

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

Дипломний проект

здобувача освіти денної форми навчання

РП.05.16.000.ДП

ПАЛЬЧЕНКА

ДАНИЛА

ОЛЕГОВИЧА

м. Одеса

2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: **«Розробка програмного забезпечення»**

Група: **4РП-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка мобільного додатку під ОС Android

Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах).

Дипломник _____ (Пальченко Д.О.)

Керівник _____ (Іванова Л.В.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « ____ » _____ 2022 р. Протокол ЕК № _____

Оцінка ЕК _____

Секретар ЕК _____

Анотація

Тема дипломного проекту «Розробка мобільного додатку для ОС Android».

У цій роботі будуть описані основні принципи розробки сучасних мобільних додатків, процес підбору інструментів, аналіз технологій розробки та впровадження програми.

Аналітична частина: аналіз використання додатків. Технічне завдання, актуальність розробки, методи і засоби розвитку. Опис життєвого циклу програми, середовища розробки IDE, графічного дизайну, стилю програми, елементів та навігації.

Розділ впровадження: описує основні етапи впровадження МД, архітектури, написання та тестування мобільного додатку, його роботи. Останніми розділами були охорона праці та економічні розрахунки, під час розробки та конструюванні програми.

Ключові слова: Мобільний додаток, смартфон, розробка, Android Studio, інструмент, методи, UI, UX, Android.

Annotation

The topic of the diploma project "Development of a mobile application for Android".

This paper will describe the basic principles of development of modern mobile applications, the process of selecting tools, analysis of technology development and implementation of the program.

Analytical part: analysis of the use of applications. Terms of reference, relevance of development, methods and means of development. Description of the program life cycle, IDE development environment, graphic design, program style, elements and navigation.

Implementation section: describes the main stages of MD implementation, architecture, writing and testing of a mobile application, its work. The last sections were labor protection and economic calculations during the development and design of the program.

Keywords: Mobile application, smartphone, development, Android Studio, tool, methods, UI, UX, Android.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.
“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Пальченко Данило Олегович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка мобільного додатку під ОС Android

затверджена наказом по коледжу від “30” грудня 2021р. № 306-А2-ОД

2. Термін здачі закінченого проекту (роботи) 20.06.2022р.

3. Вихідні данні до проекту (роботи) _____

1. Технічне завдання

2. Програмне забезпечення для створення мобільних додатків

3. Вимоги до функціоналу та графічного дизайну;

4. Засоби тестування мобільних додатків

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Вступ. 1. Інформаційні технології. Аналіз технологій розробки мобільних додатків 2. Розробка мобільних додатків під ос ANDROID 3. Реалізація мобільного додатку. Економічні розрахунки . Охорона праці. Висновок. Список використаних джерел інформації.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

1. Архітектура мобільного додатку

2. Алгоритм роботи мобільного додатку

3. Схема взаємодії контенту мобільного додатку

4. Графічний дизайн мобільного додатку

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Основний</i>	<i>Іванова Л.В.</i>	<i>30.11.2021</i>	<i>17.05.2022</i>
<i>Економіка</i>	<i>Копайгородська Т.Г.</i>	<i>4.05.2022</i>	<i>17.05.2022</i>
<i>Охорона праці</i>	<i>Чорновол В.І.</i>	<i>4.05.2022</i>	<i>17.05.2022</i>
<i>Нормоконтроль</i>	<i>Петрашова В.І.</i>	<i>4.05.2022</i>	<i>17.05.2022</i>
<i>Старший консультант</i>	<i>Скорнякова О.В.</i>	<i>4.05.2022</i>	<i>17.05.2022</i>

7. Дата видачі завдання _____ 30.11.2021 _____

Керівник

Іванова Л.В.

(підпис)

Завдання прийняв до виконання

Пальченко Д. О.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	<i>Передмова</i>	<i>4.05.2022</i>	
2.	<i>Мобільні додатки у бізнесі та суспільстві</i>	<i>8.05.2022</i>	
3.	<i>Розробка технічного завдання для додатку</i>	<i>10.05.2022</i>	
4.	<i>Розробка та реалізація мобільного додатку</i>	<i>15.05.2022</i>	
5.	<i>Тестування та публікація мобільного додатку</i>	<i>17.05.2022</i>	
6.	<i>Економічні розрахунки</i>	<i>22.05.2022</i>	
7.	<i>Розділ охорони праці</i>	<i>26.05.2022</i>	
8.	<i>Висновок</i>	<i>28.05.2022</i>	
9.	<i>Перелік літератури</i>	<i>31.05.2022</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>2.06.2022</i>	
11.	<i>Оформлення графічної частини</i>	<i>10.06.2022</i>	
12.	<i>Малий захист кваліфікаційної роботи</i>	<i>15.06.2022</i>	
13.	<i>Захист кваліфікаційної роботи</i>	<i>18.06.2022</i>	

Дипломник Пальченко Д.О.

(підпис)

Керівник Іванова Л.В.

ЗМІСТ

	Стр.
<i>ПЕРЕДМОВА</i>	9
<i>1. АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ.</i>	11
1.1 Мобільні додатки в реаліях громадського користування.	11
1.2 Актуальність розробки мобільних додатків.	13
1.3 Методи і засоби розробки мобільного додатку під ОС <i>Android</i> .	15
1.4 Технічне завдання на розробку мобільного додатку.	18
<i>2. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ПІД ОС ANDROID.</i>	19
2.1 Вибір схеми моделі життєвого циклу.	19
2.2 Визначте завдання та функції мобільного додатку.	23
2.3 Вибір середовища розробки.	23
2.4 Вибір графічного дизайну інтерфейсу користувача, дизайну і стилю програми, елементів і навігації.	29
2.5 Вибір засобів розробки та тестування мобільного приложення.	34
<i>3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ.</i>	33
3.1 Архітектура, інтерфейс і функціональність мобільного приложення.	34
3.2 Написання та тестування мобільного додатку.	34
<i>4. ЕКСПЛУАТАЦІЯ ДОДАТКУ ТА ЙОГО ЗАПУСК.</i>	43
5. ЕКОНОМІЧНІ РОЗРАХУНКИ	46
6. ОХОРОНА ПРАЦІ.	53
ВИСНОВОК	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ.	59

ПЕРЕДМОВА

Мобільний додаток — це спеціальне розроблене програмне забезпечення, призначене для запуску та функціонування на пристроях смартфонах, планшетах та інших пристроїв, розроблене для конкретної платформи ОС смартфонів (Android, IOS та ін.).

Послуги з розробки мобільних програм також можна розділити залежно від платформи, для якої буде створюватися програма. До основних послуг входить: Розробка програм для iOS, Розробка програм для Android, Розробка VR, AR, MR програм.

У призначенні такого ПЗ може бути різним, може ділитися на різні категорії, залежно для якого програма створюється, приклад таких категорій може бути: розважальне, онлайн магазини, послуги, онлайн помічники. Такі програми та програми можуть бути вже представлені / передбачені системою додатків, чи це додаток системного характеру, або звичайне WeatherApp додаток на наш смартфон, такі програми можуть бути завантажені з представлених великих майданчика додатків (маркетплейси) - Play Market, App Store, Amazon AppStore (та менш популярні аналоги).

Кожна людина користується смартфоном та представленими ними додатками, вона стала основним пристроєм контролю, який допомагає та спрощує нашу інформаційно-технологічну сферу життя, роблячи її цікавішим та захоплюючим.

За допомогою таких APP ми не тільки можемо спілкуватися з другом у месенджері, замовляти їжу через програму, здійснювати пошук туру через програму на смартфоні, бронювати місце в таксі/кіно, але й замовляємо товари з магазинів, бронюємо житло, використовуємо телефони як навігатори, фото - і відеокамери, читалки, онлайн банки, і просто як спосіб розважитися та скоротити час.

Мобільні Програми є більш популярними, ніж аналогічні веб-сайти, оскільки більш зручні. Вони забезпечують кращу взаємодію користувача, швидше завантажують контент, простіше у використанні. Сьогодні мобільні програми та їх аналоги міцно закріплені в нашому житті.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		9

Сфера загального спрямованого використання МД. Багато власників бізнесу зосереджені на розробці додатку і отриманні прибутку від своїх ідей. Такі підприємства орієнтуються на потреби клієнта, постійного прогресу і вдосконалення свого продукту. Мобільне рішення здатне охопити широкий спектр маркетингових завдань. Забезпечує покращену видимість ринку для малого, середнього та великого бізнесу. Наявність додатку на смартфоні клієнта дозволяє збирати дані про додаток, розширюючи можливості професійної взаємодії. Який стане прямим маркетинговим каналом, необхідність постійного розвитку бізнесу.

Більшість малих підприємств покладаються на традиційні методи маркетингу, бізнесу потрібен необхідний час для прийняття все нових і нових методів маркетингу, розробки стратегій для нового плану, який дозволить охопити коло потенційних споживачів. Середній і великий бізнес покладаються на прописаний план, завжди починаючи і вдосконалюючи його, це зменшує реалізацію нової ідеї, в період прийняття нових методів маркетингу. Додаток виділяється на тлі конкуренції і стимулює продажі. Впровадження для малого бізнесу допоможе продавати продукти та послуги онлайн набагато швидше. Це може допомогти у створенні популяризації бренду. Додаток може бути ключовим елементом кампанії з впізнаваності бренду. Всі дані, зібрані в додатку, можуть випередити кращу сферу прибутку, яка визначає майбутнє такої компанії. Такі знання можуть бути використані для підвищення якості обслуговування та виявлення нових можливостей. Важливо стежити за прогресом і своєчасно реагувати на потреби.

Погода в режимі реального часу. Кожен з нас перед походом на роботу, навчання, прогулку, не враховуючи цього, враховує свій зовнішній вигляд і комфорт, що носити сьогодні, чи буде день похмурим, якою буде температура, чи буде сильний вітер, піде дощ, або день буде сонячним. Прогнозні послуги - які мають точну інформацію про погоду в будь-який час і в будь-якому місці. Сервіс синоптика в додатку на телефоні, який може встановити кожен. У додатку можна дізнатися температуру, хмарність, умови протягом дня і багато іншого. Маючи таку інформацію про погоду, можна ретельно підготувати свій план протягом дня. Велика кількість користувачів користуються мобільними погодними додатками, що

зручніше, ніж Web-версія, хоча може здатися більш зручним, ніж її аналог на ОС Android. Такий додаток можна завантажити і встановити на будь-який смартфон клієнта під управлінням ОС Android. Особливістю такого додатка є програмне забезпечення загального призначення, функціоналом є формування інформації про статистику погоди. Можливість перепризначення поточного місця перегляду (зміна міста); розділений список з вибором - список, який має назви цікавих міських точок світу; сервіс навігаційної системи - GPSLocation збирає дані геолокації користувача, після оновлення доступу можна отримати локальний прогноз, і багато іншого. Додаток був розроблений на мові JAVA на платформі Android Studio.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>11</i>

1. АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1 Мобільний додатки в реаліях громадського користування

Сьогодні популярність мобільних додатків в публічному користуванні зростає з кожним роком, практично у всіх людей є смартфони, які використовуються людьми не тільки для дзвінків і обміну повідомленнями, але і для пошуку інформації, прослуховування музики, оплати комунальних послуг, бронювання послуг, замовлення продуктів харчування, таксі і багато іншого. Сфера застосування смартфонів зростає, розробники мобільного програмного забезпечення знижують такі вимоги в додатках, можна розділити такі категорії розроблених додатків, це:

1. Дитячі додатки.
2. Туризм.
3. Категорія Гурмани.
4. Комунальні послуги для бізнесу.
5. Інтернет-магазини.
6. Застосування суспільного значення.
7. Відеоігри.
8. Освітні програми.

