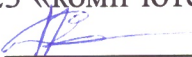


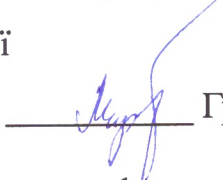
Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

Пояснювальна записка

до випускної кваліфікаційної роботи магістра на тему:
“Методи та засоби безпечного обміну та збереження зображень у
мобільному застосунку”

Виконав студент 2 курсу групи КІм-22-1
спеціальності 123 «комп'ютерна інженерія»
 Пупена Д.І.

Керівник, завідувач кафедри комп'ютерних наук,
доктор філософії

 Граф М.С.

Рецензент, професор кафедри комп'ютерної
інженерії та кібербезпеки, доктор технічних наук,
доцент

 Воротніков В.В.

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп'ютерних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Спеціальність 123 «комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерної інженерії та

кібербезпеки

 Сфіменко А.А.

13 жовтня 2023 року

ЗАВДАННЯ

На випускню кваліфікаційну роботу

Студента: Пупени Дмитра Івановича

(прізвище, ім'я, по батькові)

Тема роботи Методи та засоби безпечного обміну та збереження зображень користувачів у мобільному застосунку

Затверджені наказом університету від "13" жовтня 2023р. № 571/с

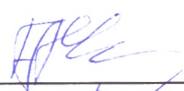
Термін здачі студентом закінченої роботи _____

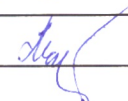
Вихідні дані роботи (зазначається предмет і об'єкт дослідження) Об'єктом дослідження є процес обміну зображень у мобільному додатку та захист цього процесу

Предметом дослідження є методи захисного обміну зображень у мобільних застосунках

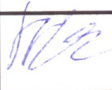
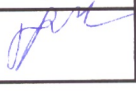
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Пірог О.В., доцент КІ та КБ	2.09.23	2.09.22
2	Пірог О.В., доцент КІ та КБ	6.10.23	6.10.23
3	Пірог О.В., доцент КІ та КБ	4.11.23	4.11.23

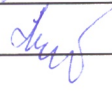
Студент  Пупена Д.І.

Керівник  Граф М.С.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формування технічного завдання. Опрацювання літературних джерел.	2 листопада 2023 - 5 листопада 2023	
2	Проектування інфраструктури додатків	6 листопада 2023 - 12 листопада 2023	
3	Налаштування захисту обміну зображеннями	13 листопада 2023 - 26 листопада 2023	
4	Тестування систем захисту	27 листопада 2023 - 4 грудня 2023	
5	Захист	27 грудня 2023	

Студент  Пупена Д.І.

Керівник  Граф М.С. ,

Реферат

Випускна кваліфікаційна робота магістра складається з проекту мобільних додатків та пояснювальної записки. Пояснювальна записка до випускної роботи містить 51 сторінку

Метою роботи є Створення мобільного застосунку де ми будемо реалізовувати та користуватись безпечним обміном зображень використовуючи можливості сервісів firebase.

В роботі визначено основні завдання на розробку додатків та інфраструктури, проаналізовано інформацію яка буде використана при розробці.

КЛЮЧОВІ СЛОВА: ДОДАТОК, ІНФОРМАЦІЙНА СИСТЕМА, БЕЗПЕЧНИЙ ОБМІН, FIREBASE.

ABSTRACT

The master's thesis consists of a mobile application project and an explanatory note. The explanatory note for the thesis is 51 pages long.

The aim of the work is to create a mobile application where we will implement and utilize secure image exchange using the capabilities of Firebase services.

The thesis defines the main tasks for the development of applications and infrastructure, and analyzes the information that will be used in the development.

KEYWORDS: APPLICATION, INFORMATION SYSTEM, SECURE EXCHANGE, FIREBASE.

Зміст

Вступ	7
Розділ 1 Сучасні технології	9
1.1. Аналіз сучасних технологій мобільних додатків та їхніх особливостей	9
1.2. Використання Firebase у мобільних додатках, зокрема для роботи з зображеннями	17
Розділ 2. Опис мобільного застосунку Whabe	19
2.1. Опис функціоналу додатка.	19
2.2. Використання технологій Swift та Kotlin для написання додатка для різних платформ.	20
2.3. Робота з Firebase: використання сервісів, таких як Firebase Database, Storage, Remote Config, Analytics, Auth, Crashlytics, тощо.	22
2.4. Особливості роботи з сервером, зокрема зберігання та обробка зображень на Firebase Storage.	24
Розділ 3. Методи захисту та безпека в мобільному додатку "Whabe"	26
3.1. Аналіз методів захисту зображень у соціальних мережах та їх можливе застосування в "Whabe".	26
3.2. Аналіз методів захисту спілкування клієнта та сервера та їх можливе застосування в "Whabe".	27
Розділ 4. Дослідження ефективності методів безпеки	33
4.1. Оцінка ефективності використаних методів та технологій захисту в мобільному додатку "Whabe".	33
4.2. Варіанти щодо подальшого підвищення рівня безпеки додатка.	35
Висновки	38
Список використаної літератури:	43

Вступ

Сучасний етап розвитку інформаційних технологій супроводжується неабиякими зрушеннями в сфері взаємодії та обміну інформацією, зокрема, широким застосуванням мобільних додатків. За останні кілька років мобільні застосунки стали невід'ємною частиною нашого повсякденного життя, пропонуючи нові можливості для соціальної взаємодії та спільного відчуття технологічного прогресу.

Однак разом із цим зростає і рівень загроз, пов'язаних із безпекою особистих даних користувачів, зокрема, зображень, які вони обмінюються в соціальних мережах та мобільних додатках. Проблема безпеки обміну та збереження зображень стає нагальною через кілька ключових аспектів.

Ми будемо детально аналізувати та вдосконалення методів та засобів безпечного обміну та збереження зображень користувачів на основі мобільному застосунку "Whabe". Робота спрямована на вирішення актуальної проблеми захисту приватності та конфіденційності в умовах поширеного використання мобільних додатків для обміну візуальним контентом.

Ключові завдання роботи включають:

Огляд літератури та існуючих методів: Провести аналіз наявних підходів до захисту особистих даних у мобільних додатках та соціальних мережах, зокрема у сфері обміну зображеннями.

Аналіз проблем безпеки в додатку "Whabe": Ретельно розглянути функціонал та архітектуру "Whabe" для виявлення можливих вразливостей та недоліків у забезпеченні конфіденційності.

Розробка заходів безпеки: На основі аналізу розробити та імплементувати методи та засоби безпеки, спрямовані на захист особистих фотографій користувачів та голосування у додатку.

Дослідження ефективності заходів: Провести тестування та оцінку запропонованих заходів безпеки для визначення їхньої ефективності та розробки рекомендацій щодо подальшого вдосконалення.

Висновки та рекомендації: Сформулювати основні висновки на основі результатів дослідження та надати рекомендації для подальшого вдосконалення безпеки у мобільних додатках для обміну зображеннями.

Мета роботи полягає в створенні практичних та ефективних рекомендацій для розробників мобільних додатків, спрямованих на підвищення рівня захисту особистих даних користувачів у сфері обміну та збереження зображень.

1. Зростання кількості користувачів мобільних додатків:

З популярністю мобільних застосунків зростає і кількість користувачів, які обмінюються зображеннями. Це робить інформацію вразливою перед різними загрозами, такими як несанкціонований доступ та можливість витоку особистих даних.

2. Специфіка соціальних додатків:

Багато мобільних додатків, зокрема соціальні мережі, спеціалізуються на обміні візуальним контентом, що робить їх особливо привабливими для атак, спрямованих на витіснення конфіденційних зображень користувачів.

3. Загроза приватності та конфіденційності:

Відсутність ефективних методів захисту може призвести до порушення приватності та конфіденційності користувачів, викладаючи їх особисті дані на ризик неправомірного використання.

