МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему

«Розробка функціональних та нефункціональних вимог до системи обміну інформацією з використанням квантового розподілу ключів»

Студентка групи КБ – 81 Панченко Ю.С.

Завідувач випускаючої кафедри Любчак В. О.

Керівник роботи Котух. Е. В.

СУМИ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

Затверджую_____

Зав. кафедрою Любчак В. О.

“_____” _____2022 р.

ЗАВДАННЯ

до випускної роботи

Студентки четвертого курсу, групи КБ-81 спеціальності “Кібербезпека” денної форми навчання Панченко Юлії Сергіївни.

Тема: “ Розробка функціональних та нефункціональних вимог до системи обміну інформацією з використанням квантового розподілу ключів”

Затверджена наказом по СумДУ

№_____ від _____2022 р.

Зміст пояснювальної записки: 1) теоретичні відомості 2) аналізування; 3) створення функціональних та нефункціональних вимог 4) концепт мессенджеру.

Дата видачі завдання “_____”_____ 2022 р.

Керівник випускної роботи _____ Котух Є.В.

Завдання прийняла до виконання _____ Панченко Ю. С.

РЕФЕРАТ

Записка: 46 стор., 20 Рис., 1 табл., 40 джерела.

Мета роботи —виявлення та структурування всіх вимог до системи обміну інформацією з використанням квантового розподілу ключів

Об'єкт дослідження — область знань бізнес аналізу, вимог до ПЗ

Предмет дослідження — система та алгоритм обміну інформацією з використанням квантового розподілу ключів

Результати — виявлення функціональних та нефункціональних вимог предмета дослідження

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення.

ЖЦ - Життєвий цикл.

FR – функціональні вимоги.

NFR – нефункціональні вимоги.

SRS - Software requirements specification.

PM - project manager.

UI - User Interface

UX - user experience

ВСТУП

Визначення вимог — завдання нетривіальне. Це робиться шляхом обговорення з розробниками того, що думає замовник про майбутню систему. Замовник висловлює свої потреби та погляди на завдання системи, надалі виражені у вигляді вимог до програмного забезпечення, які класифікуються таким чином.Розробка вимог включає наступні етапи:

• Визначити цільову аудиторію користувачів цього продукту;

• Роз'яснити вимоги кожного ЦА.

• Визначення місії та цілей ЦА та БЦ (бізнес-цілі).

• Аналіз отриманої від користувачів інформації, щоб відокремити завдання від функціональних і нефункціональних вимог, очікуваних рішень і зовнішніх даних;

• Визначити відносну важливість атрибутів якості;

• Визначити пріоритети для реалізації;

• Документувати зібрану інформацію та будувати моделі;

• Перевірка специфікації вимог, щоб переконатися, що запити користувачів чітко зрозумілі.

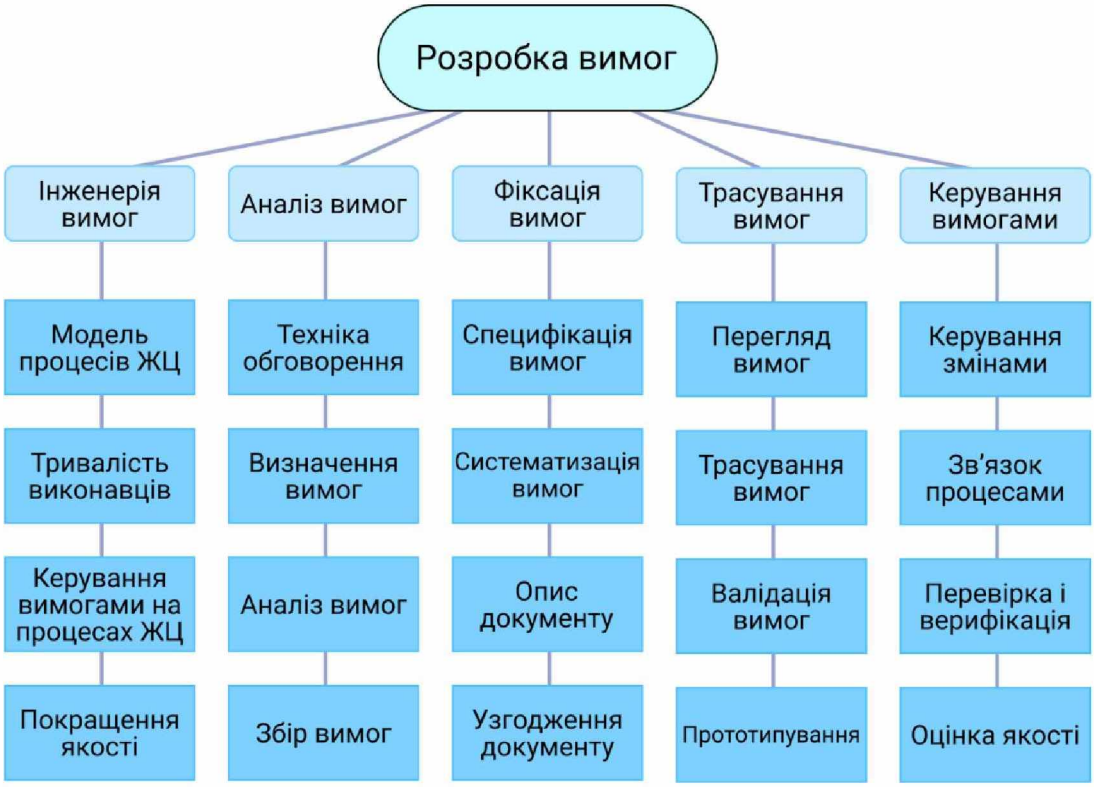


Рисунок 1.1. - Розробка вимог

Витрачання часу на визначення вимог є фактором успіху будь-якого IT-проекту. Але отримати правильну та коректну інформацію не завжди легко і це також вимагає спеціальних навичок. Отже, розробка вимог є окремою галуззю в IT.

В рамках роботи буде запропоновано загальний опис підходу до класифікації функціональних та нефункціональних вимог, та розробка технічної специфікації вимоги щодо системи передачі інформації з допомогою квантового розподілу ключів.

Розділ 1. ВИМОГИ ДО РОЗРОБКИ ПЗ

1.1 Вимоги до програмного забезпечення

Це набір властивостей, якості та функцій програмного забезпечення, які буде розроблено, або знаходиться у розробці.

Вони визначаються в процесі аналізу та подалі фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки.

Життєвий цикл ПЗ (ЖЦ) - неперервний процес з моменту прийняття рішення про створення ПЗ до вилучення його з експлуатації.

Основні процеси та етапи ЖЦ ПЗ: [1]

Проектування програми – це опис того, як програма буде працювати.

Кодування – це вихідний код і файли його конфігурації.

Тестування – це контроль відповідності програми вимогам.

Документування - документація до програми.

Супровід.

Мета розробки ПЗ – у виділений час і бюджет, розробити якісне ПЗ, яке буде задовольняти потреби користувача.

Специфікація вимог до програмного забезпечення описує функціональні можливості, які продукт повинен мати, щоб відповідати очікуванням майбутніх користувачів.

Цей документ повинен містити:

- Загальний опис
- Призначення продукту
- Специфічні вимоги до програмного забезпечення

Специфікації вимог до програмного забезпечення описують програмне забезпечення, яке буде розроблено, документ специфікації системних вимог збирає інформацію про системні вимоги.

Вимоги до програмного забезпечення є описом можливостей та функціональних можливостей цільової системи. Вимоги передають

очікування користувачів від програмного продукту. Вимоги можуть бути очевидними чи прихованими, відомими чи невідомими, очікуваними чи неочікуваними з точки зору клієнта.

В загальному випадку робота над спеціалізацією вимог починається з техніко-економічного обґрунтування.

1.2 Техніко-економічне обґрунтування

Коли клієнт звертається до організації для розробки бажаного продукту, він надає приблизне уявлення про те, які всі функції має виконувати програмне забезпечення і які всі функції очікуються від програмного забезпечення. Посилаючись на цю інформацію, аналітики детально вивчають, чи можна розробити бажану систему та її функціональні можливості.

Це техніко-економічне обґрунтування спрямоване на досягнення мети організації. У цьому дослідженні аналізується, чи можна практично матеріалізувати програмний продукт з точки зору реалізації, внеску проекту в організацію, обмежень витрат, а також відповідно до цінностей і цілей організації. Він досліджує технічні аспекти проекту та продукту, такі як зручність використання, ремонтопридатність, продуктивність та здатність до інтеграції.

Результатом цієї фази має бути звіт з техніко-економічного обґрунтування, який має містити відповідні коментарі та рекомендації для керівництва щодо обсягу, функціоналу та можливості досягнення бізнес цілей компанії за допомогою реалізації даного проекту.

За своєю суттю техніко-економічне обгрунтування лягає в основу наступного етапу розробки вимог, а саме збору вимог.

