

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра комп'ютерної інженерії

ЖИГАЛО Ярослав

**«Програмний додаток пошуку віддаленої роботи
на основі Android Studio / Remote work search
software system based on Android Studio»**

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІз-41
В. Жигало

Науковий керівник:
к.т.н. Н.Я. Савка

Кваліфікаційну роботу допущено
до захисту:

" ___ " _____ 20___ р.

Завідувач кафедри
_____ О.Л. Дубчак

Тернопіль – 2024

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний додаток пошуку віддаленої роботи на основі Android Studio» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 74 сторінки пояснюючої записки, 10 рисунків, 6 додатків. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка мобільного додатку пошуку віддаленої роботи на основі Android Studio.

Методи дослідження включають методи аналізу та синтезу, порівняння, комп'ютерного моделювання, тестування.

Розглянуто задачу пошуку віддаленої роботи на основі мобільних платформ. Проаналізовано особливості сучасних мобільних додатків та операційні системи для їх функціонування. Охарактеризовано сучасні фріланс-біржі для виконання замовлень у дистанційному форматі. Досліджено існуючі програмні засоби для проектування мобільних додатків, визначено основні їх переваги та недоліки. Розроблено алгоритм проектування мобільного додатку пошуку віддаленої роботи на основі сформульованих вимог до системи. Розроблено прототип програмного додатку із описом функцій та їх реалізації.

Розроблено архітектуру програмного забезпечення, їх зазначенням діаграми варіантів використання. Описано інтерфейс користувача мобільного додатку пошуку віддаленої роботи. Наведено результати тестування програмного додатку.

Практичне значення полягає розробці мобільного додатку пошуку віддаленої роботи на основі Android Studio, в якому реалізовано комунікацію замовника і розробника.

Ключові слова: **МОБІЛЬНИЙ ДОДАТОК, ОПЕРАЦІЙНА СИСТЕМА, ANDROID STUDIO, ФРІЛАНС-БІРЖА, KOTLIN.**

ANNOTATION

Qualification thesis “Remote work search software system based on Android Studio” in the specialty 123 "Computer Engineering" of bachelor education degree contains 74 pages of explanatory notes, 10 figures, 5 appendixes. The volume of graphic material is 2 sheets of A3 format.

The aim of qualification work is to develop remote job search mobile application based on Android Studio.

Research methods include methods of analysis and synthesis, comparison, computer modeling, testing.

The task of finding remote work based on mobile platforms is considered. The features of modern mobile applications and operating systems for their functioning are analyzed. Modern freelance exchanges for the execution of orders in remote format are characterized. The existing software for designing mobile applications were studied, their main advantages and disadvantages were determined. An algorithm for designing remote job search mobile application was developed based on the formulated requirements for system. A software prototype application with description of functions and their implementation was developed.

The software architecture is developed, indicating the use cases diagram. The user interface of remote job search mobile application is described. The results of testing the software application are presented.

The practical significance lies in the development of remote job search mobile application based on Android Studio, which implements communication between customer and developer.

Keywords: MOBILE APPLICATION, OPERATING SYSTEM, ANDROID STUDIO, FREELANCE EXCHANGE, KOTLIN.

ЗМІСТ

Вступ.....	9
1 Програмний додаток як інструмент мобільних пристроїв.....	11
1.1 Особливості сучасних мобільних додатків.....	11
1.2 Аналіз операційних систем для мобільних пристроїв.....	15
1.3 Аналіз фріланс-бірж.....	18
1.4 Постановка задачі кваліфікаційної роботи.....	22
2 Технології розробки програмних додатків.....	24
2.1 Програмні засоби розробки мобільних додатків.....	24
2.2 Алгоритм проектування мобільного додатку.....	28
2.3 Прототип мобільного додатку.....	31
3. Реалізація додатку пошуку віддаленої роботи.....	40
3.1 Архітектура програмного додатку.....	40
3.2 Інтерфейс користувача.....	43
3.3 Тестування мобільного додатку.....	49
Висновки.....	50
Список використаних джерел.....	51
Додаток А Світлокопії публікації.....	54
Додаток Б Лістинг коду програми реалізації функції реєстрації користувача ..	59
Додаток В Лістинг коду програми розробки графічного інтерфейсу....	61
Додаток Г Лістинг коду програми реалізації функції взаємодії користувачів...	63
Додаток Д Техніко-економічне обґрунтування.....	65
Додаток Е Довідка про використання....	74

