

Я.О. Гайдукевич, А.Ю. Дорошенко

ПРО РЕАЛІЗАЦІЮ ІНТЕРФЕЙСІВ МЕТЕОРОЛОГІЧНИХ ПРОГНОЗІВ ДЛЯ МОБІЛЬНИХ ПЛАТФОРМ

У статті детально викладено основні вимоги до реалізації інтерфейсів прогнозу погоди для мобільних платформ, що забезпечують максимальну зручність користувачам, а також запропоновано один метод для такої реалізації на мобільних пристроях під операційною системою OS Android. За приклад реалізації інтерфейсу прогнозування погоди на мобільній платформі представлено додаток, що надає в режимі реального часу погодні інформації, таку як температура, швидкість вітру, поточний час та прогноз погоди на наступні дні. Додаток використовує API для отримання погодних даних і відображення їх у зручному форматі. Метод дозволяє не лише робити точніші прогнози погоди та надавати користувачам персоналізовану інформацію про погоду, а й пропонує додаткові функції.

Ключові слова: прогнозування погоди, інтерфейс, API, мобільний пристрій, візуалізація, OS Android.

Вступ

Із розвитком технологій і широким розповсюдженням мобільних пристроїв багато користувачів використовують програми прогнозу погоди на своїх смартфонах і планшетах. Для зручності користувачів необхідно розробити ефективні методи реалізації інтерфейсів прогнозу погоди на мобільних платформах. Враховуючи обмеження мобільних пристроїв, ці методи мають надавати інформацію про погоду в зручній і легкій у використанні формі. Для того, щоб забезпечити максимально зручний і простий доступ до інформації про погоду, розробники створили програми та інтерфейси, які дозволяють користувачам отримувати інформацію про погоду в режимі реального часу.

Мобільні програми стали невід'ємною частиною нашого життя, і метеорологічні програми не є винятком. Вони дозволяють отримувати інформацію про погоду в будь-який час і в будь-якому місці, і це робить їх одними з найпопулярніших програм для мобільних пристроїв. Однак, для створення успішної програми, потрібно звернути увагу на різні аспекти, зокрема, на інтерфейс, який дозволяє користувачу взаємодіяти з програмою. У цій статті ми розглянемо методи реалізації інтерфейсів метеорологічних прогнозів для мобільних платформ, аби створити зручний та

інтуїтивно зрозумілий інтерфейс для користувачів. Ми розглянемо різні аспекти створення інтерфейсів, а саме, функціональність, зручність використання та можливості оптимізації для мобільних пристроїв. Також розглянемо деякі з найпопулярніших інструментів та технологій, які використовуються для створення мобільних інтерфейсів для метеорологічних прогнозів.

Метеорологічні прогнози є важливим елементом повсякденного життя багатьох людей, оскільки вони дозволяють планувати свої активності, а також готуватися до екстремальних погодних умов.

Прогнозування ґрунтується на аналізі ретроспективних даних, тобто минулих і поточних умов, з метою виявлення тенденцій у динаміці певного об'єкта або події. Тому розв'язання задачі прогнозування вимагає певної вибірки навчальних даних.

Інтерфейси прогнозування погоди для мобільних платформ можуть бути реалізовані по-різному, залежно від технології розробки, мови програмування, типу мобільної платформи та бізнес-вимог. Однак, незалежно від того, який метод використовується, кілька загальних підходів можуть забезпечити користувачів якісною і точною інформацією про погоду. Деякі з цих методів включають:

1. Використання API служб погоди: щоб отримати поточні дані про погоду, ви можете використовувати API служб погоди. Це дозволяє розробникам зосередитися на розробці інтерфейсу, а не гаяти час на збір і обробку даних.

2. Використання геолокації: геолокація може використовуватися для надання точної інформації про погоду в місці проживання користувача. Інші дані, наприклад, часові пояси, також можуть використовуватися для надання більш точної інформації про погоду.

3. Простота і зрозумілість: простий і зрозумілий дизайн допомагає користувачам отримати доступ до інформації про погоду в зручний і простий спосіб. Також можна використовувати піктограми та графіку, щоб допомогти користувачам швидко зрозуміти значення погодних показників.