Необхідність зручності використання на наших мобільних пристроях. Така популярність мобільного розвитку зростає з кожним разом, в кожній сфері діяльності, починаючи від сфери різних послуг і послуг, закінчуючи сферою освіти, великими компаніями, економічними біржами та іншими. Причина, по якій всі використовують додаток на смартфоні, ніж їх веб-аналоги і т.д., вона більш ніж зручна, встановлений на вашому смартфоні додаток не вимагає більше зусиль при запуску програми, ніж від вас вимагають інші аналоги пристрою, все просто і швидко.

Сьогодні відсоток населення, яке використовує смартфон, щоб сидіти в інтернеті і робити покупки в Інтернеті, зростає з кожним роком. Мобільні додатки сьогодні домінують у використанні браузерів на смартфонах. В епоху інформаційних технологій наявності одного сайту компанії точно недостатньо, а мобільні додатки дозволяють стати набагато ближче до своїх клієнтів, мобільні

додатки підкорили сучасний цифровий ринок, а це найбільша кількість користувачів всіх вікових категорій. Це означає, що всі вони є потенційними споживачами ринку різних послуг і товарів. Тому розробка мобільних додатків для смартфонів та інших мобільних пристроїв є найбільш перспективним і актуальним розвитком цифрової економіки і бізнесу в цілому.

1.2 Актуальність розробки мобільних додатків

Оскільки технології, що стоять за смартфонами, розвиваються, індустрія мобільних додатків також постійно розвивається. З кожним роком Android продовжує пропонувати нові ідеї, інновації, методи та інструменти в розробці мобільних додатків. Майбутнє розробки додатків для **Android** дуже перспективне.

Прийняття Android зросло, в основному завдяки поліпшенню користувацького досвіду, який відчуває користувач програми. Щоб залишатися попереду в постійно мінливій індустрії розробки мобільних додатків, дуже важливо прийняти тенденції завтрашнього дня.

Сьогодні фахівці в області інформаційних технологій розробляють мобільні додатки, які дозволяють вирішувати величезну кількість завдань, наприклад, створюючи 3D-анімацію. Деякі з технологій використовуються для встановлення підключення до мережі. Інші допомагають оптимізувати маршрут. Треті призначені для пошуку прибуткових продуктів. Четвертий призначений для покупки квитків і т.д. В основі кожної з цих програм лежать легкі певні утиліти, що в результаті дозволяє швидко вирішити завдання, заощадити час і досягти максимально комфортного рівня життя.

Користувачі мобільних пристроїв часто використовують всілякі додатки як ігрового, розважального, так і ділового характеру. Таким чином, в залежності від кінцевих цілей проекту, що розробляється, і завдань, що стоять перед ним, ділиться кілька видів мобільних додатків. Перші дозволяють бізнесменам і офісним працівникам контролювати бізнес-процеси, складати аналітичні звіти, виконувати такі завдання, розробляти дизайн фірмового стилю. До других відносяться різноманітні ігри, програмне забезпечення для перегляду фільмів і прослуховування

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

музики, засоби для спілкування. Кожен з мобільних додатків знаходить свого споживача, однак, як відзначають експерти з цієї області, найпопулярнішим є спеціалізоване програмне забезпечення, наприклад, розробка фірмового стилю, який необхідний компаніям, що працюють в різних напрямках.

В останні роки показник, що характеризує рівень попиту на мобільні пристрої, постійно значно зростає. Маючи понад 2,96 мільйона додатків у Магазині Google Play, Android набрав обертів з моменту свого створення і випередив конкуренцію з часткою ринку понад 85%. З кожним роком Android продовжує пропонувати нові ідеї, інновації, методи та інструменти в розробці мобільних додатків.. Така статистика дозволяє зробити висновок, що розробка мобільних додатків актуальна і доцільна. Головне правильно оцінити, для кого і навіщо створюється програмне забезпечення. Тільки корисна розробка отримає гідне визнання від користувачів.

Показником тенденції активного розвитку є:

- Архітектура додатків Android;
- Зростання багатоплатформної розробки;
- Інтеграція EMM та APM;
- Застосування зі штучним інтелектом і машинним навчанням;
- Блокчейн-технології;
- Миттєві програми
- Заявка за запитом;

Варто зафіксувати основні цілі активного розвитку, це:

- Можливість охопити велику цільову аудиторію користувачів Android.
- Різноманітні гаджети на Android дозволяють замінити робочий стіл, тому користувачам потрібні зручні мобільні додатки.
- Реалізуйте будь-які функції, які допоможуть підтримувати зв'язок з користувачами і збільшити продажі.
- Популярність Google Play зростає з кожним роком, що дозволяє виходити на нові сегменти ринку.

1.3 Методи та засоби розробки мобільного додатку під ОС Android

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Спосіб переоформлення всього етапу побудови методу. Перший спосіб створення мобільного програмного забезпечення називається Agile - це гнучка система, яка передбачає швидку реалізацію правок в постійно мінливий проект на етапі розробки.

У Agile методології розробки після кожної ітерації замовник може спостерігати за результатом і розуміти, задовольняє він його чи ні. Це одна з переваг гнучкої моделі. До його недоліків можна віднести те, що через відсутність конкретних формулювань результатів складно оцінити витрати на робочу силу і витрати, необхідні для розробки. Методологія підходить для великих або доциклічних проектів, які постійно адаптуються до ринкових умов. Відповідно, в процесі реалізації змінюються вимоги. Варто пам'ятати клас творчих людей, які прагнуть генерувати, випускати і пробувати нові ідеї щотижня або навіть щодня. Гнучкий розвиток найкраще підходить для цього психотипу керівників.

Коли використовувати Agile?

- Коли потреби користувачів постійно змінюються в динамічному бізнесі.
- Зміни на Agile впроваджуються за нижчою ціною через часті прирости.
- На відміну від моделі водоспаду, в гнучкій моделі достатньо лише невеликого планування для початку проекту..

Другий популярний метод розвитку називається **Waterfall**. Мається на увазі, що в кінці кожного етапу розробки фахівці переходять до наступного етапу. Від аналітики до компіляції ТК, потім до UX-дизайну і т.д. Процес створення програмного забезпечення повинен бути послідовним, структурованим і логічним. Повернутися до попередніх етапів після переходу на новий неможливо, а якщо щось потрібно змінити, то весь цикл розвитку повинен починатися заново. Agile є більш універсальним методом і підходить для всіх, як початківців, так і досвідчених веб-студій. Водоспад - це більш строгий і утилітарний підхід, він вимагає серйозного досвіду і кваліфікації від кожного учасника розробки.

У моделі Waterfall легко керувати проектом. Завдяки своїй жорсткості розвиток відбувається швидко, вартість і терміни заздалегідь визначені. Але це двосічний меч. Модель водоспаду дасть відмінний результат тільки в проектах з

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

чітко і заданими вимогами і способами їх реалізації. Зробити крок назад неможливо, тестування починається тільки після завершення розробки або майже завершені. Вироби, розроблені за цією моделлю без розумного вибору, можуть мати недоліки, які стають відомі тільки в кінці завдяки суворій послідовності дій. Вартість внесення змін висока, тому що ви повинні чекати, поки весь проект буде завершений, щоб ініціалізувати його. Коли використовувати методологію водоспаду?

- Тільки тоді, коли вимоги відомі, зрозумілі і фіксовані. Немає суперечливих вимог.
- Проблем з наявністю програмістів з необхідною кваліфікацією немає.
- У відносно невеликих проектах.

Засоби розробки:

Програми для ОС Android є програмами в нестандартному байт-кодi для віртуальної машини Dalvik.

Google пропонує для вільного завантаження інструментарій для розробки, який призначений для x86-машин під операційними системами Linux, Mac OS X (10.4.8 або вище), Windows XP, Windows Vista та Windows 7. Для розробки потрібен Java Development Kit 5 або новіший.

Розробку застосунків для Android можна вести мовою Java (не нижче Java. Офіційним середовищем розробки є Android Studio, представлене компанією Google в 2013. Крім цього існує плагін для Eclipse – Android Development Tools (ADT), призначений для Eclipse версій 3.3-3.7. Для IntelliJ IDEA також існує плагін, який полегшує розробку Android застосунків. Для середовища розробки NetBeans розроблено плагін, який починаючи з версії Netbeans 7.0 перестав бути експериментальним. Також існує Motodev Studio for Android, що являє собою комплексне середовище розробки, засноване на базі Eclipse і дозволяє працювати безпосередньо з Google SDK.

У 2009 році до ADT був опублікований Android Native Development Kit (NDK), пакет інструментаріїв і та бібліотек що дозволяє вести розробку застосунків мовою C/C++. NDK рекомендується використовувати для розробки ділянок коду, критичних до швидкості.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		16

- Доступними є наступні бібліотеки:
 - Bionic (Бібліотека стандартних функцій, несумісна з libc);
 - libc (стандартна системна бібліотека мови C);
 - мультимедійні бібліотеки (на базі PacketVideo OpenCORE; підтримують такі формати, як MPEG-4, H.264, MP3, AAC, AMR, JPEG та PNG);
 - SGL (рушій двовимірної графіки);
 - OpenGL ES 1.0 (рушій тривимірної графіки);
 - Surface Manager (забезпечує для застосунків доступ до 2D/3D);
 - WebKit (готовий рушій для Web-браузера; обробляє HTML, JavaScript);
 - FreeType (рушій обробки шрифтів);
 - SQLite (проста система керування базами даних, доступна для всіх застосувань);
 - SSL (протокол, що забезпечує безпечну передачу даних по мережі).

В порівнянні із звичайними застосунками Linux, застосунки Android підкоряються додатковим правилам:

Content Providers — обмін даними між застосунками;

Resource Manager — доступ до таких ресурсів, як файли XML, PNG, JPEG;

Notification Manager — доступ до рядка стану;

Activity Manager — управління активними застосунками.

Для Android був розроблений формат інсталяційних пакетів .apk.

Основні засоби, що використовуються на різних етапах розробки

Залежно від предметної області і завдань, поставлених перед розробниками, розробка програм може бути досить складним, покроковим процесом, в якому задіяна велика кількість учасників і різноманітні засоби. Для того щоб визначити, коли і в яких випадках використовуються які засоби, виділяють наступні основні етапи розробки програмного забезпечення:

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

1. Проектування додатку.
2. Реалізація програмного коду.
3. Тестування.

1.4 Технічне завдання на розробку мобільного додатку

Наступним етапом є виконання представлених нижче етапів складання технічного завдання на створення мобільного додатку. Формулювання завдання на розробку функцій додатку – необхідно оформити своє уявлення про технічне завдання додатка, яке регламентує всі відносини замовника і виконавця. У ТЗ зазвичай містяться вимоги по дизайну, функціям, навігації, способ подачі інформації, реалізацію к термінам і об'ємам робіт кожного етапу розробки додатку. Чітко сформульоване ТЗ допоможе уникнути конфліктів нерозуміння між виконавцем і замовником. Технічне завдання найчастіше представлено таким чином: мета створення додатку, структура додатку, функціонал, платформа, дизайн додатку, взаємодія з користувачами.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		18

2. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ОС ANDROID

2.1 Вибір схеми моделі життєвого циклу

Модель життєвого циклу програмного забезпечення - це структура, що містить процеси дій і завдань, які виконуються при розробці, використанні та обслуговуванні програмного продукту.

Сьогодні існує чимало стандартних моделей проектування, які дозволяють поетапно, крок за кроком, реалізувати будь-який проект від ідеї до її втілення. Їх вибір залежить тільки від розробників і тих цілей, які вони переслідують. Прикладами таких моделями є: “Waterfall - каскад”, модель “Спіраль” - Spiral, однією з таких моделей для проектування додатку є Гибкая модель - Agile.

Програма, як і будь-які інші додатки, сайти, мають свій життєвий цикл програмного забезпечення, який представлений в схемі життєвого циклу програми, яка реалізує рішення про необхідність створення програмного продукту і закінчується в момент його повного виведення з експлуатації. Модель багато в чому зумовлена металогією розробки програмного забезпечення.