					КР.КІ.9499965.00.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив		Жигало Я.			Літ.	Арк.	Акрушів
Перевір.		Савка Н.Я.				8	74
Консульт.		Савка Н.Я.			ЗУНУ.ФКІТ.		
Н. Контр.		Дубчак Л.О.			КІз-41		
Затвердив		Дубчак Л.О.					

ВСТУП

У наше життя стійко увійшли мобільні технології, які кардинальним способом покращують і процеси виробництва, і процеси споживання інформації. Використання мобільних технологій дозволяє бути в курсі всіх подій у світі, докладаючи при цьому мінімум зусиль. Крім того, мобільні технології дозволяють зменшити вартість продукції для кінцевих споживачів за рахунок оптимізації процесів, скорочення виробничих та невиробничих витрат.

Також за допомогою мобільних пристроїв люди не прив'язані до робочого місця та легко можуть отримати доступ до інформації в дорозі, цеху, на нараді й будь-де, витративши на це мінімум грошових коштів та часу. Це стосується і пошуку інформацій різного роду.

Всю користь мобільних технологій люди особливо відчують у форсмажорних обставинах, зокрема, наприклад, під час пандемії, війни. Адже вони уможливають здійснювати свою діяльність, не покидаючи свого місця перебування.

Останніми роками досить поширеною формою праці є дистанційна робота. Дистанційні співробітники можуть працювати повний робочий день у штаті однієї компанії, неповний робочий день – у різних компаніях або бути самозайнятими, виконуючи проекти будь-яким способом та у будь-який час. Часто за таких умов працівники себе називають цифровим кочівником – людина не прив'язана до конкретного фізичного місця та працює віддалено. Вакансії віддаленої роботи розміщують на різних інтернет-ресурсах, що потребує багато часу для їх опрацювання. В той же час, не завжди є передбачено зворотній зв'язок, що унеможливає отримати мобільні відповіді на запитання, що виникають.

Зараз у світі практично не залишилося людей, які б не користувалися мобільним пристроєм. Причинами їхньої популярності є зручний функціонал, доступ до всіх способів зв'язку і, звичайно ж, невеликий розмір пристрою. Слід зазначити, що мобільні пристрої практично замінили настільний комп'ютер.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Однак у цього є і свої недоліки, оскільки у зв'язку зі збільшенням мобільної техніки у населення з'являються деякі труднощі, що полягають у відстеженні різних потоків інформації. За таких умов в епоху цифрових технологій доцільними є програмні продукти, які об'єднують інформацію в певній галузі, зокрема розробка мобільних додатків, які б забезпечували пошук віддаленої роботи за швидко і з мінімальними зусиллями.

За даними досліджень, більшість людей користується мобільним телефонами на операційній системі (ОС) Android, що зумовлює доступність технологій розробки додатків. У зв'язку з цим, виникає задача розробки програмного додатку, який би допомагав знаходити віддалену роботу або підробіток із застосуванням мобільного телефону. Таким чином, метою роботи виступає створення мобільного додатку на базі операційної системи Android.

Об'єктом роботи є процес пошуку інформації в мобільному середовищі.

Предметом роботи є технології розробки мобільного додатку пошуку віддаленої роботи із механізмом взаємодії кандидата та роботодавця.

Відповідно до мети визначено такі задачі:

- проаналізувати існуючі мобільні додатки;
- проаналізувати сучасні операційні системи для мобільних пристроїв;
- охарактеризувати популярні фріланс-біржі;
- проаналізувати існуючі середовища проєктування мобільних додатків;
- охарактеризувати мови програмування для операційної системи Android;
- розробити прототип програмного додатку;
- здійснити реалізацію додатку пошуку віддаленої роботи;
- провести тестування програмного продукту;
- розрахувати техніко-економічні показники ефективності розробки.