4. Реалізація різноманітних інформаційних функцій і можливостей, як-от: повідомлення про погіршення погоди, графіки температури й опадів, історію погоди, таймери дощу тощо. Реалізація таких функцій зробить інтерфейс зручнішим і кориснішим для користувачів.

Загалом використання таких технологій і методів, як API метеорологічних сервісів, геолокація, методи машинного навчання і простий дизайн можуть забезпечити користувачів мобільних пристроїв точною і якісною інформацією про погоду.

1. Огляд існуючих методів і засобів для візуалізації метеорологічного прогнозування

Прогнозування погоди є одним із ключових напрямків комп'ютерних технологій та інформаційних систем. Різні методи аналізу та моделювання даних метеорологічних вимірювань можуть бути використані для прогнозування та надання своєчасної інформації про погоду [1].

Одним із ключових елементів успішного прогнозування погоди є розробка ефективних інтерфейсів даних і методів візуалізації. Ці методи дозволяють користувачам взаємодіяти з інформаційною системою в простий та інтуїтивно зрозумілий спосіб і отримувати корисну інформацію про погоду.

Зокрема, AccuWeather розглядає використання різних інтерфейсів і методів візуалізації для прогнозування погоди. Іншим поширеним прикладом використання інтерфейсу і методів візуалізації для прогнозування погоди є використання додатків і веб-сайтів для прогнозування погоди [9].

Додатки і веб-сайти використовують інтерфейс і методи візуалізації для представлення прогнозів погоди в зрозумілій і простій у використанні формі. Наприклад, зміни температури, швидкості вітру та кількості опадів у часі представлені у вигляді графіків та діаграм.

Крім того, піктограми та кольорове кодування можуть використовуватися для швидкого та простого передавання інформації про погоду, зокрема, про сонячне світло, хмарність чи сніг.

Прикладом такого додатку є Weather Channel, який надає детальні прогнози погоди на тиждень, використовуючи графіки, анімовані карти та відео. Додаток надає інформацію про погоду, таку як температура, опади і вітер, а також анімовані радарні карти, які відстежують рух хмар і опадів.

Існують також такі додатки, як Ventusky, який використовує 3D-моделі для візуалізації змін погоди в реальному часі, що дозволяє користувачам дізнатися більше про погодні умови в конкретному місці.

Ці додатки та веб-сайти є чудовими прикладами того, як можна використовувати користувацький інтерфейс і методи візуалізації для представлення складної погодної інформації у зрозумілій і доступній для всіх спосіб.

Звісно, для візуалізації метеорологічного прогнозування використовуються різні інструменти та техніки, залежно від того, наскільки детальний прогноз необхідно представити та для якої аудиторії.

Одним із найпоширеніших інструментів для візуалізації метеорологічного прогнозування є карти погоди, які надають інформацію про температуру повітря, вітер, опади та інші показники на певному територіальному просторі. Наприклад, можна побачити карту погоди на наступний тиждень для м. Києва, де кольором показано очікувану температуру повітря, а стрілками – напрям вітру та його швидкість.

2. Дослідження мобільних веб-інтерфейсів для метеорологічного прогнозування та авторський додаток Weather App

Дослідження мобільних веб-інтерфейсів для метеорологічного прогнозування є надзвичайно важливим дослідженням у сфері інформаційних технологій. З можливими мобільними пристроями та технологіями, багато людей віддають перевагу метеорологічним прогнозам на своїх мобільних пристроях.

По-перше, для реалізації інтерфейсу прогнозу погоди необхідно визначити, яка інформація буде відображатися на екрані. Автор використовуватиме Application programming interface для отримання інформації про погоду, швидкості вітру та часу; за допомогою додатку, що використовує API, є можливість отримувати дані про погоду з веб-сайту, який надає цю інформацію в режимі реального часу.

Основний принцип розробки інтерфейсу – зробити його максимально зрозумілим і простим у використанні. Наприклад, для відображення погоди можуть використовуватися піктограми, які нагадують користувачам про погоду. Для відображення швидкості вітру застосовуються числові значення, які одразу повідомляють користувачеві, з якою швидкістю віє вітер.

