

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

Навчально-науковий інститут денної освіти

дистанційного навчання

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

(підпис)

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ НА FLUTTER ДЛЯ
МОНІТОРИНГУ ПРОГНОЗУ ПОГОДИ»**

**зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра**

Виконавець роботи Гриб Денис Миколайович

_____ «___» _____ 2023 р.
(підпис)

Науковий керівник к.ф.-м.н., доцент, Черненко Оксана Олексіївна

_____ «___» _____ 2023 р.
(підпис)

Рецензент

ПОЛТАВА 2023 р.

ЗМІСТ

ВСТУП.....	3
1. ПОСТАНОВКА ЗАДАЧІ	4
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	5
2.1 Історія спостереження за погодою	5
2.2 Аналіз існуючих програмних засобів.....	8
2.3 Опис мови програмування	11
2.3.1 Dart	13
2.3.2 Flutter.....	14
3. ТЕОРЕТИЧНА ЧАСТИНА.....	16
3.1 Алгоритм роботи додатка	16
3.2 Чиста архітектура	18
3.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи	21
4. ПРАКТИЧНА ЧАСТИНА.....	23
4.1. Розробка блок-схеми, яка підлягає програмуванню.....	23
4.2. Опис процесу програмної реалізації	30
4.3 Опис роботи додатка.....	36
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43

ВСТУП

Актуальність теми. Сьогодні мобільні телефони не є просто засобами зв'язку, а зараз ці пристрої замінюють собою всі гаджети, необхідні в повсякденному житті, наприклад: плеєр, навігатор, фотоапарат, відеокамера та інші. Смартфон надає величезну кількість функцій, які можуть бути корисні в будь-якій сфері діяльності людини.

З появою смартфонів, процес отримання доступу до необхідної інформації значно спростився. Тепер, досить просто встановити додаток, що володіє необхідним набором функцій, отже, питання мобільних додатків є вкрай актуальним та має перспективи розвитку.

Мета роботи - є розробка програми, яка буде надавати користувачеві інформацію про погодні умови в обраному регіоні,

Об'єктом розробки - є розробка додатка для отримання користувачем актуальної інформації про стан погодних умов у потрібній для нього локації.

Предмет розробки - програмний продукт, що надає актуальні погодні дані, за поточним місцезнаходженням користувача, який розроблений за допомогою фреймворку Flutter.

Методи розробки - методика застосування нормальних алгоритмів та програмні засоби мови Dart.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі розглянуто: Історія спостереження за погодою, Аналіз існуючих програмних засобів та Опис мови програмування. У третьому розділі, розглянуто алгоритм роботи додатка. У четвертому, розділі розглянуто практичну частину.

Обсяг пояснювальної записки: 42 стор., в т.ч. основна частина – 19 стор., джерела – 9 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Основним завданням роботи є розробка мобільного додатка моніторингу прогнозу погоди.

Розглянемо основні функціональні можливості додатка:

- температура повітря протягом дня;
- інформація про опади протягом дня;
- вологість протягом дня;
- хмарність впродовж дня;
- атмосферний тиск протягом дня;
- час світанку і заходу;
- прогноз всієї цієї інформації протягом тижня;

Також, розроблений додаток повинен мати зрозумілий для користувача інтерфейс.

Для цього він повинен містити наступні елементи:

- стартову сторінку з відображенням погоди на поточний час;
- зручне перемикання між сторінками з інформацією про погоду;
- зручне оновлення погоди та місцезнаходження;
- перемикач між світлою та темною темою;
- локалізація додатка;

Процес розробки складається з безлічі підпроцесів, деякі з яких показані нижче:

- аналіз вимог;
- специфікація програмного забезпечення;
- проєктування додатка;
- розробка дизайну мобільного додатка;
- тестування мобільного додатка;

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Історія спостереження за погодою

Погодою ми звикли називати найрізноманітніші події природних явищ, що відбуваються на нашій планеті, головні з яких це - вітер, опади у звичних нам дощ і сніг, а також сонце і якоюсь мірою Місяць, адже вона безпосередньо впливає на багато відбуваються на землі погодних катаклізмів, таких як припливи та відливи, землетруси й виверження вулканів.

Також до погодних умов природи потрібно віднести й грози з хмарами та блискавками, тумани, град, бурі, шквали вітру, снігові й пилові хуртовини, торнадо, морські та океанські припливи з відливами, урагани та інші погодні катаклізми й примхи.

Погодні умови змінюються постійно, можна привести для повної переконливості такий факт - кожен секунду на нашій землі відбувається близько 100 блискавок, і це підтверджує той факт, що погода на планеті нестійка і дуже мінлива. У навколоземній атмосфері постійно відбуваються якісь події, формуються дощові хмари й так далі.

Погода найбезпосереднішим чином впливає на кожного з нас, наше самопочуття, нашу діяльність і активність, як і на все навколо нас живе. Від природи і її погоди з постійними умовами, що змінюються, залежить і рослинність її зростання. На багато погода впливає і багато від неї залежить, як ви вже знаєте.

Погода мінлива часом настільки, в чому міг переконатися кожен з вас, що бувають такі дні, коли всього за кілька годин ми спостерігали корінні в ній зміни, коли з ранку було дуже морозно і йшов сильний сніг, але вже в обід де не взявся налетів теплий вітер, і вже по вулицях побігли струмки, а до вечора знову "налетів" вже новий циклон, і знову завалив нас снігом.

Такі ж ситуації ми спостерігали й влітку, тільки не зі снігом, а з дощем, і таких випадків з різкими змінами погодних умов більш ніж достатньо. Тому, чекати від природи стабільної та благодатної погоди нам навряд чи варто, потрібно пристосовуватися до такої яка вона є, до тих мінливим погодних умов які даровані нам природою.

В цілому погодними умовами заведено вважати стан атмосфери на навколоземній її орбіті, в якийсь спостережуваний проміжок часу, або ж за тривалістю спостереження відразу декількох часових відрізків. Погодні умови можна схарактеризувати відразу декількома факторами та критеріями, перш за все - кількістю виділяється Сонцем сонячної енергії (радіації), температурою повітря і повітряним атмосферним тиском, а ще й вологістю атмосфери, напрямком потоку вітру і його силою, хмарністю та опадами, які ми знаємо в вигляді дощу або снігу.

Дізнатися погоду в історії Гісметео, історичний прогноз спостережень в Харкові, середній і причому безплатно на місяць або рік, тиждень, 10 або 14 днів, часом залежить від часу доби, а також року, кліматичних умов, тобто, переміщення і формування в атмосфері хмар, повітряних мас, циклонів, а також антициклонів. Важливо враховувати й те в якому місці та на якій широті проводиться спостереження за мінливими погодними умовами.

У сучасному світі рівень розвитку визначення погодних умов за допомогою супутникової навігації та метеорології проблем з точним її визначенням практично немає. Науковий прогноз погоди від історії спостережень, погодних історичних на день, рік або місяць, середню і навіть похвилинну температуру можна дізнатися на ресурсах інтернету, причому також і можливість опадів і зміни напрямку вітру, з досить високою часткою ймовірності дізнатися найбільш ймовірний розвиток погодних умов.

Вкрай важливим є точний прогноз погоди не тільки нам з вами як простим обивателям, але в ньому мають потребу і багато галузей і

напрямки науки, економіки та сільського господарства. Важливий він для всіх видів наземного автотранспорту, в тому числі та авіаційних перевізників, морським і річковим суднам і це список можна перераховувати довго.

Історичний прогноз погоди, як і історія погодніх спостережень і прогнозу від Гісметео, на кожен місяць і рік, а також температура 3, 5, 7, 10 і 14 днів, дивитися і дізнатися можна безплатно в щоденнику або онлайн можна подивитися в інтернеті в будь-який час, адже він просто необхідний нам з вами, і зовсім не для інтересу, а щоб точно знати про швидке зміну погоди що нас чекає в наступні години, чи буде дощ, як одягатися, захопити з собою на вулицю парасольку, яку одягати одяг і взуття, інше.

2.2 Аналіз існуючих програмних засобів

Додаток, що розробляється, не є унікальним у своєму роді – існують аналоги, які є фундаментом для вдосконалення та виправлення існуючих недоліків, прикладом програмного додатка буде додаток з Google Playstore Погода & Радар що надає інформацію про погоду [1] рис. 2.1



Рисунок 2.1 - Сторінка додатка Погода & Радар

Зовнішній вигляд додатка складається, зі стартової сторінки, про погоду на поточний час, сам інтерфейс додатка виконаний у вигляді прогортуючого списку з інформацією про погоду, на поточний момент з інформацією про погоду на тиждень, на 14 днів, та різна цікава інформація.



Рисунок 2.2 – Зовнішній вигляд додатка «Погода & Радар»
Погода на сьогодні



Рисунок 2.3 – Зовнішній вигляд додатка «Погода & Радар»
Погода на тиждень



Рисунок 2.4 – Зовнішній вигляд додатка «Погода & Радар»

Погода на 14 днів

Незважаючи на те, що «Погода&Радар» є найбільш популярним додатком для прогнозу погоди в Google Playstore, він не позбавлений недоліків, а саме:

- інтерфейс програми дуже перевантажений, в ньому важко розібратися, він не завжди працює швидко і гладко;
- на всіх екранах програми показується реклама.

Додаток розробляється з акцентом на виправлення перерахованих недоліків.

2.3 Опис мови програмування

Для розробки мобільного додатка я обрав мову програмування Dart та фреймворк Flutter

Чому я саме обрав мову програмування Dart та фреймворк Flutter, для розробки додатка.

Особливості мови Dart:

- Звичний і простий для вивчення синтаксис, природний для програмістів на JavaScript, C і Java;
- Забезпечення швидкого запуску і високої продуктивності для всіх сучасних веб браузерів і різних типів оточень, від портативних пристроїв до потужних серверів;
- Можливість визначення класів і інтерфейсів, що дозволяють використовувати інкапсуляцію і повторно використовувати існуючі методи і дані;
- Необов'язкове вказування типів, використовувати чи ні статичні типи вирішує розробник. Вказування типів дозволяє спростити зневадження і виявлення помилок, робить код яснішим і читаним, спрощує його доопрацювання та аналіз сторонніми розробниками.
- Серед підтримуваних типів: різні види хешів, масивів і списків, черги, числові і рядкові типи, типи для визначення дати і часу, регулярні вирази (RegExp). Можливо створення своїх типів;
- Для організації паралельного виконання пропонується використовувати класи з атрибутом `isolate`, код яких виконується повністю в ізольованому просторі в окремій області пам'яті, взаємодіючи з основним процесом через відправку повідомлень;
- Підтримка використання бібліотек, що спрощують підтримку і зневадження великих вебпроектів. Сторонні реалізації функцій можуть підключатися у вигляді поділюваних бібліотек. Застосунки можна розбити

на частини і доручити розробку кожної з частин окремій команді програмістів;

- Набір готових інструментів для підтримки розробки мовою Dart, включаючи реалізацію засобів динамічної розробки та зневадження з виправленням коду на льоту («edit-and-continue»);

- Можливість створювати однорідні системи, що охоплюють як клієнтську, так і серверну частину. Використання однієї мови та інструментарію для клієнтських і серверних компонентів спрощує процес кодування і позбавляє від постійної зміни контексту.