У проектуванні додатку використовується Гнучка схема «Agile», яка адаптована під реалізації даного проекту. До етапів методології такого ЛКП відносяться:

1. Аналіз вимог до програмного забезпечення.
2. Проекції:
 - Проектування та розробка технічного завдання – визначення структури додатку, матеріалів, вибір програмних засобів для його створення.
3. Програмування:
 - Розробка програмного коду, модулів, бази даних і інших елементів додатку необхідних в проекті;
4. Тестування та налагодження:
 - Тестування і фіксинг.
5. Звільнення та постійна підтримка:
 - Запуск та підтримка.

Изм.	Лист	№ докум.	Подпись	Дата

Модель Agile - це методологія гнучких принципів розробки, яка характеризується принципами командного підходу до розробки програмного забезпечення. В основному це рамки, які передбачають ітераційну роботу над проектом, коли основні етапи розвитку циклічно повторюються одна за одною. Перед кожним таким циклом виконана робота оцінюється і перевіряється як командою розробника проекту, так і замовником*. Це робиться за допомогою щоденних зустрічей, а також демонстрації та тестування продукту.

Життєвий цикл такого проекту - це набір ітерацій, кожна з яких включає в себе міні-версію розробки проекту командами, що використовують метод *Waterfall*. Виходить, що ітераційні методології відрізняються тим, що результатом кожної ітерації є готовий продукт, і кожна наступна ітерація підвищує його функціональність[1.1]..

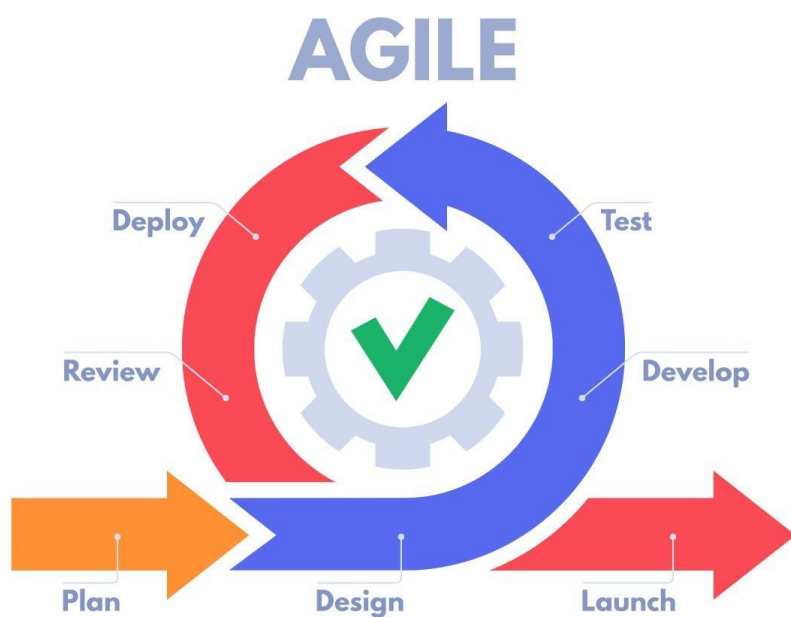


Рисунок 2.1 - Гнучка модель Agile

Переваги такої схеми AGILE

1. Збільшення прибутку.
2. Продукція виходить на ринок швидше, релізи виходять раніше і регулярно, відповідно, клієнти отримують віддачу від своїх інвестицій.

Изм.	Лист	№ докум.	Подпись	Дата

3. Якість продукції забезпечується за допомогою вбудованих випробувань і регулярних перевірок робочого продукту протягом всієї розробки.
4. Покращена прозорість для зацікавлених сторін, оскільки методологія гнучкого розвитку заохочує залучення користувачів до спільних, узгоджених зусиль.
5. Знизити ризик, оскільки команда виявляє і виправляє будь-які проблеми на ранній стадії.
6. Як клієнтоорієнтований процес, він забезпечує постійну взаємодію з клієнтами протягом усього процесу, на кожному етапі.
7. Гарантує, що якість розробки програмного забезпечення підтримується на потрібному рівні або навіть краще.
8. Клієнт (и) скористатися ранньою і частою можливістю побачити прогрес. Таким чином, ви можете змінювати рішення протягом усього процесу розробки проекту.
9. Дає сильне відчуття приналежності до клієнта (клієнтів), оскільки вони знаходяться в прямому і широкому контакті з командою розробника проекту.
10. Він може створити базову версію програмного забезпечення, що розробляється, яке може бути використане в наступних ітераціях. Це дуже корисно для проектів, де час на ринок те ж саме є проблемою великого значення.
11. Можливо, це дасть кращі результати. Це тому, що найчастіше agile-команди виключно мотивовані та самоорганізовані.
12. Знижується ризик невдачі, так як процес повністю заснований на поступовому прогресі. Отже, і клієнт(и), і команда розробників точно знають, що є повним, а що ні.

Мінуси такої схеми

1. Якщо керівник проекту не впевнений у результаті, існує підвищений ризик зриву проекту.
2. Вимагає залучення експерта для прийняття життєво важливих рішень.
3. Не підходить для невеликих проектів.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
						21
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

4. Загальна вартість впровадження гнучкого підходу трохи вище, ніж у інших підходів до розробки програмного забезпечення. Крім того, загальний прогнозований час може збільшитися в міру розвитку програмного забезпечення.

5. Характеристикою дуже високої залученості клієнтів може бути не те, про що можуть просити деякі клієнти.

При створенні додатку було використано стандарт ISO/IEC12207. На основі цього розроблено методологію додатку для роботи електронного навчального посібника.

2.2 Визначення завдань та функції мобільного додатку

На етапі планування, перш за все, слід визначити призначення основних завданнями і функціональністю майбутнього додатку. Нужно досконало определить, будет ли приложение узко или общенаправленным, каким будет приложение, будут ли иметь различные разделы, какие функции будут выполнять их особенности работы и т.д. На етапі планування можна виділити наступні задачі планування:

1. Фіксація цілей і завдання додатку;
2. максимально детальний опис майбутньої аудиторії;
3. визначення того, що ми чекаємо від користувачів додатку;
4. аналіз додатків – конкурентів;
5. планування того, чим майбутній додаток буде відрізнятися від конкурентів;
6. вибір сервісів та функцій;
7. аналіз структури і функціоналу додатку;
8. створення візуального плану ключових розділів;
9. оцінка зручності використання додатку користувачем;
10. планування способу отримання зворотного зв'язку; оцінка перспективи розвитку додатку.

2.3 Вибір середовища розробки

Середовище розробки додатку, перше що вирішують фахівці перед початком проєк і реалізації продукту – це вибір IDE розробки. До таких IDE відносяться: текстовий редактор, перекладач, інструменти автоматизації збірки, налагоджувач. На сьогоднішній день існує більше 3 популярних середовищ розробки для ОС Android.

Android Studio - є офіційним IDE для розробки додатків Android, на основі IntelliJ IDEA. На вершині можливостей, які ви очікуєте від IntelliJ, Android Студія пропонує:

Гнучка Gradle-система збірки

Побудувати варіанти і кілька АПК покоління файлу

Шаблони коду, щоб допомогти вам побудувати загальні риси додатку •
Багатий редактор макетів з підтримкою перетягування і падіння редагування теми •
інструменти, щоб фіксувати продуктивність, зручність використання, сумісність версії, і інші проблеми

• Proguard • Вбудована підтримка для Google Cloud Platform, що дозволяє легко інтегрувати Google Cloud повідомленнями. Android SDK - включає в себе різноманітні бібліотеки, документацію та інструменти, які допомагають розробляти мобільні додатки для платформи Android.

• API Android SDK-API бібліотеки Android, що надаються для розробки додатків.

• Документація SDK-включає велику довідкову інформацію, що деталізує, що включено в кожен пакет і клас і як це використовувати при розробці додатків.

• AVD (Android Virtual Device) -інтерактивний емулятор мобільного пристрою Android. Використовуючи емулятор, можна запускати і тестувати програми без використання реального Android пристрою.

• Development Tools - SDK включає кілька інструментальних засобів для розробки, які дозволяють компілювати і налагоджувати створювані додатки.

- Sample Code - Android SDK надає типові додатки, які демонструють деякі з можливостей Android, і прості програми, які показують, як використовувати індивідуальні особливості API у вашому коді.

Перед початком розробки додатків для Android корисно зрозуміти загальний підхід платформи до управління зміною API. Також важливо зрозуміти Android API Level (Ідентифікатор рівня API) і його роль у забезпеченні сумісності вашого застосування з пристроями, на яких воно буде встановлюватися.

Рівень API - цілочисельне значення, яке однозначно визначає версію API платформи Android. Платформа забезпечує структури API, які додатки можуть використовувати для взаємодії з системою Android. Кожна наступна версія платформи Android може містити оновлення API. Оновлення API-структури розроблені так щоб новий API залишався сумісним з більш ранніми версіями API. Таким чином, більшість змін в API є сукупною і вводить нові функціональні можливості або виправляє попередні. Оскільки частина API постійно оновлюється, застарілі API не рекомендуються до використання, але не видаляються з міркувань сумісності з наявними додатками. Рівень API, який використовує додаток для Android, визначається цілочисловим ідентифікатором, який вказується у файлі конфігурації кожного Android-дodatка. Нижче наведені статистичні дані API, що використовуються в розробці[1.2].

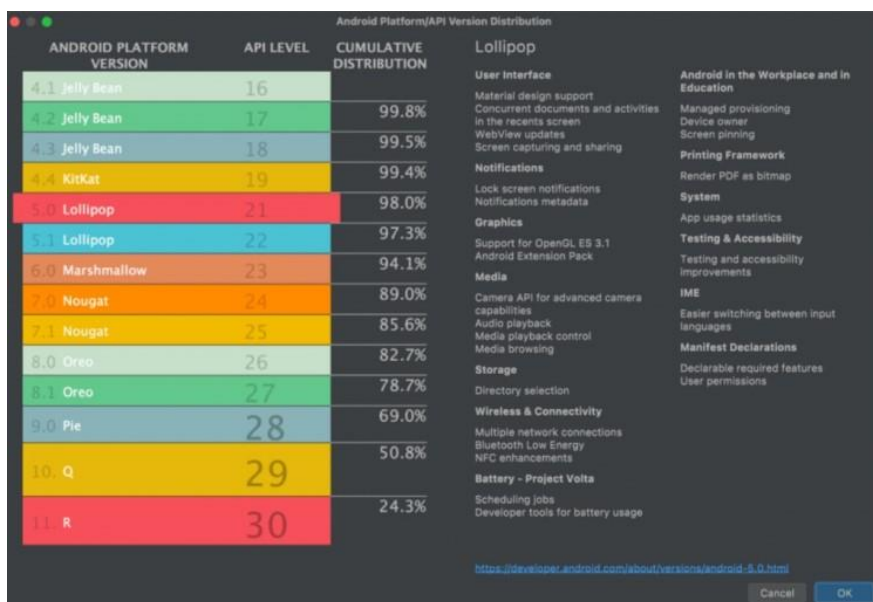


Рисунок 2.2 - Рівні API, що використовуються в розробці

Изм.	Лист	№ докум.	Подпись	Дата

Крім емулятора, SDK також включає безліч інших інструментальних засобів для налагодження та установки створюваних додатків. Якщо при розробці програми для Android за допомогою IDE Android Studio, багато інструментів командного рядка, що входять до складу SDK, вже використовуються при складанні і компіляції проекту.