Крім того, різні кольори та фонові зображення використовуються для того, щоб зробити інтерфейс більш привабливим і зручним для користувача. Наприклад, на екрані прогнозу погоди використовуються фонові зображення, що відповідають поточним погодним умовам. Також був розроблений логотип під додаток. Застосунок також реагує на час. Якщо день, то фонове зображення змінюється на яскравий малюнок, якщо ж ніч, тоді небо покривається зірками.

Реалізація інтерфейсів прогнозування погоди для мобільних платформ дуже різноманітна і залежить від типу інформації, яку необхідно виводити на екран. Дана реалізація використовує API для отримання такої інформації, як-от погода, швидкість вітру і час, що потім відображається в інтерфейсі за допомогою різних елементів, таких як текст, іконки і фонові зображення. Розробка інтерфейсів прогнозу погоди для

мобільних платформ – це складний і творчий процес, який вимагає від розробників балансу між функціональністю і зручністю використання [2].

Розглянемо додаток, представлений у статті. Написаний мовою програмування Java під операційну систему Android. Android розробники пишуть мовою Java тому, що існує велике ком'юніті, багато бібліотек, рішень і готових модулів. Іншими словами, якщо розробник стикається з якоюсь проблемою, він може швидко отримати рішення. До того ж ця мова старіша за той же Kotlin, і тому для підтримки старих додатків або для зручності обирають саме Java. Слід також зважати на ринкові умови: на Google I/O 2017, Kotlin була офіційно визнана основною мовою розробки для Android, тоді як Java перебуває на 8-му місці з деякими новими функціями (остання версія наразі 14), і її майбутнє в Android залишається невизначеним.

Одним із прийомів виведення погоди без вказання міста є – відстеження мобільного девайсу за геолокацією. На рисунку 1 продемонстровано вікно дозволу на відстеження геолокації. Додаток за етичними нормами відстежує користувача тільки з його дозволу. В такому випадку місто не доведеться вказувати вручну, це дуже зручно, якщо люди пересуваються між містами і додаток автоматично підлаштовується під зміни локації.

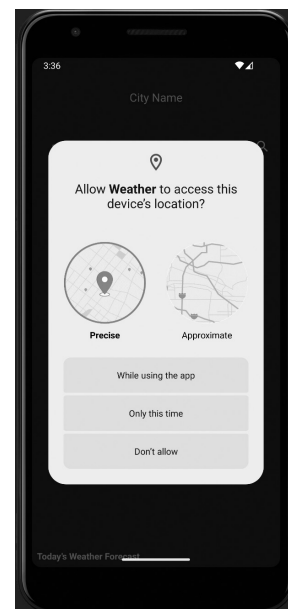


Рис. 1 Відстеження геолокації.

3. Візуалізація результатів метеорологічного прогнозування на мобільних пристроях

Візуалізація результатів прогнозу погоди на мобільних пристроях корисна для користувачів, яким потрібен швидкий доступ до даних про погоду. Існує низка додатків і сервісів, що надають таку інформацію, багато з яких використовують візуалізацію, щоб зробити дані більш зрозумілими. Для ефективної візуалізації погодних даних на мобільних пристроях необхідно враховувати безліч чинників, включно з розміром екрана, обмеженим обсягом пам'яті та низькою продуктивністю пристрою.

Візуалізація – це метод перевірки кінцевих результатів розрахунку, організації управління процесом розрахунку або навіть повернення до вихідних даних для визначення найбільш раціонального напрямку подальшого руху.

Візуалізація даних представляє дані графічно і допомагає виявити аномалії, закономірності та тенденції в процесі аналізу даних. Враховуючи завдання прогнозування, графічне представлення часових рядів було використано для виявлення сезонних елементів. Візуалізація також використовувалася для демонстрації розподілу об'єктів у двовимірному просторі [3].

Дивлячись на лінії тренду або скупчення точок на діаграмі розсіювання, аналітики можуть виявити закономірності й швидше отримати потрібне рішення. Тому в інтелектуальному аналізі даних важливо використовувати зображення, а не символи.

Основна перевага візуалізації полягає в тому, що вона майже не вимагає спеціальної підготовки користувача. Візуалізації дозволяють дуже легко зрозуміти інформацію, просто подивившись на неї.