Особливості Flutter:

- Фреймворк розроблений від компанії Google
- Має відкритий програмний код
- Дозволяє просто і швидко створювати мобільні додатки для iOS і Android

При цьому в роботі Flutter не використовує нативні компоненти зовсім. Замість цього всі UI-елементи у фреймворку створюються за допомогою власного графічного движка. Flutter дозволяє створювати всі елементи призначеного для користувача інтерфейсу додатка з готових віджетів. У цьому Flutter схожий з іншими фреймворками – React і Vue, і в той же час має ряд відмінностей від них. Так, він не використовує мову програмування Javascript, натомість Flutter використовує мову Dart.

2.3.1 Dart

Dart — мова програмування, яку розробляє компанія Google, позиціонуючи як мову структурованого програмування для Веб. Розробники вважали, що в довгостроковій перспективі Dart може стати прогресивною заміною JavaScript, котрий потерпає від наявних в даний час проблем з розширюваністю, продуктивністю і підтримкою розробки складних застосунків. Мова має схожий на Java синтаксис, не вимагає явного визначення типів і її можна використовувати для створення серверних та клієнтських застосунків.

У березні 2015 компанія Google представила оновлену стратегію просування Dart, у котрій вирішено не прив'язувати Dart до браузеру і відмовитися від ідеї інтеграції віртуальної машини Dart у Chrome. Розробку буде зосереджено на застосуванні Dart як проміжної мови, скомпільованої в JavaScript. Розвиток Dart як окремої мови, альтернативної JavaScript і безпосередньо підтримуваної у браузерах, визнано недоцільним. Замість цього Dart рухатиметься у бік якіснішої інтеграції з JavaScript і генерації оптимального JavaScript-коду. При цьому розробку віртуальної машини Dart VM буде продовжено, але вона позиціонуватиметься в основному для створення серверних і мобільних застосунків.

Влітку 2014 асоціація ECMA International, що займається стандартизацією інформаційних і комунікаційних технологій, затвердила специфікацію ECMA-408, яка стандартизує синтаксис і семантику мови Dart, а також склад базових бібліотек і супутніх мові технологій, відтоді мова Dart є офіційним стандартом Ecma. 9 Грудня 2015 в Токіо було затверджено 4-е видання Надання Dart статусу стандарту Ecma дозволить розширити область використання мови та прискорити забезпечення його підтримки в наявних на ринку браузерах і продуктах. Вибір Ecma International як організації для стандартизації обумовлений тим, що ця асоціація вже розвиває близькі до специфіки Dart стандарти для

мов JavaScript, Eiffel і C#. Просуванню Dart як стандарту сприяло надання компанією Google всіх пов'язаних з розробкою патентів у безоплатне використання, що не вимагає оплати відрахувань.

2.3.2 Flutter

Flutter — це програмний каркас із відкритим кодом для створення додатків для платформ Android та iOS, а також на вебi, розроблений компанією Google. Він є основним способом створення додатків для Google Fuchsia. Весь графічний інтерфейс Google Fuchsia створено за допомогою Flutter.

Flutter складається з:

- Flutter рушій — програмний рушій для рендерингу, написаний в основному на C++ з використанням графічної бібліотеки Google Skia. Він також використовує SDK платформ Android або iOS.
- Базової бібліотеки (Foundation library) — бібліотека складається з класів та функцій (написані на Dart), які використовують для побудови Flutter програм, для взаємодії із Flutter рушієм.
- Віджетів. Дизайн інтерфейсу користувача у Flutter будують з віджетів. Віджет у Flutter являє собою незмінний об'єкт, який описує частину інтерфейсу користувача. Вся графіка, текст, фігури та анімації створюють за допомогою віджетів. Складні віджети створюють шляхом об'єднання простих. На поточний час Flutter містить два набори віджетів, які відповідають відповідним принципам побудови:
 - віджети Material Design використовують дизайн Google;
 - віджети Cupertino імітують дизайн Apple iOS.

Flutter — молода, але дуже перспективна платформа, вже привернула до себе увагу великих компаній, які запустили свої програми. Цікава ця платформа своєю простотою в порівнянні з розробкою веб-застосунків, і швидкістю роботи на рівні з рідними додатками. Висока продуктивність програми та швидкість розробки досягається за рахунок декількох технік:

- На відміну від багатьох відомих на сьогоднішній день мобільних платформ, Flutter не використовує JavaScript ні в якому вигляді. В якості мови програмування для Flutter вибрали Dart, який компілюється в бінарний код, за рахунок чого досягається швидкість виконання операцій порівнянна з Objective-C, Swift, Java, або Kotlin.

- Flutter не використовує рідні компоненти, знову ж, ні в якому вигляді, тому не доводиться писати ніяких прошарків для комунікації з ними. Замість цього, подібно ігровим рушіям (UI ігор дуже динамічний), він малює весь інтерфейс самотійно. Кнопки, текст, медіа-елементи, фон — все це малюється всередині графічного рушія в самому Flutter. Після вищесказаного варто відзначити, що "Hello World" додаток на Flutter займає зовсім небагато місця: iOS $\approx$ 2.5 Mb і Android $\approx$ 4Mb.

- Для побудови UI під Flutter використовується декларативний підхід, натхненний веб-фреймворком ReactJS, на основі віджетів (в світі веба іменованих компонентами). Для ще більшого приросту в швидкості роботи інтерфейсу віджети перемальовуються **за** необхідності — тільки коли в них щось змінилося (подібно до того як це робить Virtual DOM в світі веб-фронтенда).

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Алгоритм роботи додатка

Крок 1: При першому запуску додатка, користувача буде зустрічати вікно з дозволом на використання геоданих, для автоматичного визначення місця знаходження.

Крок 2: Якщо користувач не надав доступу до його геоданих, додаток виводить вікно для встановлення, свого поточного місцезнаходження вручну.

Крок 3: Якщо користувач не надав доступу до геоданих, та відмовився вводити своє місце знаходження вручну, його перекине на сторінку з помилкою про відсутність даних про його місце знаходження.

Крок 4: Якщо користувач надав доступ на використання геоданих, додаток перекидає користувача на стартову сторінку з інформацією про погоду на поточний час.

Крок 5: Якщо підчас завантаження стартової сторінки, у користувача виникнуть проблему з інтернетом, або API яке надає інформацію про погоду буде не доступне, користувачу виведеться вікно з попередження про помилку та можливі її рішення, або можливість повторного завантаження інформації про погоду.

Крок 6: При вдалому завантаженні інформації про погоду, користувача зустрічає стартова сторінка, на якій знаходиться інформація про погоду, меню з перемиканням між інформацією про погоду на даний момент, погодинно, та прогнозом на 5 днів, також на цій сторінці є AppBar, на якому відображається, поточне місцезнаходження користувача, та перемикач тем та мови додатка.

Крок 7: При переході на сторінку з інформацією про погоду на тиждень, відображається список з картками з даними про погоду на всі дні тижня.

Крок 8: При переході на сторінку з інформацією про погоду на цілий день, для користувача виводиться список розбитий погодинно з періодом в 3 години, з картками в який детально описано інформацію про погоду на даний період часу.

Крок 9: Якщо в користувача виникне помилка на сторінці з інформацією про погоду на тиждень, не відпрацює API або буде поганий інтернет зв'язок, користувачу виведеться сторінка з помилкою.

Крок 10: Якщо користувач попаде на сторінку з помилкою йому виведеться детальна інформація про помилку, та можливість повторного завантаження даних про погоду.

Крок 11: По кліку на текстову кнопку поточного місцезнаходження, що знаходиться в AppBar, користувачу відобразиться вікно для зміни свого місцезнаходження вручну, та оновлення даних про погоду.

Крок 12: Після повторного завантаження даних про погоду, користувача перекидає на сторінку з інформацією про погоду.

Крок 13: При перемиканні теми, відбувається перемикання сторінки, на вибраний режим теми користувачем.

Крок 14: При виборі користувачем іншої мови, яку підтримує додаток, додаток перелаштовується під вибрану мову.

3.2 Чиста архітектура

З огляду на мету зменшення трудовитрат на розробку складного програмного забезпечення, припустимо, що необхідно використовувати готові уніфіковані рішення. Адже шаблонність дій полегшує комунікацію між розробниками, дозволяє посилаючись на відомі конструкції, знижує кількість помилок.

Патерн (англ. Design pattern) – архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого часто виникає контексту

Почнемо з головного Чиста архітектура - це концепція побудови архітектури систем, запропонована Робертом Мартіном (також відомого як "дядечко Боб"). Концепція передбачає побудову додатка у вигляді набору незалежних шарів, що полегшує тестування, зменшує зв'язність і робить додаток більш простим для розуміння.



Рисунок 3.1 – Діаграма Чистої Архітектури

Особливості Чистої Архітектури:

- Незалежність від фреймворку. Архітектура залежить від існування будь-якої бібліотеки. Це дозволяє використовувати фреймворк як інструмент замість того, щоб втискати свою систему в рамки його обмежень.

- Тестованість. Бізнес-правила можуть бути протестовані без інтерфейсу користувача, бази даних, веб-сервера або будь-якого іншого зовнішнього компонента.

- Незалежність від UI. Інтерфейс користувача можна легко змінити, не змінюючи решту системи. Наприклад, веб-інтерфейс може бути замінений на консольний без зміни бізнес-правил.

- Незалежність від бази даних. Є можливість змінити Oracle або SQL Server на MongoDB, BigTable, CouchDB або ще щось. Ваші бізнес-правила не пов'язані з базою даних.

- Незалежність від будь-якого зовнішнього сервісу. За фактом, бізнес правила просто нічого не знають про зовнішній світ.

Правило Залежностей

Концентричні кола на діаграмі зображені на рис. 4.1 є різними шарами програмного забезпечення. У загальному випадку, чим далі йти, тим загальнішим стає рівень. Зовнішні кола – механіка. Внутрішні кола – політика.

Основним правилом, що робить цю архітектуру чинною є Правило Залежностей. Це правило свідчить, що залежності у вихідному коді можуть вказувати тільки усередину. Ніщо з внутрішнього кола не може знати щось про зовнішнє коло, ніщо з внутрішнього кола не може вказувати на зовнішнє коло. Це стосується функцій, класів, змінних тощо.

