

РОЗРОБКА СИСТЕМИ ДЛЯ ІНТЕГРАЦІЇ В СОЦІАЛЬНІЙ МЕРЕЖІ ФУНКЦІОНАЛУ ВІДСТЕЖЕННЯ МІСЦЕЗНАХОДЖЕННЯ

Вінницький національний технічний університет

Анотація

Ця робота присвячена розробці системи для інтеграції в соціальній мережі функціоналу відстеження місцезнаходження. У роботі наведено процес розробки та загальний алгоритм роботи системи.

Ключові слова: GPS, місцезнаходження, пристрої, відстеження, геолокація

Abstract:

This work is dedicated to developing a system for integrating location tracking functionality into a social network. The paper outlines the development process and provides a general algorithm for the system.

Keywords: GPS, location, devices, tracking, geolocation

Вступ

У сучасному цифровому світі, де соціальні мережі стали не лише платформами для спілкування, але й невід'ємною частиною нашого повсякденного життя, виникає велика потреба в інтеграції нового функціоналу для підтримки рівня зацікавленості. Одним із ключових аспектів такої інтеграції є відстеження місцезнаходження. Система, що розробляється, має на меті не лише надати можливість ділитися місцем перебування, а й забезпечити ефективний механізм контролю та безпеки для користувачів. Розглядаючи перспективи розвитку цього напрямку, важливо звернути увагу на технологічні аспекти імплементації такої системи.

Метою даної роботи є розробка та реалізація системи, яка забезпечить інтеграцію функціоналу відстеження місцезнаходження в соціальній мережі.

Основна частина

Перш за все розглянемо декілька прикладів додатків в яких найчастіше використовується функціонал для відстеження місцезнаходження:

1. Додатки служб таксі. Геолокаційні API є основною складовою таких додатків, тому що жоден такий сервіс не зміг би повноцінно функціонувати без мапи. Наприклад не можна було б замовити поїздки без мапи та не можна було б відслідковувати поїздки без мапи.

2. Додатки для доставки їжі. Служби доставки сильно залежать від API для взаємодії з мапою, за допомогою цього люди які замовляють їжу можуть знайти найближчий ресторан, та відслідковувати стан замовлення та місцезнаходження кур'єра.

3. Додатки для відслідковування громадського транспорту. Такі додатки використовують API для відображення місцезнаходження громадського транспорту в реальному часі.

Функція активного відстеження місцезнаходження не часто зустрічається в соціальних мережах, тому розробка такого модулю може підвищити рівень зацікавленості користувачів [1].

Для кращого розуміння того, як працюють програмні модулі у системі для відслідковування людей поряд з користувачем, було прийнято рішення створити модель системи за допомогою діаграми діяльності на мові моделювання UML (рис. 1). UML-діаграма – це ефективний спосіб візуалізації систем і програмного забезпечення. Програмні інженери використовують діаграми UML для кращого розуміння дизайну, архітектури коду та потенційної реалізації складних програмних систем. Діаграма діяльності UML дозволяє візуалізувати послідовність дій та процеси, які відбуваються в системі [2]. Це допомагає краще зрозуміти логіку роботи програмних модулів і їх взаємодію.

На діаграмі можуть бути показані такі елементи системи, як користувачі, модулі відслідковування, обробники даних та інші компоненти. Вона відображає послідовність дій, що відбуваються під час взаємодії користувача з системою, а також внутрішні процеси системи.

Створення діаграми діяльності на мові UML дозволяє здійснити аналіз роботи системи на високому рівні абстракції, що сприяє кращому розумінню її функціональності та взаємодії компонентів. Такий підхід допомагає покращити процес розробки та забезпечити оптимальну архітектуру системи для відслідковування людей поряд з користувачем. Крім того, діаграми UML застосовуються для моделювання робочих та бізнес-процесів.

Згідно з цією діаграмою, користувач спочатку має автентифікуватися, щоб отримати доступ до функціоналу системи. Після успішної автентифікації користувачу відображається екран, на якому представлені місцезнаходження людей, які знаходяться поряд з ним. Така послідовність дій забезпечує зручний та ефективний спосіб використання системи для користувача.