4. Специфіка додатка "Whabe":

Враховуючи особливості функціоналу "Whabe", де користувачі вибирають між фотографіями, стає важливою розробка методів безпечного голосування та зберігання вибраних зображень.

Актуальність обраної теми полягає в тому, що безпека обміну та збереження зображень у мобільних додатках стає ключовою вимогою для забезпечення надійності та захисту особистої інформації користувачів. Розробка ефективних методів та засобів стає критичною для виведення додатків, подібних "Whabe", на новий рівень безпеки та довіри споживачів.

Розділ 1 Сучасні технології

1.1. Аналіз сучасних технологій мобільних додатків та їхніх особливостей

1. Мови програмування: Swift та Kotlin

У сучасному світі мобільних додатків мови програмування відіграють ключову роль у розробці ефективних та продуктивних застосунків. Swift та Kotlin - це мови, спеціально створені для розробки iOS та Android додатків відповідно. Обидві мови надають розробникам ряд переваг, сприяючи швидкому та ефективному процесу розробки.

Переваги:

Швидкість розробки: Swift та Kotlin вражають своєю простотою синтаксису та високим рівнем абстракції, що полегшує створення функціональних додатків.

Інтеграція з платформою: Обидві мови повністю інтегруються з відповідними платформами (iOS та Android), забезпечуючи оптимальну роботу додатків на пристроях.

Проблеми:

Неуніфікованість: Розробка для обох платформ може призвести до проблеми неуніфікованості, оскільки Swift використовується для iOS, а Kotlin - для Android. Це може вимагати додаткових зусиль для утримання та підтримки коду на обох мовах.

2. Фреймворки для кроссплатформенної розробки: React Native та Flutter

У світі розробки мобільних додатків, де час - це критичний ресурс, фреймворки для кроссплатформенної розробки стають невід'ємною частиною. React Native та Flutter - це два популярних інструменти, які дозволяють розробникам зекономити час та ресурси, пишучи код один раз для обох платформ.

React Native:

Переваги:

Ефективність розробки: Заснований на JavaScript, React Native дозволяє використовувати один і той самий код для iOS та Android, що спрощує та

прискорює процес розробки.

Широка спільнота: Велика та активна спільнота розробників React Native забезпечує підтримку, розвиток та різноманітні розширення для фреймворку.

Flutter:

Переваги:

Висока продуктивність: Використовуючи мову програмування Dart, Flutter компілює високоефективний код для обох платформ, забезпечуючи відмінну продуктивність.

Широкий вибір віджетів: Flutter пропонує розгалужений набір готових віджетів для створення різноманітних та привабливих інтерфейсів.

Проблеми:

Обмежена підтримка нативних функцій: Деякі нативні функції

3. Серверна частина

При розробці мобільних застосунків важливо визначитися з вибором серверної архітектури та послуг для забезпечення ефективного обміну даними. Існує кілька типів серверів, які зазвичай використовуються в мобільних додатках.

Види серверів:

Фізичні сервери:

Переваги:

Повний контроль над апаратною інфраструктурою.

Підходить для великих проектів з великим обсягом обчислень.

Недоліки:

Високі витрати на утримання та обслуговування.

Складно масштабувати.

Віртуальні сервери:

Переваги:

Легше масштабування та управління ресурсами.

Економія в порівнянні з фізичними серверами.

Недоліки:

Менший рівень контролю порівняно із фізичними серверами.

Хмарні сервери:

Переваги:

Висока масштабованість та гнучкість.

Платите тільки за використані ресурси.

Недоліки:

Можливі проблеми з конфіденційністю даних.

Firebase:

Firebase - це мобільна платформа, яка надає ряд сервісів для розробки мобільних та веб-додатків. Розглянемо переваги та недоліки використання Firebase.

Одні з сервісів Firebase:

Realtime database:

Миттєвий обмін даними між клієнтами.

Використовується збереження та маніпуляції з даними, як база даних.

Authentication:

Проста інтеграція системи аутентифікації.

Підтримка різних методів аутентифікації.

Хмарне сховище (Cloud Storage):

Зберігання та управління файлами.

Легка інтеграція з іншими сервісами Firebase.

Недоліки Firebase:

Вартість:

За великим обсягом даних може виникнути висока вартість.

Масштабованість:

Для деяких великих проєктів може бути не досить масштабованим.

Обмежені можливості SQL:

Для тих, хто звик до SQL, може виникнути період адаптації.

Обираючи сервер для мобільного застосунку, слід ретельно враховувати потреби проєкту, його масштаб та бюджет, щоб забезпечити оптимальну архітектуру та продуктивність.

У сучасному світі, де обмін мультимедійним контентом став невід'ємною частиною взаємодії в соціальних мережах, питання безпеки зображень стає критичним. Розглянемо існуючі методи безпечного обміну та збереження зображень у соціальних мережах, визначимо їх переваги та недоліки.

Шифрування даних:

Шифрування даних є ключовим методом для забезпечення конфіденційності та безпеки обміну зображень у мобільних застосунках. Цей метод полягає в застосуванні спеціальних алгоритмів для перетворення інформації в незрозумілу форму, яку можна читати тільки з використанням правильного ключа шифрування.

Переваги шифрування даних:

Захист конфіденційності: Шифрування гарантує, що лише авторизовані користувачі можуть отримати доступ до зображень. Навіть якщо дані потраплять у руки зломисників, вони залишаються незрозумілими без відповідного ключа.

Зменшення ризику несанкціонованого доступу: Використання шифрування важко зламати, зменшуючи ймовірність несанкціонованого доступу до особистої інформації користувачів.

Недоліки шифрування даних:

Обчислювальне навантаження: Шифрування може збільшити навантаження на обчислювальний ресурс, особливо при великому обсязі даних, що може вплинути на продуктивність додатків.

Управління ключами: Ефективне управління ключами шифрування стає ключовим аспектом для забезпечення безпеки. Втрата ключа може призвести до втрати доступу до даних.

Приклад використання шифрування даних:

Signal: Один з найпопулярніших месенджерів, Signal, використовує end-to-end шифрування для всіх типів комунікації, включаючи обмін зображеннями. Кожен етап передачі даних зашифрований, забезпечуючи високий ступінь конфіденційності.

Шифрування даних є потужним і надійним методом захисту конфіденційності, але важливо усвідомлювати його витрати на ресурси та

забезпечувати правильне управління ключами для максимальної ефективності та безпеки.

Переваги:

Забезпечує конфіденційність даних під час передачі.

Зменшує ризик несанкціонованого доступу.

Недоліки:

Призводить до збільшення обчислювального навантаження.

Потребує ефективного керування ключами шифрування.

Приклад:

Signal, який використовує end-to-end шифрування для захисту приватності користувачів.

Watermarking:

Watermarking — це метод захисту зображень шляхом внесення на них спеціальних ідентифікуючих знаків або зображень, що служать як "водяні знаки". Це може бути текстова інформація, логотип чи інші унікальні елементи, які вбудовуються в саме зображення.

Переваги Watermarking:

Додає ідентифікатор авторства: Watermark надає можливість встановлення авторства на зображення, яке може бути важливим у випадку виникнення питань про права власності.

Виявлення нелегітимного використання: Водяні знаки допомагають виявити випадки нелегітимного використання, оскільки вони залишаються на зображенні навіть після спроби його копіювання.

Недоліки Watermarking:

Зменшення якості зображення: Додавання водяних знаків може зменшити якість зображення, особливо якщо вони розташовані в видимих частинах.

Обхід водяних знаків: Зловмисники можуть вирішити видалити чи обійти водяні знаки, знижуючи ефективність заходів захисту.

Приклад використання Watermarking:

Shutterstock: Один із найбільших сервісів стокових фотографій, Shutterstock,

застосовує водяні знаки на зображеннях. Це допомагає встановити авторство та захищати від нелегітимного використання своїх фотографій.