Найпростіші типи візуалізацій з'явилися давно, але їх використання тільки зараз набуває широкого розповсюдження. Візуалізація – це не просто вдосконалення аналітичних методів, а, на думку Скотта Рівза, в деяких випадках візуалізація може навіть замінити аналітичні методи.

Візуалізація даних може мати форму графіків, діаграм, гістограм і схем.

Одним із найпоширеніших способів

візуалізації прогнозів є графічне представлення погоди на карті. Цей метод дає змогу користувачам швидко оцінити, які райони схильні до впливу таких погодних умов, як дощ, сніг, вітер і спека. Крім того, такі карти можуть допомогти мандрівникам планувати свої маршрути, даючи їм уявлення про імовірні в дорозі погодні умови. Карты можуть відображатися інтерактивно, даючи змогу користувачам натискати на різні області для отримання додаткової інформації.

Ще одним способом візуалізації прогнозу погоди є графік температури. Цей графік може показувати як поточну температуру, так і прогноз на найближчі кілька днів. Він також може показувати прогноз для різних часових поясів, наприклад, ранок, день, вечір і ніч. Така візуалізація корисна для людей, які планують свій день і вирішують, що вдягнути [4].

Використання графіків і діаграм. Цей метод часто застосовується для відображення змін таких показників, як температура, вологість і барометричний тиск за певний період часу. Графіки та діаграми дають змогу користувачам швидко та легко визначити погодні умови та майбутні прогнози.

Ще один спосіб візуалізації погодних даних на мобільних пристроях – використання анімації. Цей метод допомагає користувачам глибше зрозуміти динаміку погодних змін, відображаючи їх в анімованій формі. Такі показники, як температура, опади та вітер, можуть бути анімовані з регулярними інтервалами.

Важливим аспектом візуалізації прогнозів на мобільних пристроях є простота використання. Користувачі хочуть отримувати інформацію про погоду швидко, без необхідності шукати її на різних сторінках. Тому візуалізації мають бути простими для розуміння та використання.

Інструменти візуалізації підтримують важливі бізнес-завдання, зокрема, прийняття рішень. Тому виникає потреба в більш високій якості інструментів візуалізації, яка характеризується появою абсолютно нових інструментів візуалізації та поглядів на їхні можливості, а також розвитком різноманітних трендів у цій сфері.

Сучасні аналітичні інструменти, включаючи інтелектуальний аналіз даних,

неможливі без якісної візуалізації. Інструменти візуалізації повинні надавати чіткі, переконливі, зрозумілі та прості зображення з використанням різноманітних аспектів, із кольором, контрастністю, межами, пропорціями та масштабом зображення включно [5].

Зростання попиту на інструменти візуалізації та необхідність порівнювати їх між собою привели в останні роки до розробки набору принципів якісного візуального представлення інформації.

Повертаючись до додатку, можна провести аналогію з вищесказаним та побачити візуалізацію інтерфейсу на прикладі розробки. Інтерфейс складається з декількох частин: вибору міста, градусів за Цельсієм, показників атмосфери та логотипів до них, фону, а також погодинного відображення погодних умов і швидкості вітру. На рисунку 2 представлено два результати: м. Київ і м. Токіо. За місцевим часом 15:29 в Україні.

Загалом візуалізація прогнозів погоди є важливим інструментом донесення погодинної інформації у зручний для користувачів спосіб. Візуалізація дозволяє користувачам швидко та ефективно отримувати інформацію про прогнози погоди, відстежувати зміни погодних умов та здійснювати необхідну підготовку для подолання потенційних негативних наслідків, спричинених погодними явищами.



Рис. 2. Інтерфейс додатку.

4. Розробка інтерфейсу та отримання даних за допомогою API

Android – це платформа для мобільних пристроїв, яка уможливорює розробку додатків для смартфонів, планшетів та інших пристроїв. Розглянемо, як розробити інтерфейс на Android та отримати дані за допомогою API.

Розробка інтерфейсу на Android передбачає використання різних елементів інтерфейсу, таких як кнопки, тексти, зображення, списки та інші [8].