1.3 Збір вимог

Аналітики та інженери спілкуються з клієнтом і кінцевими користувачами, щоб знати їхні ідеї щодо того, що програмне забезпечення має

забезпечувати та які функції вони хочуть включати в програмне забезпечення. Результатами такого спілкування є створення специфікації вимог до ПЗ.

Специфікація вимог до програмного забезпечення

SRS (Software requirements specification) — це документ, створений системним аналітиком після збору вимог від різних зацікавлених сторін.

SRS визначає, як призначене програмне забезпечення буде взаємодіяти з обладнанням, зовнішніми інтерфейсами, швидкість роботи, час відгуку системи, переносимість програмного забезпечення на різні платформи, ремонтопридатність, швидкість відновлення після збою, безпека, якість, обмеження тощо.

Вимоги, отримані від замовника, написані природною мовою. Обов'язок системного аналітика задокументувати вимоги технічною мовою, щоб вони могли бути зрозумілі та корисні команді розробників програмного забезпечення.

SRS повинен мати такі функції:

- Вимоги користувача виражені природною мовою.
- Технічні вимоги виражаються структурованою мовою, яка використовується всередині організації.
- Опис дизайну має бути написаний псевдокодом.
- Формат форм і екранних відбитків графічного інтерфейсу.
- Умовні та математичні позначення для DFD(data flow diagram) тощо.

Вимоги можна перевірити за такими умовами: Якщо їх можна практично реалізувати, вони дійсні та відповідають функціональності та домену програмного забезпечення, вони повні та їх можна продемонструвати.

1.4 Процес визначення вимог:

- **Збір** - розробники обговорюють з клієнтом і кінцевими користувачами і знають їхні очікування від програмного забезпечення.

- **Вимоги до організації.** Розробники встановлюють пріоритети та впорядковують їх в порядку важливості, терміновості та зручності.
- **Переговори та обговорення** – якщо вимоги неоднозначні або є певні конфлікти в них різних зацікавлених сторін, то це обговорюється із зацікавленими сторонами. Тоді вимоги можуть бути визначені пріоритетами та обґрунтовано скомпрометовані.

Вимоги надходять від різних зацікавлених сторін. Щоб усунути неоднозначність і конфлікти, їх обговорюють для ясності та коректності.

Документація. Усі формальні та неформальні, функціональні та нефункціональні вимоги задокументовані та доступні для обробки на наступному етапі.

Саме специфікація вимог до ПЗ використовується при розробці та гарантії якості продукту. Також при управлінні проектом і зв'язаних з проектом функціях. Крім функціональних вимог, в специфікації містяться нефунціональні вимоги, які описують цілі і атрибути якості.

1.5 Функціональні вимоги

Вони описують, що конкретно потрібно реалізувати в тій чи іншій системі або продукті, які дії повинні виробляти користувачі відносно даної розробки.

Вони також задокументовані в Специфікації вимог до програмного забезпечення, що описує поведінку s-mi. Це може бути будь-яке сховище даних в інструменті управління вимогами бізнесу.

З усіх видів вимог функціональні вимоги є найбільш привабливими для кінцевих користувачів. Вони описують конкретні функції, які повинна виконувати система.

Типи функціональних вимог

- Сумісність – ця вимога описує, чи сумісна програмна система з іншими різними системами.
- Безпека – саме ця вимога описує аспекти безпеки вимог до ПЗ.
- Точність- ця вимога визначає, чи дані, які вводяться в систему, чи правильно обчислюються та використовуються системою.
- Відповідність – це саме функціональні вимоги щодо відповідності, які підтверджують відповідність розробленої с-ми промисловим стандартам

1.6 Нефункціональні вимоги

Вони часто випливають з функціональних вимог, так само, як вони ґрунтуються на внесенні клієнтів та інших зацікавлених сторін. Деталі та процес реалізації нефункціональних вимог задокументовані в архітектурі файлової системи.

Ці вимоги пояснюють якісні аспекти створюваної системи, а саме: продуктивність, портативність, простоту використання тощо. Вони на відміну від функціональних, впроваджуються поступово в будь-якій системі. [6]

Підтипи (невичерпні) нефункціональних вимог:

- Продуктивність – це тип атрибута продуктивності нефункціональної вимоги, який вимірює продуктивність системи.
- Юзабіліті - вимірює (зручність використання) програмної системи, що розробляється.
- Технічне обслуговування – це ремонтопридатність програмної системи, та легкість її підтримки. Якщо середній час між відмовами низький або високий для системи, що розробляється, то ремонтопридатність вважається низькою.

- Надійність – це один із аспект доступності. Цей атрибут якості підкреслює доступність системи за певних (визначених) умов.
- Переносимість – ця вимога означає здатність програмної с-ми працювати в іншому середовищі.
- Підтримка – це справність програмної системи та здатність сервісного/технічного експерта встановлювати програмну систему у реальному часі, контролювати її та виявляти технічні проблеми в с-мі та надавати рішення для вирішення проблеми.
- Адаптованість – ця вимога визначається як здатність програмної с-ми пристосовуватися до змін у середовищі без змін у її поведінці.

Таблиця 1. - відмінності функціональних та нефункціональних вимог [2].

	Функціональні вимоги (FR functional requirements)	Нефункціональні вимоги (NFR non-functional requirement)
1	Функціональні вимоги формують фундамент впровадження програмної с-ми	Нефункціональні вимоги доповнюють с-ми SW.
2	Ці вимоги описують, що повинна робити система.	Ці вимоги описують, якою має бути система.
3	FR детально описані в документі Проектування системи.	NFR детально описані в документі архітектури системи.
4	Говорять про поведінку функції.	Говорять про робочу поведінку цілої системи або компонента системи, а не про певну функцію.

5	У веб-програмі користувач повинен мати можливість входу за допомогою аутентифікації FR	У веб-програмі, скільки часу потрібно для входу на веб-сайт, вигляду та відчуття сторінки входу, простоти використання веб-сторінки тощо є частиною NFR
6	FR визначаються з вимог до ПЗ.	NFR є похідними від функціональних вимог.
7	FR можуть існувати без NFR.	NFR не можуть існувати без FR.
8	Надає конкретну інформацію про особливість.	Вимога може мати багато атрибутів нефункціональних вимог.
9	Вивести функціональні вимоги з вимог SW можливо майже для всіх вимог бізнесу	NFR часто пропускають для документування, оскільки відповідні запитання щодо FR не задаються.
10	Впровадження FR здійснюється в одній збірці програмного забезпечення.	NFR застосовуються протягом усього ЖЦ до досягнення необхідної поведінки.

11	Більш за все помітні замовнику.	Найчастіше, їх не видно замовнику, але можуть бути досвідченими в довгостроковій перспективі.

Підсумовуючи загальний досвід з написання вимог до ПЗ треба зазначити, що в кожній окремій області знань вимоги до ПЗ можуть суттєво відрізнятися. Таким чином, слід зазначити, що отримання практичних навичок щодо визначення та формалізації вимог до ПЗ з функціональністю криптографічного захисту інформації є актуальною задачею в рамках написання цієї роботи.

1.7 Квантовий розподіл ключів

Квантовий розподіл ключів - це безпечний метод зв'язку, який реалізує криптографічний протокол із залученням компонентів квантової механіки . Це дозволяє двом сторонам створити спільний випадковий секретний ключ, відомий лише їм, який потім можна використовувати для шифрування та розшифрування повідомлень . Його часто неправильно називають квантовою криптографією , оскільки це найвідоміший приклад квантової криптографічної задачі.

Важливою та унікальною властивістю розподілу квантових ключів є здатність двох користувачів, що спілкуються, виявити присутність будь -якої третьої сторони, яка намагається отримати знання про ключ. Це є результатом фундаментального аспекту квантової механіки: процес вимірювання квантової системи загалом порушує систему. Третя сторона, яка намагається підслухати ключ, повинна якимось чином виміряти його, вносячи таким чином виявлені аномалії. За допомогою квантових суперпозицій або квантових переплетень та передачі інформації в квантових станах, може бути реалізована система зв'язку, яка виявляє підслуховування. Якщо рівень прослуховування нижче певного порогу, може бути виготовлений ключ, який гарантовано буде безпечним (тобто підслуховувач не має інформації про це), інакше неможливий захищений ключ і зв'язок припиняється. [3]

Безпека шифрування, що використовує розподіл квантових ключів, спирається на основи квантової механіки, на відміну від традиційної криптографії з відкритим ключем , яка спирається на обчислювальну складність певних математичних функцій і не може надати жодного математичного доказу щодо дійсної складності зміни використовуються односторонні функції. QKD (Quantum key distribution) має перевірену безпеку, засновану на теорії інформації та секретності вперед .

Основним недоліком розподілу квантових ключів є те, що він зазвичай спирається на наявність автентифікованого класичного каналу зв'язку. У сучасній криптографії наявність автентифікованого класичного каналу означає, що він або вже обмінявся симетричним ключем достатньої довжини або відкритими ключами достатнього рівня безпеки.