Більше того, структури даних, що використовуються у зовнішньому колі, не повинні бути використані у внутрішньому колі, особливо якщо ці

структури генеруються фреймворком у зовнішньому колі. Ми не використовуємо нічого із зовнішнього кола, щоб могло вплинути на внутрішній.

Сутності

Сутності визначаються бізнес-правилами підприємства. Сутність може бути об'єктом з методами або вона може бути набором структур даних і функцій. Не має значення, як довго сутність може бути використана в різних додатках.

Сценарії

У цьому прошарку реалізується специфіка бізнес-правил. Він інкапсулює та реалізує всі випадки використання системи. Ці сценарії реалізують потік даних і з шару Сутностей для реалізації бізнес-правил.

Ми не очікуємо зміни в цьому шарі, що впливають на Сутність. Ми також не очікуємо, що цей шар може бути порушений зовнішніми змінами, таких як бази даних, інтерфейсом користувача або фреймворком. Цей шар ізольований від таких проблем.

Ми, однак, очікуємо, що зміни в роботі програми вплине на сценарії. Якщо будуть якісь зміни в поведінці програми, то вони, безперечно, торкнуться коду в даному шарі.

Інтерфейс-адаптери

Програмне забезпечення в цьому шарі є набором адаптерів, які перетворюють дані з формату найбільш зручним для Сценаріїв і Сутностей, у формат найбільш зручний для подальшого використання, наприклад в БД. Саме цей шар, наприклад, повністю міститиме архітектуру MVC. Моделі є, швидше за все, структурами даних, які передаються від контролерів до Сценарій, а потім назад зі Сценаріїв до Уявлень.

Так само, дані перетворюються, у цьому шарі, з форми найбільш зручним для Сценаріїв і Сутностей, у форму, найбільш зручною для постійного зберігання, наприклад, у базі даних. Код, що знаходиться

всередині цього кола, не повинен знати щось про БД. Якщо БД - це SQL база даних, то будь-які SQL-інструкції не повинні бути використані на цьому рівні.

Фреймворки та драйвери.

Зовнішній шар зазвичай складається із фреймворків, БД, UI тощо. Як правило, у цьому шарі не пишеться багато коду, за винятком коду, який спілкується із внутрішніми колами.

Це шар накопичення деталей. Інтернет – деталь, БД – деталь, ми тримаємо ці штуки зовні для зменшення їхнього впливу.

Дотримуючись цих простих правил не важко і вони збережуть багато часу в майбутньому. Розділяючи ПЗ на шари і дотримуючись Правилу Залежностей систему можна зробити легко тустуємою, з усіма перевагами. Коли будь-яка із зовнішніх підсистем застаріє, чи то БД, чи веб-фреймворк, можна буде легко замінити їх.

3.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи

Зважаючи на потреби та вимоги до додатка для моніторингу прогнозу погоди, я обрав мову програмування Dart та фреймворк Flutter для його розробки. Я обґрунтовую свій вибір наступними технічними та бізнес-аспектами:

- Швидкість розробки:

Flutter має швидку ітераційну розробку, що дозволяє розробляти додатки швидше порівняно з іншими фреймворками.

- Кросплатформеність:

Flutter дозволяє розробляти додатки для iOS та Android на одній мові програмування та з одним кодом, що зменшує затрати на розробку та підтримку додатків. Це дозволить мені ефективно використовувати свій

час та збільшити свою аудиторію, надаючи додаток користувачам обох платформ.

- Розширюваність та гнучкість: Dart має високу розширюваність та дозволяє розробникам створювати власні пакети, що значно полегшує розробку та підтримку додатків. Це дозволить мені швидко та ефективно вирішувати завдання під час розробки додатка та забезпечить його більшу гнучкість в майбутньому.

- Популярність та підтримка: Flutter є одним з найбільш популярних фреймворків для розробки мобільних додатків, тому він має велику спільноту розробників та підтримку від компанії Google. Це дозволить мені легко знайти відповіді на запитання та отримати підтримку у разі потреби.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Розробка блок-схеми, яка підлягає програмуванню

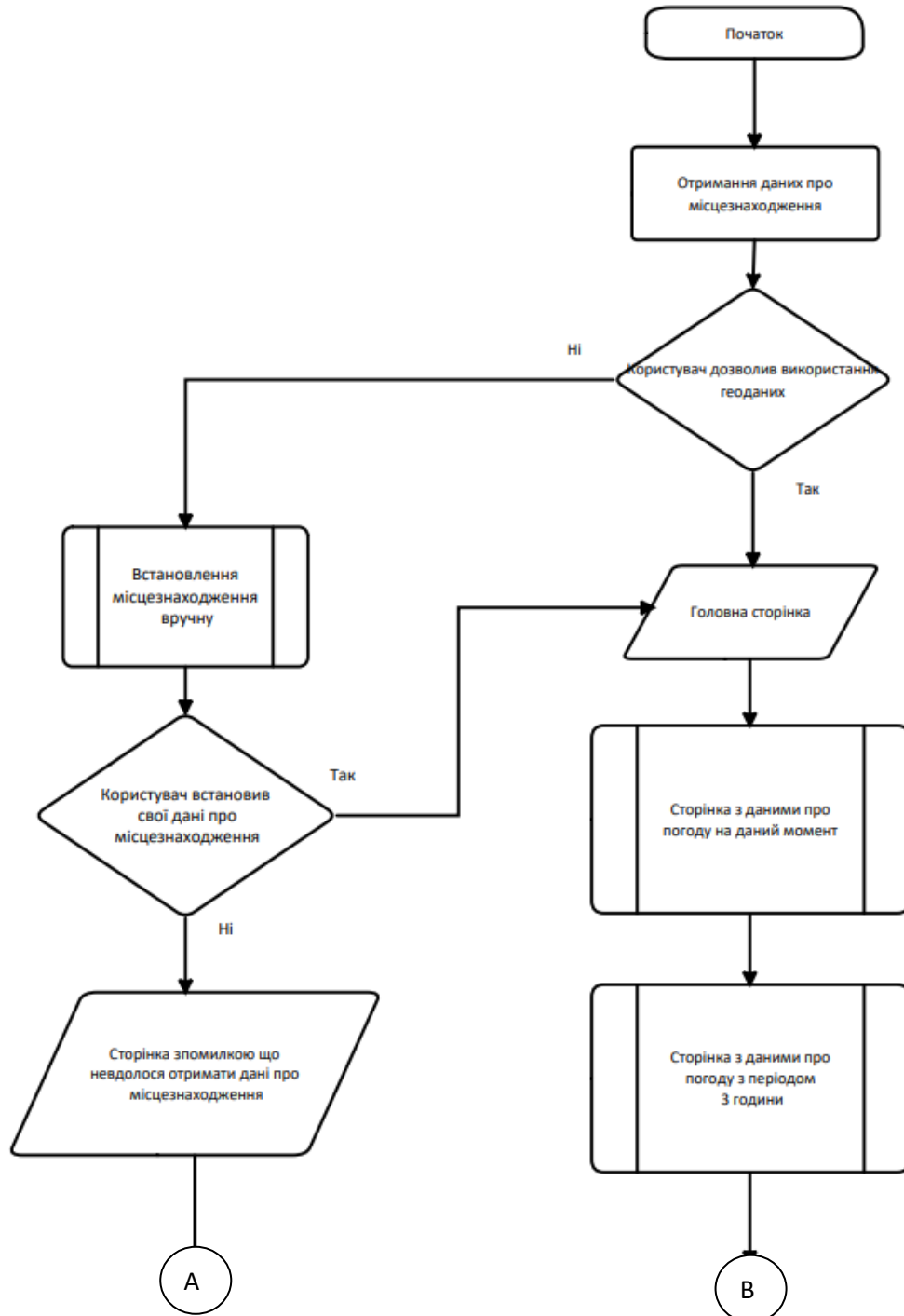


Рисунок 4.1 – Блок-схема роботи додатка для Моніторингу прогнозу погоди (Перший аркуш)

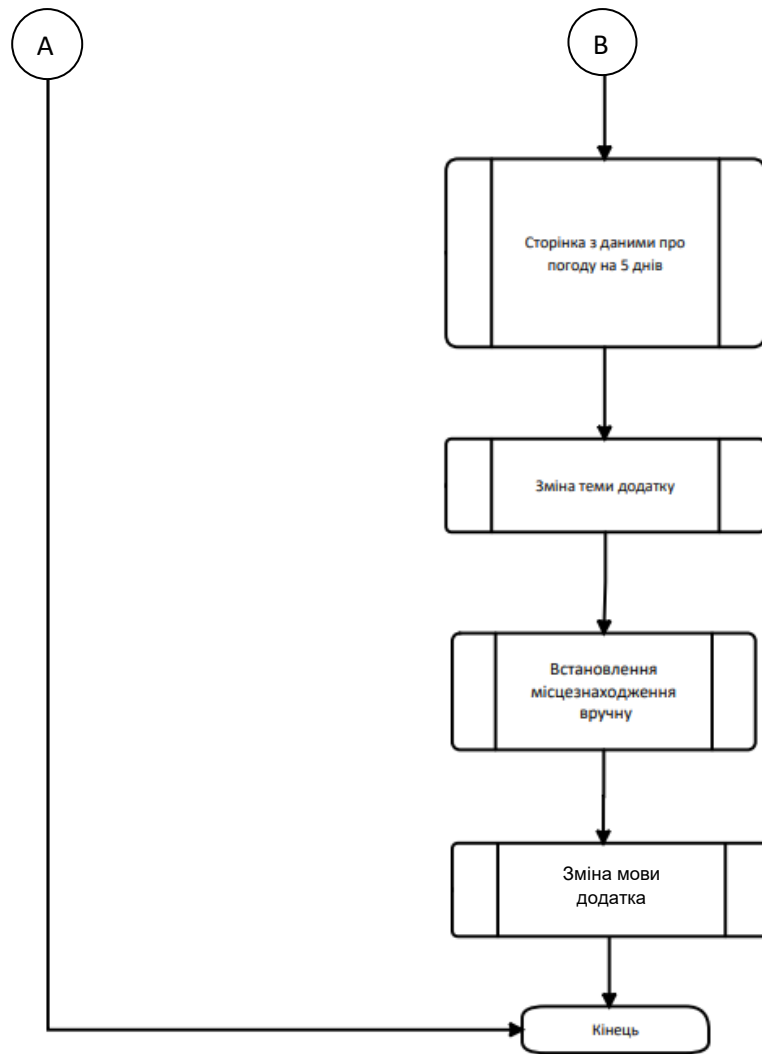


Рисунок 4.2 – Блок-схема роботи додатка для Моніторингу прогнозу погоди (Другий аркуш)



