

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

Розробка офлайн месенджера на React Native

Текстова частина до курсової роботи

за спеціальністю «Комп'ютерні науки» - 122

Керівник курсової роботи

асистент

Калітовський Б.В.

(Підпис)

“ ___ ” _____ 2022 року

Виконала студентка КН-3

Блавт К.-М.В.

“ ___ ” _____ 2022 року

Київ – 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

зав.кафедри

_____ Жежерун О.П

“ ___ ” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу студентці Блавт Катерині-Марії Володимирівні
факультету інформатики 3 курсу бакалаврської програми

ТЕМА: Розробка офлайн месенджера на React Native

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

РОЗДІЛ 1. Теоретична частина

РОЗДІЛ 2. Проектування додатку

Висновки

Список літератури

Дата видачі “ ____ ” _____ 2022 р.

Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Календарний план виконання курсової роботи

Тема: Розробка офлайн месенджера на React Native

Календарний план виконання курсової роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	січень 2022	
2.	Огляд літератури за темою роботи.	березень 2022	
3.	Аналіз актуальності та дослідження аналогів	березень 2022	
4.	Вибір програмного забезпечення для роботи.	березень 2022	
5.	Написання текстової частини роботи.	травень 2022	
6.	Надання роботи науковому керівнику для перевірки.	кінець травня 2022	
7.	Коригування роботи за результатами попереднього захисту.	кінець травня 2022	
8.	Захист курсової роботи	7-14 червня 2022	

Студентка Блавт К.-М.В.

Керівник Калітовський Б.В.

“ ___ ” _____ 2022 р.

Зміст

Анотація	6
Вступ	7
РОЗДІЛ 1. Теоретична частина	9
Аналіз додатків обміну повідомленнями без мережі інтернет	9
Принципи роботи обміну повідомленнями через Wi Fi та Bluetooth	12
Сітчаста мережа	12
Сітчаста мережа на основі Bluetooth (Bluetooth mesh)	13
З'єднання через підключення до Wi Fi	14
Захист та шифрування даних при обміні через Wi Fi та Bluetooth	15
Висновки до розділу 1	16
РОЗДІЛ 2. Проектування додатку	17
Технічне завдання та вимоги до застосунку	17
Опис обраного стеку з аргументами	18
UML діаграми роботи додатку	19
Діаграма прецедентів	19
Діаграма активностей	20
Діаграма послідовності	20
Діаграми станів	21
Архітектура додатку	22
Архітектура додатків в React Native	22
Архітектура додатку, що розробляється	23
Схема бази даних	24
Порівняння баз даних для розробки мобільних додатків	24
База даних для застосунку, що розробляється	24
Висновки до розділу 2	25
Висновки	26
Список літератури	27

Анотація

Курсова робота присвячена проектуванню додатку для обміну текстовими повідомленнями в умовах відсутності мобільного зв'язку та інтернету. Такі умови можливі через різні причини - перебування в місцях, де немає покриття, наприклад на природі, пошкоджена інфраструктура або сильне перевантаження мережі або спеціальне глушіння зв'язку. З'єднання здійснюється за допомогою технології Bluetooth mesh. Проаналізовано схожі додатки, що вже є на ринку, розглянуто принципи роботи технології Bluetooth mesh, побудовано архітектуру додатку та обрано стек аргументів для розробки. React Native - фреймворк для розробки кросплатформенних застосунків, що дозволяє реалізувати його і для Android, і для IOS, що важливо, оскільки, серед тих додатків, що є на ринку, не всі є для IOS або працюють не досконало.

Вступ

В сучасному світі якісна та швидка комунікація — основа успіху будь-якого процесу. Щоденно мільйони людей обмінюються повідомленнями та дзвінками одні з одним та отримують миттєві відповіді навіть з інших куточків планети. Існує багато додатків, розроблених під різні потреби з корисними особливостями під кожен з них. Більшість людей на своєму телефоні зараз має більше двох-трьох різних додатків для спілкування, одні використовуються для роботи, інші більш зручні для приватного спілкування. В основному всі вони працюють завдяки інтернету або мобільному зв'язку. Проте, що робити, якщо він зникне? Причини можуть бути різні: найчастіше — технічні неполадки, які спеціалісти стараються виправити чим швидше. Але події останніх місяців в Україні показують, що в умовах війни під час військових дій проблеми можуть статись будь-коли. Зв'язок задежить він стабільності постачання електроенергії та цілісності інфраструктури, яка забезпечує його поширення, проте ми знаємо, що саме вони часто є ціллю ворожих атак. Також в місцях активних боїв його можуть глушити, а значить отримати чи передати важливу інформацію навіть на близьку відстань стає неможливо. Через це противник може спробувати цензурувати вільний обмін інформації вдаючись до інформаційної війни, тим самим маніпулювати мирним населенням та унеможливити його координацію чи мобілізацію. Проте існують додатки, які не використовують інтернет або мобільний зв'язок для передачі, та можуть якісно виконувати свою роботу в екстремальних умовах. В основному принципи роботи обміну повідомленнями в них відбувається через Wi Fi або Bluetooth, тому дальність таких передач близько 100 м, проте є варіанти для подовження. Такі додатки вже підтвердили свою дієвість та ефективність в найрізноманітніших ситуаціях. Наприклад стихійних лих в Ірані та

Мексичі, великих музичних фестивалях по всьому світу, під час протестів в Індії та Гонконгу. Отже, є доцільно розібратись в роботі таких додатків. Технології зв'язку, передачі та обміну інформації не забуваючи про безпеку, простоту налаштування та застосування.