Квантовий розподіл ключів використовується тільки для виробництва та розповсюдження ключа, а не для передачі будь -яких даних повідомлень. Квантова комунікація передбачає кодування інформації в квантових станах або кубітах , на відміну від використання бітами класичної комунікації . Зазвичай для цих квантових станів використовують фотони . Розподіл квантових ключів використовує певні властивості цих квантових станів для забезпечення його безпеки. Існує кілька різних підходів до розподілу квантових ключів, але їх можна розділити на дві основні категорії залежно від того, яку властивість вони використовують.

Якщо використовувати квантові частинки як носії інформації, то при спробі перехопити повідомлення воно змінює стан частинки, що дозволить виявити порушення передачі. Крім того, неможливо отримати повну інформацію про квантовий об’єкт, тому його неможливо відтворити. Ці властивості квантових об’єктів роблять їх «невловимими». Ідея використання квантових об’єктів для захисту інформації від підробки та несанкціонованого доступу вперше була запропонована Стівеном Веснером у 1970 році. Через десять років Беннет і Брасард, знайомі з роботами Веснера, запропонували використовувати квантові об'єкти для передачі ключів. У 1984 році вони опублікували статтю з описом ключового протоколу квантового поширення BB84. Носіями інформації в протоколі BB84 є фотони, поляризовані під кутами 0, 45, 90, 135 градусів. Однак неможливо точно відрізнити вертикально поляризовані фотони від 45 градусів поляризованих фотонів. Ці поведінкові характеристики квантових об’єктів становлять основу протоколів розподілу квантових ключів.

Розглянемо один з методів розподілу ключів який використовується також в квантовій комунікації.

1.8 Протокол Діффі-Хелмана

Обмін ключами за допомогою протоколу Діффі-Хеллмана – це метод безпечного обміну криптографічними ключами загальнодоступним каналом. Метод є одним із перших практичних прикладів обміну відкритими ключами. Сьогодні він використовується для багатьох програм, таких як, наприклад, Proton Mail, SSH, GPG і так далі. [7]

При розгляду методу будемо використовувати такі значення - Аліса (А), Боб (Б) та інша людина (це третя особа, яка не приймає участь у спілкуванні).

Діффі-Хеллман працює за принципом неповного обміну ключем шифрування через мережу. Кожна сторона має відкритий ключ (який може бачити кожен) та закритий ключ (його може бачити лише користувач комп'ютера).

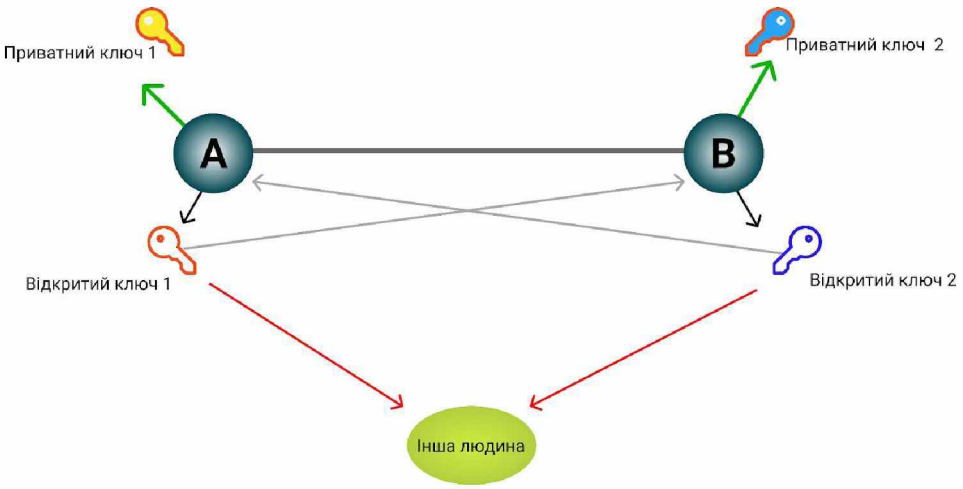


Рисунок 1.2 - обмін ключами

Створюємо частковий ключ шифрування, використовуючи 3 доступні одиниці інформації: Аліси закритий та відкритий ключі та відкритий ключ Боба. Алгоритмічно ми погодилися використовувати відкритий ключ Аліси як базу та його відкритий ключ як модуль.