Watermarking — це ефективний метод для захисту авторських прав та виявлення нелегітимного використання зображень, але важливо збалансувати його використання так, щоб не порушити якість контенту.

Переваги:

Додає ідентифікатор авторства до зображення.

Допомагає у виявленні та відслідковуванні нелегітимного використання.

Недоліки:

Може зменшити якість зображення.

Може бути обхідно, якщо зловмисники вирішать видалити водяний знак.

Приклад:

Shutterstock використовує водяні знаки для захисту авторських прав на фотографії.

Права доступу:

Метод прав доступу полягає в регулюванні та обмеженні доступу до зображень на різних рівнях. Це включає в себе установку прав для користувачів або груп користувачів щодо перегляду, редагування, чи завантаження зображень.

Переваги Прав доступу:

Гнучкість управління доступом: Права доступу дозволяють точно налаштовувати, які користувачі мають доступ до конкретних зображень та які операції вони можуть виконувати.

Захист особистої інформації: Застосування прав доступу забезпечує захист особистої інформації користувачів, уникнення несанкціонованого доступу та використання.

Недоліки Прав доступу:

Складність управління: У складних системах прав доступу може виникати проблема складності управління, особливо коли маємо велику кількість користувачів та зображень.

Потенційні вразливості: Якщо не враховувати всі можливі вектори атак або якщо настройки не встановлені належним чином, система прав доступу може бути вразливою.

Приклад використання Прав доступу:

Facebook: Соціальна мережа Facebook використовує складну систему прав доступу, дозволяючи користувачам встановлювати налаштування приватності для своїх альбомів та фотографій. Це дозволяє обмежити доступ до зображень лише для обраних користувачів або груп.

Використання прав доступу є важливим елементом для забезпечення безпеки та конфіденційності зображень у соціальних мережах. Важливо правильно конфігурувати та управляти цими правами для максимальної ефективності.

Переваги:

Користувачі можуть встановлювати обмеження на доступ до своїх зображень. Збільшує контроль над приватністю.

Недоліки:

Невідповідальне управління правами може призвести до небезпеки.

Обмеження можуть бути обійдені в деяких випадках.

Приклад:

Instagram дозволяє користувачам встановлювати рівень приватності своїх фотографій.

Аналіз метаданих:

Метадані — це інформація, яка надається разом із зображенням і містить різноманітні дані, такі як час зйомки, географічні координати, модель камери тощо. Аналіз метаданих зображень є важливим елементом забезпечення безпеки та приватності користувачів у соціальних мережах.

Переваги Аналізу метаданих:

Ідентифікація джерела: Метадані можуть містити інформацію про джерело зображення, таке як модель камери чи пристрою, що може бути корисним для визначення його автентичності.

Геотегування для взаємодії: Географічні координати можуть служити для

покращення соціального взаємодії, але важливо управляти доступом до цієї інформації.

Недоліки Аналізу метаданих:

Порушення приватності: Неправильне управління метаданими може вести до порушення приватності, особливо коли вони містять конфіденційні дані.

Спотворення даних: Зміна метаданих може бути використана для спотворення даних або приховування дійсної інформації.

Приклад використання Аналізу метаданих:

Instagram: При завантаженні фотографій у Instagram, користувачі можуть вибрати, чи дозволити виводити географічні координати, пов'язані із зображеннями. Це надає можливість взаємодіяти з контентом, пов'язаним із конкретними місцями, але також потребує обережності щодо конфіденційності.

Аналіз метаданих зображень є важливою частиною стратегії безпеки у соціальних мережах, спрямованою на забезпечення взаємодії та захист особистої інформації користувачів.

Переваги:

Допомагає виявляти та вивчати ризики безпеки.

Забезпечує можливість відновлення втрачених або видалених зображень.

Недоліки:

Метадані можуть бути використані для створення стежачої інформації.

Деякі платформи можуть збирати та використовувати метадані для маркетингових цілей.

Приклад:

Adobe Lightroom дозволяє переглядати та редагувати метадані фотографій.

Методи безпечного обміну та збереження зображень у соціальних мережах представляють важливу складову для забезпечення конфіденційності та захисту особистої інформації користувачів. За вивченням різноманітних підходів та технік можна зробити кілька ключових висновків.

Шифрування даних є ефективним засобом для захисту зображень від несанкціонованого доступу. Важливо враховувати вибір алгоритмів шифрування та управління ключами для максимізації безпеки.

Водяні знаки відіграють важливу роль у протидії крадіжкам та незаконному використанню зображень. Розробка та використання унікальних водяних знаків дозволяє ідентифікувати авторство та джерело зображень.

Ефективне використання прав доступу дозволяє користувачам контролювати, хто має доступ до їхніх зображень. Належне конфігурування та управління правами є ключовим для забезпечення конфіденційності.

Ретельний аналіз метаданих дозволяє визначити джерело та інші важливі відомості про зображення. Це може бути використано для виявлення автентичності та управління географічним контекстом.

Загальний висновок полягає в тому, що безпека зображень у соціальних мережах вимагає комплексного підходу, включаючи застосування різних технік та стратегій. Забезпечення конфіденційності та захист особистої інформації користувачів визначається правильним використанням інструментів та технологій, а також усвідомленням ризиків та потенційних загроз.

1.2. Використання Firebase у мобільних додатках, зокрема для роботи з зображеннями

Firebase, як інтегрована платформа від Google, надає зручні та потужні інструменти для розробки мобільних додатків. У цьому розділі ми детально розглянемо використання Firebase для роботи з зображеннями в мобільних додатках, зокрема в контексті застосунку "Whabe."

Огляд Firebase

Firebase включає широкий спектр сервісів, і для нашого прикладу ми розглянемо на таких сервісах, як Firebase Storage, Firebase Realtime Database, та Firebase Authentication.

Firebase Storage:

Збереження зображень: Firebase Storage надає простий та ефективний спосіб зберігання зображень у хмарі. Завдяки цьому сервісу, ми можемо легко керувати завантаженням та завантаженням зображень, а також забезпечити доступ до них з різних пристроїв.

Інтеграція з мобільним додатком: Firebase Storage дозволяє нашому додатку безперебійно взаємодіяти з віддаленим сховищем зображень, надаючи необхідну гнучкість та швидкодійність.

Firebase Realtime Database:

Збереження метаданих: Firebase Realtime Database може використовуватися для зберігання метаданих про зображення, таких як дата публікації, власник, кількість голосів тощо. Це забезпечить швидкий та узгоджений доступ до важливих даних.

Синхронізація в реальному часі: Однією з переваг Firebase Realtime Database є можливість отримувати оновлення даних в реальному часі. Це стає особливо корисним для відслідковування змін в зображеннях та їхніх метаданих.

Firebase Authentication:

Захист доступу: Firebase Authentication дозволяє нам забезпечити безпеку нашого застосунку, використовуючи різні методи автентифікації, такі як

електронна пошта, Google або Facebook. Це допомагає уникнути несанкціонованого доступу до зображень та інших особистих даних користувачів.

Використання Firebase у застосунку "Whabe":

У нашому додатку "Whabe," Firebase використовується для ефективного управління зображеннями та забезпечення їх безпеки. Кожне завантажене зображення зберігається в Firebase Storage, а його метадані, такі як голоси та коментарі, у Firebase Realtime Database.

Firebase Authentication гарантує, що тільки автентифіковані користувачі мають доступ до функціоналу додатку, а їхні дані залишаються конфіденційними. Це дозволяє створити безпечне та надійне середовище для обміну та зберігання зображень.

Переваги та Виклики:

Використання Firebase у застосунку "Whabe" дозволяє забезпечити високу продуктивність, надійність та зручність у роботі з зображеннями. Однак важливо враховувати можливі виклики, такі як обмежена масштабованість при великому обсязі даних або необхідність додаткових заходів забезпечення безпеки для даних користувачів.