РОЗДІЛ 1. Теоретична частина

Аналіз додатків обміну повідомленнями без мережі інтернет

Розглянемо декілька популярних додатків, які вже є на ринку і користуються великою популярністю.

Перший напевно найбільш відомий — це Bridgefy. Bridgefy — мексиканська компанія заснована в Мексиці у 2014 році. Bridgefy — мобільний застосунок, який дозволяє обмінюватися текстовими повідомленнями створюючи сітчасті мережі на основі Bluetooth, які забезпечують пряме з'єднання між пристроями замість того, щоб обробляти інформацію через віддалений сервер. Відповідно дальність передачі для двох користувачів не більше 100 м, проте якщо у цих межах є ще один користувач Bridgefy, то телефон «посередника» стане підсилювачем сигналу збільшить далекодію застосунку. Кількість таких посередників не обмежена, а саме тому й дальність також. Застосунок можна використовувати у двох режимах — приватне та публічне спілкування. Для приватного спілкування, абонент має бути обов'язково одним із ваших телефонних контактів. Якщо ви хочете повідомити інформацію усім навколо, її отримають усі, з чийми телефонами ви маєте зв'язок, в свою чергу вони, як посередники, передадуть її наступним, тим, хто подалі від вас і так далі, в цьому випадку алгоритми Bridgefy розроблені так, щоб повідомлення не дублювались. Технологія Bridgefy заснована на спонтанних сітчастих мережах (mesh network), створених мобільними пристроями. У випадку Bridgefy сітчаста мережа — це група пристроїв, які під'єднані один до одного завдяки з'єднанням Bluetooth.

У будь-якому застосунку для комунікації — безпека один з найважливіших факторів. Раніше, до квітня 2020 року, компанія була звинувачена у небезпідставних прогалинах в безпеці, проте згодом, у жовтні того ж року,

оголосила, що вони прийняли протокол Signal як у своїй програмі, так і в SDK, щоб зберегти інформацію конфіденційною.

Варто зауважити, щоб використовувати додаток без підключення до мережі, перший вхід має бути здійснений при підключенні до інтернету. Воно потрібне для активації унікального програмного забезпечення, завдяки якому застосунок працює, тому подбати про це треба заздалегідь. На рисунку 1 показана можливість писати в публічний чат і за умови, якщо абонент є в списку контактів — надсилати приватне повідомлення.

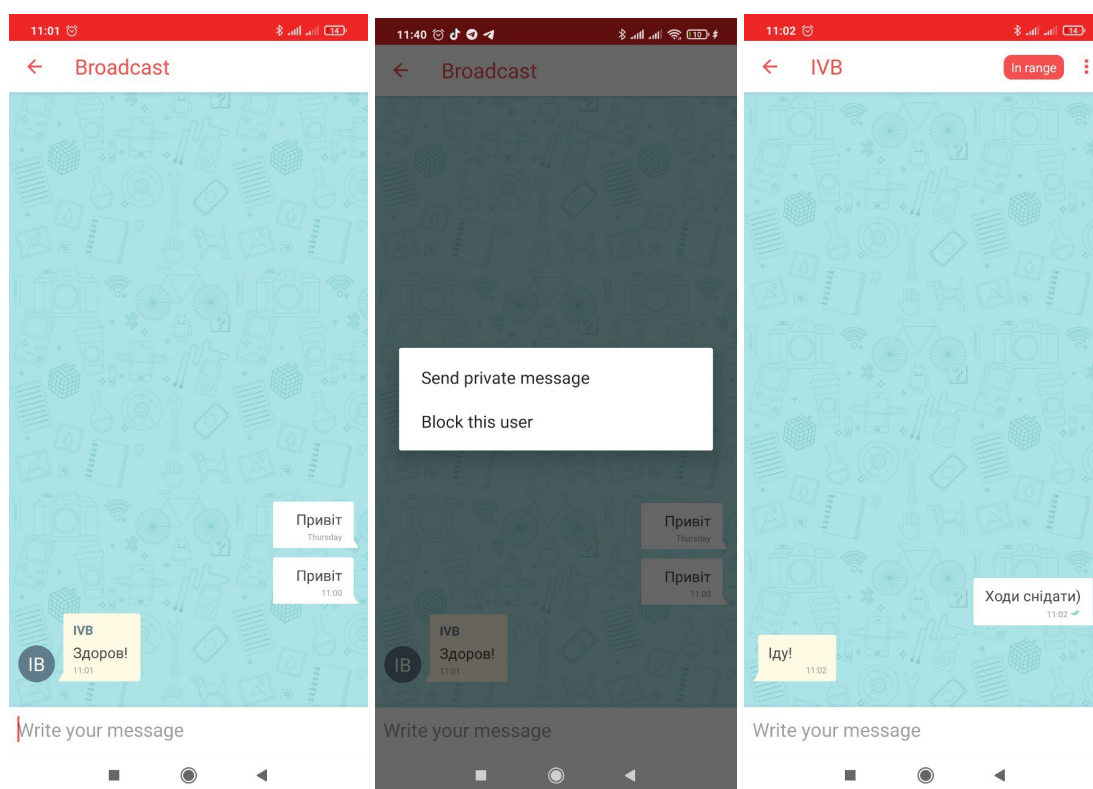


Рисунок 1 — Приклад роботи Bridgefy

Наступний за популярністю Briar, розроблений у 2017 році. Briar — це програмна комунікаційна технологія з відкритим вихідним кодом, призначена для забезпечення безпечного та стійкого однорангового (peer-to-peer) зв'язку без централізованих серверів і мінімальної залежності від зовнішньої інфраструктури. Повідомлення можна передавати через Bluetooth, Wi Fi (будучи підключеними до однієї мережі),