Кваліфікаційну роботу підготовлено згідно поставленої задачі, беручи до уваги вимоги, що зазначено у [9-11].

Основні результати роботи опубліковано у тезах доповіді на Міжнародній науковій конференції «Організація наукових досліджень у сучасних умовах» 2024» [32]. Копії публікації поміщено у додатку А.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ПРОГРАМНИЙ ДОДАТОК ЯК ІНСТРУМЕНТ МОБІЛЬНИХ ПРИСТРОЇВ

1.1 Особливості сучасних мобільних додатків

Розвиток інформаційних технологій (ІТ) і масова дистрибуція мобільних пристроїв дозволили сформувати новий ринок – ринок мобільних додатків. Мобільні додатки є одним з основних трендів останніх років у сфері ІТ. Коло розробників зростає, кількість додатків і користувачів збільшується, і все більше компаній усвідомлюють необхідність створення власних додатків для підвищення конкурентоспроможності. Таке динамічне зростання мобільних користувачів, а також постійний розвиток технологій свідчать про те, що світовий ринок мобільних додатків знаходиться в фазі активного зростання. Це один з найперспективніших ринків, який відкриває широкі можливості для ІТ-компаній.

Мобільний додаток – це спеціально розроблене прикладне програмне забезпечення для конкретної мобільної платформи (iOS, Android і т.д.). Багато мобільних додатків попередньо встановлені на самому пристрої або можуть бути завантажені на нього з інтернет-магазинів, зокрема, таких як App Store, Google Play Market та інші, безкоштовно або за окрему плату [26]. Поширення мобільних додатків покликане полегшити життя користувачам мобільних пристроїв, а також урізноманітнити його.

Мобільний додаток має суттєві переваги в порівнянні з мобільною версією сайту:

- більш зручний і інтуїтивно зрозумілий інтерфейс;
- онлайн-програми лояльності;
- постійний зв'язок з користувачем;
- геолокація;
- найточніший збір даних про цільову аудиторію.

Мобільний додаток – це свого роду адаптер, який допомагає користувачу взаємодіяти з різного роду інформацією. У зв'язку з цим розрізняють [7]:

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- заявки на події, призначені для трансляції спортивних або інших заходів;
- сервісні додатки, які аналогічні сайтам, що відображають діяльність організацій.
- ігри, в тому числі освітні та виховні;
- інтернет-магазини, розроблені для онлайн-покупок;
- рекламні додатки, що використовуються для реклами різних брендів;
- бізнес-додатки, які дозволяють оптимізувати процес організації, забезпечуючи доступ до ділової інформації та інтеграцію з базами даних;
- системні додатки, які використовують додаткові налаштування і опції телефону і його програмного забезпечення;
- навігаційні та пошукові сервіси, які використовують GPS-модуль, що дозволяє використовувати телефон в якості повноцінного навігатора;
- мультимедійні додатки, які розширюють можливості телефону при роботі з відео- та аудіоінформацією;
- соціальні мережі, які є онлайн-сервісами для спілкування, поширення інформації та організації соціальних відносин;
- контент-додатки.

Це поділ безпосередньої сфери застосування. Така класифікація використовується не надто часто. Вона більше підійде звичайним користувачам. Розробників має цікавити поділ на різні типи програмного забезпечення щодо способу програмування. У відповідному сенсі виділяють [16]:

- нативні програми;
- гібридні додатки;
- веб-контент;
- мобільні програми React Native.

Кожен вид запропонованих додатків мають свої переваги та недоліки. Про них потрібно знати перед тим, як приступати до безпосередньої розробки.

Native App – найпоширеніший варіант. Такі програми створюються окремо для кожної операційної системи (Андроїд, iOS тощо). При згадці мобільних

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

програм, зазвичай, мається на увазі такий варіант. Прикладами нативних програм для смартфонів та планшетів можуть бути такі відомі продукти як Pokemon Go та Spotify.