Розділ 2. Опис мобільного застосунку Whabe

2.1. Опис функціоналу додатка.

Додаток "Whabe" представляє собою українську соціальну платформу, спрямовану на спрощення процесу вибору, дозволяючи користувачам порівнювати та голосувати за найкращі фотографії. Розглянемо детально його функціонал та особливості.

Публікація фотографій:

Користувачі можуть легко завантажувати свої фотографії на платформу, використовуючи простий та інтуїтивно зрозумілий інтерфейс додатка.

Голосування за фотографії:

Основна функція "Whabe" - голосування за фотографії. Користувачі можуть виражати свої уподобання та визначати, яка фотографія є найкращою, обираючи

одну з двох опцій.

Додаток має всі функції які мають бути у соціальній мережі:

Взаємодія: Додаток пропонує можливість підписуватися на інших користувачів, обмінюватися коментарями та взаємодіяти в чатах.

Розширена соціальна мережа: "Whabe" надає користувачам можливість не лише порівнювати фотографії, але й будувати свою соціальну мережу за інтересами.

Механізм визначення переможців:

На підставі голосів користувачів формується рейтинг фотографій, і переможці отримують визнання у спільноті.

Технічні аспекти:

Мови програмування: Додаток написаний на Swift для iOS-платформи та Kotlin для Android, що забезпечує високу продуктивність та швидкодію.

Backend на Firebase: Зображення зберігаються в Firebase Storage, а дані - в Realtime Database. Використання Firebase Authentication дозволяє забезпечити безпеку та автентифікацію користувачів.

Додаток "Whabe" є має оригінальний та інноваційний підхід до соціальної мережі. Голосування за фотографії, взаємодія та будівлення власної спільноти роблять його цікавим та захоплюючим для користувачів. Використання технологій Swift, Kotlin та Firebase дозволяє досягти високої продуктивності та забезпечити безпеку особистих даних. В цілому, "Whabe" є унікальним інструментом для тих, хто бажає не лише ділитися своїми зображеннями, але й отримувати визнання від спільноти.

Посилання на додаток: <https://apps.apple.com/ua/app/whabe/id1628410216>
https://play.google.com/store/apps/details?id=com.whabelc.whabe&hl=en_US

2.2. Використання технологій Swift та Kotlin для написання додатка для різних платформ.

Додаток "Whabe" використовує мови програмування Swift для iOS та Kotlin для Android. Розглянемо, як ці технології сприяють розробці додатка для різних платформ.

Swift для iOS:

Переваги використання Swift:

Швидкість та Ефективність: Swift визначається високою швидкістю виконання завдяки вбудованим оптимізаціям та низькому рівню абстракції, що робить його відмінним вибором для розробки продуктивних додатків.

Безпека та Надійність: Типізація Swift дозволяє виявляти та виправляти помилки на етапі компіляції, що підвищує надійність додатків та спрощує їх підтримку.

Інтероперабельність з Objective-C: Swift легко інтегрується з існуючим кодом на Objective-C, що дозволяє використовувати бібліотеки та фреймворки, створені для попередніх версій iOS.

Активна Спільнота та Підтримка: Швидкий розвиток та підтримка Swift з боку Apple забезпечують актуальність та інновації в екосистемі iOS.

Легкий Синтаксис: Синтаксис Swift є простим та лаконічним, що робить код зрозумілим та його легко читати.

Інтеграція з платформою: Мова програмування Swift ідеально інтегрується з функціоналом iOS, забезпечуючи високий рівень продуктивності та оптимальну роботу додатка на пристроях Apple.

Недоліки використання Swift:

Недоступність на Інших Платформах: Swift розроблена виключно для платформи iOS, тому використання її для розробки додатків для інших платформ може виявитися важкодоступним.

Нестабільність API: Оскільки мова була випущена нещодавно, деякі API можуть бути менш стабільними та піддаватися змінам у нових версіях мови.

Великий Розмір Вихідного Коду: Додатки, написані на Swift, можуть мати

більший розмір вихідного коду порівняно з додатками, написаними на Objective-C.

Помірна Підтримка для Старших Версій iOS: Нові функції Swift можуть бути недоступні для старших версій iOS, що ускладнює підтримку старих пристроїв

Kotlin для Android:

Сумісність з Java: Kotlin повністю сумісний з Java, що дозволяє розробникам використовувати існуючий Java-код, покращуючи його та додаючи нові функції.

Безпека та Чистий Синтаксис: Kotlin пропонує безпеку типів, що робить код більш надійним. Його синтаксис чистий та лаконічний, спрощуючи написання коду.

Короткий Код: Kotlin може значно скоротити кількість коду порівняно з Java, завдяки власним функціям та удосконаленням.

Розширена Функціональність: Мова надає багато функціональних можливостей, включаючи лямбда-вирази, функції вищого порядку та інші конструкції, що полегшують розробку.

Офіційна Підтримка Google: Kotlin став офіційною мовою програмування для розробки Android-додатків, отримавши підтримку від Google.

Недоліки використання Kotlin:

Час Навчання: Для розробників, які раніше працювали з Java, може знадобитися час для вивчення нових конструкцій та особливостей Kotlin.

Потенційна Вартість Міграції: Перехід з Java на Kotlin може вимагати часу та ресурсів, особливо для великих проєктів.

Неідеальна Інтеграція в Інші Мови: Інтеграція Kotlin з іншими мовами може бути менш ефективною, особливо порівняно з іншими мовами, такими як Java.

Потрібен Kotlin Runtime: Додаток на Kotlin може потребувати включення Kotlin Runtime, що може збільшити розмір додатку.

Загальні переваги використання Swift та Kotlin:

Обидві мови програмування мають ефективне керування пам'яттю та

ресурсами, що забезпечує стабільну роботу додатка без перебоїв.

Swift та Kotlin надають широкі можливості для реалізації різноманітних функціональних можливостей додатка.

2.3. Робота з Firebase: використання сервісів, таких як Firebase Database, Storage, Remote Config, Analytics, Auth, Crashlytics, тощо.

Firebase Database:

Опис: Firebase Database - це реальний час та облікова база даних, яка дозволяє зберігати та синхронізувати дані між різними пристроями.

Переваги:

Реальний час: Зміни в базі даних відображаються миттєво на всіх пристроях.

Простота використання: Легко інтегрується з Android застосунком, надаючи зручний API для роботи з даними.

Недоліки:

Вартість: Додаткові функції можуть вимагати платних тарифів.

Firebase Storage:

Опис: Firebase Storage - це хмарний сховище для зберігання та обміну медіафайлами, такими як зображення та відео.

Переваги:

Простота завантаження: Легко завантажувати та отримувати файли з Firebase Storage.

Інтеграція з іншими сервісами: Пов'язано з Firebase Database для зручного управління файлами.

Недоліки:

Обмеження обсягу: Безкоштовний тариф може мати обмеження по обсягу збережених файлів.

Firebase Remote Config:

Опис: Firebase Remote Config - це сервіс для динамічного конфігурування параметрів додатка без потреби в оновленні.

Переваги:

Швидке оновлення: Зміни в конфігурації негайно відображаються на пристроях.

Таргетинг: Можливість налаштовувати конфігурації для різних користувацьких груп.

Недоліки:

Потребує підключення до Інтернету: Для отримання оновлень конфігурації необхідне Інтернет-підключення.

Firebase Analytics:

Опис: Firebase Analytics - сервіс для аналізу користувацької активності та збору даних для поліпшення додатка.

Переваги:

Великий обсяг даних: Забезпечує детальну статистику використання додатка.

Інтеграція з іншими Firebase сервісами: Аналізує дані щодо взаємодії користувачів з різними аспектами додатка.

Недоліки:

Потребує дотримання правил конфіденційності: Збір та аналіз особистих даних повинен дотримуватися відповідних нормативів.