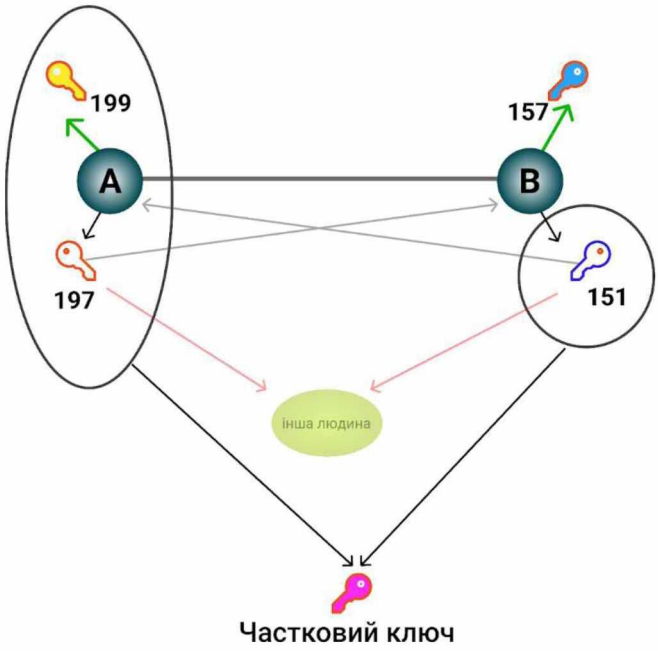


Рисунок 1.3 - частковий обмін ключами

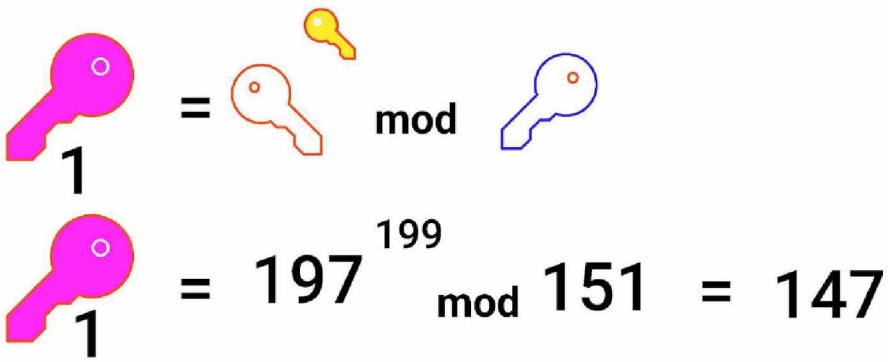


Рисунок 1.4 - знаходження часткового ключа 1

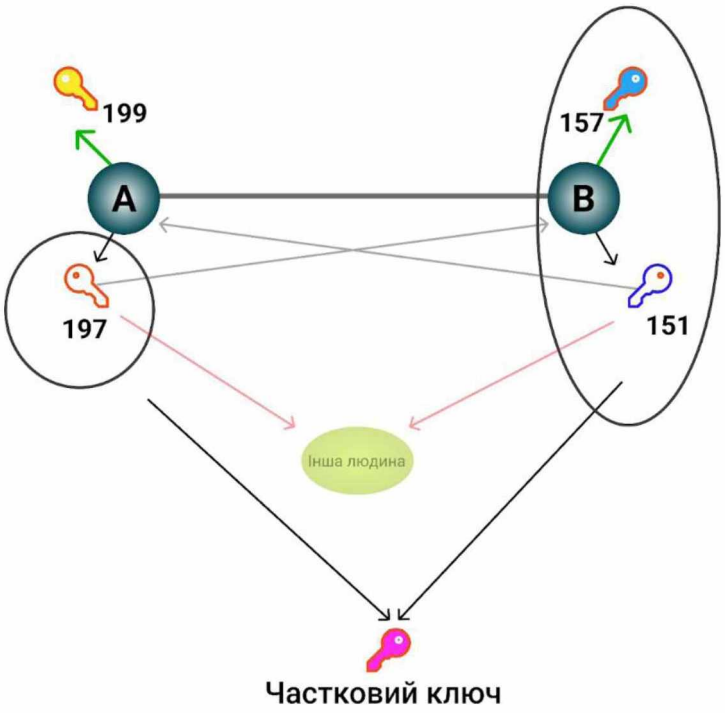


Рисунок 1.5 - частковий обмін ключами

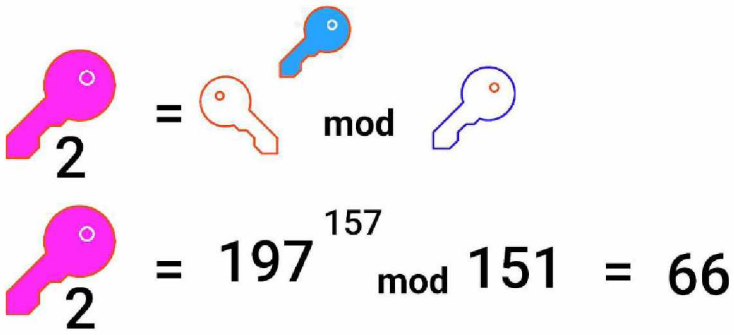
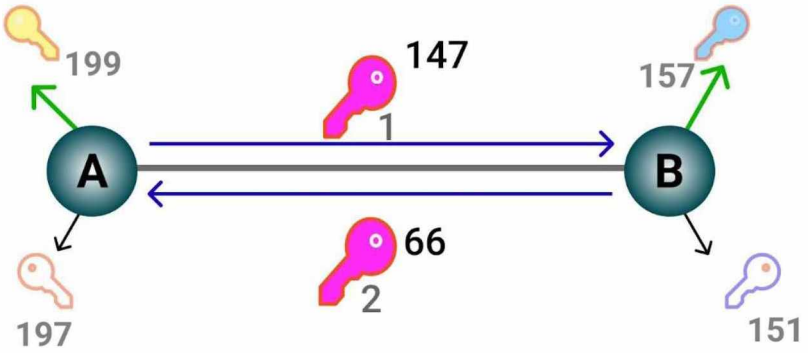


Рисунок 1.6 - знаходження часткового ключа 2

Інша людина може побачити обмін частковими ключами. Він також може знати обидва відкриті ключі. Але йому для повного перехоплення потрібно зламати відповідні закриті ключі.



$$197^{x} \bmod 151 = [1 - 150]$$

Рисунок 1.7 - відбувається частковий обмін ключами

Особливість обчислення за модулем у тому, що функція змушує значення циклічно змінюватися. Якщо у вас є модуль 151, значення буде між 0 та 151

Існує нескінченна кількість можливих значень, які за модулем 151 дорівнюють або 66, або 147.

В результаті інформації для отримання закритих ключів користувачів недостатньо (якщо не брати до уваги грубий перебір шалено великої кількості всіляких варіантів).

Крім того, у прикладі використовуються невеликі числа для закритих та відкритих ключів, що робить вибір реальним. У реальному житті це майже неможливо.

Тепер, коли ми успішно обмінялися частковими ключами, давайте знову застосуємо оператори ступеня та поділу із залишком до часткового ключа іншої людини, використовуючи наші власні закриті ключі таким чином (рис 1.8).

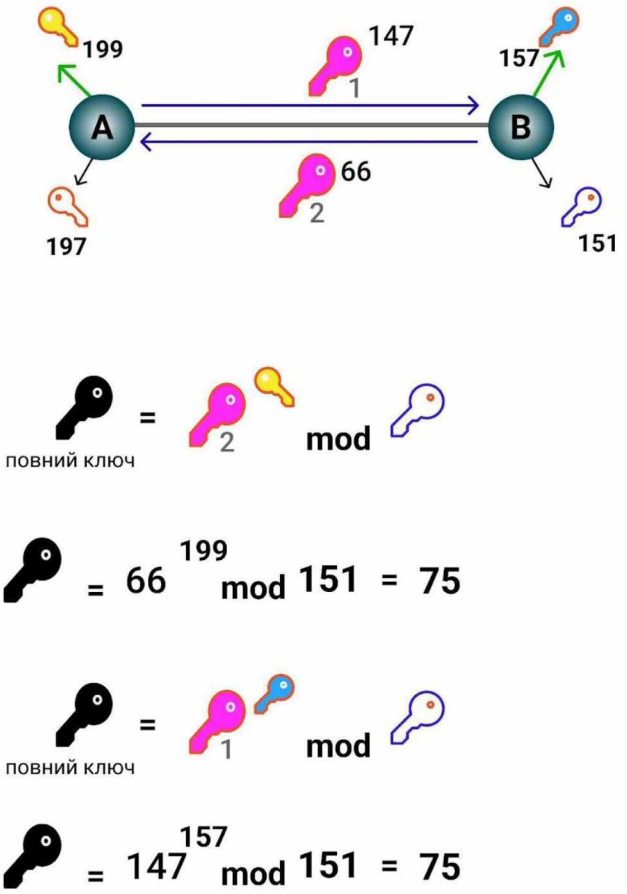


Рисунок 1.8 - генерація повного ключа

Ми отримали два однакових повних ключа, кожен з яких дорівнює 75. Причина полягає в наступному математичному співвідношенні (рис 1.9).

$$(g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p$$

Рисунок 1.9 - співвідношення

Де,a и b — закриті ключі, a g и p — відкриті.

Нам вдалося обмінятися по мережі достатньою кількістю інформації, щоб згенерувати загальний ключ шифрування, не ставлячи під загрозу закриті ключі.

Розділ 2. ФУНКЦІОНАЛЬНІ ТА НЕФУНКЦІОНАЛЬНІ ВИМОГИ ДО СИСТЕМИ ПЕРЕДАЧІ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ КВАНТОВОГО КАНАЛУ

При розробці програмного забезпечення функціональні вимоги визначають функції, які має виконувати вся програма або лише один із його компонентів. Функція складається з трьох кроків: введення даних – поведінка системи – виведення даних. Він може обчислювати, маніпулювати даними, виконувати бізнес-процеси, встановлювати взаємодію користувача або виконувати будь-які інші завдання.[9].

Нефункціональні вимоги визначають стандарти продуктивності та атрибути якості програмного забезпечення, наприклад зручність використання системи, ефективність, безпека, масштабованість тощо.

2.1 Функціональні вимоги до системи передачі інформації з використанням квантового каналу

Функціональні вимоги є важливими, оскільки вони показують розробникам програмного забезпечення, як повинна поводитися система. Якщо система відповідає функціональним вимогам, отже, вона працює належним чином.

Розглянемо та проаналізуємо функціональні вимоги до системи передачі інформації з використанням квантового каналу (мессенджер). Для зручності та подальшої ідентифікації вимог будемо використовувати умовне кодування кожної вимоги з використанням “FR”.

FR1. При запуску мессенджера користувач повинен бути повідомлений про те, який мессенджер він використовує та его поточну версію. Також можна зазначити що, він працює за допомогою квантового обміну ключів (акцент на унікальності).

FR2. Месенджер повинен предоставити вибір способу доступу до додатку:

Код-пароль

Face ID

Touch ID

Без додаткової авторизації

FR3. Месенджер повинен предоставити вибір способу авторизації через соціальні мережі (Google, Facebook, E-mail).

FR3. Двухфакторна авторизація (за допомогою смс на мобільний телефон, або електронну пошту приходить код авторизації ,або додаткового паролю)

FR4. Мессенджер повинен при реєстрації користувачу присвоїти особистий(уникальний) client ID, який в подальшому будет ідентифікувати аккаунт.

FR5. Мессенджер повинен зробити запит у користувача на дозвіл:

Синхронування контактів (для швидкої взаємодії з іншими користувачами).

Використання камери, мікрофона (для відправки фото, аудіо\відео дзвінку та повідомлень).

Відправка повідомлень (в фоновому режимі).

Використання геолокації (для відправки).

FR6. Мессенджер повинен мати функціонал заповнення особистої інформації в обліковому записі.

FR7. У користувача повинна бути можливість редагування та збереження особистих даних.

FR8. Мессенджер повинен створити запит на дозвіл на видимість статусу (online, offline) для інших користувачів або окремо вибраної групи людей.

FR9. Мессенджер повинен мати можливість запрошувати нових користувачів (автоматично створене повідомлення із ссилкою на додаток, та одразу відправка через звичайне СМС).

FR10. Мессенджер повинен мати функціонал відображення списку контактів\ друзів, які online у розділі контактів.

FR11. Мессенджер повинен давати можливість блокувати користувачів. Обмежувати їх дії стосовно особистого профілю.

FR12. Мессенджер повинен мати функціонал можливість пересилання файлів за межі мессенджеру (інші додатки).

FR13. Мессенджер повинен повідомляти користувача про програми які будуть працювати у фоновому режимі.

FR14. Користувач повинен бути повідомлений про нові оновлення. Якщо мессенджер автоматично оновлюється, повинно бути повідомлення про те, що змінилося. Якщо, оновлення встановлено не автоматично, користувачу необхідно запропонувати це зробити, і одразу зазначити, що було змінено\ виправлено\додано).

FR15. Мессенджер повинен показати користувачу коротку інструкцію по взаємоді з пз (Рис.3.4-3.7).

FR16. Мессенджер повинен мати можливість здійснювати аудіо\відео дзвінок в гарній якості при достатній швидкості інтернету

FR17. Користувач володіє інформацією чи є чат або дзвінок зашифрованим. (Для цього у кожного учасника дзвінка або листування на екрані відображається певний набір цифр та/або символів у певному порядку. При захищеності спілкування вони будуть співпадати)

FR18. Мессенджер повинен мати функціонал оформлення чату, повідомлень, шрифтів з наглядними прикладами.