через Інтернет через Тог (можлива комунікація на далекі відстані) або знімний накопичувач, наприклад USB-накопичувачі. Вся комунікація наскрізно зашифрована. Відповідний вміст зберігається в зашифрованому вигляді на пристроях-учасниках. Спосіб додавання контактів і загалом комунікації дещо відрізняється на відміну від попереднього додатку. Якщо абонент знаходиться поруч достатньо просканувати QR-коди один одного. Якщо ж абонент на відстані — за допомогою згенерованого посилання, яке дійсно 48 годин, яке можна передати іншим каналом комунікації. Функціонал також дещо відрізняється. Є можливість створювати приватні групові бесіди, блоги та публічні форуми. Додаток також може підписуватись на оновлення і розсилки RSS. Тобто мати доступ до інформації, навіть якщо з'язок глушать або пошкоджена інфраструктура. Оскільки додаток задумувався, як такий, цільова аудиторія якого включає в себе активістів, журналістів та громадянське суспільство, отже безпека та збереження конфіденційності для нього дуже важлива. Саме тому додатково було додано можливість натискання тривожної кнопки, яка назавжди видаляє ваші особисті дані, контакти та повідомлення. Проте варто пам'ятати, що вони залишаються на носіях ваших співрозмовників. Безпечаними перевагами Vgig є відсутність прив'язки до номеру телефону або електронної пошти, встановлення зразу навіть без підключення до інтернету, вхід в додаток лише через пароль, проте при видаленні застосунку з телефону або якщо ви забули пароль, відновити доступ буде неможливо. Варто розуміти, що ваш 'акаунт' прив'язаний до вашого пристрою. Серед недоліків варто зауважити, що застосунок доступний лише для пристроїв з операційною системою Android, а також у ньому немає можливості надіслати повідомлення всім доступним пристроям, зараз розробники працюють над цим функціоналом.

Третій варіант застосунку, які працюють без підключення до інтернету або мобільного зв'язку — це FireChat, розроблений ще у 2014 році і є першим месенджером, що як і Bridgefy застосовує сітчасті мережі на основі Bluetooth. Розроблявся він для комунікації під час відпочинку на природі або масових заходах, де можуть бути проблеми із мережею, проте він також набув популярності під час мітингів у Гонконзі у 2014 році. На відміну від попередніх додатків, цей дозволяє обмінюватись зображеннями та відео, проте не надто великими. Тут ви можете надсилати повідомлення будь-кому з доступних навколо користувачів, а також є можливість створювати групові чати до 80 людей. Також є глобальний чат, але доступний він лише при підключенні до інтернету.

Отже, ми розглянули найпопулярніші додатки, за допомогою яких ми можемо спілкуватися та передавати інформацію без підключення до інтернету. Кожен з них був розроблений з певною метою, а тому має власні особливості у функціонуванні.

Принципи роботи обміну повідомленнями через Wi Fi та Bluetooth

Сітчаста мережа

При аналізі застосунків, які вже активно використовуються в світі, ми бачимо різні види бездротового зв'язку та передачі повідомлень між різними користувачами без виходу в інтернет. Це може бути спільна мережа Wi Fi, навіть, якщо у ній немає доступу до інтернету, або через Bluetooth: напрямую одноранговий зв'язок (peer-to-peer) або створена сітчаста мережа (mesh-network).

Для початку треба розібратися що таке сітчаста мережа. Сітчаста мережа — це топологія комп'ютерної мережі, в якій кожен вузол передає дані по мережі і виступає в ролі комутатора. Всі вузли на рівних правах

співпрацюють у розподілі даних в мережі, тобто кожен вузол бере участь у передачі даних. Історично склалось, що спочатку вони застосовувались у військовій сфері, сьогодні ця технологія активно застосовується ще у медицині та комерційних проектах. Багато новинок у сфері інтернету речей ґрунтується на ній.

Варто розуміти термінологію. Вузол — пристрій, який вже є частиною мережі. Кожен вузол в свою чергу може складатись з декількох елементів.