Firebase Auth:

Опис: Firebase Auth - сервіс для аутентифікації користувачів за допомогою різних методів, таких як електронна пошта, Google, Facebook тощо.

Переваги:

Різні методи аутентифікації: Надає різноманітні способи для легкої ідентифікації користувачів.

Захист персональних даних: Забезпечує безпеку під час обміну конфіденційною інформацією.

Недоліки:

Залежність від сторонніх служб: Для деяких методів аутентифікації може бути необхідна реєстрація в сторонніх службах.

Firebase Crashlytics:

Опис: Firebase Crashlytics - сервіс для виявлення та відстеження кешів додатка для підвищення його стабільності.

Переваги:

Негайне сповіщення: Повідомляє про крах додатка негайно після виникнення.

Аналіз причин: Надає відомості про причини та обставини кешів.

Недоліки:

Залежність від активності користувачів: Деякі кеші можуть залишитися невиявленими, якщо вони не відбуваються часто.

2.4. Особливості роботи з сервером, зокрема зберігання та обробка зображень на Firebase Storage.

Firebase Storage:

Firebase Storage виступає як розширення облачного сховища для медіафайлів у додатку "Whabe". Його використання має ряд переваг та особливостей.

Переваги:

Масштабованість: Firebase Storage гарантує високу масштабованість для зберігання зображень, забезпечуючи оптимальну продуктивність при будь-якому розмірі користувацької бази.

Зручний доступ: Легкий доступ до зображень забезпечує ефективну роботу із завантаженням, відображенням та обміном контентом у додатку.

Інтеграція з іншими Firebase сервісами: Firebase Storage легко інтегрується з іншими сервісами Firebase, такими як база даних та аналітика.

Недоліки:

Вартість: При великому обсязі даних може виникнути проблема з вартістю зберігання в Firebase Storage.

Обмеження функцій обробки: Обробка зображень на стороні сервера може бути обмеженою, і для деяких завдань може знадобитися додатковий backend.

Зберігання та обробка зображень:

Однією з ключових функціональних особливостей "Whabe" є ефективно

зберігання та обробка зображень. Цей процес включає в себе кілька етапів.

Етапи обробки зображень:

Завантаження на сервер: Коли користувач додає нове зображення, воно завантажується на сервер у Firebase Storage.

Оптимізація розміру: Зображення може бути автоматично оптимізоване для зменшення обсягу даних та підвищення швидкодії.

Створення ескізів: Для прискорення завантаження і покращення взаємодії з користувачем можуть створюватися ескізи або попередні перегляди зображень.

Переваги:

Швидке завантаження: Збереження зображень на Firebase Storage дозволяє ефективно завантажувати їх у додаток, забезпечуючи швидку відповідь.

Оптимізація обсягу даних: Зменшення розміру файлів сприяє економії трафіку та поліпшує продуктивність.

Недоліки:

Втрата якості: Оптимізація розміру може призвести до втрати деякої якості зображення, що важливо враховувати при виборі рівня стиснення.

Висновок:

Використання Firebase Storage у поєднанні з ефективною обробкою зображень гарантує надійну та продуктивну роботу мобільних додатків і "Whabe", як приклад. Розглянуті переваги та недоліки допомагають зрозуміти, як удосконалити взаємодію з сервером та оптимізувати роботу з медіафайлами.

Розділ 3. Методи захисту та безпека в мобільному додатку "Whabe"

3.1. Аналіз методів захисту зображень у соціальних мережах та їх можливе застосування в "Whabe".

В даному розділі обговоримо методи та стратегії, які використовують мобільний додаток для забезпечення безпеки користувачів та захисту їхніх особистих даних. Розглянемо важливі аспекти, такі як автентифікація, захист від

несанкціонованого доступу та механізми реагування на інциденти.

Шифрування зображень:

Одним з ключових елементів безпеки є шифрування зображень. Цей процес забезпечує захист конфіденційності та цілісності зображень, які користувачі обмінюються у додатку.

Застосовані методи:

AES-шифрування (Advanced Encryption Standard): один з найбільш надійних стандартів симетричного шифрування - AES. Кожне зображення шифрується унікальним ключем, що забезпечує високий рівень безпеки.

Процес шифрування:

При завантаженні фотографії на сервер, вона автоматично шифрується сильним ключем AES.

Ключ шифрування унікальний для кожного файлу, забезпечуючи індивідуальний захист для кожного зображення.

Переваги використання AES:

Надійність та стійкість:

AES вважається одним з найбільш безпечних алгоритмів шифрування, здатний витримати атаки із застосуванням потужних обчислювальних ресурсів.

Швидкодія:

Симетричне шифрування, таке як AES, забезпечує швидкість обробки, що важливо для реагування на запити користувачів у реальному часі.

Унікальний ключ для кожного зображення:

Один і той самий ключ не використовується для шифрування різних зображень. Це запобігає ситуаціям, коли розкриття ключа одного файлу може призвести до доступу до інших.

Переваги індивідуальних ключів:

Додатковий рівень безпеки:

Кожне зображення має унікальний ідентифікатор, що підвищує рівень безпеки шляхом ускладнення атак, спрямованих на отримання ключів.

Моніторинг доступу та аудит:

Система має вести журнал доступу до зашифрованих зображень та аудиту подій, пов'язаних із розшифруванням.

Моніторингові заходи:

Журналювання:

Запис подій, пов'язаних із доступом до зашифрованих зображень, з метою виявлення несанкціонованого доступу.

Аудит ключів:

Перевірка та аудит використаних ключів для виявлення будь-яких unauthorized modifications.

3.2. Аналіз методів захисту спілкування клієнта та сервера та їх можливе застосування в "Whabe".

SSL-захист:

SSL (Secure Socket Layer) - це протокол, який забезпечує захищену комунікацію між користувачем та сервером. Використання SSL має кілька ключових аспектів безпеки.

Застосовані методи:

Шифрування даних в транспортному шарі:

Усі дані, які передаються між мобільним додатком та сервером, шифруються, завдяки чому вони залишаються конфіденційними під час транзиту через мережу.

Процес шифрування:

Початок з'єднання:

Коли клієнт і сервер встановлюють з'єднання, вони домовляються про використання протоколу SSL/TLS.

SSL-розширення:

Клієнт та сервер обмінюються даними, включаючи параметри шифрування та ключі для безпечного обміну інформацією.

Ключі для шифрування:

Використовуючи узгоджені параметри, генеруються унікальні ключі для шифрування і розшифрування даних.

Шифрування та розшифрування:

Після встановлення з'єднання всі дані, які передаються між клієнтом і сервером, шифруються за допомогою використовуваних ключів.

Використані методи шифрування:

Симетричне шифрування: Використовує один ключ для шифрування і розшифрування. Це ефективно, але вимагає безпечної передачі цього ключа.

Асиметричне шифрування: Використовує пару ключів - публічний та приватний. Публічний ключ використовується для шифрування, а приватний - для розшифрування.

Ризики та їх уникнення:

Атака "Man-in-the-Middle": Злоумішленник може намагатися перехопити комунікацію. SSL використовує сертифікати для підтвердження автентичності сервера, унеможливлюючи атаки MITM.

Подвійна витрати даних: Криптографічне шифрування може збільшити обсяг переданих даних. Однак це компенсується захистом від несанкціонованого доступу.

Приклад:

Розглянемо сценарій передачі логіну та пароля від користувача до сервера:

Клієнт і сервер обмінюються ключами шифрування.

Користувач вводить свій логін та пароль.

Ці дані шифруються на клієнтському пристрої за допомогою публічного ключа сервера.

Зашифровані дані надсилаються на сервер.

Сервер розшифровує отримані дані за допомогою свого приватного ключа.

Цей процес забезпечує безпеку персональних даних під час їх транзиту через мережу.

Сертифікати SSL/TLS:

Застосування SSL-сертифікатів для аутентифікації сервера та забезпечення безпеки під час обміну даними.