FR19. Можливість видаляти, закріпити, пересилати, виділяти повідомлення/файли в будь-якому форматі.

FR20. Мессенджер повинен надавати фунціонал для повного видалення повідомлень/файлів без можливості відновлення.

FR22. Мессенджер повинен мати функціонал відкриття посилань у безпечному режимі.

FR23. Мессенджер повинен мати функціонал помітки, що користувач отримав\прочитав повідомлення.

FR24. Мессенджер повинен мати функціонал обмеження на відправку повідомлень (наприклад, тільки з 10:00 інші користувачі можуть відправити вам повідомлення).

FR25. Мессенджер повинен мати функціонал автоматичного перекладу повідомлень.

FR26. Мессенджер повинен мати функціонал попередження: чи дійсно користувач хоче відправити особоту інформацію (номер картки, телефону, паспортні дані)

FR27. Мессенджер повинен мати функціонал для відправки двух і більше файлів альбом або кожен окремо.

FR28. Мессенджер повинен мати функціонал бачити кількість непрочитаних повідомлень. (Загалом, в окремих чатах).

FR29. Мессенджер повинен мати функціонал пересилання повідомлень\файлів двум и більше людям одночасно.

FR30. Мессенджер повинен мати функціонал відправки файлів у сжатому\повному розмірі.

FR31. Мессенджер повинен мати можливість використання системних смайлів.

FR32. Мессенджер повинен мати функціонал бачити, що співбесідник набирає повідомлення.

FR33. Мессенджер повинен мати функціонал при відправці файлів, додавати опис.

FR34. Мессенджер повинен мати можливість завантаження даних\ файлів на свій пристрій.

FR35. Мессенджер повинен мати функціонал відповідного елементу, якщо повідомлення не відправлене.

FR36. Мессенджер повинен мати можливість спілкування із службою підтримки та перелік найчастіших запитань.

FR37. Мессенджер повинен предостовляти можливість для пошуку інформації за:

Ключовими словами (серед усіх чатів\ у конкретно выбранному чаті)

Датою

Контактам

FR38. Мессенджер повинен мати функціонал створення груп з контактів.

FR39. Мессенджер повинен мати функціонал надавання різних прав (адміністрування) учасникам групи.

FR40. Мессенджер повинен мати функціонал налаштування доступу до груп. Публічним або приватним.

FR41. Мессенджер повинен в групі, мати можливість, бачити котрий саме співбесідник набирає повідомлення

FR42. Мессенджер повинен мати функціонал закріпити чати\користувачів.

FR43. Мессенджер повинен мати фільтри\ сортування контактів за іменем (контактом).

FR44. Мессенджер повинен мати функціонал запису аудіо повідомлень в MP3 форматі продовженістю 30 хвилин. Та прослуховування голосових повідомлень у фоновому режимі.

FR45. Мессенджер повинен мати функціонал перегляду відправлених\отриманих файлів у окремій вкладці.

FR46. Мессенджер повинен мати можливість налаштування звукових повідомлень з чатів.

FR47. Мессенджер повинен мати можливість очистити кеш.

FR48. Мессенджер повинен мати функціонал синхронування декількох пристроїв.

FR49. Мессенджер повинен мати інформацію про синхониховані пристрої (пристрій, дата останнього заходу)

FR50. Мессенджер повинен бути адаптивним для ПК, планшету, телефону.

FR51. Мессенджер повинен мати бути адаптивним до вертикально та горизонтального положення гаджету.

2.2 Нефункціональні вимоги до системи передачі інформації з використанням квантового каналу

Нефункціональні вимоги є важливими, оскільки вони допомагають розробникам програмного забезпечення визначати можливості та обмеження системи, які необхідні для розробки високоякісного програмного забезпечення. Отже, нефункціональні вмоги так само важливі, як і функціональні, для успішного впровадження продукту.

Якщо програма не відповідає нефункціональним вимогам, вона продовжує виконувати свої основні функції, однак не зможе забезпечити зручність для користувача.

NFR1. Для повноцінного функціонування системи версія Android повинна бути не нижче Android 10, на IOS не нижче iOS 12.

NFR2. ПЗ повинно функціонувати безперебійно.

NFR3. Дані має бути збережено на сервері при не запланованій відмові обслуговування на клієнтському пристрою, виходу з особистого запису чи закриття додатку. (при настанні нештатних ситуацій, наприклад, автоматичний перезапуск, відновлення роботи, дублювання важливих даних, резервування логіки).

NFR4. ПЗ, що переривається за будь-яких обставин має зберігати початковий стан і повертатися до того ж стану/сторінці, який був до того, як він був перерваний.

NFR5. ПЗ має зберігати працездатність при збільшенні кількості користувачів, чи об’єму даних, які працюють.

NFR6. ПЗ має підтримувати необхідну кількість одночасно працюючих користувачів.

NFR7. ПЗ має при активному сеансі використовувати не більше ніж 512 мб оперативної пам’яті пристрою, де виконується сесія обміну інформацією.

NFR8. ПЗ має бути адаптивним до горизонтального та вертикального положення програми.

NFR9. ПЗ має підтримувати засоби автоматичного та ручного тестування за наявність необхідного інструментарію.

NFR10. ПЗ має підтримувати локалізацію з переліком таких мов: Українська, Англійська, Німецька, Польська.

NFR11. ПЗ має підтримувати можливість видалення облікового запису з видаленням всіх даних користувача при відсутності активності 1 рік.

NFR12. Система ПЗ повинна отримувати оповіщення не пізніше ніж через 0,5 секунд після виникнення події.

NFR13. Всі повідомлення ПЗ повинні бути надіслані не пізніше ніж за 10 секунд після отримання оповіщення про подію.

NFR14. Базовий вектор для формування ключа повинні передаватися квантовим каналом зв’язку. Зашифрована інформація повинна передаватися відкритим каналом зв’язку.

NFR15. ПЗ має при недостатній кількості пам’яті, повинно згружати не використовуванні файли за період обраний користувачем.

NFR16. ПЗ має оновлювати список користувачів які online кожних 2 секунди.

NFR17. ПЗ має справлятися зі зростаючим робочим навантаженням при одночасному використанні різних додатків. Повинно працювати у режимі багатозадачності.

NFR18. ПЗ повинно не надавати доступу, доки користувач не створить надійний пароль. (Стосується створення додаткового пароля для двофакторної авторизації)

NFR19. ПЗ має обробляти дані користувача (кеші, збережені дані і т. д.), коли вони збільшуються, без затримки, оптимізуючі спосіб зберігання і доступу до них.

NFR20. ПЗ має використовувати бази даних для довготривалого зберігання результатів роботи програми.

Написання вимог є важливим процесом для створення будь-якого програмного забезпечення. Вони повинні бути чітко прописані та зрозумілі. Потрібно розуміти чому написання вимог є складним процесом, тому вони повинні бути однозначними.

Розділ 3 КОНЦЕПТ МЕССЕНДЖЕРУ

Для отримання дизайн-концепт ПЗ має пройти етапи проектування дизайну.

1. Аналіз

Робота дизайнера починається з вивчення та аналізу вихідних даних. І часто таке буває, що на цьому етапі у нього виникає велика кількість питань. Всі ці питання він зазвичай ставить PM (project manager). Адже саме PM писав специфікацію.

2. Розробка UX інформаційної структури- порядок організації контенту та всіх основних елементів проекту.

Розробка ІС включає визначення мети проекту, аналіз ЦА (цільової аудиторії), аналіз конкурентів, складання фокус-групи (для докладного дослідження). Результатом виконання цієї стадії є варфрейми.

Зазвичай цей етап входить ретельне тестування кожної функції проекту. Чим більше тестів, тим краще. Оскільки повинна бути впевненість, що користувач зможе розібратися у всіх функціях та логіці вашого проекту. [8].

3. UI - Результат цього етапу - готові кінцеві прев'ю сторінок (jpg або png), вихідники (psd макети). А також UI set (UI kit) для великих проектів - набір всіх елементів сайту (додатки) з усіма існуючими станами).

4. Прототипування -тестування інтерактивного (клікабельного) зразка.

Всі функції та фішки (цікаві анімації та ефекти) необхідно реалізувати у прототипах. На те вони й прототипи, щоби показувати основну візуальну складову проекту. Рекомендують також робити прототипи ще на стадії розробки UX, коли ви вже є визначенність з інформаційною структурою проекту і варфрейми. Тоді є можлтвість відразу зібрати з них клікабельний прототип та предомонструвати його.

Головною ідеєю мессенджеру було: зробити його максимально захищениим та зрозумілим у використанні.

На рисунку 11 відображено головний екран при запуски мобільного додатку. На ньому зображено логотип та версія додатку. Його назва “MessTo” складається з двох слів messenger together.