Сітчаста мережа на основі Bluetooth (Bluetooth mesh)

Bluetooth Mesh — це сітчаста мережа на основі Bluetooth з низьким енергоспоживанням (BLE) і сумісний з Bluetooth v4.0 і вище. Bluetooth SIG випустила стандарт Bluetooth Mesh 13 липня 2017 року. Її основною ціллю були - збільшення дальності зв'язку мережі, що використовує Bluetooth. Вона дозволяє з'єднуватись зв'язком багато-до-багатьох, тоді як раніше існували лише два типи зв'язку: один-до-одного та один-до-багатьох. Оскільки всі учасники рівноправні, вони можуть транслювати повідомлення на далекі відстані. Хоча Bluetooth Mesh базується на BLE, з'єднання пристроїв дещо відрізняється від звичного. Час сканування та представлення (адвертайзингу) вони використовують для ретрансляції повідомлень іншим пристроям. Кожен вузол описується станом та власними властивостям, але для теми нашого дослідження нас цікавить саме передача повідомлень. Проте, якщо розглядати технологію Bluetooth mesh для інтернету речей, наприклад освітлення в домі, саме стани і властивості відіграють важливу роль у правильному функціонуванні системи. В Bluetooth Mesh вузли посилають повідомлення іншим вузлам для керування їхніми станами, або передачі інформації іншим вузлам і саме другий випадок нас цікавить. Всі повідомлення в сітчастій мережі мають пряму адресуватися. Кожен вузол в мережі має власну унікальну адресу, яка надається йому при додаванні до сітчастої мережі. Проте

існують і групові адреси — адреси, які використовуються, якщо ми хочемо звернутися до конкретної групи чи типу вузлів, це може бути фізичний, або логічний поділ, наприклад вузли-ретранслятори. Також можна звернутись до абсолютно всіх вузлів в мережі.

Існує чотири види вузлів: вузли-ретранслятори, проксі-вузли, дружні вузли та малоспоживаючі вузли. Варто зауважити, що кожен вузол одночасно може належати різним типам, наприклад проксі та ретранслятор. Вузли-ретранслятори — може передавати повідомлення іншим вузлам, на цьому й ґрунтується видовження нашої мережі. Проксі-вузли - ті, хто може запрошувати нові пристрої до сітчастої мережі.

Алгоритм масштабування мережі складається з п'яти кроків.

Перший — розсилка mesh-маячків при адвертайзингу.

Другий — запрошення

Третій — обмін публічними ключами

Четвертий — аутентифікація

П'ятий — поширення даних, після аутентифікації кожен пристрій генерує ключ сесії, використовуючи власний приватний ключ та публічний ключ, отриманий від іншого пристрою. Ключ сесії потім використовується для захисту з'єднання під час передачі додаткових даних забезпечення, там також використовуються ключ мережі, ключ пристрою, параметри безпеки та індивідуальна адреса, яка присвоюється пристроєм, який запросив.

Лише після цього новий пристрій стає частиною сітчастої мережі.

З'єднання через підключення до Wi Fi

Важливо зауважити, що передача повідомлень можлива лише, якщо обидва користувачі під'єднані до однієї і тої самої точки доступу, навіть, якщо в ній немає доступу до інтернету. Уявімо, що пошкоджений інтернет кабель або інфраструктура, яка відповідає за нього, проте живлення та електропостачання є, тоді роутери і маршрутизатори працюватимуть. Коли

пристрої під'єднані до однієї точки доступу Wi Fi, то вони є в одній підмережі. Ця точка доступу грає роль роутера. Коли при підключенні, як клієнт пристрій отримує IP адресу, яку знає роутер також. Таким чином він знає IP адреси усіх, хто підключений до нього. Через спеціальне програмне забезпечення в застосунку і стає можлива комунікація між різними пристроями.

Захист та шифрування даних при обміні через Wi Fi та Bluetooth

Розуміємо, що безпека при обміні повідомленнями в додатках грає важливу роль, а отже в таких, додатка, які ми розглядаємо, приватність та конфіденційність грають важливу роль.

Для початку розглянемо питання безпеки в Bluetooth mesh.

Варто розуміти, що всі повідомлення в середині мережі зашифровані та аутентифіковані, а ключі безпеки можуть мінятися впродовж сесії. Безпека окремо гарантується з трьох різних сторін: мережі, програмного забезпечення та кожного пристрою індивідуально. Саме тому існує три види ключів для кожного з рівнів.

Мережевий ключ є у того, хто вже став частиною сітчастої мережі. З нього добуваються ключ конфіденційності та мережевий ключ шифрування, який дозволяє передавати повідомлення далі, але не розшифрувати його.

Ключ програмного забезпечення використовується для розшифрування та аутентифікації повідомлень на рівні застосунку.

Ключ пристрою належить конкретному пристрою та використовується для захисту зв'язку між пристроями, що не є вузлами даної сітчастої мережі.

Шифрування самих даних в застосунках, які нас цікавлять також є важливим моментом. Розглянемо криптографічний протокол Signal, який на даний момент вважається одним з найбезпечніших.

Протокол Signal — криптографічний протокол, створений компанією Open Whisper Systems для забезпечення наскрізного шифрування голосових викликів, відеодзвінків і миттєвих повідомлень. Його використовують такі популярні додатки, як WhatsApp та Telegram. Наскрізне шифрування — це найбезпечніший спосіб приватного та безпечного спілкування в Інтернеті. Шифруючи повідомлення на обох кінцях розмови, наскрізне шифрування не дозволяє нікому посередині читати приватні повідомлення.