До сильних сторін нативних додатків відносять:

- продуктивність та безпека. Пов'язано це з тим, що програма створюється для кожної операційної системи, враховуючи наявні особливості платформи;
- функціональність. Нативні мобільні програми покращують користувацький досвід. Вони намагаються використати всі можливості ОС для повноцінної роботи;
- високий рівень безпеки. Зламати таке програмне забезпечення нелегко.

Основний мінус нативної розробки – це вартість. Для кожної мобільної платформи потрібно складати окрему версію програми, підтримувати її. Це тягне за собою додаткові витрати. Цей вид програм займає багато місця в пам'яті пристрою. По міру виходу нових версій обсяг програми може зростати.

Веб-програми випускаються у вигляді адаптованих сайтів, які відкриваються через браузер. Користувач не завантажує та не зберігає його на своєму пристрої. Якщо потрібно завантажити відповідну програму, доцільно говорити про її додавання до закладок інтернет-браузера.

Прикладами веб-застосунків можуть бути PWA – прогресивні веб-програми, які представлені нативним програмним забезпеченням усередині браузера. Деякі відповідні категорії приписують Google Maps.

До переваг веб-програм відносять:

- відсутність необхідності налаштування під операційну систему, що робить розробку більш простою, швидкою, а також менш витратною;
- програми не вимагають попереднього завантаження. Вони не займають простору у пам'яті мобільного пристрою;
- немає необхідності оновлювати. Їх набагато простіше підтримувати з боку програміста, а користувачу не доведеться при доопрацюванні проекту нічого додатково встановлювати.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Недоліком є залежність від браузера, що впливає на функціональність. Те, що в одному браузері працює нормально, в іншому не підтримується або відображається з помилками. Залежність від Інтернету – без підключення до мережі активувати мобільний веб-додаток ніяк не вийде.

Гібридний вид. До відповідної категорії відносять веб-програми, які виглядають як нативні, а також мають їх ознаки: висока продуктивність, іконки на робочому столі, підтримка функціонування в автономній формі. Приклад такої програми є Uber, деякі відносять до цієї категорії Instagram []. Такі мобільні програми є бюджетними і швидко реалізованими, покращують користувацький досвід. Така категорія підтримує взаємодію з операційною системою пристрою. Продуктивність є нижчою, ніж у нативних додатків.

До окремої групи відносять програми React Native, заснований компанією Meta (Facebook у минулому) [30]. Це відкрита і проста у освоєнні бібліотека для створення програм, які однаково добре запускаються як на смартфонах/планшетах, так і на комп'ютерах/ноутбуках.

До переваг кросплатформової розробки відносять [23]:

- вартість;
- швидкість освоєння фреймворку та безпосереднього створення коду;
- функціональність React Native.

Таким чином, розібратися із принципами створення подібних програм зможе навіть новачок, але функціональність, продуктивність та безпека підсумкового продукту виявляться під питанням.

Оскільки мобільні додатки розгортають на операційній системі, яка є ключовою складовою сучасних девайсів, які забезпечують функціональність, безпеку та досвід користувача, то у наступному підрозділі розглянемо детальніше існуючі операційні системи для мобільних пристроїв.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз операційних систем для мобільних пристроїв

Більшість операційних систем є вільним і відкритим вихідним кодом програмного забезпечення, що свідчить про відсутність обмежень на те, хто може його завантажити. Операційна система смартфона, іноді звана мобільною ОС, є свого роду ОС, призначеною для роботи на мобільних пристроях, таких як смартфони, планшети, смарт-годинники та інші подібні пристрої.

В останні роки на ринку ОС для мобільних пристроїв не відбулося кардинальних змін. Зміни в рейтингу стосуються в основному тільки трьох лідерів ринку: Android, iOS, ОС Windows Mobile.