- Dalvik Debug Monitor Service (DDMS) - інтегрований з Dalvik Virtual Machine, стандартної віртуальні машини платформи Android, цей інструмент дозволяє управляти процесами на емуляторі, а також допомагає в налагодженні додатків. Можна використовувати цей сервіс для завершення процесів, вибору певного процесу для налагодження, генерування трасувань даних, перегляду "купи" або інформації про потоки, робити скріншоти емулятора та багато іншого;

- Hierarchy Viewer - візуальний інструмент, який дозволяє налагоджувати і оптимізувати користувальницький інтерфейс розробляється. Він показує візуальне дерево ієрархії уявлень, аналізує швидкодію перемальовування графічних зображень на екрані і може виконувати ще багато інших функцій для аналізу графічного інтерфейсу додатків;

- Layoutopt - інструмент командного рядка, який допомагає оптимізувати схеми розмітки та ієрархії розміток в створюваному додатку. Необхідний для вирішення проблем при створенні складних графічних інтерфейсів, які можуть зачіпати продуктивність програми;

- Draw9-patch - графічний редактор, який дозволяє легко створювати NinePatch графіку для графічного інтерфейсу розроблюваних додатків;

- sqlite3 - інструмент для доступу до файлів даних SQLite, створених і використовуваних додатками для Android;

- Traceview - цей інструмент видає графічний аналіз трасувань логів, які можна генерувати з додатків;

- mkshcard - інструмент для створення образу диска, який ви можете використовувати в емуляторі для симуляції наявності зовнішньої карти пам'яті (наприклад, карти SD).

Найбільш важливий з них цих інструментів - є емулятор мобільного пристрою, однак до складу SDK входять і інші інструменти для налагодження, упаковки та інсталяції ваших додатків на емулятор.

- Eclipse - це інтегроване середовище розробки, яке досить легко налаштувати і може бути інтегроване з Android SDK, рідним набором розробників, який вам потрібно розробити. Інструментом для будівельних проектів є Ant (популярний інструмент для створення програмного забезпечення, написаний на Java). Eclipse використовує Java як мову програмування. Цей IDE дуже простий у використанні, має простий інтерфейс, тому підходить для початківців.

Простота використання Eclipse забезпечується плагінами, що дає можливість розвиватися не тільки на Java, але і на інших мовах, таких як C / C ++, Python Perl, Ruby, Groovy, PHP, Erlang, компонент Pascal, Zonnon та інші. Для цього середовища розробки існує цілий набір комерційних і безкоштовних модулів. Eclipse має проект BIRT для розробки, розробки та звітності аналітичних додатків. Існують також різні модулі для створення графічних інтерфейсів.

У цього проекту для реалізації програми використовується середовище **Android Studio**. На даний момент Android Studio являється популярною і офіційною середовищем розробки ОС Android.

2.4 Вибір графічного дизайну інтерфейсу користувача, дизайну і стилю програми, елементів і навігації

Наступним етапом розробки додатку є розробка інтерфейсу та створення дизайну основних розділів. UI/UX-дизайн (від англ. User Experience Design) – в перекладі означає «досвід взаємодії» і включає в себе різні UX-компоненти: інформаційну архітектуру, проектування взаємодії, графічний дизайн і контент. В цілому, UX дизайн увазі комплексний підхід до взаємодії користувача з інтерфейсом, будь-то веб-додаток, мобільний додаток або будь-яка інша програма.

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

Інтерфейс програми є інструментом взаємодії між користувачем і програмою. Користувачеві важливо легко зрозуміти функціональність, щоб візуальний був приємним і не відштовхуючим, а на будь-якому пристрої програма працювала правильно. Загальна межа між двома функціональними об'єктами, вимоги до яких визначаються стандартом. Сукупність засобів, методів і правил взаємодії між елементами системи. У повсякденному житті ми постійно стикаємося з інтерфейсами, в наших смартфонах, ПК і всьому іншому, що нас оточує.

Інтерфейс мобільного додатку, основні вимоги

1. Використання знайомих елементів інтерфейсу користувача;
2. Високий рівень візуалізації, що дозволяє гармонійно перебувати в додатку;
3. Всі важливі іконки повинні бути чітко видні і зручно розташовані;
4. Зручний вихід даних;
5. Відображати всі діалогові вікна, що запитують персональні дані;
6. явна демонстрація можливостей;
7. Настроювання та настроювання.

Основні питання, які вирішуються UX-дизайном.

Постановка цілей і завдань - чого в підсумку нам необхідно досягти?

Підбір відповідних UX інструментів для реалізації цілей

Розробка продукту, максимально зручного і легкого в сприйнятті цільовою аудиторією

Аналіз кінцевого результату - відповідає продукт очікуванням замовника і наскільки високий рівень задоволеності користувачів.

User Interface Design [UI] або користувальницький інтерфейс - це більш широке поняття, яке включає в себе певний набір графічно оформлених технічних елементів (кнопки, чекбокси, селектори та інші поля). Його завдання - допомогти користувачеві організувати взаємодію з програмою.

На сьогоднішній момент існують деякі правила UI-дизайну

- Організованість елементів інтерфейсу. Це означає, що вони повинні бути логічно структуровані і взаємопов'язані.
- Угруповання елементів інтерфейсу. Об'єднання в групи логічно пов'язаних елементів (меню, форми).
- Вирівнювання елементів інтерфейсу. Складно уявити, що погано вирівняний інтерфейс може бути для когось зручним!
- Єдиний стиль елементів інтерфейсу. Стильове оформлення грає не останню роль, адже саме воно зберігається в пам'яті користувача.
- Наявність вільного простору. Це дозволяє розмежовувати інформаційні блоки, зосереджуючи увагу на чомусь одному. Розроблений по всі правилам користувальницький інтерфейс значно підвищує ефективність ресурсу і дає йому конкурентні переваги. При створенні дизайну додатку необхідно керуватися одним принципом: "Створювати дизайн необхідно тільки для ключових екранів". Ключові екрани – це ті екрани, які суттєво відрізняються від інших. Зазвичай, створюється один або декілька варіантів дизайну, відповідно до технічного завдання. При цьому окремо створюється дизайн головної сторінки, і дизайни типових сторінок. Власне «дизайн сторінки» представляє собою графічний файл, листковий малюнок, що складається з найбільш дрібних картинок елементів загального малюнка. Унікальний дизайн коштує дорожче, але й передбачає розробку з нуля, повністю унікальну розробку під конкретне замовлення. Залежно від професіоналізму та/або політики замовника веб-дизайнер або розробляє ідею і концепцію дизайну повністю самостійно, або одержує ряд вимог (колір, стиль і т.п.), очікувань та ідей від замовника і намагається триматися цього напрямку при розробці макета[23]. Іноді пропонуються дизайн-рішення на основі шаблонів, що прискорює роботу. Кінцевим продуктом роботи є дизайн-макет: картинка, що представляє передбачуваний майбутній зовнішній вигляд сторінок додатку, розміром приблизно 960x640 px (пікселів) – розмір, відповідний середньому стандарту, пов'язаний з необхідністю подальшої прив'язки до різних

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		28

- **Прототипування, дизайн і демонстрація**

Розробляється прототип, згідно з яким дизайнер переходить до детального креслення кожного компонента; Доопрацювання обраної концепції проектування прикладних інтерфейсів; цей метод показує структуру і опис взаємодії користувача з мобільним додатком. Одним з останніх етапів є клікабельний прототип. Це більш детальна форма фінального проекту. Він імітує взаємодію користувача з інтерфейсом, тобто дозволяє натискати на всі елементи управління, перевіряти анімацію і відео. Цей етап вимагає багато часу.

2.4.1 Мобільна навігація в додатку

Хороший дизайн мобільної навігації повинен бути інтуїтивно зрозумілим і простим, щоб допомогти користувачеві якомога легше знайти те, що їм потрібно. Для створення мобільного додатку дизайнер використовує безліч інструментів для розробки навігаційної системи, відокремлюю 3 основні:

- ✓ Navigation graph [NG];
- ✓ Navigation Editor [NE];
- ✓ NavHostFragment [NHF];

Проектування навігації будь-якого продукту є одним з основних пунктів розробки. В якості мети навігації багато хто прописує рух користувачів в найкоротші терміни, але шлях користувача повинен бути не тільки швидким, але і логічним і простим. Мобільна навігація повинна бути візуально та інтуїтивно зрозумілою. З точки зору зручності досить, щоб навігація була простою і зрозумілою.

Посилання та кнопки в додатку повинні бути досить великими. Навігація повинна мати правильний дисплей, розмір іконок і тексту. Навігація повинна мати не більше 5 пунктів. Використовуючи не більше п'яти елементів, це знижує втому користувача і економить місце на екрані. Також важливо враховувати відстань між навігаційними елементами. Користувачі повинні мати достатньо вільного місця, щоб їм не довелося натискати інші кнопки поблизу. Розміри в мобільній навігації мають значення, тому вам потрібно витратити час на тестування та перевірку того, як користувачі взаємодіють з додатком.

Розвиток візуального дизайну, етап на якому додаються анімація, логотипи, кольори, користувальницькі шрифти та інше. Вихідний екран не повинен бути сильно перевантажений. Інформація повинна бути гідною і легкою для читання.

2.5 Вибір засобів розробки та тестування мобільного додатку

2.5.1 Вибір операційної системи

Android — операційна система і та платформа для мобільних телефонів і планшетних комп'ютерів, створена компанією Google на базі ядра Linux та підтримується альянсом Open Handset Alliance. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення та застосування спираються на цю реалізацію Java.

Перша версія Android була випущена 23 вересня 2008 року і носила назву 1.0 Astroboy, а наступна — 1.1 Bender. Від назв на честь відомих роботів згодом довелося відмовитися через розбіжності з правовласниками.

2008 року Android пережив численні оновлення, які поступово покращували операційну систему, додаючи нові функції, та виправляючи помилки у попередніх випусках.

Операційна система - являє собою основу будь-якого пристрою, що містить набір програм, які вирішують 2 важливих завдання такої системи, вона являє собою:

- Інтерфейсний зв'язок між пристроями обчислювальної системи та прикладними програмами.
- Керування пристроями та робочими процесами шляхом ефективного розподілу системних ресурсів.

Операційна система Android є найпопулярнішою в світі і встановлена на найбільшій кількості пристроїв у світі. Виходячи з представленої статистики операційних систем за останні 2019 - 2020 роки, статистика наведена нижче [1.3].

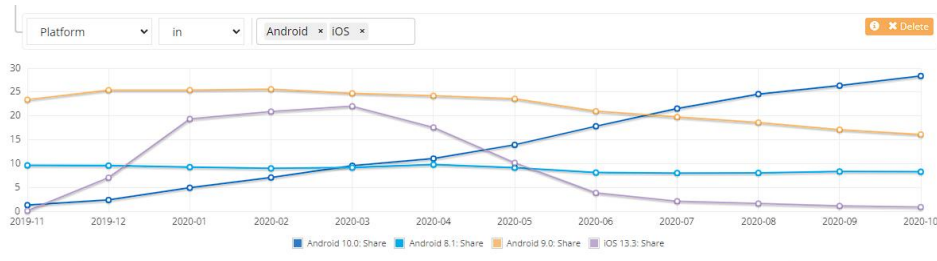


Рисунок 2.3 - iOS & Android Market

З огляду на популяцію ANDROID в нашій країні, в основу ОС був обраний Android..

Після того, як ви вибрали операційну систему, для якої буде розроблений додаток, вам потрібно вирішити, для яких версій буде розроблений додаток, нижче наведені статистичні дані від Google [1.4].

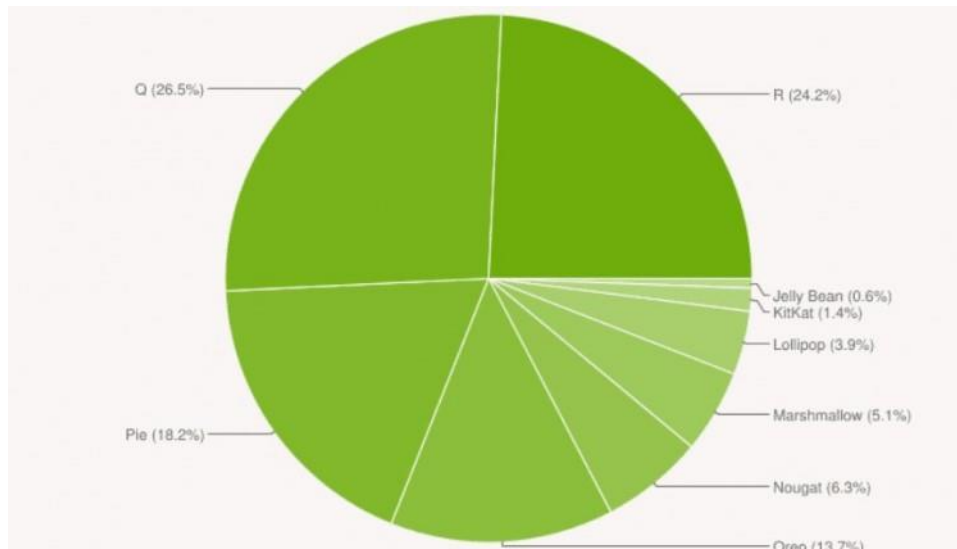


Рисунок 2.4 - Статистика використовуваних версій ОС Android

За основу розробки була обрана прохідна версія API 24: Android 7.0+ (Nougat), яка сумісна з 91,7% пристроїв.