Головний екран реалізує такі FR: При запуску мессенджера користувач повідомлений про те, який мессенджер він використовує та его поточну версію.

Рис. 3.1 - головна сторінка завантаження

На малюнках 3.1 і 3.2 зображено екран авторизациії. На Рисунку 3.1 екран в неактивному стані (жодне з полей для заповнення неактивне користувачем).

На рисунку 3.2 активне поле для заповнення номеру телефону. Воно відрізняється від початкового стану, коли не активне. Така реалізація даного поля є більш зручною для заповнення та перевірки введеної інформації користувачем.

Концепція даних екранів реалізує таку функціональну вимогу, як вибір способу авторизації через соціальні мережі (Google, Facebook, E-mail)

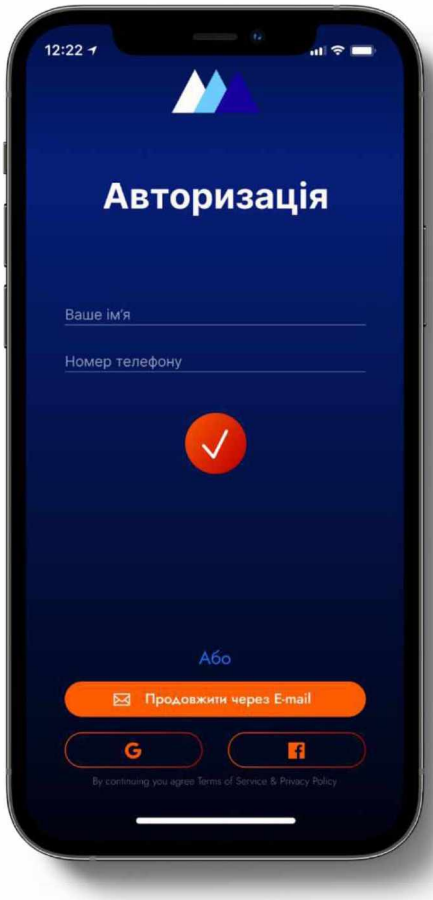


Рис. 3.2 - сторінка авторизації

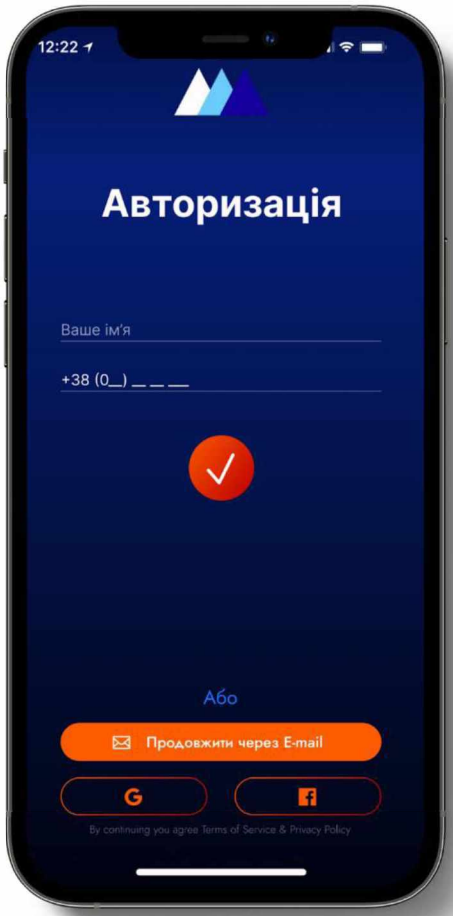


Рис. 3.3 - сторінка авторизації (при введенні номеру телефона)

На рисунках 3.4 -3.7 зображена коротка інструкція по елементам мессенджеру.