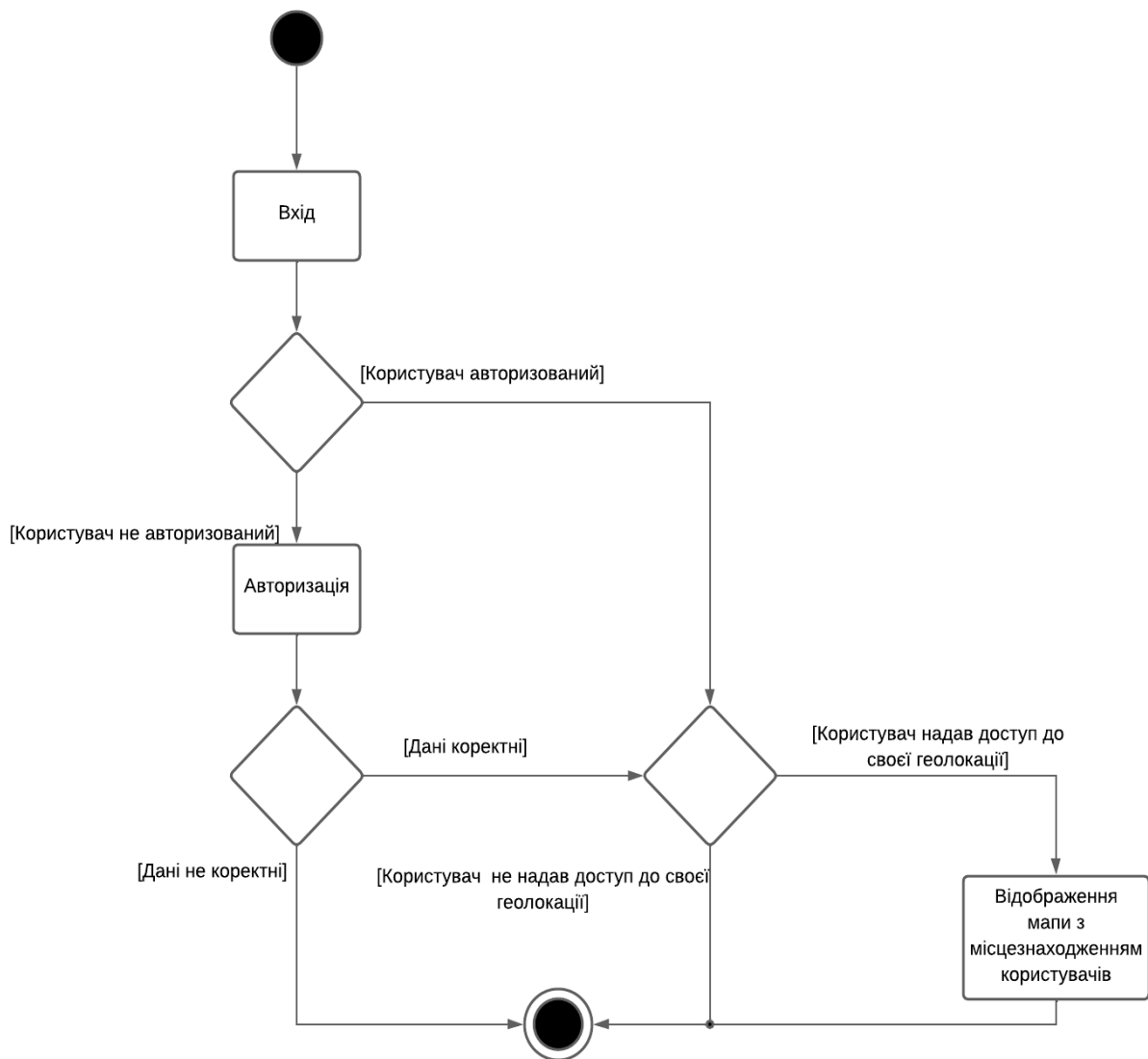


Рисунок 1 – UML-діаграма роботи системи

Для розробки системи відслідковування людей поряд з користувачем необхідно створити загальний алгоритм, який буде керувати роботою мобільного застосунку [3] (рис. 2). Згідно з діаграмою, перший крок – це автентифікація користувача, щоб вони могли увійти до системи і скористатися її функціоналом.

Після успішної автентифікації система перевіряє, чи надав користувач доступ до своєї геолокації. Це важливий крок, оскільки для ефективної роботи застосунку необхідно мати доступ до географічних даних користувача.

Якщо доступ до геолокації надано, користувачу відображається екран з мапою, на якій показані місцезнаходження інших людей, які знаходяться поряд. Це дозволяє користувачеві бачити, де знаходяться його друзі або родичі в реальному часі.

Одночасно з передачею місцезнаходження користувача у хмару для зберігання, інші користувачі можуть запитати цю інформацію з хмари. Усі дані, які передаються між мобільним застосунком та хмарним сервісом, підлягають шифруванню для забезпечення безпеки та конфіденційності.

Такий алгоритм забезпечує зручний та ефективний спосіб використання системи для користувачів, а також забезпечує захист їх приватності та безпеки даних.