Протокол Signal використовує так звану систему «храповика», яка змінює ключ після кожного повідомлення. В додаток до постійних ключів, він створює колекцію тимчасових пар ключів для кожного користувача. Коли хтось надсилає повідомлення контакту через програму за допомогою протоколу Signal, програма поєднує тимчасові та постійні пари відкритих та закритих ключів для обох користувачів, щоб створити спільний секретний ключ, який використовується для шифрування та розшифровки цього повідомлення. Оскільки для створення цього секретного ключа потрібен доступ до приватних ключів користувачів, він існує лише на двох їх пристроях. А система тимчасових ключів протоколу Signal, які він постійно поповнює для кожного користувача, дозволяє йому згенерувати новий спільний ключ після кожного повідомлення.

Висновки до розділу 1

Отже, ми проаналізували застосунки, які є на ринку та вирішують завдання комунікації без мережі. Розібрались з технологією передачі повідомлень через Bluetooth, сітчасті мережі Bluetooth mesh та через спільну точку доступу Wi Fi. Також розібрались особливостями шифрування даних за допомогою протоколу Signal та захисту їх передачі.

РОЗДІЛ 2. Проектування додатку

Технічне завдання та вимоги до застосунку

Планується розробити мобільний додаток для обміну текстовими повідомленнями в умовах без доступу до мобільної мережі чи інтернету за допомогою сітчастої мережі Bluetooth mesh (аналог Bridgefy, але з певними удосконаленнями).

Цільова аудиторія — люди, які перебувають в місцях, де відсутній звичний зв'язок з тих чи інших причин: просто не має покриття, неполадки в інфраструктурі, перевантаження мережі, спеціальне глушіння зв'язку, і яким потрібно зв'язатись з близькими знайомими або всіма людьми навколо і повідомити певну інформацію. Водночас важлива конфіденційність, тобто можливість використання без прив'язки до номеру телефону або електронної пошти, лише до приладу. Вся інформація зберігається на пристрої користувача і може бути екстрено видалена за потреби.

Вимоги:

- Можливість роботи лише через Bluetooth.
- Зручний та зрозумілий інтерфейс для користувача.
- Легкість реєстрації.
- Швидкість передачі повідомлень.
- Можливість писати повідомлення у публічний чат, його отримають усі вузли сітчастої мережі, в якій перебуває абонент.
- Можливість бачити список усіх доступних абонентів та надсилати їм запит на “дружбу”, лише тоді, коли інший пристрій підтвердив запит, можливе приватне спілкування.
- Можливість створювати групові чати та запрошувати “друзів”, лише власник чату може їх додавати нових учасників.
- Можливість покидати чати.

- Можливість видаляти повідомлення, які автоматично видаляються і з приладу користувача.
- Можливість натискання тривожної кнопки, яка видалить всю інформацію з пристрою і (опціонально) сам додаток.

Опис обраного стеку з аргументами

Було вирішено розробляти на React Native. React Native — це сучасний JavaScript опенсорс фреймворк для написання кросплатформених мобільних додатків створений у 2015 році Facebook. Кросплатформеність дозволяє створювати єдиний проект під IOS і Android. Якраз з цим є проблема у застосунків, які вже є на ринку. Не всі вони навіть є на IOS, а ті, що є, працюють не завжди так, як мають. Багато великих компаній, чийми застосунками ми користуємось щодня написані саме на React Native, серед таких Facebook, Instagram, Skype, Pinterest, Uber, Tesla, SoundCloud, Uber, Discord, Bloomberg, Airbnb і навіть Microsoft OneDrive. React Native відомий своєю легкістю і швидкістю розробки.

Переваги та особливості React Native:

- Одна кодова база для IOS і Android, тобто швидкість розробки збільшується майже вдвічі.
- Завдяки використанню нативних API і модулів, які приєднуються автоматично, розробка швидша й зручна, а користувацький досвід на хорошому рівні.
- Оскільки цей фреймворк став дуже популярним в останні роки, з'явилося багато хороших та якісних рішень для більшості проблем.
- TypeScript — статична типізація зменшує кількість багів і полегшує підтримку проекту.
- Fast Refresh — код миттєво оновлюється, як тільки в нього внесли зміни.

- Користувацький інтерфейс може бути розроблений абсолютно по новому, так, як потрібно вашому застосунку.
- Легкий доступ до апаратної частини пристрою, наприклад Bluetooth (саме те, що ми використовуватимемо в нашому застосунку).

UML діаграми роботи додатку

Діаграма прецедентів

Діаграма прецедентів містить два структурні елементи: учасник та прецедент. В нашому випадку учасник — користувач застосунку, який хоче комунікувати з іншими користувачами, теж акторами. Прецедент — що саме має відбуватись в застосунку. На даній діаграмі зображені прецеденти для Користувача 1. Більшість прецедентів описані у вимогах до застосунку.