Android – це ОС для смартфонів, планшетних комп'ютерів, електронних книг, медіаплеєрів, розумних годинників, ноутбуків, окулярів Google, телевізорів, автоматичних систем управління автомобілем та інших пристроїв. Вона заснована на ядрі Linux і власній реалізації Google віртуальної машини Java. Розроблений компанією Android Inc., яка була викуплена Компанією Google в 2005 році [8].

Система пропонує дуже зручний інструментарій і максимальну гнучкість налаштувань, що дозволяє кожному користувачу смартфона або планшета на Android повністю налаштувати його під свої потреби. Зазначена ОС завжди позиціонувалася її розробниками як система з відкритим вихідним кодом. Це дозволяє будь-кому створювати власні додатки, ігри та інші додатки для розширення можливостей Android-гаджетів.

Розробники спочатку все продумали так, щоб ОС працювала максимально швидко навіть на самому «бюджетному» обладнанні. Це, безсумнівно, плюс, адже користуватися сучасними смартфонами можуть навіть люди, які не мають великого запасу грошей. З мінусів варто відзначити погану автономність пристроїв. При активному використанні смартфон або планшет доведеться заряджати щодня, а в деяких випадках і 2 рази на день. Виробники активно працюють над усуненням цього недоліку і останнім часом на ринку почали

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

з'являтися пристрої з помітно підвищеною автономністю. За статистикою на ринку мобільних ОС Android лідирує і становить 72%.

iOS – закрита операційна система корпорації Apple, встановлена виключно на смартфонах iPhone, планшетах iPad, а також музичному плеєрі – iPod. Розробка iOS вимагає від розробника або апаратного забезпечення Apple і середовища розробки XCode, або використовувати сторонні сервіси для компіляції додатків. Також зараз існує безліч фреймворків для написання рідних кросплатформних додатків, зокрема, іonic [7].

Ionic – один з найвідоміших фреймворків для кросплатформної розробки, побудований на основі Apache Cordova, який забезпечує доступ до різних функцій пристроїв, таких як геолокація, push-сповіщення, камера та багато іншого, що дозволяє розробникам створювати додатки для iOS та Android за допомогою інструментів веб-розробки, таких як HTML, CSS та JavaScript [5].

Всі додатки для iOS завантажуються в Apple App Store і повинні бути підписані сертифікатом розробника. Таким чином, користувачі App Store можуть бути впевнені, що завантажують конкретну програму від конкретного розробника, а не підробку від імені когось іншого. Процедура підписання заявки дозволяє ОС зрозуміти, хто такий розробник і переконатися, що додаток не був змінений з моменту його створення. За статистикою на світовому ринку iOS займає близько 27%.

Крім вищесказаного, існують і інші ОС для мобільних пристроїв, наприклад: Windows Mobile OS, BlackBerry, Symbian, Bada, Palm, Meego, Open webOS та інші. Вони не дуже популярні і займають дуже невелику частку на світовому і внутрішньому ринку – не більше 1% всього ринку.

Розробка мобільного додатку для відразу двох операційних систем вимагає більше часу і грошей, тому доцільно почати з однієї платформи. Щоб зробити правильний вибір, потрібно враховувати різні особливості iOS і Android.

Android є найпопулярнішою мобільною ОС в світі. В Україні ця платформа займає 84% ринку, тоді як iOS становить лише 15%. Отже, додаток для Android може охопити більш широку аудиторію [3].

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Обидві ОС є багатообіцяючими. За останніми даними, частка Android і iOS на світовому ринку в сукупності становить 99%. Спостереігаючи за тенденцією, у найближчі кілька років ці платформи не втратять актуальності.

Знання цільової аудиторії є одним з найважливіших факторів при розробці додатку. Інформація про вік, дохід, сімейний стан, професію та інтереси допоможе визначити, якій ОС віддають перевагу майбутні користувачі програми.

iOS має обмежену файлову систему, тому деякі ідеї не можуть бути реалізовані. Android дає розробникам більше свободи, оскільки Google надає доступ до файлів і апаратного забезпечення ОС.