API (Application Programming Interface) – це інтерфейс, який надається для взаємодії з додатком. Для отримання даних за допомогою API на Android можна використовувати посилання на сайт, з якого додаток буде брати інформацію, а також згенерувати API ключ для доступу до цієї інформації. Зазвичай ключ надходить на пошту після реєстрації на сайті. Додаток використовує дані з сайту OpenWeather..

Що ж до класів Java для розробки додатків для Android, то існує багато класів, які надають різні функціональні можливості для створення додатків для платформи Android. У цьому розділі описано деякі важливі класи, що використовуються у розробці Android.

Activity: Цей клас використовується для створення користувацького інтерфейсу програми та взаємодії з користувачем. Кожна активність відображає окремий екран і може містити елементи керування, як-от кнопки, текст і списки. Активності також відповідають за обробку подій, що відбуваються під час взаємодії з користувачем, зокрема, натискання кнопок і переходи між екранами.

Fragment: фрагмент є розширенням класу дії і є частиною користувацького інтерфейсу або підекраном у межах дії. Фрагменти можливо використовувати для створення більш гнучких і розширюваних інтерфейсів, які легко комбінуються за допомогою перетягування. Віджети за потреби можна додавати до дії або видаляти з неї.

Intent: це класи, що використовуються для взаємодії між різними компонентами програми, такими як дії, віджети

та служби. Вони можуть застосовуватися для запуску нових дій, передавання даних між діями, запуску служб або виклику інших додатків. Наміри можна використовувати для виконання різних дій. Зокрема, для надсилання повідомлення, фотографування з камери або перехід на веб-сторінку.

Layout: У розробці Android використовуються різні класи, пов'язані з компонуванням, які визначають структуру і зовнішній вигляд користувацького інтерфейсу. Наприклад, клас `LinearLayout` використовується для розміщення елементів управління в горизонтальних або вертикальних колонках; клас `RelativeLayout` дозволяє задати відносне положення елементів за допомогою правил розміщення; клас `Layout` використовується для створення макета інтерфейсу користувача.

Adapter: адаптери – це класи, що використовуються для надання даних в елементах списків, таких як `ListView` і `RecyclerView`, і для керування їхнім відображенням. Адаптери дозволяють прив'язувати дані з різних джерел (наприклад, масивів або баз даних) до елементів списку і керувати їхнім відображенням [10].

Це лише кілька прикладів класів, які зазвичай використовуються у розробці Android-додатків із використанням Java. Існує багато інших класів і пакетів, які надають багатофункціональності для розробки різних додатків на цій платформі.

Також представлені класи, які були розроблені для додатку:

Клас `MainActivity` відображає прогноз погоди. Розглянемо докладніше, які класи та методи використовуються в цьому коді:

```
import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
```

Секція імпорту з'єднує різні класи і пакети, необхідні для роботи програми. Це пакети підтримки, класи, які обробляють дані про місцезнаходження, класи, що обробляють дані JSON тощо.

Метод `getCityName`:

```
private String getCityName(double
longitude, double latitude) {
// ...
return cityName;
}
```

Цей метод використовує задані координати (широту і довготу) для отримання назви міста.

Метод `getWeatherInfo`:

```
private void getWeatherInfo(String
city) {
// ...
}
```

Цей метод застосовує зовнішній API для отримання інформації про погоду в конкретному місті.

Інші методи в коді виконують різні завдання. Це й обробка результатів дозволів, налаштування елементів інтерфейсу та взаємодія з мережею.

Ці методи є відправною точкою для програми для Android, яка відображає погоду і отримує дані з відповідного API; він містить логіку для управління елементами інтерфейсу, виклику API, отримання інформації про місцезнаходження користувача і обробки дозволів.

Клас `Adapter` використовується для відображення погодних даних у вигляді списку за допомогою `RecyclerView`. Використовується для встановлення елементів списку та відображення даних в об'єктах.

```
@NonNull
@Override
public Adapter.ViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
View view = LayoutInflater.
from(context).inflate(R.layout.weather_rv_
item, parent,false );
return new ViewHolder(view);
}
```

Цей метод створює новий екземпляр `ViewHolder` відповідно до макету

weather_rv_item. Викликається, у разі потреби створення нового елемента списку.