Рисунок 4.2 – Блок-схема зміни мови додатка



Рисунок 4.3 – Блок-схема зміни теми додатка

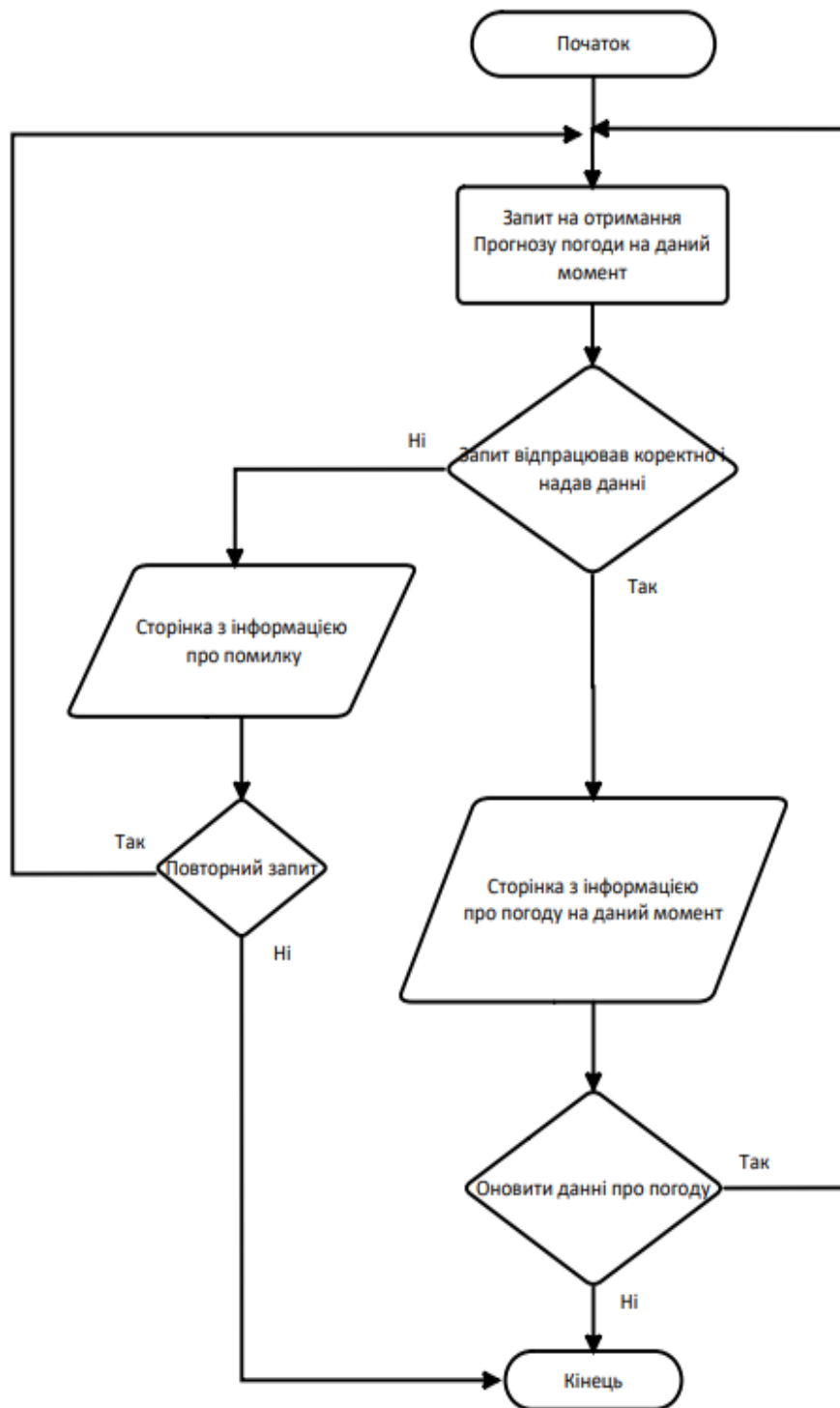


Рисунок 4.5 – Блок-схема відкриття сторінки з даними про погоду на даний момент

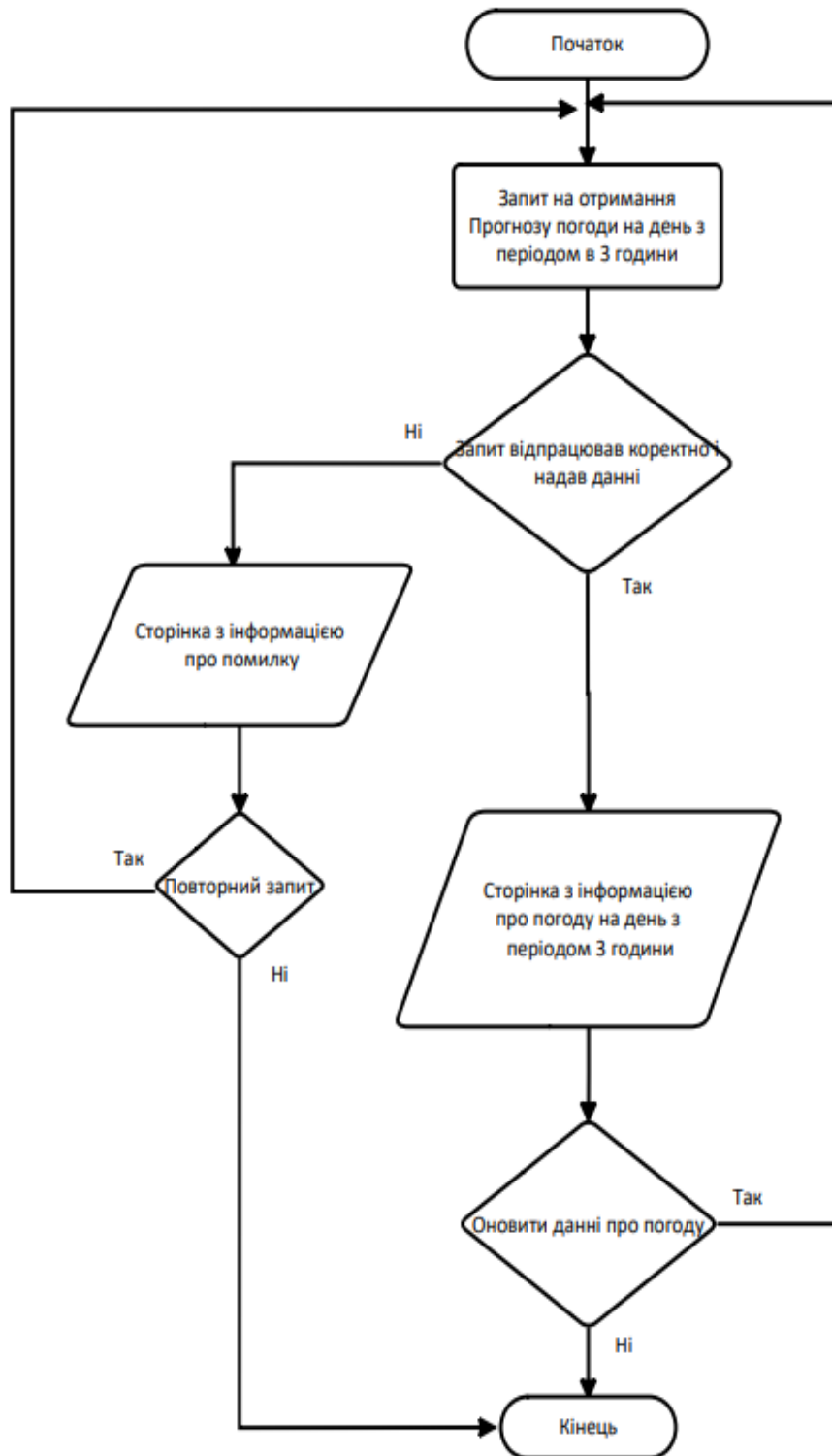


Рисунок 4.6 – Блок-схема відкриття сторінки з даними про погоду на день з періодом 3 години