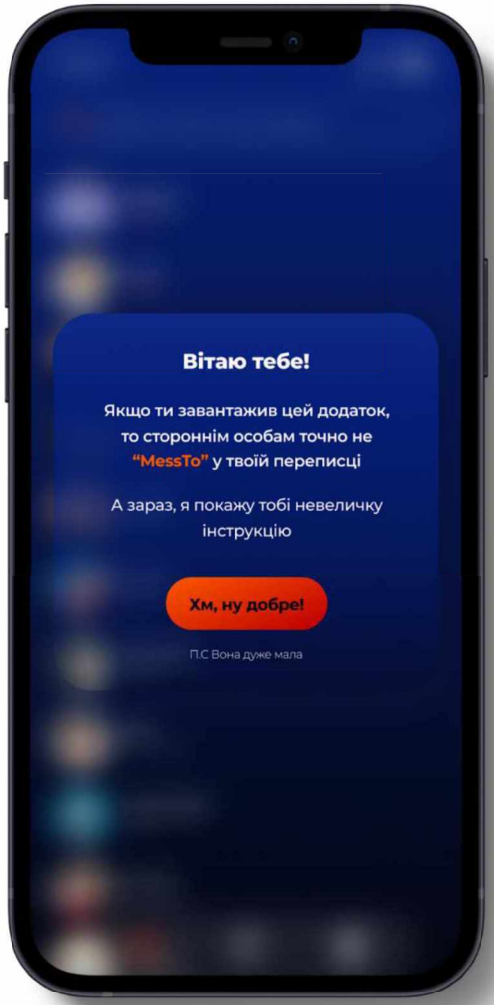


Рис. 3.4 - повідомлення користувачу

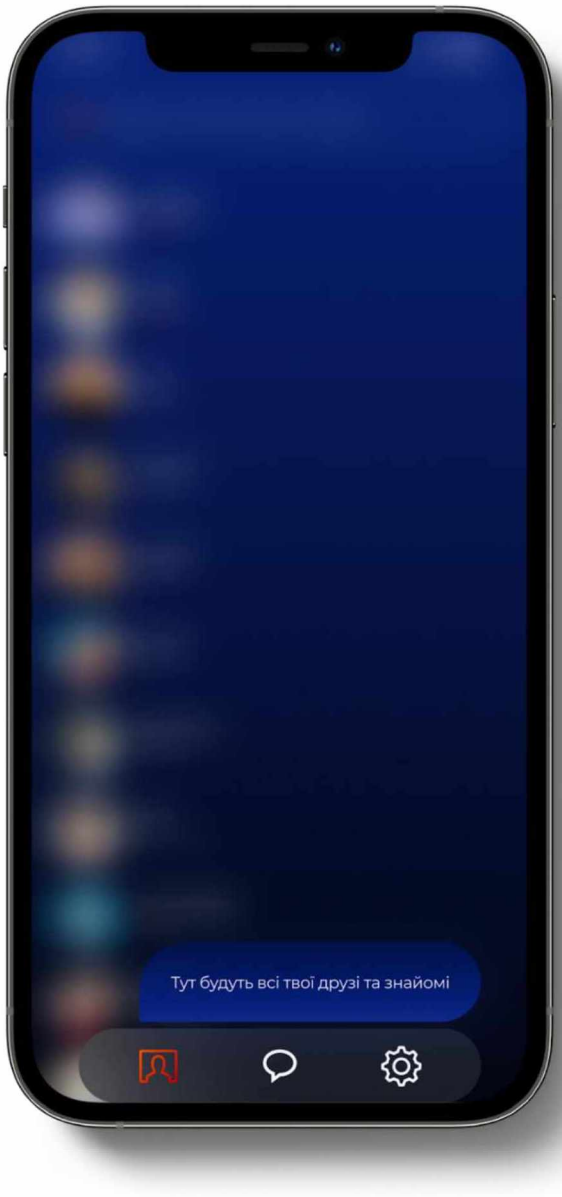


Рис. 3.5 - інформація про контакти

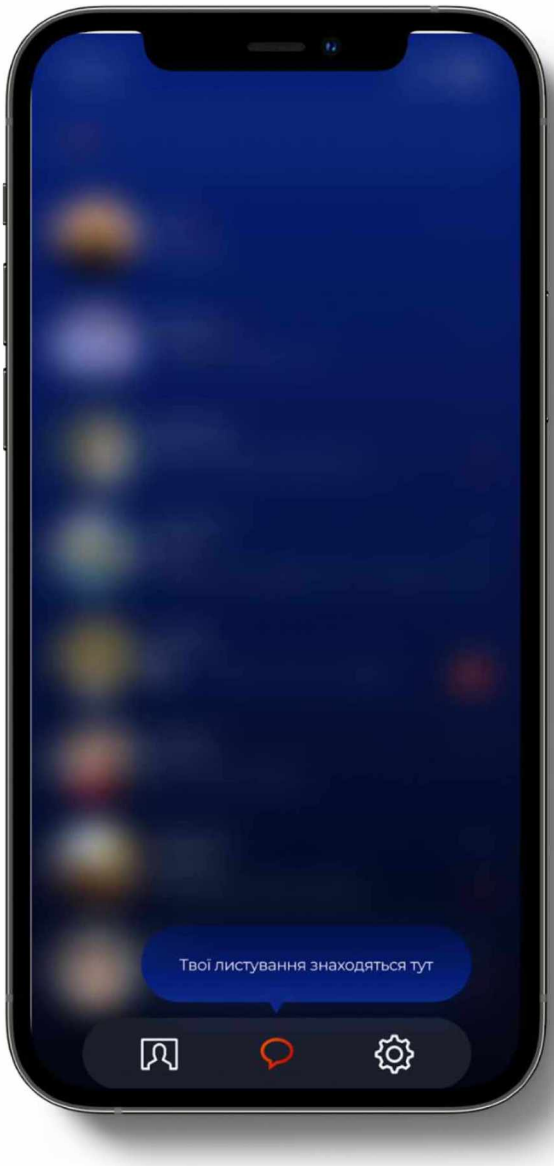


Рис. 3.6 - інформація про чати

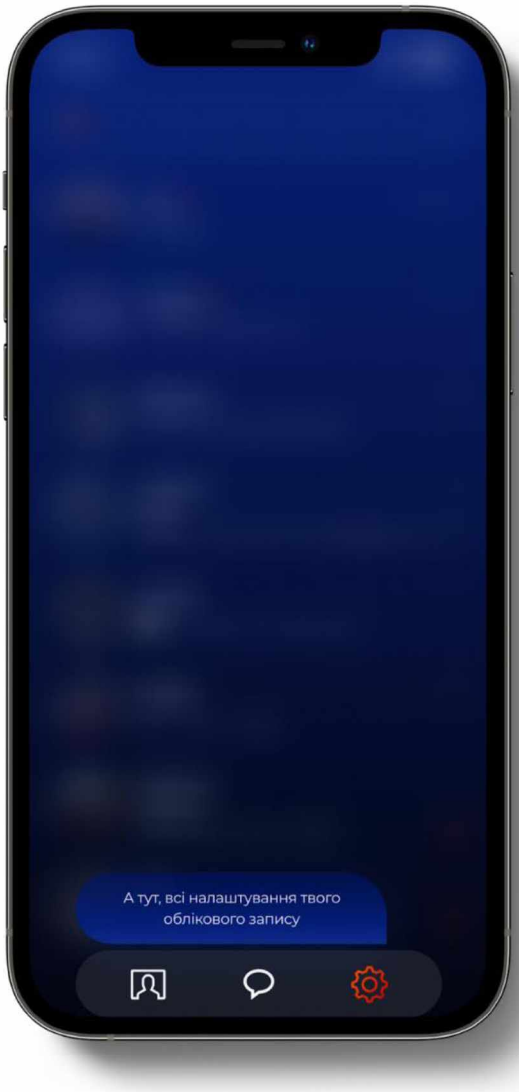


Рис. 3.7 - інформація про налаштування

На рисунку 3.8 реалізована вкладка контактів. На ній зображено список контактів\ друзів, які online\ offline, без статусу. Та пошук контактів за іменем.

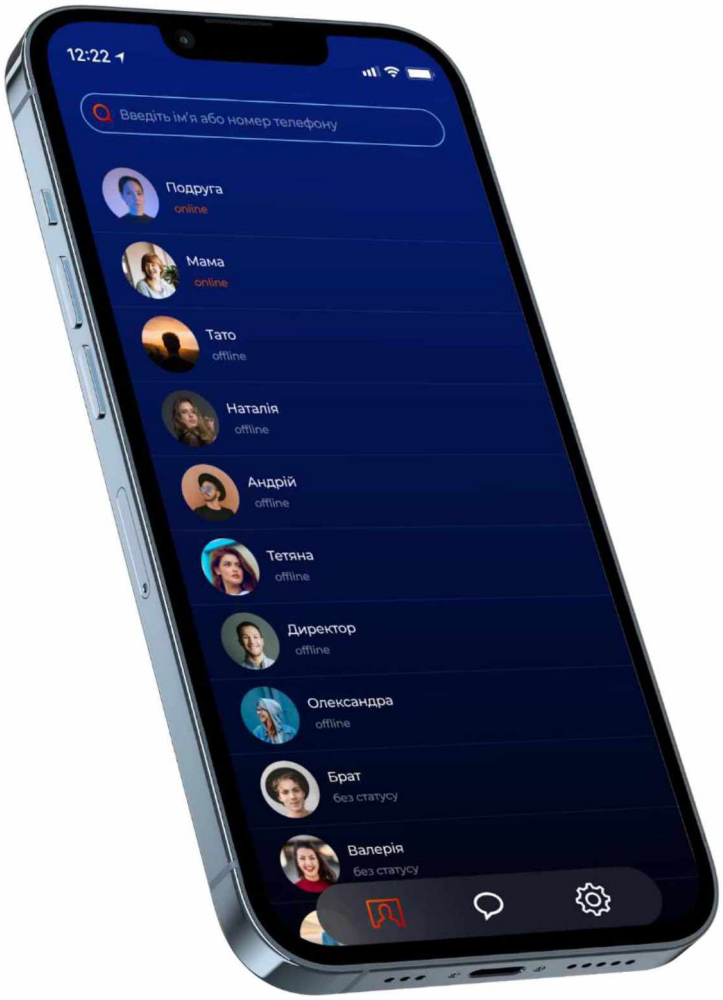


Рис. 3.8 - контакти

На рисунку 3.9 концепт розділу з діалогами. Він реалізує такі FR:

Видимість непрочитаних повідомлень (Загалом, в окремих чатах). Непрочитані повідомлення зображуються помаранчем кругом навпроти відповідного чату. Якщо, повідомлень більше ніж 1, відображається їх кількість як у групі “Мемаси”.

Можливість бачити як контакт “Подруга” набирає повідомлення.

Також є поле для пошуку інформації за:

Ключовими словами (серед усіх чатів\ у конкретно выбранному чаті)

Датою

Контактам

Є можливість закріпити чати\користувачів. Контакт “Тато” закріплений - це видно по відповідній позначки, яка виглядає як канцелярська кнопка. Також цей чат завжди буде першим, незважаючи на інші повідомлення.

Також налаштування звукових повідомлень з чатів. В чаті “Новини” звукові оповіщення відключені. На це вказує відповідна позначка у вигляді перечеркнутого рупору.

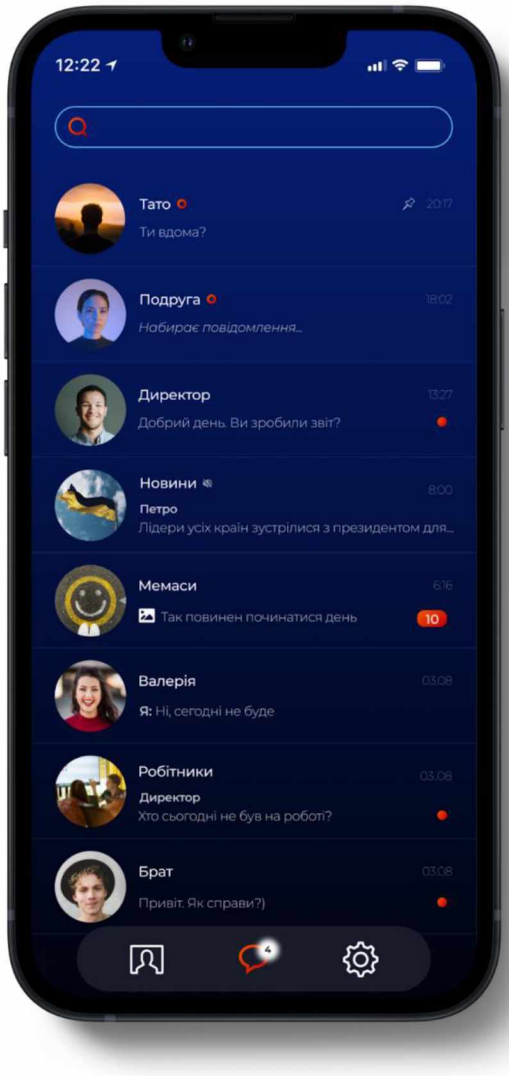


Рис. 3.9 - чати

На рисунку 3.10 відкритий діалог. В ньому реалізовані такі вимоги:

Помітки, що користувач отримав\прочитав повідомлення. (Помаранчева - користувач отримав и прочитав повідомлення, сіра - повідомлення не прочитане)

Помітка, що повідомлення не відправлене. (червоне коло із знаком оклику “!”)

Запис аудіо повідомлень. Знак мікрофону.

Та прикріплювати файли, відображається як скріпка.