Метод `onBindViewHolder`:

```
@Override
public void onBindViewHolder(@NonNull Adapter.ViewHolder holder, int position) {
    Model model = Model.ArrayList.get(position);
    holder.temperatureTV.setText(model.getTemperature()+ «°c»);
    Picasso.get().load(«http:».concat(model.getIcon())).into(holder.conditionIV);
    holder.windTV.setText(model.getWindSpeed()+ «Km/h»);
    SimpleDateFormat input = new SimpleDateFormat(«yyyy-MM-dd hh:mm»);
    SimpleDateFormat output = new SimpleDateFormat(«hh:mm aa»);
    try{
        Datet=input.parse(model.getTime());
        holder.timeTV.setText(output.format(t));
    }catch(ParseException e){
        e.printStackTrace();
    }
}
```

Цей метод прив'язує дані з об'єкта Model до відповідного ViewHolder. Встановлює значення тексту, зображень та інших елементів списку.

Метод `getItemCount`:

```
@Override
public int getItemCount() {
    return ModelArrayList.size();
}
```

Цей метод повертає загальну кількість об'єктів Model в список.

Внутрішній клас `ViewHolder`:

```
public class ViewHolder extends RecyclerView.ViewHolder{
    private TextView windTV,temperatureTV, timeTV;
    private ImageView conditionIV;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        windTV = itemView.findViewById(R.
```

```
id.idTVWindSpeed);
        temperatureTV = itemView.findViewById(R.id.idTVTemperature);
        timeTV = itemView.findViewById(R.id.idTVTime);
        conditionIV = itemView.findViewById(R.id.idTVCondtiton);
    }
}
```

Цей внутрішній клас є об'єктом ViewHolder, який містить посилання на елемент макета item. Це забезпечує доступ для ефективного керування елементами у списку та для подальшої конфігурації.

Беручи дані з об'єкта моделі, клас Adapter може конфігурувати відповідні елементи інтерфейсу для кожного елемента списку і відображати погодні дані у списку елементів.

Останній клас Model використовується для представлення моделі погодніх даних, що відображаються у списку за допомогою класу Adapter. Клас містить користувацькі поля для зберігання даних про час, температуру та швидкість вітру. Він також містить геттери та сеттери для доступу до цих полів.

Отримати значення поля моделі можна за допомогою геттера і встановити значення поля за допомогою сеттера. Це дозволяє зберігати та отримувати погодні дані для кожного об'єкта Model.

Клас Model використовується для створення об'єктів, що містять погодні дані, і передачі цих об'єктів адаптеру (клас Adapter). Адаптер використовує дані з об'єкта Model для відображення інформації про погоду у списку елементів RecyclerView.

Також не варто забувати про front-end частину розробки додатку, до неї належить розробка макету (кнопки, поля для вводу тощо), логотип додатку, іконки, підбір кольору. На рисунку 3 продемонстровано логотип додатку.

Додаток використовує бібліотеку Picasso для фонового зображення, тому зображення зберігається на веб-ресурсі і не займає місце в проєкті. Приклад коду наведено нижче: `if(isDay==1){`

```
    Picasso.get().load(«https://imgur.com/EIacjLW.png»).into(backIV);
```

```
}else{  
Picasso.get().load(«https://images.  
unsplash.com/photo-1532074534361-  
bb09a38cf917»).into(backIV);  
}
```

У кодї використовується оператор if-else, який, залежно від значення isDay, визначає зображення, яке слід завантажити за допомогою бібліотеки Picasso для завантаження зображення з URL-адреси та встановлення його в ImageView з ім'ям backIV.

Таким чином, в залежності від значення isDay до ImageView встановлюються різні зображення.

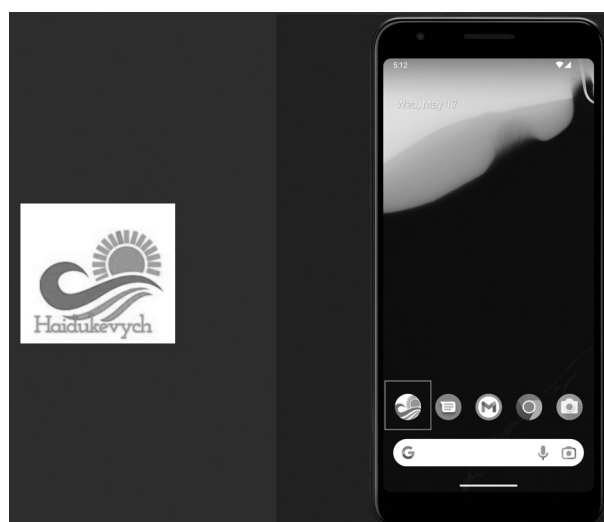


Рис.3. Логотип.