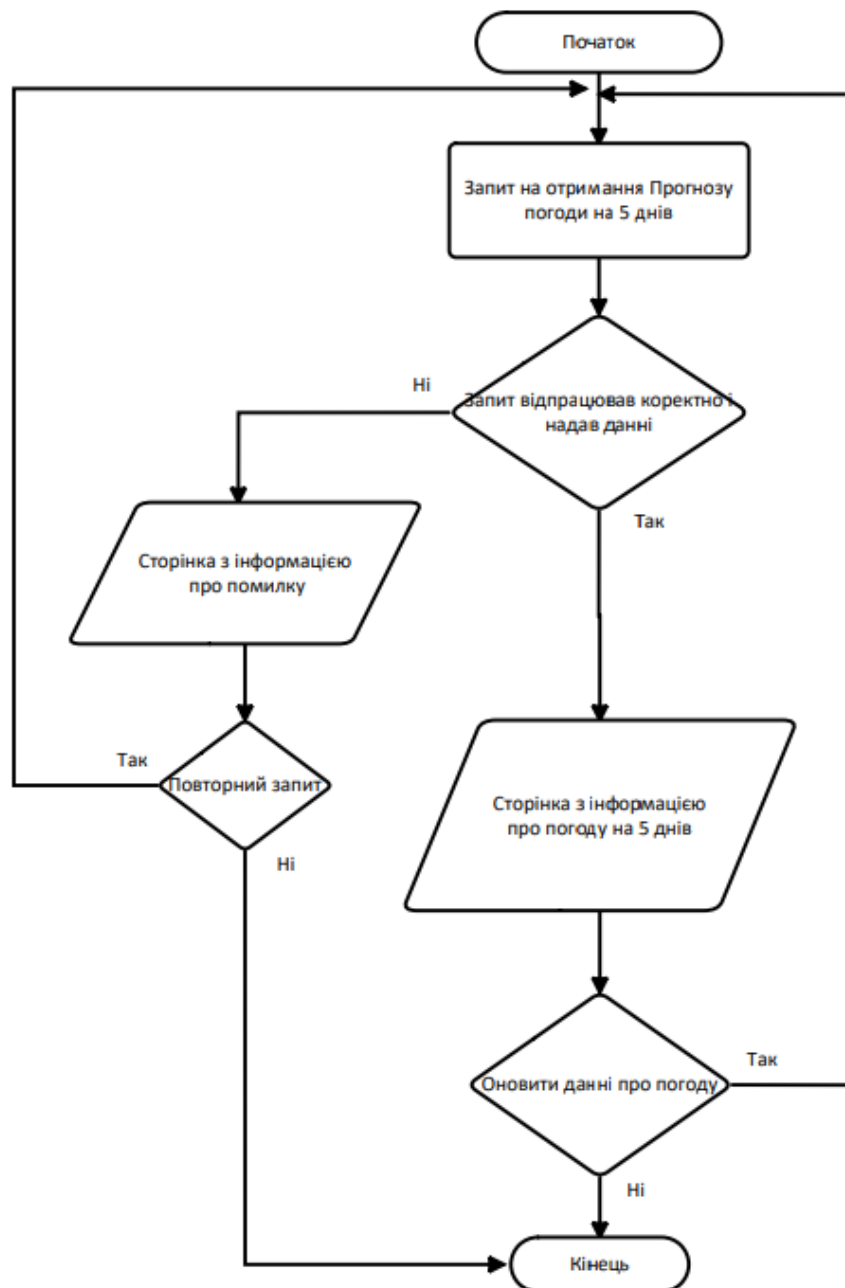


Рисунок 4.7 – Блок-схема
відкриття сторінки з даними про погоду на 5 днів

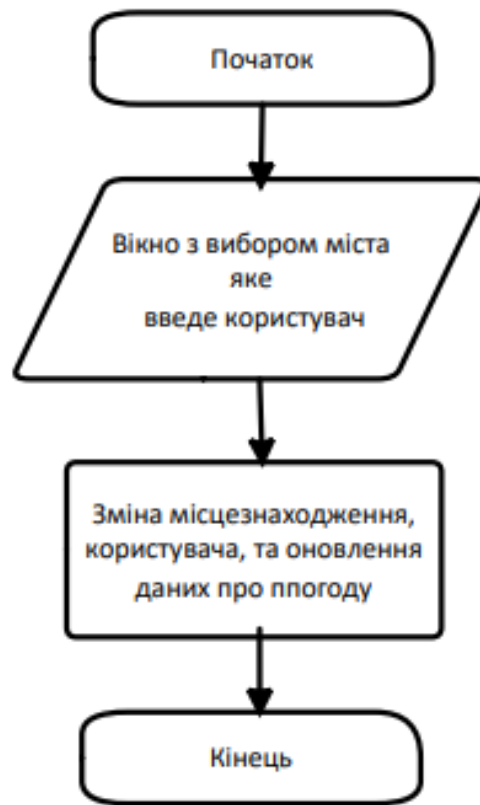


Рисунок 4.8 – Блок-схема встановлення місцезнаходження вручну

4.2. Опис процесу програмної реалізації

Розробка зовнішнього вигляду додатка відбувалася з використанням сучасних стандартів і принципів побудови призначеного для користувача інтерфейсу. В першу чергу орієнтованість спрямована на Material Design [2].

Це мова візуальних образів, який створила корпорація Google для уніфікації інтерфейсів її продуктів і сервісів.

Матеріальний дизайн (англ. Material design) – принципи дизайну сайтів, програмного забезпечення і додатків, а також правила дизайну інтерфейсів для операційної системи Android від компанії Google.

Вперше представлений на конференції Google I / O 25 червня 2014 року 49 Ідея дизайну полягає в інтерфейсі, поведінку і вигляд якого слідує правилам поведінки і виду паперових карт в реальному житті.

Дослідження, зроблене дизайнерами Google змінює підхід до інтерфейсу і робить його реальним. Інтерфейс перетворюється в живу приємну людині взаємодія.

Основні компоненти матеріального дизайну наведені на рис. 4.8:

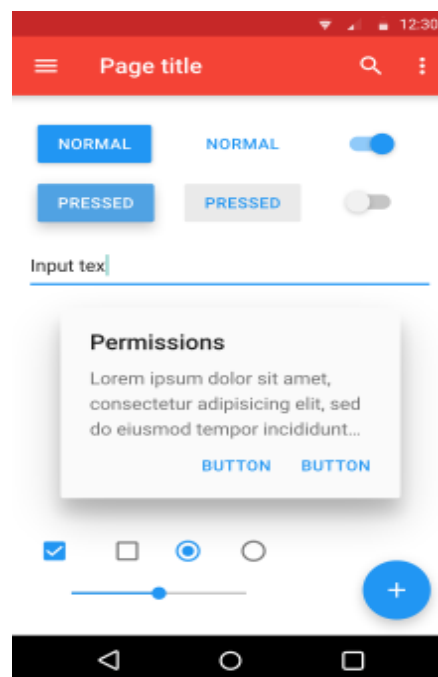


Рисунок 4.9 – Основні компоненти Material Design

Матеріальний дизайн за допомогою системи управління тінями створює візуальну ілюзію простору між додатком і екраном пристрою.

Основні принципи матеріального дизайну:

- була збільшена реалістичність анімації за допомогою прискорення і пригальмовування рухів, пружного стрибки об'єктів;
- розвинулися інтерактивні іконки;
- з'явилася об'єктна хореографія, при якій елементи здатні впливати на сусідні елементи.

Material design ґрунтується на принципах друкованого дизайну. І не тільки для краси, а й для розстановки акцентів і фокусування уваги користувача на потрібному елементі, для спрощення навігації серед ієрархії конструкцій інтерфейсу, для інтуїтивної передачі їх сенсу. На діях користувача сфокусовано основну увагу. Взаємодією з дизайном керує призначений для користувача досвід, а не навпаки. Всі дії відбуваються в одному оточенні, інтерактивні об'єкти без переривання переходять з одного середовища в інше.

Крім того додаток розроблявся відповідно до вимог зручності та використовуваних призначеного для користувача інтерфейсу, ґрунтуючись на UX– дизайні і естетику зовнішнього вигляду.

Нижче будуть наведені скріншоти екранів додатка.

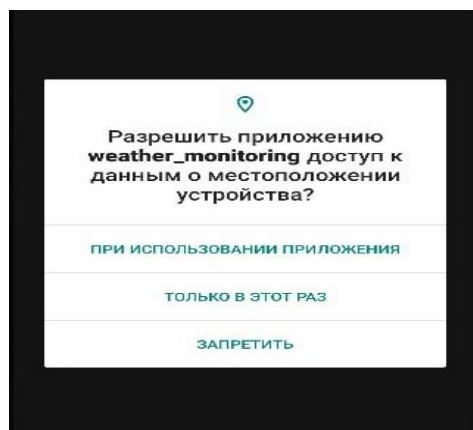


Рисунок 4.10 – вікно дозволу на використання геоданих.

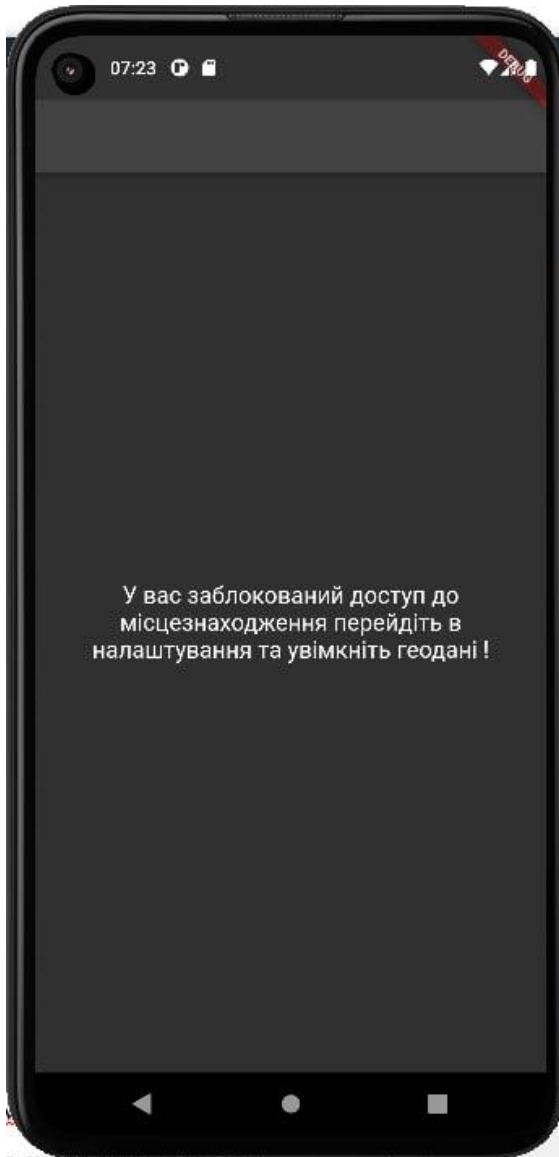


Рисунок 4.11 – Вікно попередження про заблоковані геодані.

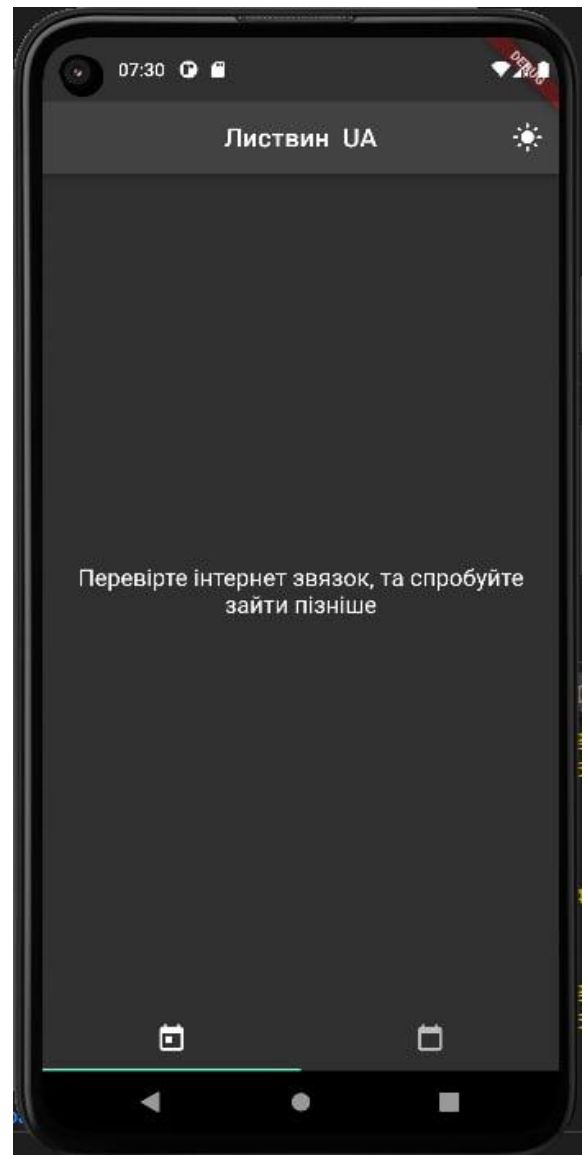


Рисунок 4.12 – Вікно попередження можливу відсутність інтернету.