Проаналізувавши представлені дані, можна зробити висновок, що найоптимальнішим вибором ОС для мобільного додатку є ОС Android. Такий вибір обґрунтовується не тільки часткою ринкової зайнятості, але і повною відкритістю системи, що дозволяє розробляти продукти без порушення будь-яких ліцензійних угод.

Виділяючи особливості системи, можна перерахувати позитивні характеристики ОС Android:

- відкрита операційна система;
- просте налаштування системи для середнього користувача;
- великий вибір безкоштовних додатків;
- існує режим «Для розробників», який дає можливість легко протестувати розроблений додаток, не вимагаючи підпису і розблокування на Windows Mobile і iOS;
- широкий спектр плагінів бібліотек, які спрощують розробку додатків.

Одним з найважливіших і значущих критеріїв для розробника при виборі ОС є її широка поширеність на пристроях різних компаній, наприклад, на Samsung, HTC, Xiaomi, Huawei, і т.д. Таким чином, розробник може домогтися більш широкого поширення свого додатку.

Як і будь-який інший продукт, Android має й недоліки, зокрема:

- повна відкритість системи може заплутати користувача через неможливість налаштувати систему під себе;

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- високі витрати пристрою, що вимагає від розробника подальшої оптимізації розробленого додатку;
- високі витрати оперативної пам'яті пристрою, який з кожною новою версією ОС тільки зростає.

Проте вказані недоліки не суттєво впливають на вибір, оскільки технології розробки програмних додатків на ОС Android доступніші, дешевші, поширеніші та легші у вивченні та використанні.

1.3 Аналіз фріланс-бірж

Фріланс-біржі – це спеціальні веб-сервіси для пошуку віддалених співробітників або виконавців для конкретного проекту або завдання. На такому сервісі можна опублікувати опис проекту або задачі і отримати ряд пропозицій від фрілансерів [17]. Щоб визначити, якому фрілансеру довірити свою роботу, можна не тільки порівняти запропоновані ними ціни і терміни, але і вивчити профілі кандидатів на біржі. Профіль, як правило, містить коротке резюме, рейтинг фрілансера на біржі (виставлені клієнтами), історію виконаної роботи, відгуки, портфоліо.

Для того, щоб зробити мобільний додаток найбільш актуальним і затребуваним проведено порівняльний аналіз послуг вже представленого в мережі інтернет сервісу фрілансерського обміну.

У Kabanchik.ua функціонал спрямований на зручний пошук спеціалізованих майстрів для виконання різних напрямків послуг – від прибирання до фрілансінгу [6, 17]. Потрібно швидко доставити покупку з магазину або знайти майстра для ремонту ноутбука? Досить встановити додаток на смартфон або планшет, створити завдання і через пару хвилин сам майстер зв'яжеться для обговорення деталей. Інтерфейс вибору категорії додатків показаний на рисунку 1.1.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 Інтерфейс мобільного додатку Kabanchik.ua

Користувачі програми можуть створити завдання і знайти фахівця в більш ніж 700 категоріях. Найпопулярнішими з них є:

- будівельно-ремонтні роботи: від склеювання шпалер до ремонту «під ключ»;
- кур'єрська доставка і транспортні послуги: від доставки квітів до допомоги в переїзді квартири;
- послуги з прибирання – від прибирання вікон до комплексного генерального прибирання;
- домашній майстер – сантехніка, електромонтажні роботи, монтаж меблів і невеликі домашні справи;
- ремонт обладнання – побутового, комп'ютерного, мобільного;
- послуги тренерів та репетиторів;
- freelance exchange – послуги копірайтерів, дизайнерів, розробників

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

сайтів та інших фрілансерів.

Інтерактивні меню і великі кнопки дозволяють створювати завдання буквально в дорозі. Щоб знайти виконавця, достатньо вибрати відповідну категорію та внести необхідні дані:

- коротко описати роботу;
- вказати дату й час виконання завдання;
- вибрати місто;
- запропонувати ціну;
- при необхідності завантажити фото для більш точного опису необхідних робіт;
- прикріпити файл з детальною технічною специфікацією.