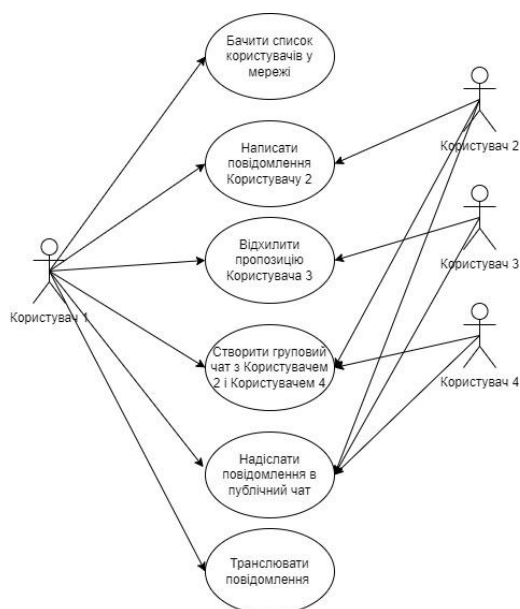


Рисунок 2 — Діаграма прецедентів

Діаграма активностей

Діаграма активностей описує поведінку застосунку в зрозумілому вигляді за допомогою блок-схеми. За її допомогою ми можемо зрозуміти логіку продій та перехід від однієї активності до іншої.

Оскільки наш застосунок не має єдиного сервера і працює без доступу до мережі, всі процеси в нас відбуваються між фізичними пристроями, що приймають всю доступну інформацію. Дана діаграма для користувачів, які вже є частиною мережі.

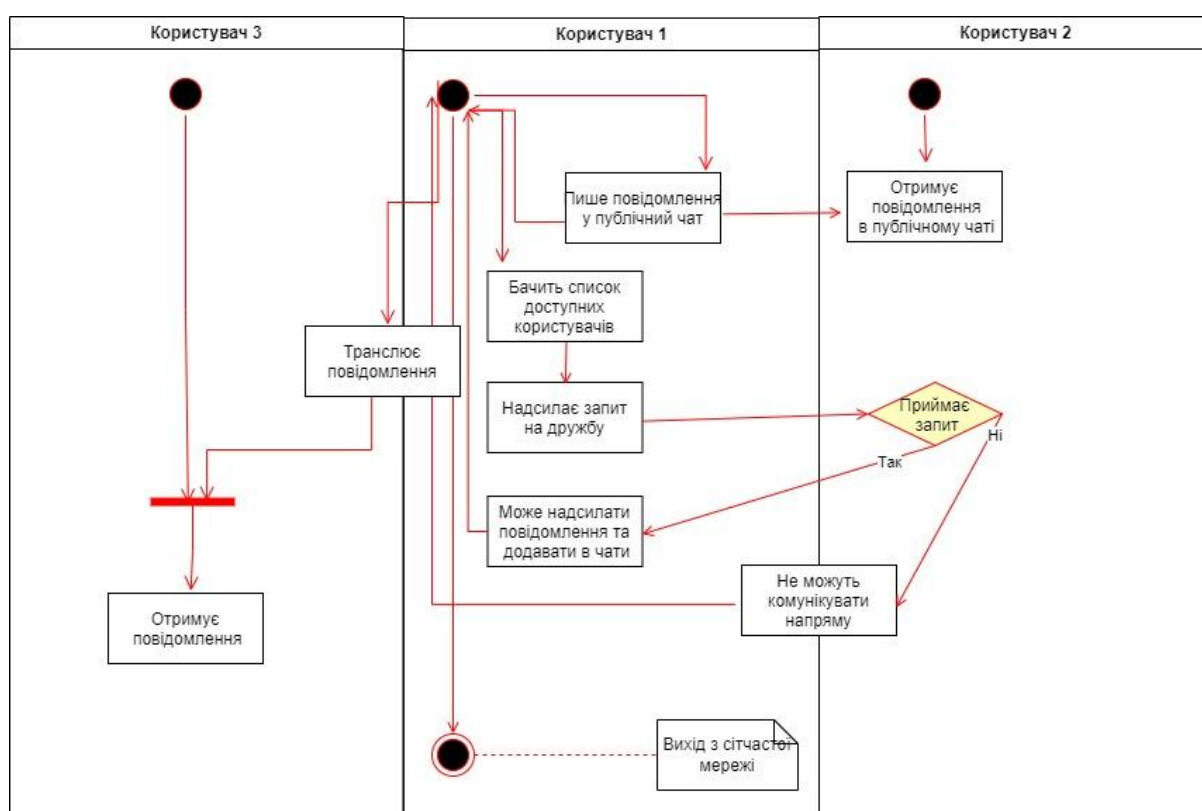


Рисунок 3 — Діаграма активностей

Діаграма послідовності

Діаграма послідовності динамічно показує взаємодію об'єктів в часі.

Бачимо, що основний функціонал — можливість писати в публічний чат будь-коли, так само як і шукати користувачів і спілкуватись в приватних повідомленнях.