Рисунок 4.13 – Стартова сторінка з інформацією про погоду, в світлій темі

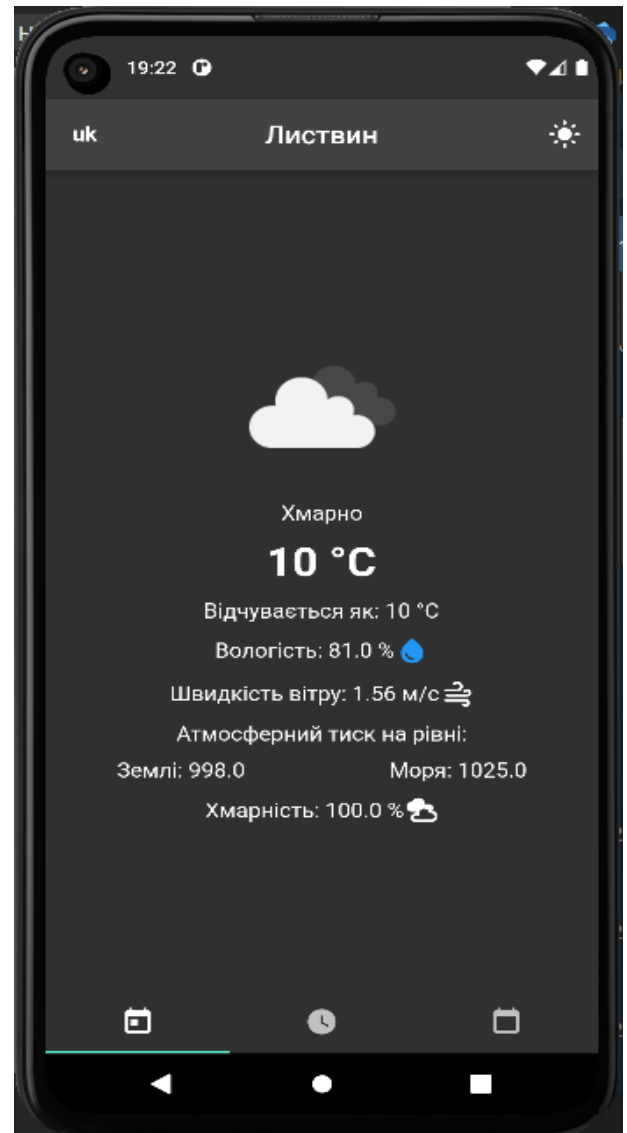


Рисунок 4.14 – Стартова сторінка з інформацією про погоду, в темній темі

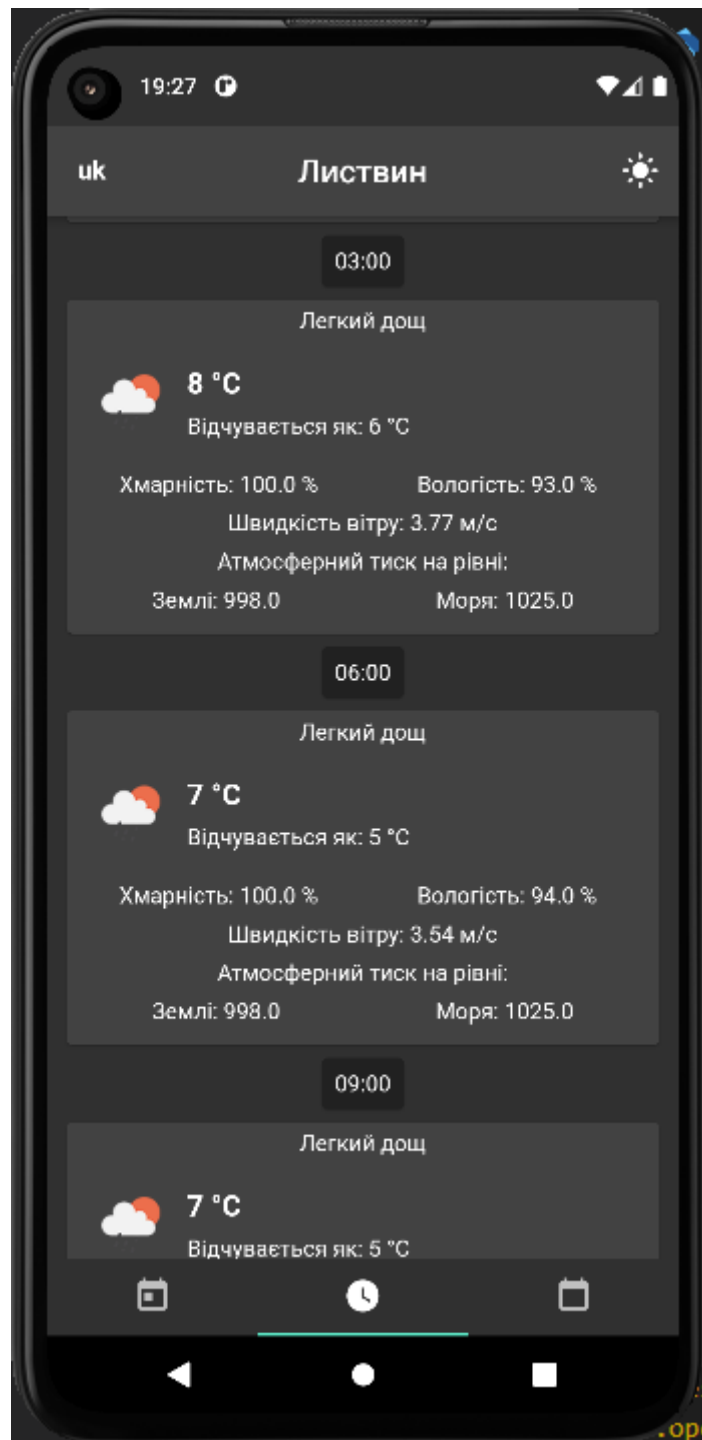


Рисунок 4.15 - Сторінка з інформацією про погоду на день з періодом кожні 3 години

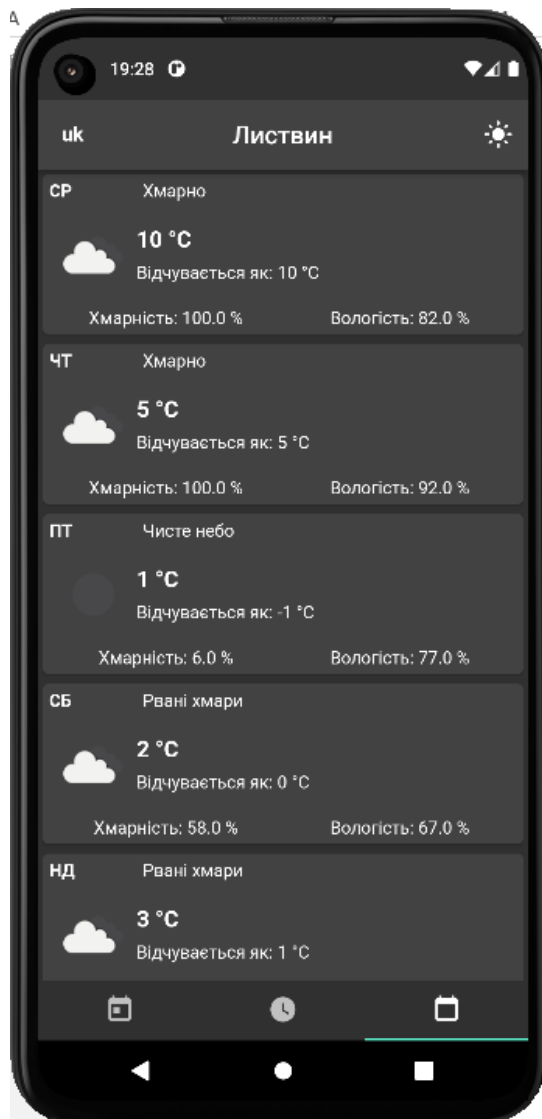


Рисунок 4.16 - Сторінка з інформацією про погоду на 5 днів

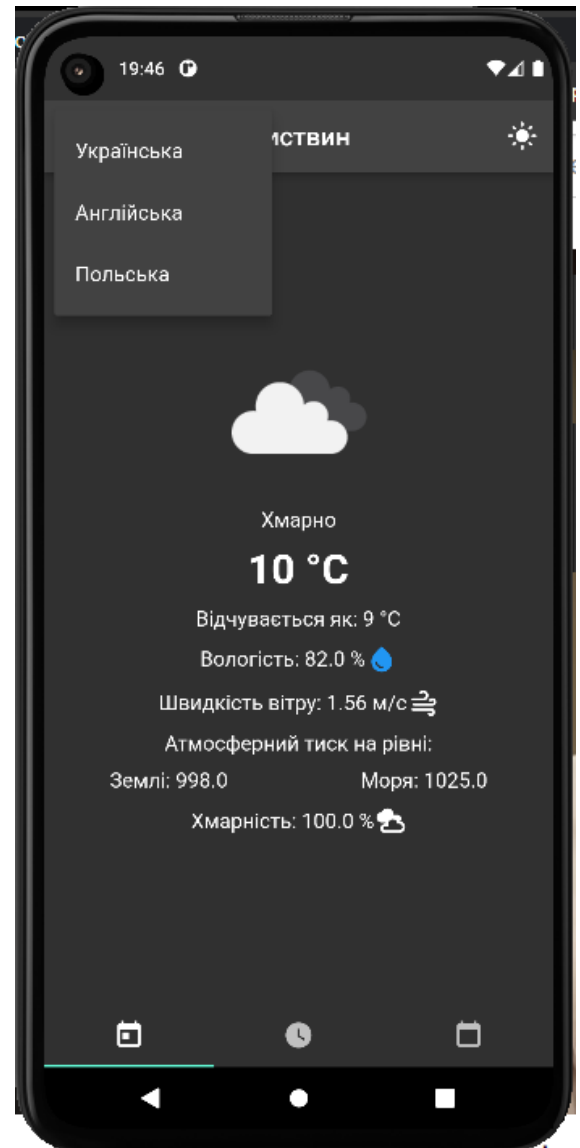


Рисунок 4.17 – Попап зі списком підтримуємих мов додатка

4.3 Опис роботи додатка

При першому завантаженні додатка користувача зустрічає вікно з дозволом на використання даних про місцезнаходження (Рис. 4.18), якщо користувач заборонить використання його перекине на сторінку з помилкою (Рис 4.19).