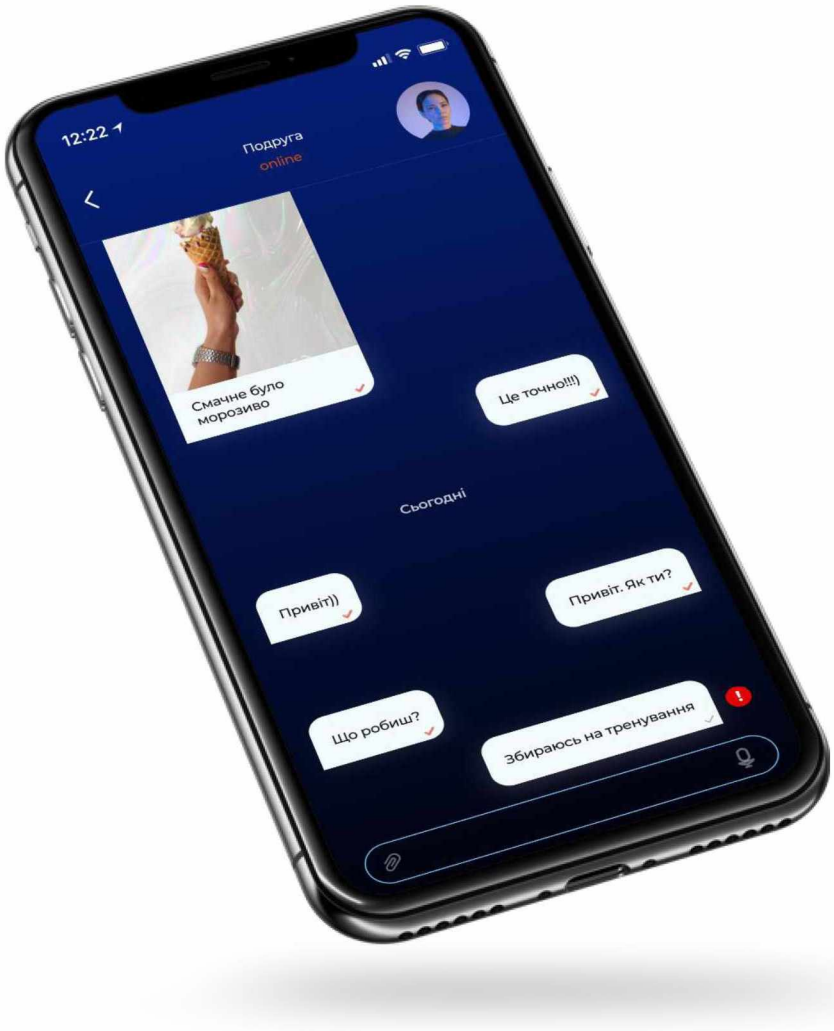


Рис. 3.10 - листування

ВИСНОВКИ

У випускній роботі досліджено область знань бізнес аналізу (БА) та вимог до програмного забезпечення (ПЗ). Було детально проаналізовано джерела інформації щодо виявлення та структурування всіх вимог до системи обміну інформацією. Досліджено етапи розробки вимог та їх особливості. Також в ході роботи були визначенні функціональні та нефункціональни вимоги до ПЗ.

Вивчаючи інформацію детальніше було розглянуто квантовий обмін ключей, та один із методів - протокол Діффі-Хелмана.

Для подальшої реалізації було стисло досліджено та описано функціональні та нефункціональні вимоги до системи обміну інформацією з використанням квантового розподілу ключів. Більшість функціональних вимог для данного ПЗ не криточно відрізняються від вимог для звичайного мессенджера.

Було реалізовано концепт мессенджеру реалізований в програмі Figma - векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи, що розробляється однойменною компанією. Був проаналізований ринок конкурентів, виявлені сильні та слабкі сторони, проведений порівняльний аналіз.

В дизайні використані сучасні тренди типографіки, інтерфейсу та кольорів. Інтерфейс є легким та інтуїтивно зрозумілим для користувача. В розробці концепту проаналізована психологія користувача та цільової аудиторії.

СПИСОК ЛІТЕРАТУРИ

1. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ [Електронний ресурс]. – Режим доступу : http://baklaniv.at.ua/ANALIZ_VYMOG/lekcija_1-2.pdf

2. Опорний конспект лекцій з курсу “Аналіз вимог до програмного забезпечення” [Електронний ресурс]. – Режим доступу : http://dspace.wunu.edu.ua/bitstream/316497/7281/1/FCIT_kKN_sPZS_dAVPZ_%20LEC.pdf

3. Нефункціональні вимоги до системи: поняття та приклади: [Електронний ресурс]. – Режим доступу : https://ukr.kagutech.com/3985531-non-functional-system-requirements-concept-and-examples

4. Види і властивості вимог: [Електронний ресурс]. – Режим доступу : https://studopedia.org/8-131361.html

5. Фаза вимог: [Електронний ресурс]. – Режим доступу : https://uk.itpedia.nl/2016/12/19/sisp-2-1-de-requirements-fase/

6. Особливості функціональних вимог та не функціональних вимог: [Електронний ресурс]. – Режим доступу : https://uk.myservername.com/features-functional-requirements

7. Як працює обмін ключами в протоколі Діффі-Хелмана: [Електронний ресурс]. – Режим доступу : https://tproger.ru/translations/diffie-hellman-key-exchange-explained/

8. Жизненный цикл проекта. Этапы проектирования дизайна.

[Електронний ресурс]. – Режим доступу : https://telegra.ph/ZHiznennyj-cikl-proekta-EHtapy-proektirovaniya-dizajna-04-19 What are Business Requirements?: [Електронний ресурс]. – Режим доступу : https://requirements.com/Content/What-is/what-are-business-requirements-1

9. Функциональные и нефункциональные требования: полное руководство [Електронний ресурс]. – Режим доступу

: https://bestprogrammer.ru/izuchenie/funktsionalnye-i-nefunktsionalnye-trebovaniya-polnoe-rukovodstvo

10. Етапи розробки мобільного додатку: аналітика та технічне завдання: [Електронний ресурс]. – Режим доступу : https://vc.ru/dev/142571-etapy-razrabotki-mobilnogo-prilozheniya-analitika-i-tehnicheskoe-zadanie

11. Визначення вимог до програмних систем [Електронний ресурс]. – Режим доступу : https://studfile.net/preview/10015384/

12. Технічне завдання на створення мобільного додатку: [Електронний ресурс]. – Режим доступу : https://www.hse.ru/mirror/pubs/share/190980326

13. Вимоги до мобільних додатків: [Електронний ресурс]. – Режим доступу : https://apiok.ru/res/files/apps/mobile_rules.pdf

14. Как мы делаем мобильные приложения: аналитика и техническое задание: [Електронний ресурс]. – Режим доступу : https://wnfx.ru/kak-my-delaem-mobilnye-prilozheniya-analitika-i-tehnicheskoe-zadanie/

15. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ МОБИЛЬНОГО ПРИЛОЖЕНИЯ: [Електронний ресурс]. – Режим доступу : https://businessarchitecture.ru/tehnicheskoe-zadanie-na-razrabotku-mo/

16. Разработка технического задания (ТЗ): [Електронний ресурс]. – Режим доступу : https://ios-lab.ru/razrabotka_tz/

17. Нефункциональные требования к программному обеспечению. [Електронний ресурс]. – Режим доступу : https://habr.com/ru/post/231961/

18. Нефункциональные требования к программному обеспечению. [Електронний ресурс]. – Режим доступу : https://myalm.ru/news/%D0%9E%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BD%D0%B5%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8

%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D1%85-%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9

19. Understanding Functional and Non-Functional Requirements in App Development: [Електронний ресурс]. – Режим доступу : https://www.moveoapps.com/blog/functional-and-non-functional-requirements-in-app-development/

20. 9 Nonfunctional Requirements Examples: [Електронний ресурс]. – Режим доступу : https://www.indeed.com/career-advice/career-development/non-functional-requirements-examples

21. Non-Functional Requirement of the Mobile Development system.: [Електронний ресурс]. – Режим доступу : https://medium.com/@vishwasng/non-functional-requirement-of-the-mobile-development-system-e0ed98f2a872

22. Non-functional Requirements: Examples, Types, How to Approach: [Електронний ресурс]. – Режим доступу : https://www.altexsoft.com/blog/non-functional-requirements/

23. Addressing Non-Functional Requirements in Mobile Applications [Електронний ресурс]. – Режим доступу : http://web.cse.ohio-state.edu/~champion.17/5236/Lecture8_NFRTesting.pdf

24. Non-Functional Requirements in Mobile Applications [Електронний ресурс]. – Режим доступу : https://sachinsdate.wordpress.com/2013/04/27/non-functional-requirements-in-mobile-applications/

25. [Електронний ресурс]. – Режим доступу : https://www.pcquest.com/non-functional-requirements-mobile-apps-1/

26. Нефункциональные требования к программному обеспечению. [Електронний ресурс]. – Режим доступу : https://habr.com/ru/post/231961/

27. [Електронний ресурс]. – Режим доступу : [Електронний ресурс]. – Режим доступу : https://wblog.wiki/uk/Quantum_key_distribution

28. Документ вимог до мобільного додатку: [Електронний ресурс]. – Режим доступу : https://punicapp.com/blog/pages/1327/1327