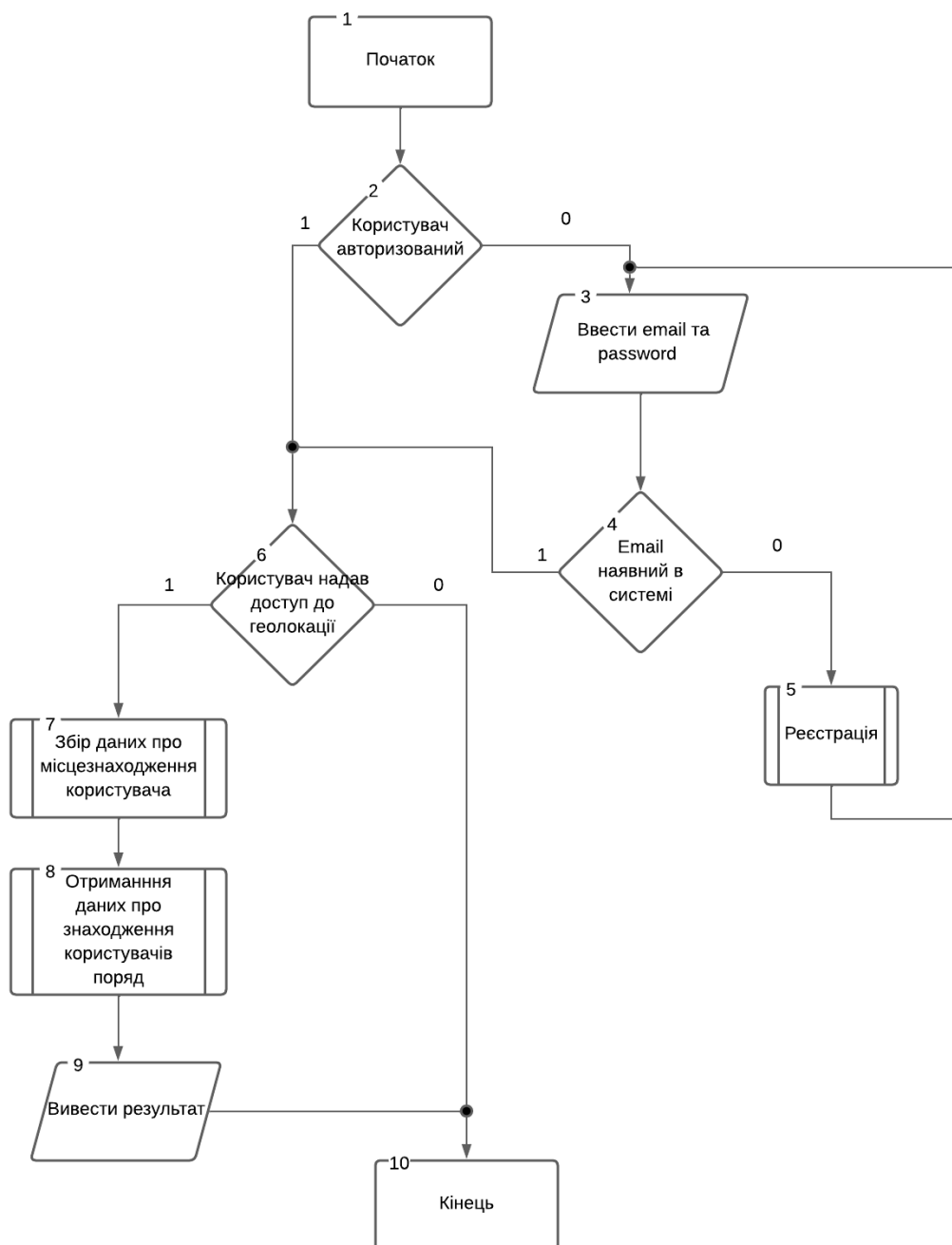


Рисунок 2 – Блок-схема алгоритму роботи системи

Головною функціональністю програмного продукту є отримання інформації про місцезнаходження користувача за допомогою GPS. Для досягнення цієї мети необхідно спочатку ініціалізувати «слухача» для відстеження змін у геолокації. Це дозволить програмі автоматично отримувати оновлені координати користувача, коли він переміщується.

Реалізація екземпляра "слухача" для відстеження геолокації користувача шляхом імплементції вже існуючого інтерфейсу та перевизначення його методів у класі, що створюється, є досить поширеним підходом у програмуванні. Цей підхід дозволяє використовувати готові інтерфейси або абстрактні класи та адаптувати їх під конкретні потреби проєкту.

Інтерфейс може визначати методи для відстеження геолокації, наприклад, методи для отримання координат, визначення розташування на карті тощо. Перевизначення цих методів у класі дозволить забезпечити власну реалізацію функціональності відстеження геолокації, яка відповідає конкретним потребам проєкту.

Цей підхід дозволяє розділити логіку відстеження геолокації на окремий компонент, що полегшує його використання та підтримку у системі. Крім того, він сприяє збереженню принципів об'єктно-орієнтованого програмування, а саме поліморфізм – об'єкти різних класів можуть мати однакові методи, але реалізовувати їх різними способами. Програма працює з усіма об'єктами, використовуючи загальний інтерфейс, що робить код більш гнучким і універсальним.