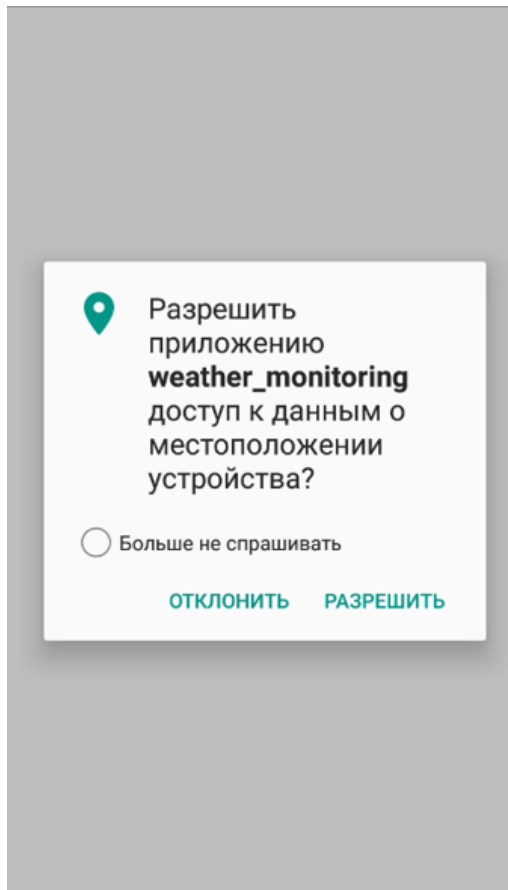


Рисунок 4.18 – Вікно на підтвердження використання даних про місцезнаходження

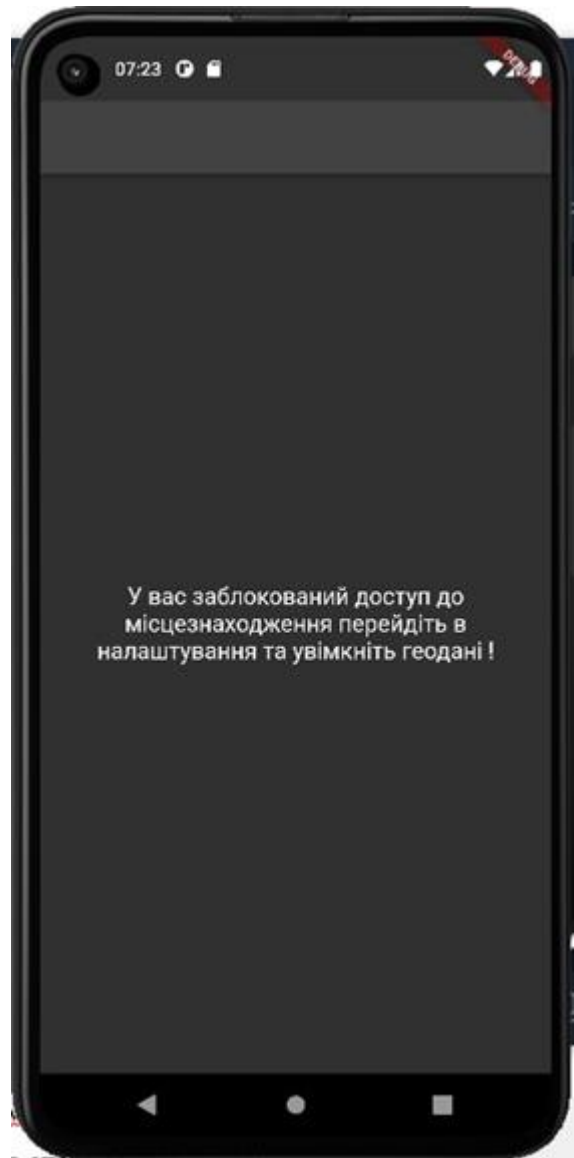


Рисунок 4.19 – Вікно попередження про заблоковані геодані.

Після завантаження додатка, користувача зустрічає головна сторінка додатка, по замовчуванню відображається сторінка з прогнозом погоди на даний момент (Рис. 4.20), також на головній сторінці є перемикач між сторінками (Рис. 4.21), в користувача є можливість отримати дані про погоду на даний момент, прогноз на день з періодом в 3 години (Рис. 4.22), та прогноз погоди на 5 днів (Рис. 4.23).



Рисунок 4.20 – Сторінка з прогнозом погоди на даний момент



Рисунок 4.21 – перемикач між сторінками

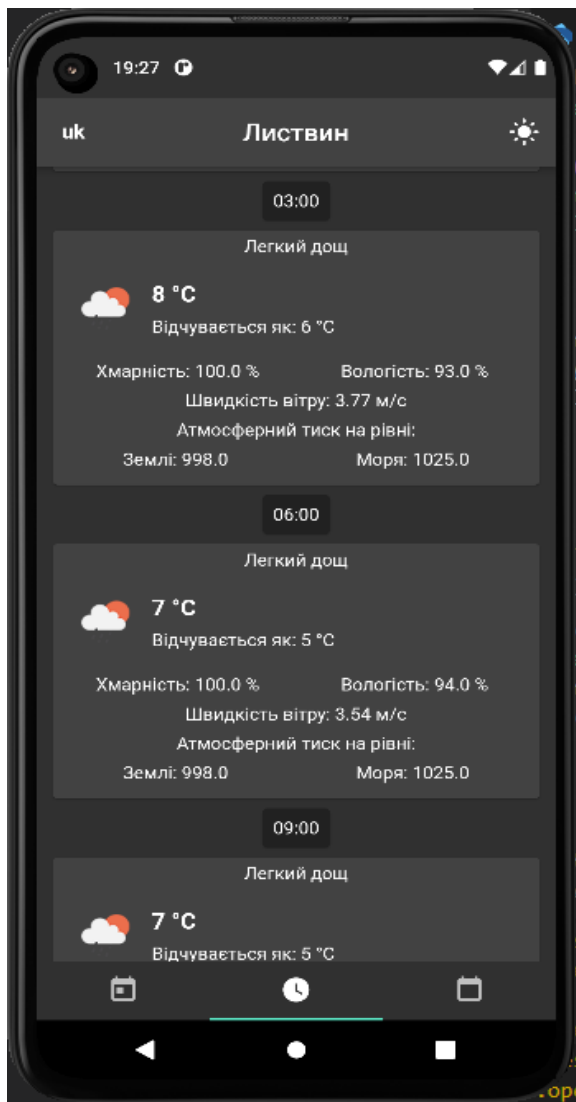


Рисунок 4.22 - Сторінка з інформацією про погоду на день з періодом кожні 3 години



Рисунок 4.23 – Сторінка з інформацією про погоду на 5 днів

Також в додатка присутній перемикач тем з світлої на темну (4.24).

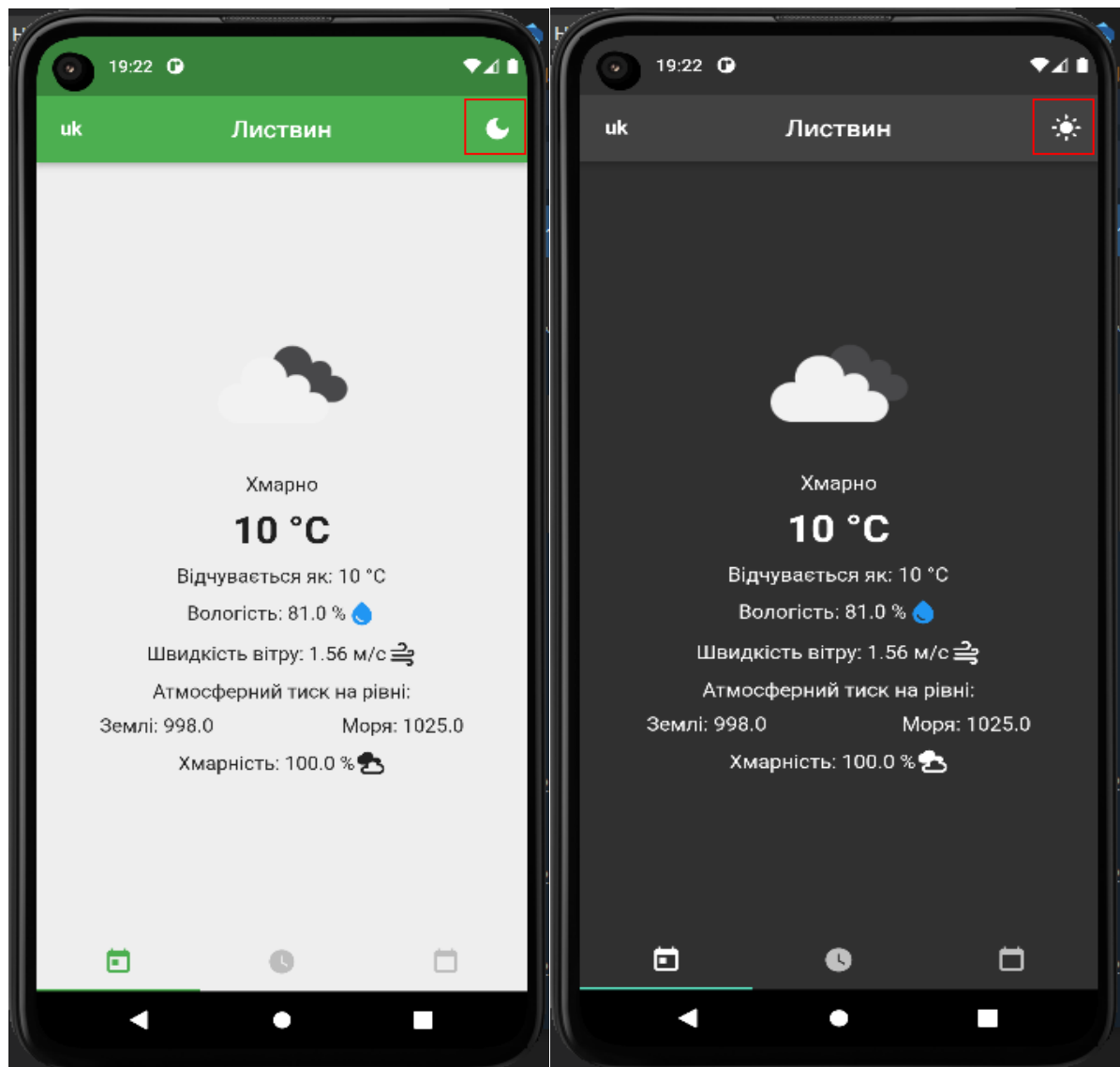


Рисунок 4.24 – Перемикач між світлою та темною темою

Щоб змінити мову додатка, на головному екрані присутній спливаючий список з доступними мовами (Рис. 4.25), на даний момент додаток підтримує такі мови як: Українську, Англійську, Польську.