Процес використання сертифікатів SSL/TLS:

Запит на сертифікат:

Коли сервер встановлює з'єднання з клієнтом, він представляється та надсилає запит на сертифікат. Цей сертифікат містить публічний ключ та іншу інформацію про сервер.

Видання сертифіката:

Центр сертифікації (ЦС) аутентифікує сервер та, якщо він вважає його достовірним, видає сертифікат. ЦС використовує свій приватний ключ для підпису сертифіката.

Використання сертифіката:

Сервер відправляє свій сертифікат клієнту. Клієнт може перевірити автентичність сертифіката, використовуючи публічний ключ ЦС.

Шифрування даних:

Коли з'єднання встановлено, дані між клієнтом і сервером шифруються за допомогою симетричного шифрування.

Використані методи та компоненти:

Центр сертифікації (ЦС): Організація, яка підтверджує автентичність сервера та видання сертифікатів.

Публічний та приватний ключі: Сервер має приватний ключ, яким він підписує свій сертифікат. Публічний ключ використовується для перевірки підпису та шифрування.

Ризики та захист:

Підrobка сертифікатів: Атаки, де атакувальник видає себе за інший сервер, використовуючи підроблений сертифікат. Ризик зменшується за допомогою надійних Центрів сертифікації.

Видалення сертифікатів: Якщо сертифікат був вилучений або недійсний, можливість захисту від використання застарілого ключа.

Приклад:

Розглянемо сценарій фішингового атак, де зловмисник намагається підробити додаток:

Отримання фальшивого сертифіката: Зловмисник отримує фальшивий

сертифікат та встановлює його на свій сервер.

Користувач і "фішинговий" сайт: Користувач переходить на "фішинговий" додаток, і браузер отримує сертифікат.

Перевірка сертифіката: Браузер перевіряє сертифікат і виявляє, що він не відповідає нашому додатку.

Попередження користувача: Ми попереджаємо користувача про потенційну небезпеку, допомагаючи уникнути атаки.

Цей процес забезпечує безпеку користувачів і виявляє можливі атаки на автентичність

Аутентифікація сервера:

SSL використовує цифрові сертифікати для підтвердження автентичності сервера. Це перешкоджає можливості "Man-in-the-Middle" атак, забезпечуючи користувачам впевненість у тому, що вони спілкуються саме із запланованим сервером додатку.

Процес аутентифікації сервера:

Запит на аутентифікацію: Коли клієнт підключається до нашого сервера, сервер відправляє свій сертифікат клієнту.

Перевірка сертифіката: Клієнт перевіряє, чи відповідає сертифікат сервера його очікуванням. Це може включати перевірку підпису, дійсності та відповідності імені домену.

Отримання публічного ключа: Якщо сертифікат валідний, клієнт отримує публічний ключ сервера з сертифіката.

Шифрування симетричним ключем: Клієнт створює симетричний ключ, який шифрується публічним ключем сервера та надсилається серверу.

Розшифрування симетричного ключа: Сервер використовує свій приватний ключ для розшифрування симетричного ключа, який тепер буде використовуватися для шифрування даних, які передаються між клієнтом і сервером.

Інші важливі елементи:

Цифровий підпис: Сервер підписує свій сертифікат своїм приватним

ключем. Клієнт може перевірити відповідність підпису за допомогою публічного ключа сервера.

Міжнародні стандарти: Аутентифікація сервера в SSL відповідає міжнародним стандартам та рекомендаціям щодо забезпечення безпеки передачі даних в Інтернеті.

Ризики та захист:

Сертифікатні атаки: Зловмисники можуть використовувати скомпрометований або підроблений сертифікат для атаки "Man-in-the-Middle."

Недійсні сертифікати: Застосування недійсних або вже відкликаних сертифікатів може збільшити ризик.

Приклад:

Розглянемо сценарій атаки "Man-in-the-Middle":

Запит на аутентифікацію: Зловмисник впроваджується між клієнтом і сервером під час початкового з'єднання.

Напад на сертифікат: Зловмисник використовує свій фальшивий сертифікат та представляється клієнту.

Клієнт і атакуючий сервер: Клієнт отримує сертифікат, перевіряє його, але він є фальшивим.

Захоплення даних: Зловмисник може переглядати та модифікувати дані, які передаються між клієнтом і "фішинговим" сервером.

Непомічений зловмисник: Без ефективної аутентифікації сервера, клієнт може навіть не підозрювати про атаку.

Це демонструє важливість правильної аутентифікації сервера для запобігання атакам у середовищі Інтернету.

Застосування в Мобільних додатках:

Захист особистих даних:

SSL використовується для шифрування особистих даних користувачів у додатку під час авторизації та обміну інформацією між клієнтом та сервером.

Безпечний обмін медіафайлами:

Шифрування SSL застосовується для захисту конфіденційності медіафайлів,

які обмінюються користувачами додатку.

Переваги:

Захист від перехоплення даних:

SSL-захист унеможливлює зловмисникам перехоплення та читання чутливої інформації.

Аутентифікація сервера:

Використання сертифікатів SSL гарантує користувачам, що вони спілкуються саме із справжнім сервером.

Методи захисту та безпека в мобільних додатках враховують сучасні стандарти безпеки для забезпечення захищеності особистих даних та безпеки користувачів. Використання сучасних методів аутентифікації, контролю доступу та реагування на інциденти гарантує безпечне та надійне використання додатку.

Розділ 4. Дослідження ефективності методів безпеки

4.1. Оцінка ефективності використаних методів та технологій захисту в мобільному додатку "Whabe".

Розгляд безпеки в мобільних додатках стає критично важливим у світлі зростаючих загроз та ризиків. У цьому розділі буде проведено детальний аналіз та оцінка ефективності застосованих методів та технологій захисту в мобільному додатку "Whabe".

Застосування SSL-захисту:

SSL (Secure Socket Layer) в "Whabe" відіграє ключову роль у забезпеченні безпеки комунікацій між користувачем та сервером. SSL шифрує дані, що передаються між пристроями, запобігаючи можливості перехоплення чи модифікації інформації. Основні переваги SSL-захисту включають:

Шифрування даних: SSL використовує криптографічні протоколи для шифрування даних під час їхньої передачі, забезпечуючи конфіденційність.

Аутентифікація сервера: Механізми SSL підтверджують автентичність сервера, уникати можливості атак "людина посереднику".

Оцінка ефективності: SSL в "Whabe" є важливим заходом для забезпечення

безпеки транзакцій та комунікацій. Висока ефективність SSL базується на його надійному шифруванні та можливості перевірки автентичності сервера.

Захист зображень у Firebase Storage:

У "Whabe" зображення зберігаються в Firebase Storage, що відомий своєю надійністю та швидкістю. Сервіс автоматично шифрує дані в спокої та відновлює їх у разі втрати. **Заходи безпеки Firebase Storage:**

Шифрування в спокої: Всі дані, що зберігаються в Firebase Storage, автоматично шифруються, забезпечуючи конфіденційність даних.

Резервне копіювання та відновлення: В разі втрати даних чи відмови пристрою можна відновити інформацію з резервних копій.

Оцінка ефективності: Використання Firebase Storage у "Whabe" є дієвим заходом для забезпечення доступу до зображень та їхньої надійної збереження. Шифрування та резервне копіювання гарантують безпеку та цілісність файлів.

Методи безпечного обміну зображень:

В "Whabe" застосовуються різні методи для забезпечення безпечного обміну зображень між користувачами. Оцінка ефективності включає:

Шифрування даних: Всі зображення шифруються перед відправкою та розшифровуються лише на стороні отримувача.

Контроль доступу: Забезпечення доступу лише авторизованим користувачам до зображень та виключення незаконного доступу.

Оцінка ефективності: Застосування шифрування та контролю доступу в "Whabe" забезпечує конфіденційність та безпеку зображень під час обміну, запобігаючи можливим загрозам.