Після отримання інформації про місцезнаходження, необхідно її зашифрувати, щоб забезпечити конфіденційність та безпеку даних. Шифрування інформації про місцезнаходження є важливим кроком для захисту приватності користувачів та запобігання несанкціонованому доступу до їх особистої інформації. Під час шифрування дані перетворюються в криптографічно безпечний формат, який неможливо прочитати без відповідного ключа.

Це дозволяє зберігати інформацію про місцезнаходження в зашифрованому вигляді, що робить її недоступною для будь-яких несанкціонованих осіб чи програм.

Програмну реалізацію процесу шифрування даних про місцезнаходження користувача зображено на рисунку 3.

```
1 usage
public List<byte[]> encryptData(String lon, String lat){
    List<byte[]> encrypted = new ArrayList<>();
    try {
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding");
        SecretKey secretKey = new SecretKeySpec(secretString.getBytes(StandardCharsets.UTF_8),
            algorithm: "AES");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        encrypted.add(cipher.doFinal(lon.getBytes()));
        encrypted.add(cipher.doFinal(lat.getBytes()));
    } catch (InvalidKeyException | NoSuchPaddingException | NoSuchAlgorithmException |
        IllegalBlockSizeException | BadPaddingException e) {
        throw new RuntimeException(e);
    }
    return encrypted;
}
```

Рисунок 3 – Програмний код методу для шифрування інформації про місцезнаходження

Алгоритм шифрування інформації про місцезнаходження реалізований за допомогою бібліотеки Cipher, яка надає різні алгоритми шифрування. Першим кроком є створення секретного ключа. Для обраного алгоритму шифрування AES ключ може мати різну довжину, наприклад, 128, 192 або 256 біт. Програмний продукт, що розробляється використовує алгоритм AES з довжиною секретного ключа 256 біт. Потім ініціалізується об'єкт Cipher, який буде відповідальний за шифрування інформації. Після ініціалізації об'єкту Cipher можна використовувати методи класу для шифрування.

Після шифрування отримані дані потрібно передати в сервіс (рис. 4), який відповідає за збереження та обробку місцезнаходження користувачів. Цей процес відбувається через відправку запиту на сервіс за допомогою Retrofit. Даний сервіс запущений на віддаленому сервері та відповідає за зберігання та оновлення даних, а також за забезпечення доступу до них відповідно до правил безпеки та приватності.

```
public class LocationServiceImpl implements LocationService {  
  
    2 usages  
    private final LocationServiceApi locationServiceApi;  
  
    1 usage  
    public LocationServiceImpl() {  
        Retrofit retrofit = RetrofitUtil.getRetrofitInstance(url "https://philosophers.party/");  
        this.locationServiceApi = retrofit.create(LocationServiceApi.class);  
    }  
  
    1 usage  
    public void sendLocationInfo(Double latitude, Double longitude){  
        LocationDto locationDto = new LocationDto();  
        locationDto.setLatitude(latitude);  
        locationDto.setLongitude(longitude);  
  
        Call<Void> call = locationServiceApi.postLocation(locationDto);  
  
        try {  
            call.execute();  
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```

Рисунок 4 – Передача повідомлення на сервіс для роботи з місцезнаходженням

Після успішного збереження даних у хмарному середовищі, система може відобразити користувачеві інформацію про місцезнаходження інших користувачів та відобразити ці дані на мапі.

Фінальним кроком є розшифровка даних про місцезнаходження інших користувачів та відображення цих даних на мапі.

Висновок

В ході роботи було розглянуто процес розробки системи для інтеграції в соціальній мережі функціоналу відстеження місцезнаходження. Також було розроблено модель роботи системи за допомогою UML діаграми. Ця діаграма візуалізує структуру та взаємодії між різними компонентами системи. Для демонстрації принципу роботи системи було створено загальний алгоритм. Як результат було розроблено програмний модуль для інтеграції в соціальній мережі функціоналу відстеження місцезнаходження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кондратюк О.О., Романюк О.В. Дослідження різних API для відстеження місцезнаходження пристрою : Матеріали конференції «ЛПІ Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2024)». – Вінниця : ВНТУ, 2024.
2. What is UML Diagram? URL: <https://miro.com/diagramming/what-is-a-uml-diagram/> (date of access 09.05.2024).
3. What is programming algorithm. URL: <https://www.indicative.com/resource/programming-algorithm> (date of access 09.05..2024).

Кондратюк Олександр Олександрович – студент групи ІПІ-206, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: kondralex222@gmail.com

Романюк Оксана Володимирівна – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksanav@gmail.com

Kondratiuk Oleksandr – student of group 1PI-20b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: kondralex222@gmail.com

Oksana Romaniuk – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksanav@gmail.com