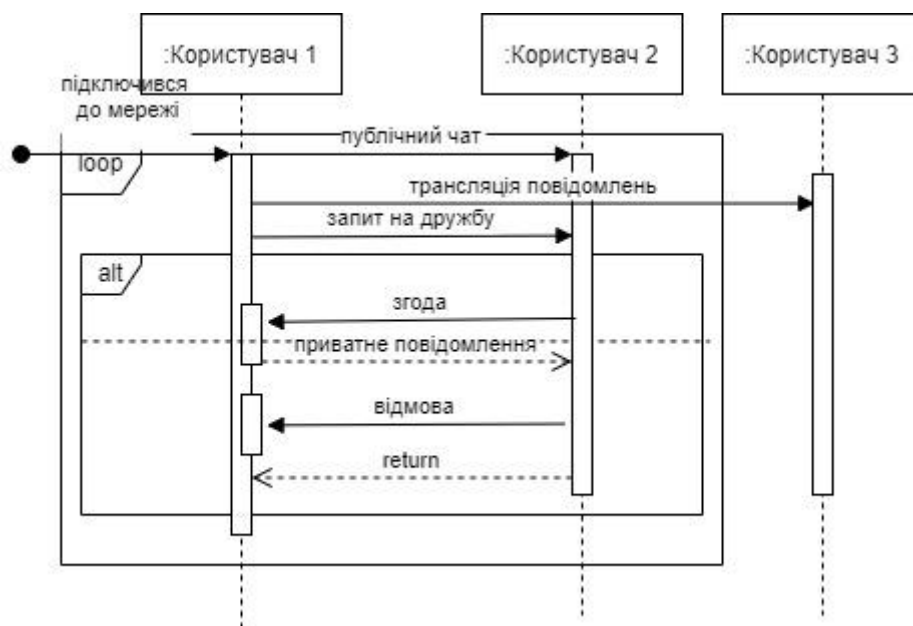


Рисунок 4 — Діаграма послідовності

Діаграми станів

На діаграмі станів показують різні стани об'єкта під роботи і причини через які він змінює їх.

Перш за все треба розуміти, що деякі пристрої виконують роль транслятора, оскільки так працює сітчаста мережа, відповідно до того як вона розділить ролі, тому вони можуть перебувати в декількох станах одночасно.

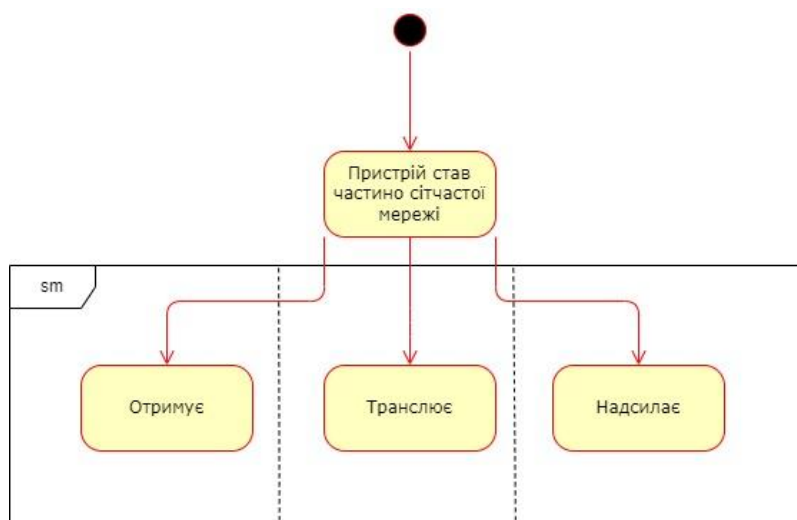


Рисунок 5 — Діаграма станів

Архітектура додатку

Архітектура додатків в React Native

Як вже згадували, React Native дозволяє написання кросплатформених застосунків. А це означає, що основний код бізнес-логіки - спільний та написаний на JavaScript, а кожна з частин проекту IOS і Android бере свої native інструменти.

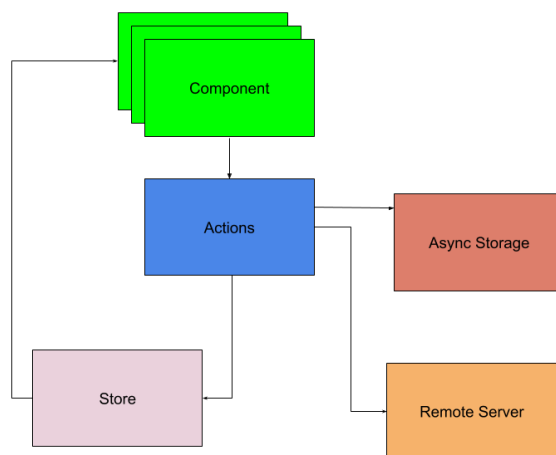


Рисунок 6 — Архітектура додатків на React Native

Стан застосунку зберігається в store. При дії components на actions і відповідно будь-якій зміні в store потрібні частини повторно візуалізуються.

Важливо розуміти, як відбувається зв'язок між спільною бізнес-логікою (JavaScript core) та native інструментами операційної системи пристрою. Це відбувається завдяки так званому React Native Bridge.

Варто зауважити, що робота ґрунтується на трьох основних стовпах:

- Main Thread (або Native Thread) створений операційною системою при відкритті додатку, так званий UI Thread.
- JavaScript Thread, в якому лежить вся логіка.

- Shadow Thead (або Shadow Tree) — створює макет результату від JavaScript Thread і передає в Main Thread для візуалізації.

React Native Bridge — це можливість JavaScript Thread ефективно доступатись до native інструментів за допомогою JSON запитів та відповідей на них, серед відповідей можуть бути і цілі функції.

Архітектура додатку, що розробляється

У випадку нашого застосунку немає централізованого сервера. Вся інформація обробляється і зберігається в базі даних на пристрої користувача-клієнта, тому внутрішня архітектура додатку трішки не схожа до стандартної.