Використання мов програмування та фреймворків:

Використання Swift для iOS та Kotlin для Android у "Whabe" вказує на високий рівень безпеки розробки. Ці мови програмування надають:

Швидкість розробки: Чистий синтаксис та високий рівень абстракції сприяють швидкій розробці без втрати безпеки.

Інтеграція з платформою: Висока інтеграція Swift та Kotlin з платформами iOS та Android забезпечує оптимальну роботу додатку на обох пристроях.

Оцінка ефективності: Використання Swift та Kotlin у "Whabe" дозволяє надійно та ефективно розробляти додаток для обох мобільних платформ.

Firestore сервіси:

Використання різних сервісів Firestore (база даних, зберігання, аутентифікація, аналітика) в "Whabe" визначає його стабільність та високу швидкість. Кожен сервіс має свої переваги та спрямований на покращення відповідної функціональності додатку.

Оцінка ефективності: Firestore демонструє свою ефективність у "Whabe," надаючи зручний інтерфейс для роботи з базою даних, зберіганням, аутентифікацією та аналітикою.

4.2. Варіанти щодо подальшого підвищення рівня безпеки додатка.

Неперервний аудит безпеки - є важливою складовою стратегії забезпечення безпеки інформаційних систем. В основі його концепції лежить постійний моніторинг та аналіз заходів безпеки для виявлення потенційних загроз та вразливостей у реальному часі. Цей метод дозволяє автоматично ідентифікувати та реагувати на можливі атаки, забезпечуючи високий рівень безпеки та ефективну захисту інформаційних ресурсів.

Переваги:

Реальний час виявлення загроз: Метод неперервного аудиту безпеки дозволяє миттєво реагувати на потенційні загрози та інциденти, забезпечуючи виявлення у реальному часі.

Зменшення часу реакції: Завдяки автоматизованому аудиту та системам моніторингу, час між виявленням загрози та прийняттям відповіді зменшується, що покращує загальну безпеку системи.

Збільшення ефективності: Автоматизовані інструменти аналізу безпеки дозволяють ефективно обробляти великі обсяги даних та виявляти аномалії, які можуть залишитися непоміченими при ручному аудиті.

Контроль конфігурації: Метод дозволяє постійно перевіряти конфігурацію системи, виявляючи можливі вразливості та незахищені конфігураційні

параметри.

Покращена виявлення штучного інтелекту: Інтеграція систем штучного інтелекту дозволяє покращити точність виявлення загроз та аномалій завдяки постійному вдосконаленню моделей.

Недоліки:

Високі витрати на розгортання: Заведення автоматизованих систем вимагає великих витрат на розгортання та конфігурацію спеціалізованих інструментів.

Фальшиві позитиви: Алгоритми виявлення можуть генерувати фальшиві позитиви, що призводить до витрат ресурсів на перевірку невірних сигналів.

Залежність від ресурсів: Ефективність методу залежить від доступності достатніх ресурсів для аналізу та моніторингу.

Інтеграція в "Whabe":

Автоматизований аналіз поведінки користувачів: Використання систем неперервного аудиту для аналізу поведінки користувачів допоможе виявити незвичайні активності та вчасно реагувати на можливі порушення безпеки облікових записів.

Миттєве виявлення незвичайних запитань: Інтеграція аудиту може виявляти незвичайні запитання, які можуть бути підозрілими, такі як спроби неправильної аутентифікації чи несподівані запити до сервера.

Оновлення безпеки в реальному часі: Використання методу дозволить негайно реагувати на нові загрози та впроваджувати оновлення безпеки в реальному часі.

Приклади використання:

Виявлення аномальної активності: Система аудиту може виявити незвичайні підключення або спроби доступу до даних, що допоможе вчасно виявити можливі атаки.

Моніторинг конфігурацій: Неперервний аудит може слідкувати за змінами в конфігурації додатка, надаючи можливість виявлення вразливостей

Удосконалення методів шифрування:

Метод "Удосконалення методів шифрування" спрямований на підвищення рівня захисту даних шляхом вдосконалення використовуваних алгоритмів шифрування. Основною метою є забезпечення високої конфіденційності інформації та запобігання несанкціонованому доступу до неї.

Переваги:

Високий рівень безпеки: Удосконалені методи шифрування забезпечують більшу стійкість до криптографічних атак, зменшуючи ймовірність розшифрування даних злоумисниками.

Забезпечення конфіденційності: Застосування потужних шифрів дозволяє ефективно захищати конфіденційні дані користувачів від несанкціонованого доступу.

Відповідність стандартам безпеки: Удосконалені методи шифрування можуть сприяти відповідності різним стандартам безпеки, таким як GDPR або HIPAA.

Недоліки:

Високе обчислювальне навантаження: Деякі потужні шифри можуть вимагати значних обчислювальних ресурсів, що може вплинути на продуктивність додатку.

Сумісність із старішими версіями: У деяких випадках, удосконалення шифру може призвести до проблем сумісності зі старішими версіями додатку чи платформи.

Інтеграція в додаток "Whabe":

Удосконалення методів шифрування може бути інтегроване в "Whabe" через оновлення криптографічних бібліотек та вдосконалення алгоритмів шифрування для забезпечення більшої безпеки збереження та обміну зображеннями користувачів.

Приклад інтеграції:

Застосування апдейту додатку з новими версіями шифруючих алгоритмів, які відповідають сучасним стандартам безпеки, забезпечить високий рівень конфіденційності та захист даних користувачів.

Захист від фішингу:

Захист від фішингу - спрямований на запобігання атакам, спрямованим на користувачів з метою вибуху їхньої конфіденційної інформації, таких як логіни, паролі або особисті дані, шляхом імітації довірених джерел.

Переваги:

Зменшення ризику інформаційного шахрайства: Захист від фішингу допомагає уникнути витоку конфіденційної інформації користувачів, забезпечуючи їм безпечні умови використання додатку.

Підвищення довіри користувачів: Активні заходи захисту від фішингу сприяють збереженню довіри користувачів до додатку та платформи.

Збереження репутації додатку: Захист від фішинг-атак допомагає уникнути негативного впливу на репутацію додатку та його розробника.

Недоліки:

Можливі помилки впізнавання: Деякі заходи захисту можуть визначати легітимний трафік як потенційно небезпечний, що може викликати помилкові блокування для користувачів.

Постійне вдосконалення технік фішингу: Зловмисники постійно модернізують методи фішингу, тому методи захисту повинні постійно адаптуватися.

Інтеграція в додаток "Whabe":

Інтеграція методів захисту від фішингу в "Whabe" передбачає використання антиспамових та антифішингових алгоритмів, які аналізують повідомлення та сповіщення, перевіряючи їх на ознаки фішингових атак. Наприклад, ідентифікація неперсональних сповіщень та посилань на підозрілі джерела може служити індикатором фішингу.

Приклад інтеграції:

Застосування механізмів миттєвого виявлення та блокування потенційно небезпечних сповіщень або посилань у повідомленнях може допомогти уникнути фішинг-атак та забезпечити безпеку використання додатку "Whabe" для користувачів.

Висновки

У цьому дослідженні ми ретельно розглянули різні методи та технології, спрямовані на забезпечення безпеки мобільних додатків, зокрема, додатка "Whabe". Наш огляд включав такі ключові аспекти:

Шифрування зображень:

Опис: Використовується для забезпечення конфіденційності зображень у мобільному додатку "Whabe".

Застосування: Захист особистої інформації, зберігання на сервері додатка.

Ефективність: Надійний захист від несанкціонованого доступу.

SSL-захист:

Опис: Застосовується для безпечного обміну даними між мобільним додатком та сервером.

Застосування: Захист від атак посередника та забезпечення конфіденційності при передачі даних.

Ефективність: Забезпечує криптографічний захист на рівні транспортного рівня.

Шифрування для забезпечення конфіденційності зображень:

Опис: Використовується для додаткового захисту конфіденційності зображень у системі.