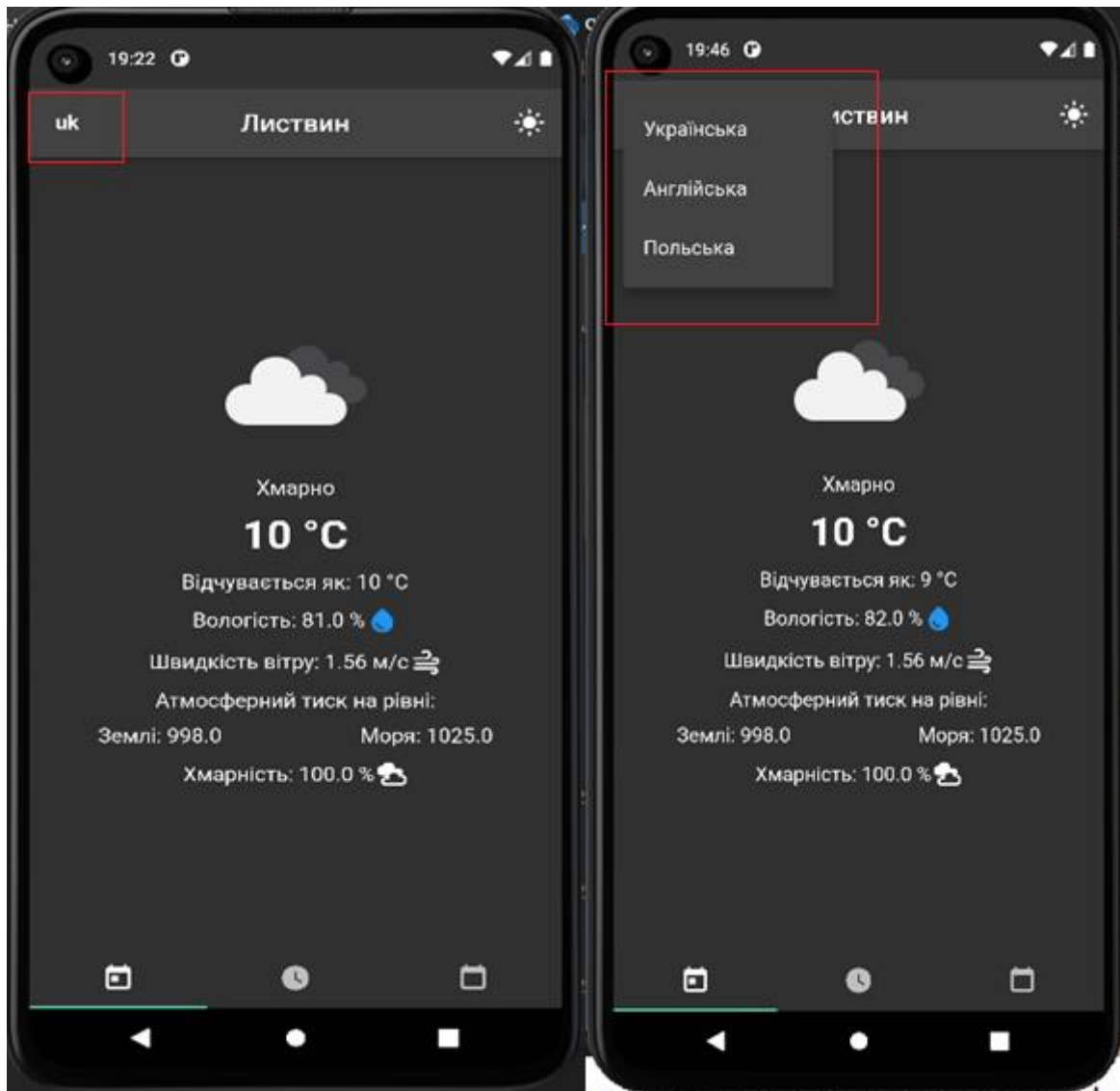


Рисунок 4.25 – Спливаючий список з доступними мовами додатка

Також в додатка присутня можливість зміни місце знаходження вручну, та оновлення даних по погоді, щоб змінити місто, користувач повинен натиснути на текстову кнопку що знаходиться в AppBar (Рис. 4.26), після чого перед ним з'явиться вікно для введення міста, та список з результатами пошуку (Рис. 4.27), після вибору міста із списку пошуку, додаток оновлює дані про погоду.

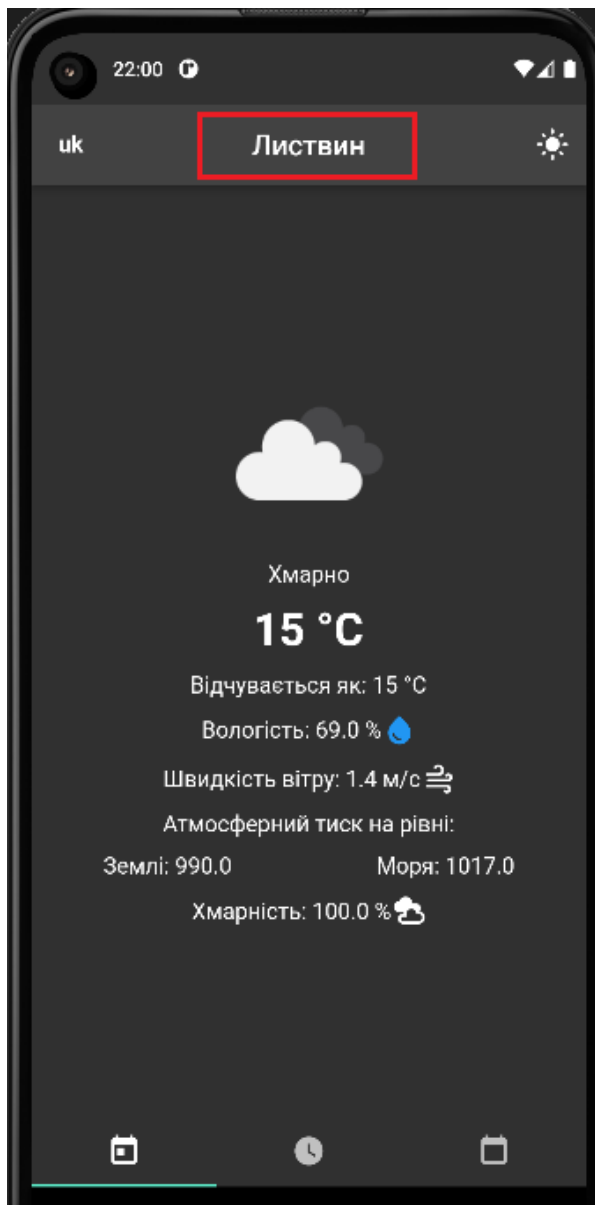


Рисунок 4.26 – Текстова кнопка для виклику вікна для зміни місце знаходження вручну

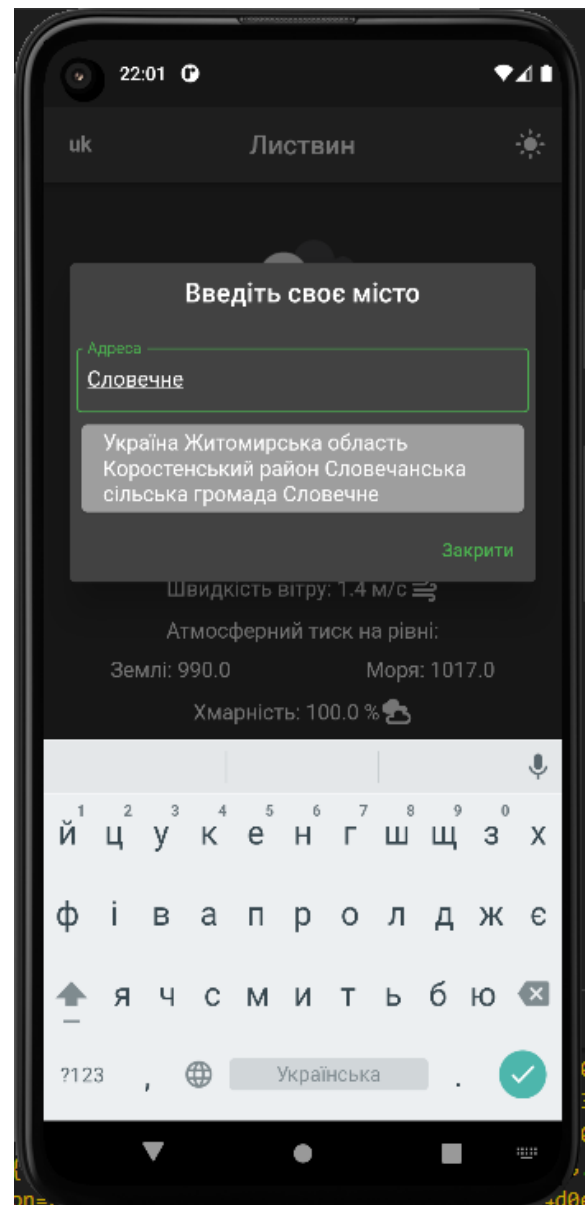


Рисунок 4.27 – Вікно з пошуком міста, та вибором результату, для оновлення місце знаходження

ВИСНОВКИ

У ході створення додатка були виконані наступні дії:

- проаналізована та описана предметна область;
- розглянуті програмні аналоги, що існують на ринку;
- підтверджена актуальність додатка;
- встановлено мету;
- розглянуті, проаналізовані та обрані програмні засоби для реалізації додатка;
- розроблено алгоритм роботи додатка;
- розроблена візуальна частина додатка;
- протестовано роботу додатка;

Для реалізації програмного забезпечення було проведено об'ємну роботу по взаємодії між елементами середовища Android Studio. Для цього була використана допоміжна література. За допомогою інструментів, що надаються Android Studio, був розроблено дизайн, та структура додатка.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сторінка додатка Погода&Радар в Google Playstore. Режим доступу: <https://play.google.com/store/apps/details?id=de.wetteronline.wetterapp&hl=ru&gl=US>
2. Material design [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії. — Режим доступу: https://uk.wikipedia.org/wiki/Material_design
3. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра / О. В. Ольховська, О. О. Черненко. — Полтава : ПУЕТ, 2022. — 67 с. — 1 електрон. опт. диск (CVD-ROM).
4. Історія розвитку метеорології і кліматології [Електронний ресурс] — Режим доступу: <https://jak.koshachek.com/articles/istorija-pogodi-sposterezhennja-za-pogodoju-i.html>
5. Dart [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії. — Режим доступу: <https://uk.wikipedia.org/wiki/Dart>
6. Flutter [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії. — Режим доступу: <https://uk.wikipedia.org/wiki/Flutter>
7. Інформація про основи Flutter [Електронний ресурс] — Режим доступу: <https://devzone.org.ua/post/pro-flutter-korotko-osnovi>
8. Роберт Сесіл Мартін. Чиста архітектура. Мистецтво розробки програмного забезпечення. 2018. 410с. [Електронний ресурс] - Режим доступу: http://loveread.ec/view_global.php?id=88325
9. Олександр Швець. Занурення в ПАТЕРНИ ПРОЕКТУВАННЯ. 2021. 396с.