Додаток складається з користувацького інтерфейсу, через який відбувається взаємодія з користувачем, який в свою чергу запускає бізнес-логіку, має доступ до сховища та взаємодіє з іншими пристроями через Bluetooth та Bluetooth mesh.

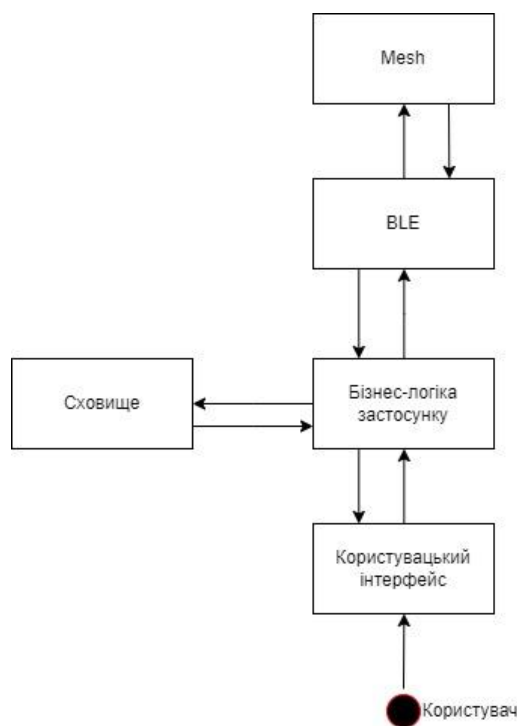


Рисунок 7 — Архітектура додатку

Схема бази даних

Порівняння баз даних для розробки мобільних додатків

Оскільки додаток працює лише офлайн і не має віддаленого сервера, база даних необхідна лише на клієнті. Існує декілька варіантів таких баз даних, що часто використовуються при розробці мобільних додатків, наприклад, Realm та SQLite. Незважаючи на популярність SQLite має деякі обмеження. Перш за все треба розуміти, що SQLite — це система управління реляційною базою даних, а тому запити можуть бути повільними і може бути важко керувати великим набором даних. Realm — це проста у використанні альтернатива SQLite з відкритим вихідним кодом. Основна відмінність між ними полягає в тому, що Realm — це нереляційна база даних (NoSQL), а система керування об'єктно-орієнтованою базою даних і часто використовується для заміни SQLite. Основна її перевага для нас — швидкість та легкість налаштування для роботи на React Native.

База даних для застосунку, що розробляється

Оскільки наш застосунок має в собі не багато елементів, зручно використовувати нескладну базу даних Realm. Дві основні переваги: швидкість та простота в налаштуванні. Оскільки даних і повідомлень в нашому додатку може бути справді багато в залежності від ситуацій, чим швидше база даних може записувати та витягувати інформацію, тим краще.

Нижче наведено схему бази даних, яка матиме лише одну таблицю. Оскільки основне, що ми зберігаємо в додатку це повідомлення, від кого і коли вони прийшли, а також тип цього повідомлення (приватне або публічне).

Тип повідомлення	User ID	Текст повідомлення	Дата та час
------------------	---------	--------------------	-------------

Рисунок 8 — База даних

Висновки до розділу 2

В цьому розділі ми розписали технічне завдання та вимоги до застосунку, обрали та описали фреймворк, на якому писатимемо - React Native. зрозуміли його особливості та переваги. Розробили UML діаграми: преудентів, активностей, послідовностей та станів. Розібрались з особливостями архітектури застосунків на React Native та розробили одну для нашого завдання. Проаналізували популярні бази даних для розробки мобільних додатків та обрали Realm для нашого застосунку.

Висновки

Отже, було проаналізовано застосунки, що вже є на ринку, розглянуто принципи роботи технології Bluetooth mesh, побудовано архітектуру додатку та обрано стек аргументів для подальшої розробки самого додатку для обміну текстовими повідомленнями в умовах відсутності мобільного зв'язку та інтернету. Було ознайомлено з принципами роботи фреймворку React Native та розробки додатків на ньому.

Список літератури

1. <https://bridgefy.me/>
2. <https://briarproject.org/>
3. Мобільний зв'язок та інтернет: якщо зникне, що робити – кілька додатків. Інфографіка – 2022. – [Електронний ресурс] – Режим доступу до ресурсу: https://www.ukrinform.ua/rubric-other_news/3420033-mobilnij-zvazok-ta-internet-akso-znikne-so-robiti-kilka-dodatkiv-infografika.html
4. Diving deep into Briar at XMPP Meetup Berlin – 2021. – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=sKuljekMzTc>
5. Signal Protocol – 2022. – [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Signal_Protocol
6. Mesh-мережі – 2022. – [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Mesh-%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96#%D0%92%D0%B8%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F_Mesh-%D0%BC%D0%B5%D1%80%D0%B5%D0%B6
7. Bluetooth Low Energy: подробный гайд для начинающих. Bluetooth Mesh – 2021. – [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/547620/>
8. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ НА REACT NATIVE – 2022. – [Електронний ресурс] – Режим доступу до ресурсу: <https://brander.ua/what-we-offer/application-development/rozrobka-mobilnykh-dodatkov-na-react-native>
9. Realm vs SQLite: Which Database to Choose in 2022 – 2022. – [Електронний ресурс] – Режим доступу до ресурсу: <https://orangesoft.co/blog/realm-vs-sqlite>