Застосування: Захист зображень, які можуть містити особисту чи конфіденційну інформацію.

Ефективність: Двоюрисний захист та перешкоджання несанкціонованому доступу до зображень.

Методи шифрування та їх використання в розробці мобільних додатків:

Методи шифрування, такі як шифрування зображень та використання SSL-захисту, є ключовими складовими безпеки мобільних додатків. Їх використання в розробці "Whabe" дозволяє:

Захищати приватні дані користувачів, що є надзвичайно важливим у соціальному додатку.

Забезпечувати конфіденційність інформації, що обмінюється між додатком та

сервером.

Унеможливлувати несанкціонований доступ до зображень та інших конфіденційних даних.

Переваги та недоліки використання методів шифрування:

Переваги:

Забезпечення конфіденційності даних.

Захист від атак та несанкціонованого доступу.

Недоліки:

Можливість складнішого управління та обслуговування шифрування.

Додаткове навантаження на ресурси системи.

Дослідження дозволяє зрозуміти, що застосування відповідних методів шифрування є необхідністю для забезпечення безпеки мобільних додатків, особливо у контексті соціальних мереж.

Загальний висновок щодо ефективності застосованих методів та їх відповідності поставленим цілям:

Розглянуті методи та технології безпеки, використані у мобільному додатку "Whabe", виявилися ефективними у досягненні основних цілей щодо забезпечення конфіденційності, цілісності та доступності даних.

Конфіденційність: Методи шифрування, такі як шифрування зображень та SSL-захист, дозволяють ефективно захищати конфіденційні дані користувачів. Реалізація двошарового шифрування для зображень сприяє їхньому надійному захисту.

Цілісність: Застосування SSL-захисту для безпечного обміну даними між додатком та сервером гарантує цілісність передачі інформації. Шифрування зображень допомагає уникнути втрати цілісності графічного контенту.

Доступність: Правильна інтеграція шифрування та SSL-захисту не суттєво ускладнює доступ користувачів до додатку. Вони працюють у фоновому режимі, не впливаючи на швидкість або функціональність.

Переваги використання застосованих методів:

Надійність: Застосовані методи забезпечують надійний рівень безпеки, що є важливим для соціальних додатків із високим рівнем конфіденційності.

Ефективність: Шифрування зображень та SSL-захист працюють ефективно та не значно впливають на продуктивність додатку.

Недоліки та можливі вдосконалення:

Управління ключами: Справжній виклик може виникнути управлінням ключами шифрування, особливо при масштабуванні. Розробники повинні докладати зусиль для забезпечення безпеки ключів.

Оптимізація продуктивності: Хоча вплив на продуктивність мінімальний, можливості оптимізації завжди існують для покращення швидкодії додатку.

Загальний висновок щодо ефективності:

Застосовані методи безпеки виявилися успішними у досягненні заявлених цілей. Вони створюють надійний бар'єр для захисту важливих даних користувачів, забезпечуючи стабільність та безпеку функціонування мобільного додатка "Whabe".

Перспективи подальшого розвитку та вдосконалення безпеки мобільного застосунку "Whabe":

1. Розширення методів шифрування:

Шифрування звукових даних: Розгляд можливостей шифрування для звукового контенту, що дозволить розширити сферу захисту інформації.

Індивідуальне шифрування: Розробка механізмів для індивідуального шифрування окремих елементів, таких як повідомлення або фотографії.

2. Підвищення безпеки ключів шифрування:

Використання апаратних ключів: Дослідження можливостей використання апаратних ключів для забезпечення ще більшої безпеки ключових елементів системи шифрування.

Удосконалення алгоритмів: Постійне вдосконалення алгоритмів шифрування для забезпечення стійкості до сучасних криптографічних атак.

3. Розвиток системи виявлення аномалій:

Машинне навчання та AI: Впровадження технологій машинного навчання та штучного інтелекту для розпізнавання аномальних ситуацій та підозрілих активностей.

Багаторівневий аналіз: Розробка системи, яка здатна виявляти аномалії на різних рівнях, від користувача до серверної частини додатку.

4. Сервіси біометричної ідентифікації:

Розпізнавання обличчя та відбитків пальців: Використання сучасних технологій для вдосконалення інтерфейсу та забезпечення безпеки за допомогою біометричних даних користувачів.

5. Оптимізація роботи з сервером:

Використання CDN: Впровадження Content Delivery Network (CDN) для покращення швидкодії завантаження та зберігання зображень на Firebase Storage.

6. Активна участь у галузевих стандартах:

Долучення до ініціатив: Участь у створенні та впровадженні галузевих стандартів безпеки додатків для забезпечення відповідності найвищим стандартам.

Загальний висновок:

Подальший розвиток та удосконалення безпеки мобільного застосунку "Whabe" вимагатиме поєднання новітніх технологій, поглибленого дослідження та активної участі в галузевих ініціативах. Це дозволить зберігати високий стандарт безпеки та забезпечувати задоволення користувачів від використання додатку в умовах надзвичайно зростаючих вимог до безпеки в інтернет-середовищі.

Список використаної літератури:

1. Маккартні П. Розробка мобільних додатків для iOS: Програмування на Swift: Видавництво "Діалектика", 2020.
2. Маклафлін Б. Розробка мобільних додатків для Android: Програмування на Kotlin: Видавництво "Діалектика", 2021.
3. Шарман А. Розробка мобільних додатків з React Native: Видавництво "Діалектика", 2019.
4. Феллер Б., Чарльз Г. Професійна розробка мобільних додатків для Android з використанням Java: Видавництво "Кодекс", 2018.
5. Бурлі Е. Розробка мобільних додатків з використанням React Native: Кроссплатформенні рішення для iOS та Android: Видавництво "Кодекс", 2020.
6. Філіпс Б. Розробка Android-додатків за допомогою Android Studio: Повний посібник: Видавництво "Кодекс", 2019.
7. Хадсон П. Swift: Розробка додатків для iOS та macOS: Видавництво "Діалектика", 2021.
8. Гуманіцька І. Android-програмування: Високий ступінь важкості: Видавництво "Діалектика", 2019.
9. Борден Н., Пілес Г. React Native в дії: Видавництво "Діалектика", 2018.
10. Дженкінс Л. Розробка мобільних додатків на Kotlin: Реальні приклади з Google Play: Видавництво "Діалектика", 2020.
11. Філіпс Б., Стюарт К., Гарднер Б. Програмування для Android: Повний посібник від Big Nerd Ranch: Видавництво "Кодекс", 2017.
12. КонSOLE Д., Хільгесс А., Ландау К. Програмування для iOS: Повний посібник від Big Nerd Ranch: Видавництво "Кодекс", 2019.
13. Масіелло Е., Фрідманн Дж. Володіння React Native: Видавництво "Діалектика", 2018.
14. Гріффіт К. Розробка мобільних додатків з Ionic 2: Кроссплатформенні додатки з Ionic 2, Angular 2 і Cordova: Видавництво "Діалектика", 2017.

15. Біессек А. Flutter для початківців: Вступ до створення кросплатформених мобільних застосунків з використанням Flutter і Dart 2: Видавництво "Діалектика", 2020.
16. Клейтон К. Програмування на Swift UI для iOS: Вступ до Swift з використанням Xcode для початківців: Видавництво "Діалектика", 2019.
17. Еленков Н. Внутрішня безпека Android: Глибокий посібник до архітектури безпеки Android: Видавництво "Кодекс", 2015.
18. Клейтон К. Програмування на iOS 13 для початківців: Початок роботи з побудови iOS-додатків за допомогою Swift 5.1 і Xcode 11, 4-е видання: Видавництво "Діалектика", 2019.
19. Мейєр Р. Професійний Android, 4-е видання: Видавництво "Кодекс", 2018.
20. Він Т. ін. Swift в глибину: Видавництво "Діалектика", 2015.