Висновки

У статті запропоновано реалізацію інтерфейсу прогнозу погоди для мобільних платформ, що має важливе значення для створення зручного та функціонального додатку для прогнозування погоди. У виборі способу реалізації інтерфейсу слід враховувати характеристики мобільного пристрою, такі як розмір екрану, продуктивність і наявність мережевого підключення.

Огляд існуючих методів показує, що наразі найпоширенішими мобільними додатками є ті, що мають картографічний та графічний інтерфейси. Ці методи дозволяють користувачам легко орієнтуватися в погодній інформації, надають візуальні інструменти для аналізу даних та швидкого реагування на основі прогнозів погоди.

Загалом вибір методу реалізації ін-

терфейсу прогнозу погоди для мобільних платформ залежить від вимог і потреб кінцевого користувача. Необхідно врахувати кілька факторів, зокрема, розмір екрану, наявність підключення та продуктивність пристрою. Однак у всіх випадках кінцевою метою є надання користувачам зручної, зрозумілої та актуальної інформації про погоду, яка допоможе їм планувати свої дії відповідно до прогнозів погоди.

Для досягнення оптимального користувацького досвіду розробники використовують різні функції, такі як адаптивний дизайн, графіку, інтерфейсні технології. Зокрема, карти та графіки, інформацію про місцезнаходження та інтеграцію з іншими сервісами.

Тож, розробка ефективних і зручних інтерфейсів прогнозування погоди має важливе значення для надання користувачам точної та своєчасної інформації про погоду. Різноманітні методи і техніки, як-от візуалізація даних, алгоритми машинного навчання та інтерактивні користувацькі інтерфейси, можуть значно підвищити точність і надійність прогнозів погоди та покращити користувацький досвід. Крім того, включення даних у режимі реального часу та функцій персоналізації може ще більше підвищити ефективність інтерфейсу прогнозу погоди. Оскільки попит на інформацію про погоду на мобільних платформах зростає, вкрай важливо розробляти інтерфейси, оптимізовані для цих пристроїв і зручні для користувачів. Використовуючи правильні методи і технології, розробники можуть створювати інформативні та привабливі інтерфейси прогнозу погоди, які допомагають користувачам безпечно діяти за мінливих погодних умов.

Також розглянуто можливість розробки інтерфейсу на OS Android та отримання даних за допомогою API. Використання цих прийомів дозволяє розробляти функціональні та зручні додатки для мобільних пристроїв.

Література

1. А.Ю. Дорошенко, Р.В. Кушніренко, О.А. Яценко, Проектування програми

візуалізації поверхні Землі з використанням алгебраїко-алгоритмічних засобів // Проблеми програмування. – 2019, № 2. – С. 3-10.

2. В.А. Прусов, А.Ю. Дорошенко. Ефективний обчислювальний метод мезомасштабного прогнозу погоди. Допов. Нас. акад. наук. укр. 2020. № 3. Р. 10—18.
3. Gupta, A., Prakash, S., & Kumar, S. (2018). Design and Development of Mobile Application for Weather Forecasting. *International Journal of Innovative Technology and Exploring Engineering*, 7(4S), 106-111.
4. Zhao, L., Yu, Y., & Wang, C. (2019). Design and Implementation of a Mobile Weather Forecasting System Based on Baidu Map. In *2019 International Conference on Computer Engineering and Application* (pp. 61-66). IEEE.
5. Chen, S., & Xu, Y. (2020). Design and Implementation of Mobile Weather Forecasting System Based on GIS. In *2020 International Conference on Mechatronics, Control and Robotics* (pp. 213-217). IEEE.
6. Wang, J., Zhang, Y., & Li, Y. (2018). Design and Implementation of Mobile Weather Forecasting System Based on Android. In *Proceedings of the 2018 International Conference on Computer, Information and Telecommunication Systems* (pp. 23-27). IEEE.
7. Alhasanat, M. A., Al-khader, R., & Bani Younes, A. (2019). A Mobile Application for Weather Forecasting using Neural Networks. In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence* (pp. 95-100). ACM.
8. Goyal, A., & Sharma, R. (2020). Design and Implementation of Mobile Weather Forecasting System using OpenWeatherMap API. In *Proceedings of the International Conference on Advanced Computing Technologies and Applications* (pp. 279-288). Springer.
9. Belhaj, M., & Bouziane, M. (2021). Mobile weather applications: A review of the main design and usability principles. *International Journal of Human-Computer Interaction*, 37(5), 434-448.
10. Kaur, P., & Singh, K. (2021). Design and development of a weather forecasting mobile application. *International Journal of Emerging Technologies and Innovative Research*, 8(1), 39-44.

