fanatic*, [online] 25 June. Available at: <<https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art080>> [Accessed 01 August 2023].
- Deep Knowledge Analytics, n.d. *Messaging Apps & Communication Platforms Comparative Analysis. Methodology*. [online] Available at: <<https://analytics.dkv.global/messengers-methodology.pdf>> [Accessed 01 August 2023].
- DSTU ISO/IEC 25012:2016 [DSTU ISO/IEC 25012:2016], 2016. Inzheneriia system i prohramnykh zasobiv. Vymohy do yakosti system i prohramnykh zasobiv ta yii otsiniuvannia (SQuaRE). Model yakosti danykh (ISO/IEC 25012:2008, IDT) [Systems and software engineering. Requirements for the quality of systems and software tools and its evaluation (SQuaRE). Data quality model (ISO/IEC 25012:2008, IDT)]. *BUDSTANDART Online*, [online] 27 December. Available at: <[https://online.budstandart.com/ua/catalog/doc-page?id\\_doc=69135](https://online.budstandart.com/ua/catalog/doc-page?id_doc=69135)> [Accessed 01 August 2023].
- Dzogbenuku, S.A., n.d. Strategy, Analytics and M&A. *Deloitte*. [online] Available at: <[https://www.deloitte.com/gh/en/services/consulting/services/strategy-analytics-manda.html?icid=top\\_strategy-analytics-manda](https://www.deloitte.com/gh/en/services/consulting/services/strategy-analytics-manda.html?icid=top_strategy-analytics-manda)> [Accessed 01 August 2023].
- Horbenko, I. D. and Horbenko, Yu. I., 2012. *Kryptolohiia. Teoriia. Praktyka. Zastosuvannia* [Cryptology. Theory. Practice. Application]. Kharkiv: Fort.
- Jorgensen, P.C., 2008. *Software Testing: A Craftsman's Approach*. 3rd ed. [e-Book] New York: Auerbach Publications. <https://doi.org/10.1201/9781439889503>
- Khare, R. and Lawrence, S., 2000. Upgrading to TLS Within HTTP/1.1. *RFC Editor*, [e-journal] RFC 2817. <https://doi.org/10.17487/RFC2817>
- Kuznetsov, O.O. and Semenov, S.H., 2009. *Protokoly zakhystu informatsii u kompiuternykh systemakh ta merezhakh* [Information protection protocols in computer systems and networks]. Kharkiv: Kharkiv National University of Radio Electronics.
- Mahapatra, S., 2023. Multithreading in C++. *GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/multithreading-in-cpp/>> [Accessed 01 August 2023].
- QSqlDatabase Class, n.d. *The Qt Company*. [online] Available at: <https://doc.qt.io/qt-6/qsqldatabase.html> [Accessed 01 August 2023].
- Qt Designer, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/qt-5/qtdesigner-index.html>> [Accessed 01 August 2023].
- Reddy, M., 2011. API Design for C++. [e-Book] *Elsevier*. Available at: <[https://books.google.com.ua/books?id=IY29LyIT85wC&redir\\_esc=y](https://books.google.com.ua/books?id=IY29LyIT85wC&redir_esc=y)> [Accessed 01 August 2023].
- Saha, A. and Praharaj, T., 2023. Introduction to Pattern Designing. *GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/design-patterns-set-1-introduction/>> [Accessed 01 August 2023].
- Sharma, K., 2022. SQL injection and cross-site scripting: The differences and attack anatomy. *ManageEngine Log360*, [online] 15 November. Available at: <<https://www.manageengine.com/log-management/cyber-security/sql-injection-web-application-attack-and-cross-site-scripting-the-differences-and-attack-anatomy.html>> [Accessed 01 August 2023].
- Sherman, M. and Schiela, R., 2018. Building Secure Software for Mission Critical Systems. *Carnegie Mellon University*. [online] Available at: <<https://apps.dtic.mil/sti/trecms/pdf/AD1087868.pdf>> [Accessed 01 August 2023].

SSL/TLS in Detail, 2009. *Microsoft*. [online] Available at: <[https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785811\(v=ws.10\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc785811(v=ws.10))> [Accessed 01 August 2023].

The future is written with Qt, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io>> [Accessed 01 August 2023].

The Meta-Object System, n.d. *The Qt Company*. [online] Available at: <<https://doc.qt.io/qt-6/metaobjects.html>> [Accessed 05 August 2023].

## UDC 004:316.444]:316.77

### **Oleksandr Tkachenko,**

*PhD in Physics and Mathematics, Associate Professor,  
Associate Professor at the Department of Information Technologies,  
State University of Infrastructure  
and Technologies,  
Kyiv, Ukraine  
aatokg@gmail.com  
<https://orcid.org/0000-0001-6911-2770>*

### **Denys Honcharenko,**

*Master's student, Department of Information Technologies,  
State University of Infrastructure  
and Technologies,  
Kyiv, Ukraine  
rogalikck@gmail.com  
<https://orcid.org/0009-0002-3648-5882>*

## SOME ASPECTS OF THE MESSENGERS DEVELOPMENT – MEANS OF SOCIAL COMMUNICATION

**The purpose of the article** is to study and analyse various desktop messengers, identify existing problems and ways to solve them, and apply innovative approaches to the implementation of a desktop messenger that combines a high level of security and excellent performance with a wide range of functionalities that meet modern user requirements.

Desktop messengers have become an important component of everyday communication for many categories of users, providing a convenient and fast way to exchange various information. The article examines, in particular, how the integration of modern technologies into the communication process contributes to a change in the approach to communications, increasing the amount of information transmitted, reducing the time to receive a response, and providing the user with information about the receipt of the transmitted information by his or her “interlocutor” (or “interlocutors”).

**The research methods** are the main methodological approaches and technological tools for creating high-performance and user-friendly messengers.

**The scientific novelty of the study** is the analysis of modern technologies for developing platforms for supporting the online communication and communication process, the results of which can be used in the development of an own software product – a messenger.

The article substantiates the relevance of developing a desktop messenger, which is due to the fact that fast and secure messaging is of great importance today since modern messengers have become one of the most important means of communication, facilitating remote work and so-called “social” communication.

**Conclusions.** The paper investigates the known views on the modern approach to the development of desktop messengers. The analysis has led to the choice of the concept of developing a modern author’s messenger based on the use of C++ and Qt and, as a result, the prototype of the author’s messenger developed on its basis.

The development of the messenger using C++ and Qt helps to solve the problems of security and privacy of communications (in particular, by protecting personal information distributed in the messenger). Encryption support, data protection, and access rights management are implemented using the capabilities of C++ and Qt.

The proposed approach to messenger development provides one of the best solutions, reducing the separation between front- and back-end parts and facilitating development flexibility. This makes it possible to create scalable and high-performance messengers. In addition, this approach guarantees the security and performance optimization of desktop messengers and improves their functionality and usability.

**Keywords:** communications; communication; user interface; desktop messenger; data security; frontend; backend; SSL/TLS.

28.08.2023