Изм.	Лист	№ докум.	Подпись	Дата

3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Архітектура, інтерфейс і функціональність мобільного додатку -

Архітектура клієнт-сервер - це обчислювальна архітектура, де завдання і мережеве навантаження розподіляються між програмним забезпеченням, яке є постачальником ресурсів і послуг - сервером і замовником ресурсів і послуг - клієнтом.

Сервер - це сховище даних і надає доступ до цих даних іншим об'єктам мережі за їх запитом. Клієнт - це робоча станція, яка використовує серверні ресурси та надає користувачеві зручний інтерфейс для керування цими ресурсами. Клієнт не повинен мати прямих посилань на базу даних для забезпечення безпеки даних і повинен бути завантажений основною логічною складовою програми для можливої реалізації масштабованості системи. Сервер працює за клієнтськими запитами і керує обробкою цих запитів. Після виконання кожного запиту, незалежно від результату обробки, сервер надсилає відповідь клієнту, який надіслав запит. Як правило, клієнт і сервер розташовані на різних комп'ютерах, але також можуть бути виконані на одній машині [3.1].



Рисунок 3.1 - Архітектура клієнт-сервер

У нашому випадку архітектура API клієнт-сервер дозволяє значно зменшити обсяг мережевого трафіку, надавши відповідь на запит, а не повну передачу даних.

3.2 Написання та тестування мобільного додатку

3.2.1 Розробка дизайну програми

Для розробки дизайну був використаний онлайн-сервіс для розробки інтерфейсів і прототипування Figma. Один з найпопулярніших кросплатформних редакторів зображень. Figma вважається спільним програмним забезпеченням. Він пропонує безліч інструментів для створення найважких та привабливих дизайнів з можливістю потім експортувати якісь частини дизайну у вигляді векторної графіки, чи навіть увесь дизайн цілком

Для початку роботи потрібно перейти за посиланням на сайт www.figma.com, після завантаження натиснути кнопку «Try Figma for free», у вікні, що відкриється, проходимо процедуру реєстрації/авторизації, тільки після цього ви зможете працювати зі службою.

Після створення нового «Design Fiel» відкриється середовище для роботи. Для початку проектування створюємо «Рамку», монтажну область, після чого починаємо працювати [4.1!].

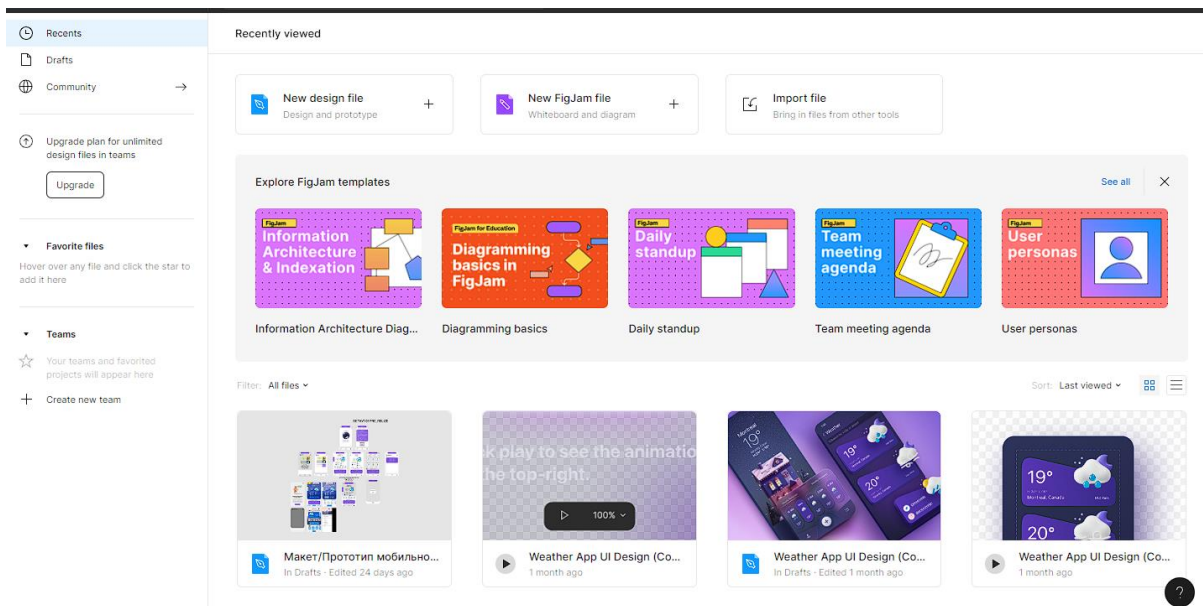


Рисунок 3.2 – Головне меню створення проекту Figma.

Перед початком розробки були створені ескізи перших прототипів конструкції [4.2!].

Изм.	Лист	№ докум.	Подпись	Дата

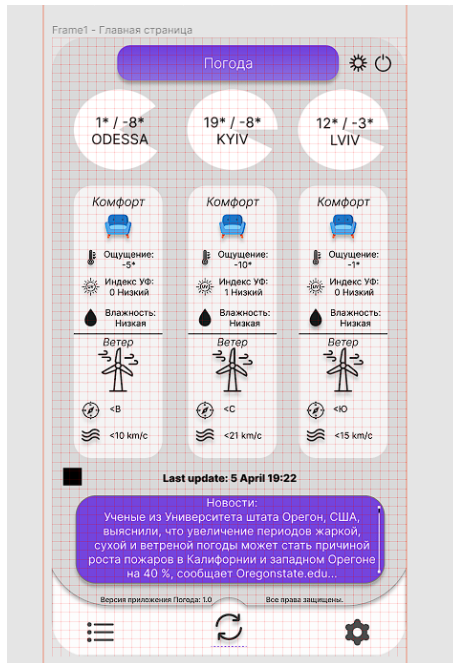


Рисунок 3.3 – Створення першого дизайну

Постійне вдосконалення програмних можливостей вимагало все більшого простору для інформації та функціональності, це вимагало вдосконалення та оновлення старих.



Рисунок 3.4 – Друге створення дизайну

Изм.	Лист	№ докум.	Подпись	Дата

Надалі був прийнятий титульний дизайн, на якому був побудований весь подальший дизайн програми, вибір став через її унікальності стилю, інформативності, приємного зовнішнього вигляду дизайну [4.4].



Рисунок 3.5 – Створення третього дизайну

У візуальному оформленні компонентів були задіяні маркування типу «Corners» округлення, кути і тіні. Колірна палітра була обрана в особливому стилі програми, для цього використовувалися Adobe Color і її аналоги..

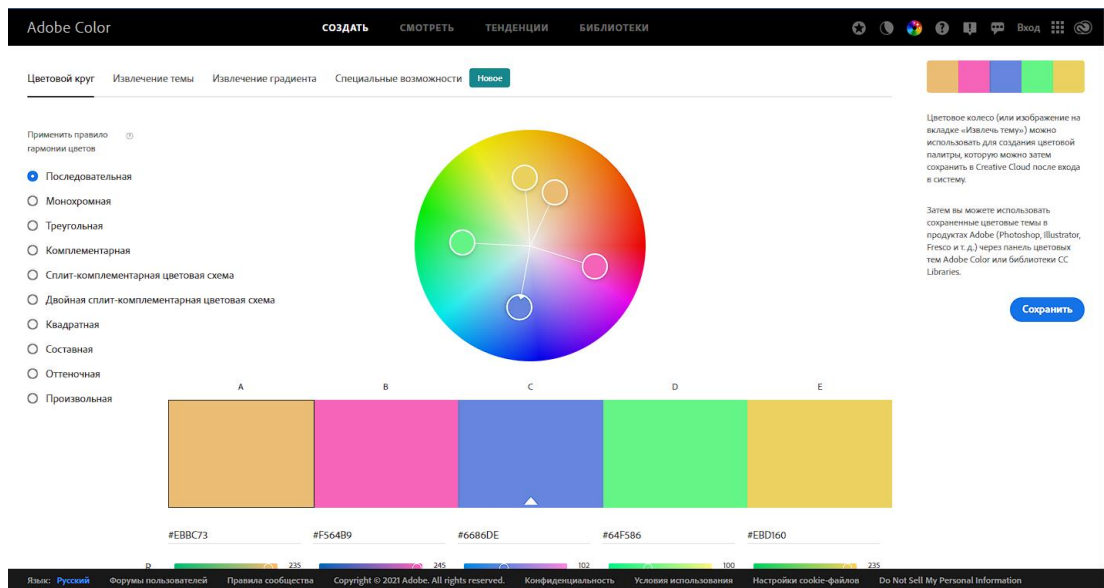


Рисунок 3.6 – Колір Adobe

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

3.2.2 Робота з API

API є основою цієї програми. Для створення програми було прийнято рішення отримувати дані з сервера, в якості сервера я вибрав популярний сервіс - OpenWeatherMap.

Платформа OpenWeather - це набір елегантних і широко впізнаваних API. Оснащений згортковими рішеннями машинного навчання, він здатний доставити всю інформацію про погоду, необхідну для прийняття рішень в будь-якій точці земної кулі..

API - це виклик функції блоку необхідного коду, для відображення його обраної інформації на екрані програми.

Щоб почати роботу з API, спочатку нам потрібно буде зайти на сайт <https://openweathermap.org> – API, там ми створюємо обліковий запис, заходимо в особистий кабінет - Мої КЛЮЧІ API, копіюємо свій унікальний ключ* і заходимо в програму (див.нижче).