References

1. A.Y. Doroshenko, R.V. Kushnirenko, O.A. Yatsenko, Designing a program for visualization of the Earth's surface using algebraic-algorithmic tools // *Programming problems*. – 2019, No. 2. – P. 3-10.
2. V.A. Prusov, A.Y. Doroshenko, An Efficient Computational Method for Mesoscale Weather Forecasting. *Dopov. Nac. acad. nauk. Ukr.* 2020. № 3. P. 10—18.
3. Gupta, A., Prakash, S., & Kumar, S. (2018). Design and Development of Mobile Application for Weather Forecasting. *International Journal of Innovative Technology and Exploring Engineering*, 7(4S), 106-111.
4. Zhao, L., Yu, Y., & Wang, C. (2019). Design and Implementation of a Mobile Weather Forecasting System Based on Baidu Map. In *2019 International Conference on Computer Engineering and Application* (pp. 61-66). IEEE.
5. Chen, S., & Xu, Y. (2020). Design and Implementation of Mobile Weather Forecasting System Based on GIS. In *2020 International Conference on Mechatronics, Control and Robotics* (pp. 213-217). IEEE.
6. Wang, J., Zhang, Y., & Li, Y. (2018). Design and Implementation of Mobile Weather Forecasting System Based on Android. In *Proceedings of the 2018 International Conference on Computer, Information and Telecommunication Systems* (pp. 23-27). IEEE.
7. Alhasanat, M. A., Al-khader, R., & Bani Younes, A. (2019). A Mobile Application for Weather Forecasting using Neural Networks. In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence* (pp. 95-100). ACM.
8. Goyal, A., & Sharma, R. (2020). Design and Implementation of Mobile Weather Forecasting System using OpenWeatherMap API. In *Proceedings of the International Conference on Advanced Computing Technologies and Applications* (pp. 279-288). Springer.
9. Belhaj, M., & Bouziane, M. (2021). Mobile weather applications: A review of the main design and usability principles. *International Journal of Human-Computer Interaction*, 37(5), 434-448.
10. Kaur, P., & Singh, K. (2021). Design and development of a weather forecasting mobile application. *International Journal of Emerg-*

ing Technologies and Innovative Research,
8(1), 39-44.

Одержано: 18.05.2023

Про авторів:

Дорошенко Анатолій Юхимович,
доктор фізико-математичних наук,
професор, завідувач відділу теорії
комп'ютерних обчислень
Інституту програмних систем
НАН України,
професор кафедри автоматки і
управління
в технічних системах НТУУ
«КПІ імені Ігоря Сікорського».
Кількість наукових публікацій в
українських виданнях – понад 200.
Кількість наукових публікацій
в іноземних виданнях – понад 100.

Індекс Гірша – 6. <http://orcid.org/0000-0002-8435-1451>,

Гайдукевич Ярослав Олегович,
студент кафедри інформаційних систем
та технологій КПІ імені Ігоря Сікорсько-
го.

Кількість наукових публікацій
в українських виданнях – 2.
<http://orcid.org/0000-0002-6300-1778>

Місце роботи авторів:

Інститут програмних систем
НАН України,
03187, м. Київ-187, проспект Академіка
Глушкова, 40.
Тел.: (044) 526 3559.
e-mail: doroshenkoanatoliy2@gmail.com,
yarmcfly@gmail.com