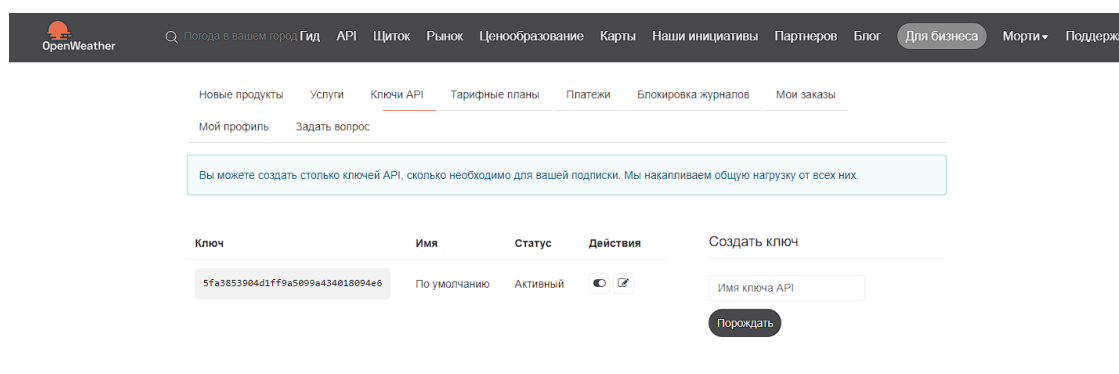


Рис. 3.7 – Створення облікового запису

Щоб мати можливість викликати API, потрібно реалізувати це в проєкті, це робиться наступним методом.

```
else {
    String city = input.getText().toString();
    String key = "212095f94561d3d9f278fdec515927e1";
    String url = "https://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid=" + key + "&units=metric&lang=ru";
    String time = "https://api.sunrise-sunset.org/json?lat=36.7201600&lng=-4.4203400&date=today";
    new GetURLData().execute(url);
}
```

Рисунок 3.8 – Створення посилання та ключа в програмі

Тепер для роботи з блоками запитів с API сервера, потрібен адрес для запитів обробки URL, це поле «String url=»». Для здійснення виводу створимо новий об'єкт JSON з виходом Result. Далі виводимо JsonObject. Приклад коду:

```

@SuppressLint("SetTextI18n")
@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    if ( !isOnline() ){
        Toast.makeText(getApplicationContext(), text: "No internet connection!",
            Toast.LENGTH_LONG).show();
        return;
    }
    try {
        JsonObject jsonObject = new JsonObject(result);
        result_info.setText(jsonObject.getJsonObject("main").getDouble( name: "temp") + ("°C"));
        result_info1.setText("Давление: " + jsonObject.getJsonObject("main").getDouble( name: "pressure") + " pHa");
        result_info2.setText("Ощущения: " + jsonObject.getJsonObject("main").getDouble( name: "feels_like") + "°C");
        String City = jsonObject.getString( name: "name");
        city_result.setText("Город | Страна: " + City + ", " + jsonObject.getJsonObject("sys").getString( name: "country"));
        JSONArray jsonArray = jsonObject.getJSONArray( name: "weather");
        JsonObject jsonObject1 = jsonArray.getJSONObject( index: 0);
        String icon = jsonObject1.getString( name: "icon");
        String description = jsonObject1.getString( name: "description");
        String main = jsonObject1.getString( name: "main");
        result_info6.setText(description);
        result_info4.setText("Влажность: " + jsonObject.getJsonObject("main").getDouble( name: "humidity") + "%");
        result_info5.setText("Ветренность: " + jsonObject.getJsonObject("wind").getDouble( name: "speed") + " км/ч");
        result_info3.setText("Облачность: " + jsonObject.getJsonObject("clouds").getString( name: "all") + "%");
        visibility.setText("Видимость: >" + jsonObject.getDouble( name: "visibility") + " км");
        long unixSeconds = jsonObject.getJsonObject("sys").getLong( name: "sunset");
        long unixSeconds1 = jsonObject.getJsonObject("sys").getLong( name: "sunrise");
        Date dat = new Date(unixSeconds*1000L);
        Date dat1 = new Date(unixSeconds1*1000L);
        SimpleDateFormat sdf = new SimpleDateFormat( pattern: "HH:mm");
        sdf.setTimeZone(TimeZone.getDefault());
        Date dat2 = new Date(unixSeconds1*1000L);
        SimpleDateFormat sdf1 = new SimpleDateFormat( pattern: "HH:mm");
        sdf1.setTimeZone(TimeZone.getDefault());
        String formattedDate = sdf.format(dat);
        String formattedDate1 = sdf1.format(dat1);
        sunrise_text.setText("Восход: " + jsonObject.getJsonObject("sys").getLong("sunrise")/1000 + data);
        sunnise_text.setText("Восход: " + formattedDate1 + ", " + data);
        sunset_text.setText("Закат: " + formattedDate + ", " + data);
        TimeZone tz= TimeZone.getDefault();
        sunset_text.setText("Time: " + tz.getDisplayName(false, TimeZone.SHORT));
        teststend2.setText("Time zone: " + tz.getDisplayName(false, TimeZone.SHORT));
        teststend.setText("Lat: " + jsonObject.getJsonObject("coord").getDouble( name: "lat") +
            "\n" + "Lon: " + jsonObject.getJsonObject("coord").getDouble( name: "lon"));
        double lat = jsonObject.getJsonObject("coord").getDouble( name: "lat");
        double lon = jsonObject.getJsonObject("coord").getDouble( name: "lon");
        String weatherDate = jsonObject.getString( name: "weather");
        Log.i( tag: "WEATHER DATA: ", weatherDate);
        teststend1.setText(" " + lat + lon);
        DateFormat df = DateFormat.getDateTimeInstance();
        String updatedOn = df.format(new Date());
        TextUpdate.setText("Last update: " + updatedOn);
    }
}

```

Рисунок 3.9 – Виведення блоків запитів.

Частина коду, яка відповідає за обробку всіх вищезазначених дій запити.
Знімок екрана: подання нижче.

```

private class GetURLData extends AsyncTask<String, String, String> {
    protected void onPreExecute() {
        super.onPreExecute();
        result_info.setText("Поиск... ");
    }

    @Override
    protected String doInBackground(String... strings) {
        HttpURLConnection connection = null;
        BufferedReader reader = null;
        try {
            URL url = new URL(strings[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.connect();
            InputStream stream = connection.getInputStream();
            reader = new BufferedReader(new InputStreamReader(stream));
            StringBuffer buffer = new StringBuffer();
            String line = "";
            while ((line = reader.readLine()) != null)
                buffer.append(line).append("\n");
            return buffer.toString();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (connection != null)
                connection.disconnect();
            if (reader != null) {
                try {
                    reader = new BufferedReader(new InputStreamReader(stream));
                    StringBuffer buffer = new StringBuffer();
                    String line = "";
                    while ((line = reader.readLine()) != null)
                        buffer.append(line).append("\n");
                    return buffer.toString();
                } catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                } finally {
                    if (connection != null)
                        connection.disconnect();
                    if (reader != null) {
                        try {
                            reader.close();
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }
                }
            }
        }
        result_info.setText("ERROR");
        city_result.setText("Город | Страна: Does not exist");
        return toString();
    }
}

```

Рисунок 3.10 – Код обробки запитів.

3.2.3 Реалізація метода `getLocation`.

Створення `getLocation` дає змогу отримувати дані геолокації за допомогою служб `Google Location Ahead`.

По-перше, давайте створимо кнопку. Зайдіть в `Activity_main.xml`, додайте кнопку для обробки запиту.

```
<Button
    android:id="@+id/b_location"
    android:layout_width="25dp"
    android:layout_height="25dp"
    android:layout_marginTop="650dp"
    android:layout_marginBottom="5dp"
    android:autoText="true"
    android:background="@drawable/ic_free_icon_font_refresh_3917293"
    android:textColor="@color/black"
    android:textSize="10sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Рисунок 3.11 – кнопка `b_location`

Переходимо «Manifest» – в `AndroidManifest.xml` потрібно прописати допуск до `USES_PERMISSION_FINE_LOCATION`, `INTERNET`, `COARSE_LOCATION` (див.нижче).

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Рисунок 3.12 – Дозволи, запитані програмою

Для того, щоб заявка запитала дозвіл на надання персональних даних, потрібно натиснути на «КНОПКУ ОНОВЛЕННЯ» ПІДТВЕРДИТИ або ВІДХИЛИТИ передачу даних. Код дивіться нижче.

```
b_location.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
            getLocation();
        } else {
            ActivityCompat.requestPermissions(MainActivity.this, new
String[] {Manifest.permission.ACCESS_FINE_LOCATION}, 44);
        }
    }
});
```

Перевірка `ACCESS_FINE_LOCATION`:

```
fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
    != PackageManager.PERMISSION_GRANTED &&
```

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

```

ActivityCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
    return;
}

```

Якщо користувач видав PERMISSION_ACCESS виконується наступна кодова

частина:

```

fusedLocationClient.getLastLocation()
    .addOnSuccessListener(this, new OnSuccessListener<Location>() {
        @SuppressWarnings("SetTextI18n")
        @Override
        public void onSuccess(Location location) {
            if (location != null) {
                try {
                    Geocoder geocoder = new Geocoder(MainActivity.this,
Locale.getDefault());
                    List<Address> addresses =
geocoder.getFromLocation(location.getLatitude(),
                        location.getLongitude(), 1);
                    teststend2.setText(addresses.get(0).getLocality());
                    String lat = (" " + addresses.get(0).getLatitude());
                    String lon = (" " +
addresses.get(0).getLongitude());
// teststend4.setText("GPS STATUS: ", "= " +
LocationRequest);
                    boolean requestingLocationUpdates =
Boolean.parseBoolean(null);
                    if (requestingLocationUpdates) {
                        startLocationUpdates(); }
                    requestingLocationUpdates = true;
                    teststend4.setText("GPS STATUS = " +
requestingLocationUpdates);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    });
b_location.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (user_field.getText().toString().trim().equals("")) {
            String city_location = teststend2.getText().toString();
            String key = "212095f94561d3d9f278fdec515927e1";
            String url = "https://api.openweathermap.org/data/2.5/weather?q=" +
city_location + "&appid=" + key + "&units=metric&lang=ru";
            new GetURLData().execute(url);
        }
    }
});

```

Рис.3.13 – Кодування доступу

3.2.4 Установка додатку

Після завершення усіх кориговок я можу згенерувати арк-файл для встановлення додатку на мій смартфон та мануально протестувати додаток. Для цього у верхній панелі обираємо BUILD → Build_Bundle/APK(s) → Build APK(s), файли будуть у папці AndroidStudioProjects\ users\ app\ build\ outputs\ apk\ debug\ . Файл типу арк треба скачати на Android-смартфон та запустити, після цього додаток встановиться на ваш смартфон.

					<i>РП.05.16.000. 00 ДП</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		42

4. ЕКСПЛУАТАЦІЯ ДОДАТКУ ТА ЙОГО ЗАПУСК

Після завантаження і установки програми на телефон додаток з'явиться на робочому столі. У разі емуляції компілюємо додаток і чекаємо. Після запуску програми відкриється головний екран.

Головний екран: є відправною точкою для доступу до всіх функцій пристрою.

Примітка*: служить основним і стартовим фрагментом програми.

Наступні кроки*: *передбачено користувачем*.

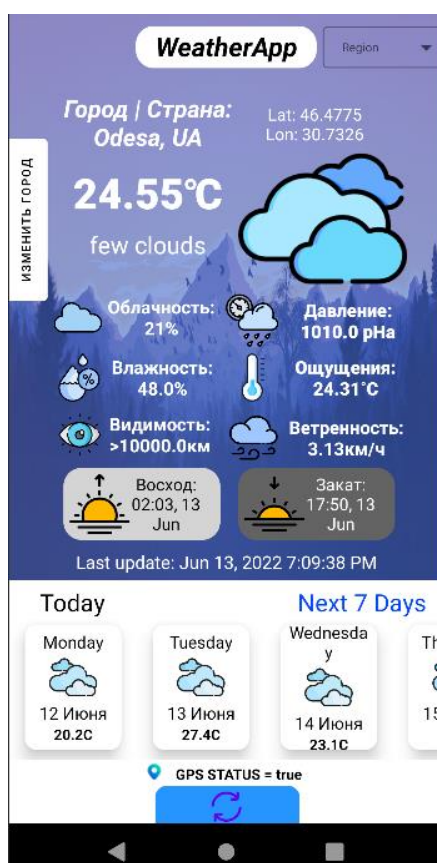


Рисунок 4.1 – Головний користувацький екран програми.

- « **Region** »

***Призначення*:** Натискання на компонент "Region" у верхньому правому куті відкриє випадаючий список у правій частині екрана, та підтвердження вибору [1.8].

Изм.	Лист	№ докум.	Подпись	Дата

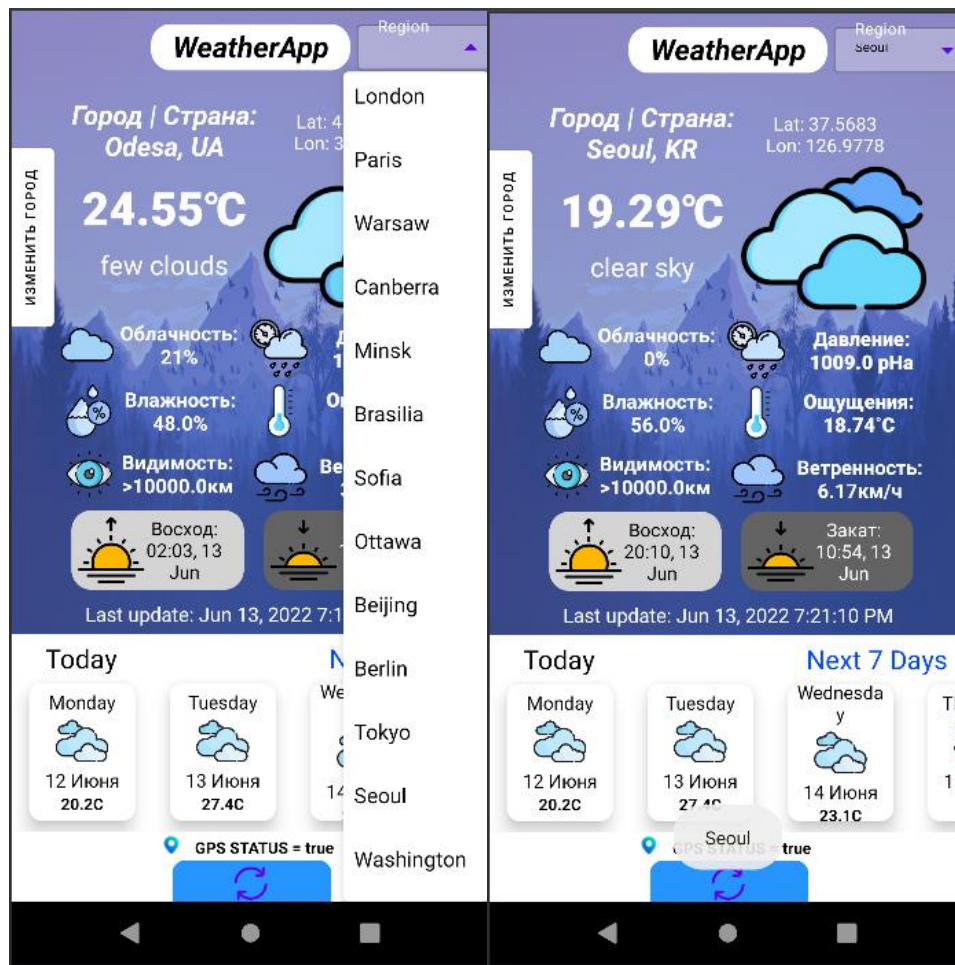


Рисунок 4.2 - Випадаючий список. Та підтвердження вибору

Зі списку виберіть будь-які та попередньо встановлені імена для підтвердження - тикаємо по імені. Інформація була оновлена [1.9]!

- « **Змінити місто** »

***Призначення*:** Натиснувши на бічну вертикальну кнопку «Змінити місто», відкривається діалогове вікно з текстовим полем, де ми вводимо назву міста, перш ніж почати читаємо примітку, для підтвердження вибору натискаємо «Пошук». Готово! "Відхилити" закриває діалогове вікно [2.1, 2.2].

Изм.	Лист	№ докум.	Подпись	Дата

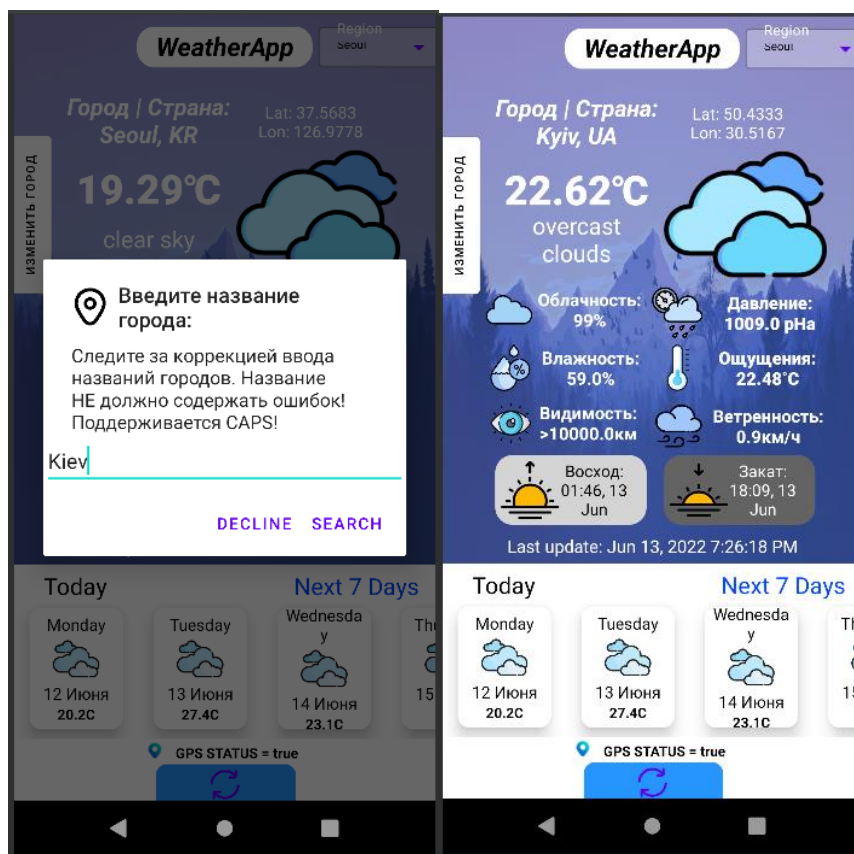


Рисунок 4.3 - Відкрите діалогове вікно та підтвердження.

- « **Reload(кнопка)** »

У текстовому полі "GPS STATUS = TRUE" відображаються значення TRUE/FALSE, вказується, чи включена функція передачі геоданих.

***Призначення*:** Натисканням на кнопку відкриється вікно підтвердження "Передача даних геолокації", -Підтвердити вибір-. Після наступного клацання будуть відображатися дані вашої поточної позиції.

***Примітка*:** [Location] Передача даних про місцезнаходження мають бути ввімкнені.

- « **Today & Next 7 Days** »

***Призначення*:** Інформаційні блоки, відображення інформації Щоденного прогнозу за 7 днів (результат висновку: день тижня, дата, температура), жест вліво скролить блоки;

***Примітка*:** Остальной функционал который задумывался не был реализован, часть такого контента было вырезано, из-за переробіток и вчасності ненадобности.

Изм.	Лист	№ докум.	Подпись	Дата

5. ЕКОНОМІЧНІ РОЗРАХУНКИ

У цьому дипломному проєкті створено мобільний погодні додаток загального використання. Додаток дозволить отримувати необхідну актуальну інформацію про погоду влюбій точці світу с високою точністю.

При оцінці ефективності кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускною спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

5.1 Визначення трудомісткості розробки ПЗ.

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку.

Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

Аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт, приведені на таблиці [5.1].

Таблиця 5.1- Аналоги ПЗ

Найменування ПП	Обсяг функції ПП – V_0 , Усл.машинних командах.
1. ПП JSON API	2500 – 6800
3. ПП введення інформації	1060 – 4750
4. ПП оптимізації розрахунків	1300 – 4200

На підставі значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки в умовах комп'ютера, $K_k=0,7\div 0,8$): $T^a P = 244 * 0,7 = 170,8$ (люд/годин).

Трудомісткість програмного продукту

Визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

- **Розробка технічного завдання**

$$T_{ТЗ} = T^a P \times L_1 \times K_H \quad (1)$$

- **Розробка технічного проекту**

$$T_{ТП} = T^a P \times L_2 \times K_H \quad (2)$$

- **Розробка робочого проекту**

$$T_{РП} = T^a P \times L_3 \times K_H \times K_T \quad (3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага і-го етапу розробки [див 5.2];

K_H – поправочний коефіцієнт, що враховує ступінь новизни [див 5.3];

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм

Таблиця 5.2. - Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
-			
ТЗ (L_1)	0,17	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Таблиця 5.3. - Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Таблиця 5.4 - Значення коефіцієнта ступеня використання в розробці типових програм.

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер потрібно розрахувати трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання:

$$T_{TЗ} = T_a * L_1 * K_n = 170,8 * 0,12 * 0,7 = 14,3 \text{ (люд/годин)}; \quad (4)$$

Трудомісткість розробки технічного проекту:

$$T_{ТП} = T_a * L_2 * K_n = 170,8 * 0,11 * 0,7 = 13,15 \text{ (люд/годин)}; \quad (5)$$

Трудомісткість розробки робочого проекту:

$$T_{РП} = T_a * L_3 * K_n * K_T = 170,8 * 0,61 * 0,7 * 0,7 = 51,05 \text{ (люд/годин)}; \quad (6)$$

Для подальших розрахунків необхідно визначити кількість папера, витраченого на кожен етап.

$N_{ТЗ} = 6$ (стр), $N_{ТП} = 17$ (стр), $N_{РП} = 9$ (стр), $N_{ПЗ} = 48$ (стр) – технічне завдання, розробка ТП, розробка робочого проекту, пояснювальна записка відповідно [див 5.5].

Таблиця 5.5. - Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, ч.		
	Розробка ПП	Контроль керівника	Нормоконтроль
ТЗ	$T_{РТЗ}=14,3$	$T_{КК}=0,7*$ $N_{ТЗ}=0,7*6=4,2$	$T_{НК}=0,15* N_{ТЗ}$ $=0,15*6=0,9$
Розробка ТП	$T_{РТП}=13,15$	$T_{КК}=0,7*N_{ТП}=0,7*17$ $=11,9$	$T_{НК}=0,15*N_{ТП}=$ $0,15*17=2,55$
Розробка РП	$T_{РРП}=51,05$	$T_{КК}=0,7*$ $N_{РП}=0,7*9=6,3$	$T_{НК}=0,15* N_{РП}$ $=0,15*9=1,35$
Розробка пояснювальної записки	$T_{РПЗ}=1,5*N_{ПЗ}$ $48 =72$	$T_{КК}=0,7*$ $N_{ПЗ}=0,7*48=33,6$	$T_{НК}=0,15*N_{ПЗ}$ $=0,15*48=7,2$
Усього, в т.ч.:	Т		
на розробку	$ST_{Р}=150,5$		
Контролькерівника		$ST_{КК}=56$	
Нормоконтроль			$ST_{НК}=12$

5.2 Розрахунок ціни програмного продукту.

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 5.6. Відповідно до статті 8 «Закону про Державний бюджет України на 2022» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39.26 грн.

Таблиця 5.6 - Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	174.23	39.26	6840.26
2.Контроль керівника	30.3	50.00	1515
3.Нормоконтроль	14.85	50.00	742.5
Усього	-	-	$\Sigma Z_0 = 9097.76$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю [див 5.7].

Таблиця 5.7- Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна, грн.	Вартість, грн.
Папір А4	Лист	60	2.50	150
Разом	-	-	-	$B_{M1} = 150$
Транспортно – заготівельні витрати (10%)			-	$B_{тр_з} = 0,1 \times B_{M1} = 0,1 * 150 = 15$
Усього:			-	$B_M = B_{M1} + B_{тр_з} = 150 + 15 = 165$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці [див 5.8].

Таблиця 5.8. - Розрахунок витрат планової собівартості

Стаття витрат	Значення, грн	Формула розрахунку
1. Матеріали	165	B_M (див табл. 5.7)
2. Основна заробітна плата	9097.76	Z_0 (див. табл. 5.6)

3. Накладні витрати	5458.65	$V_{\text{нак.}} = 0,6 \times 3_0 = 0,6 * 9097.76$ $= 5458.65$
4. Повна собівартість	14759.41	$C_{\text{пов}} = B_{\text{м}} + 3_0 + V_{\text{нак.}} = 165 + 909$ $7.76 + 5486.65$

Розмір прибутку, що включається в ціну, визначається по наступній формулі:

$$П = (C_{\text{пов}} * P) / 100 = (14759.41 * 15) / 100 = 2213.85 \text{ (грн)}. \quad (7)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_0 = C_{\text{пов}} + П = 14759.41 + 2213.85 = 16973.26 \text{ (грн)}; \quad (8)$$

Податок на додану вартість визначається по наступній формулі:

$$ПДВ = 0.2 * Ц_0 = 0,2 * 16973.26 = 3394.65 \text{ (грн)}; \quad (9)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_0 + ПДВ = 16973.26 + 3394.65 = \underline{20367.91} \text{ грн}; \quad (10)$$

Изм.	Лист	№ докум.	Подпись	Дата

6. ОХОРОНА ПРАЦІ

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням.

В Україні діє Закон "Про охорону праці", який є одним з найважливіших актів законодавства про охорону праці.

Умови праці на робочому місці, безпека технологічних процесів, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам законодавства. Роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці. Охорона праці під час створення мобільного додатку ОС Android.

6.1 Аналіз умов праці на робочому місці:

На робочому місці користувача персонального комп'ютера виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, підвищений рівень електромагнітних випромінювань, висока напруга електричної мережі, статична електрика та інші. Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи.

Аналізуючи умови праці на готовому місці програміста, можна негативно сплести фактори:

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

- ✓ Підвищена або знижена температура повітря робочої зони;
- ✓ Вплив мікроклімату зони робоча;
- ✓ Недостатня освітленість робочого місця;
- ✓ Підвищений рівень шуму і вібрації на робочому місці;
- ✓ Інфразвуковий вплив;
- ✓ Електромагнітне випромінювання;
- ✓ Електробезпека робочого місця;

6.2 Організація робочого місця і виробничі приміщення:

Кожне робоче місце обладнане робочим столом площею 1,2 м², стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші. Відстань від тильної поверхні одного персонального комп'ютера до екрана іншого 1,5 – 2м; За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 - 2м. Конструкція робочого місця користувача персонального комп'ютера має забезпечення підтримання оптимальної робочої пози офісного працівника. Висота робочої поверхні робочого столу має регулюватися в межах 680 – 800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6м², а об'єм не менший за 20м³, тобто площі та об'єму даного приміщення не вистачає для розташування 7 робочих місць операторів ПК.

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600мм, завширшки не менше ніж 500мм, завглибшки (на рівні колін) не менше ніж 450мм, на рівні простягнутої ноги не менше ніж 650мм.

Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися.

Монітор має розташовуватися на оптимальній відстані від очей користувача становить 600-700мм, монітор має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 градусів до нормальної лінії погляду працівника.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300мм від краю, звернутого до працюючого. Розташування пристрою ведення – виведення інформації має забезпечувати добру видимість монітору, зручність ручного керування в зоні досяжності моторного поля і за висотою – 900 – 1300мм, за шириною 400-500мм

Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6м², а об'єм не менший за 20м³, тобто площі та об'єму даного приміщення не вистачає для розташування 7 робочих місць операторів ПК.

6.3 Мікроклімат робочої зони програміста:

Робота програміста за енерговитратами відноситься до категорії легких робіт, тому повинні дотримуватися наступні вимоги:

- + *оптимальна температура повітря – 22°C (допустима – 20-24°C);*
- + *оптимальна відносна вологість – 40-60% (допустима – не більш 75%);*
- + *швидкість руху повітря від 0,1-0,2м/с, допустима –не більше 0,2м/с.*

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пару і пилу використовують вентиляцію.

6.4 Освітлення робочого місця:

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні бокового освітлення КПО=1,5%. Для штучного освітлення нормованим параметром виступає Емін – мінімальний рівень освітленості, та Кп – коефіцієнт пульсації світлового потоку, який не повинний бути

більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк.

Для освітлення приміщення, у якому працює програміст, використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Природне освітлення - здійснюється через вікна в зовнішніх стінах будинку. Штучне освітлення - використовується при недостатньому природному освітленні й здійснюється за допомогою двох систем: загального й місцевого освітлення.

Для загального освітлення приміщення, де перебуває робоче місце програміста, використовуються газорозрядні лампи типу ЛД. Нормами для даних робіт встановлена необхідна освітленість робочого місця $E_{н}=300$ лк (для робіт високої точності, коли найменший розмір об'єкта розрізнення дорівнює 0,3 – 0,5 мм).

6.5 Вплив шуму і вібрації:

Робоче місце з ПК, має бути устатковане монітором, вінчестером в системному блоці, трьома вентиляторами системи охолодження ПК та клавіатурою. Крім того поряд працює периферійна техніка. Таким чином у приміщенні мають місце шуми механічного і аеродинамічного походження, широкосмугові із аперіодичним підсиленням при роботі принтерів.. Допустимий еквівалентний рівень шуму для робочого місця програміста складає 65 дБА.

Зниження рівня шуму у приміщенні можна здійснити шляхом використанням блоків живлення ПК з вентиляторами на гумових підвісках. Джерелами шуму при роботі з ПК є жорсткий диск, вентилятор блока живлення мережі, вентилятор, розташований на процесорі, механічні сканери, пересувні механічні частини принтера та ін..

Шум, що створюється працюючими ПК, є широкосмужним, постійним з аперіодичним посиленням при роботі принтерів. Тому шум необхідно оцінювати загальним рівнем звукового тиску за частотним коригуванням «А» і вимірювати в дБА.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ПК, мають відповідати вимогам ДСанПіН 3.3.2.007-98, ДСН 3.3.6-037-99.

Під час виконання робіт з ПК у виробничих приміщеннях значення характеристик вібрації на робочих місцях мають не перевищувати допустимі відповідно до ДСанПіН 3.3.2.007-98, ДСН 3.3.6-039-99.

6.6 Електробезпека:

Приміщення з ПК за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

На робочому місці програміста з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартам фірми ІВМ, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

Щоб запобігти електронебезпеці не слід: зафарбовувати або забілювати електропроводку, чіпляти предмети на неї, класти її на газові та водопровідні труби; не допускати торкання з телефонними проводами, не використовувати в якості заземлення водопровідну або теплову мережі. Небезпечно користуватися ручним, незаземленим електроінструментом з напругою більше 42 Вольт.

Обов'язковим заходом безпеки від дії електричного струму є заземлення та занулення. В разі ураження необхідно негайно надати допомогу - відокремити враженого від джерела струму за допомогою діелектричного предмета, викликати лікаря і терміново надати долікарняну допомогу.

6.7 Пожежна безпека:

Пожежна безпека входить в комплекс заходів з охорони праці, і організаційна робота в цій сфері на об'єктах господарювання включає широкий спектр заходів, а саме:

- створення умов для безпечної праці, мінімізації ризику виникнення пожеж,
- своєчасне і повноцінне забезпечення технічними засобами для запобігання займанню та усунення самих пожеж та їх наслідків,
- розробка і впровадження регламентів по гасінню пожеж, евакуації та порятунку з місць пожежі й задимлення людей і майна (матеріальних цінностей), внутрішнє і зовнішнє навчання співробітників.
- Первинні засоби пожежогасіння. Призначення та місцезнаходження первинних засобів пожежогасіння.

Будинки, споруди, приміщення, технологічні установки повинні бути забезпечені первинними засобами пожежогасіння: вогнегасниками, ящиками з піском, покривалами з негорючого теплоізоляційного полотна, грубововняної тканини чи повсті, іншим пожежним інструментом, які використовуються для локалізації і ліквідації пожеж у початковій стадії їхнього розвитку.

Коли від пожежі захищаються приміщення з персональними комп'ютерами, то слід враховувати специфіку вогнегасних речовин у вогнегасниках, які призводять під час гасіння до псування обладнання. Ці приміщення рекомендується оснащувати вуглекислотними вогнегасниками з урахуванням граничнодопустимої концентрації вогнегасної речовини. Для зазначення місцезнаходження первинних засобів пожежогасіння слід установлювати відповідні знаки згідно з чинними державними стандартами. Знаки слід розміщувати на видних місцях на висоті 2-2,5 м від рівня підлоги як у середині, так і поза приміщеннями (у разі потреби).

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

ВИСНОВОК

Метою дипломного проекту було створення мобільного погодного додатку, для отримання необхідних статичних даних з використанням серверної API. Додаток був створений у середовищі розробки Android Studio с використанням мови програмування Java.

При створенні програмного забезпечення були взяті основні методи розробки Android, це ідея, реалізації, потреби користувача, зовнішній дизайн, функціональність, етап розробки і тестування, публікація. В ході завершення розробки був отриманий готовий функціональний додаток, який на момент публікації готовий до використання, виконує заявлені розробником задачі.

Програмне застосування такої програми, орієнтоване на загальний спектр використання. В ході виконання загальних алгоритмів, завдань і функціональності описаних в матеріалі дипломної роботи, користувач отримує необхідні йому дані о погоді місцезнаходження..

Під час вирішення завдання вимог проєтування, було вивчено теоретичні методи уявлення та аналізу вимог до створення програмного забезпечення. В результаті роботи, отримав розроблений додаток який вимогає потребам дипломного проекту, методів та засобів розробки, дизайну, та функціоналу. Використовувалися сучасні технології розробки.

Подальша розробка і доопрацювання програми можливо завдяки гнучкій моделі. Повна можливість розробки коду з метою подальшої трансформації та просування.

					<i>РП.05.16.000. 00 ДП</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Дослідження мобільних додатків (глобальний ринок) (tadviser.ru)// Режим доступу:
[https://www.tadviser.ru/index.php/Мобильные_приложения_\(мировой_рынок\)](https://www.tadviser.ru/index.php/Мобильные_приложения_(мировой_рынок))
2. Дослідження: Як розвивається ринок мобільних додатків? - Webpromo // Режим доступу: <https://web-promo.ua/blog/issledovanie-kak-razvivaetsya-rynok-mobilnyh-prilozhenij/>
3. Ринок мобільних додатків: досягнення та перспективи | (woxapp.com) // Режим доступу: <https://woxapp.com/ru/our-blog/market-of-mobile-apps-achievements-and-prospects/>
4. APIAndroid, розробка API // Режим доступу:
<http://developer.android.com/about/versions/android-4.0.html>
5. Бойчик І. М. Економіка підприємства : навчальний посібник для студентів економічних спеціальностей вищих навчальних закладів I-IV рівнів акредитації. Третє видання, випр. і доп. / І. М. Бойчик, П. С. Харів., М. І. Холчан, Ю. В. Піча. – К. : Каравела, 2016. – 328 с. // Режим доступу:
<http://dspace.wunu.edu.ua/bitstream/ЕП-ПІДРУЧНИК%201%20файл.pdf>
6. Актуальність розробки мобільних додатків для певного виду бізнесу | SafeZone - форум допомоги // Режим доступу: <https://appcraft.pro/blog/12-idej-dlya-mobilnyh-prilozhenij-v-2021-godu/>
7. Актуальність розробки мобільних додатків для андроїд - HTMLer.ru. // Режим доступу: <https://htmler.ru/2020/12/29/aktualnost-razrabotki-mobilnyh-prilozhenij-pod-android>
8. Тенденції розвитку додатків для Android у 2022 році // Режим доступу:
<https://infopro54.ru/news/4-tendencii-razrabotki-prilozhenij-dlya-android-v-2022-godu>
9. Розробка та створення мобільних додатків для Android // Режим доступу:
<https://appcraft.pro/blog/razrabotka-prilozhenij-dlja-android/>

Изм.	Лист	№ докум.	Подпись	Дата

РП.05.16.000. 00 ДП

Лист

59

10. Порівняння гнучкого та водоспадного управління проектами // Режим доступу: <https://www.atlassian.com/ru/agile/project-management/project-management-intro>
11. Вибір середовища для розвитку // Режим доступу: <https://studfile.net/preview/8033758/page:7/>
12. Підручник на тему "Розробка мобільних додатків, Соколов а В.В., 2019. // Режим доступу: <https://obuchalka.org/20191113115520/razrabotka-mobilnih-prilozenii-sokolova-v-v-2019.html>
13. Де я можу отримати актуальну статистику щодо використовуваних операційних систем? — // Режим доступу: <https://qna.habr.com/q/30229>
14. Операційна система Android. Короткий огляд найпопулярніших ... | by Chistyakov V | NOP::Nuances of Programming | Medium // Режим доступу: <https://medium.com/nuances-ofprogramming/операционная-система-android-826fb74c5af9>.