Після створення програми інформація автоматично відправляється виконавцям і через кілька хвилин починають з'являтися перші пропозиції. В один клік клієнт може побачити інформацію про кожного кандидата – відгуки, рейтинги, портфоліо, характеристики. Замовник усвідомлює кожну зміну задачі: додаток відправляє повідомлення про нові додатки, стан завдання, новий зворотний зв'язок від виконавця.

Недоліки такого сервісу:

- паспортні дані потрібні під час реєстрації;
- системний інтерфейс не достатньо гнучкий.

Freelancehunt – це біржа фрілансу від українських розробників. Останнім часом він набирає популярність і займає перші місця в рейтингах. Це пов'язано з високим рівнем професіоналізму адміністрації проекту, яка приділяє достатньо уваги своїй розробці.

На Freelancehunt практично всі клієнти і виконавці ввічливі і уважні до деталей. Звичайно, що це призводить до підвищення загального рівня сайту. Людям комфортно працювати з цим обміном, тому його часто рекомендують в блогах і порталах [18, 25].

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

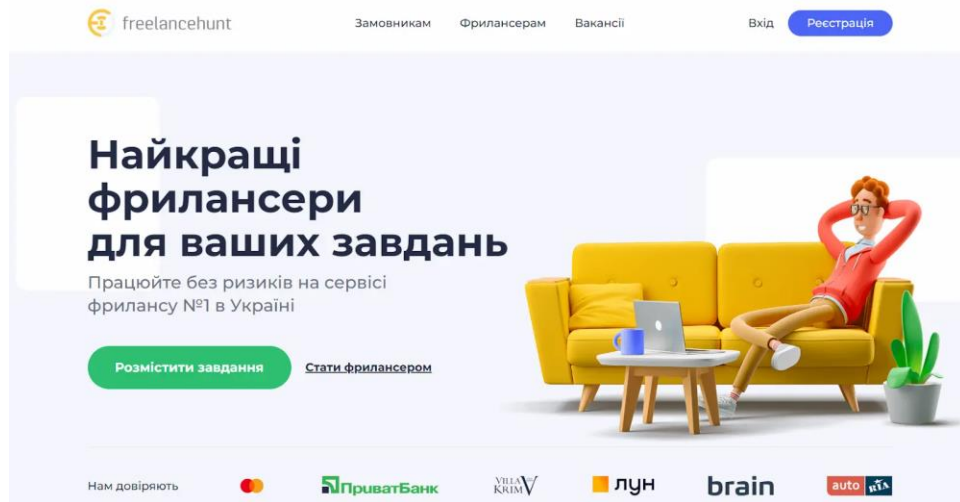


Рисунок 1.2. – Інтерфейс платформи Freelancehunt

На біржі є замовлення для: програмістів, копірайтерів, маркетологів, SEO-спеціалістів, дизайнерів, контент-менеджерів, веб-майстрів, аудіо/відео-редакторів, фахівців в рекламних мережах, перекладачів.

Переваги такого сервісу:

- низький поріг входження;
- більша кількість замовлень для новачків;
- зручний інтерфейс;
- мало посередників;
- доступні ціни на замовлення;
- низька ціна на PRO-акаунт;
- порівняно просте просування по рейтингу.

Недоліки:

- взяти замовлення без PRO-акаунту практично неможливо;
- висока конкуренція на деякі види робіт;
- взяти замовлення без рейтингу і портфолію практично неможливо.

Weblancer третій варіант біржі для пошуку проектів створення готових сайтів в Україні. Виконавець на біржі може розмістити профіль та резюме для пошуку постійної роботи. З кожної операції біржа бере комісію 5%. Після реєстрації фрілансер може надіслати 5 безкоштовних заявок. За додаткові дії фрілансер отримує додаткові безкоштовні відгуки (наприклад, заповнення

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

профілю) [17]. Екранну форму вікна інтерфейсу платформи показано на рисунку 1.3.

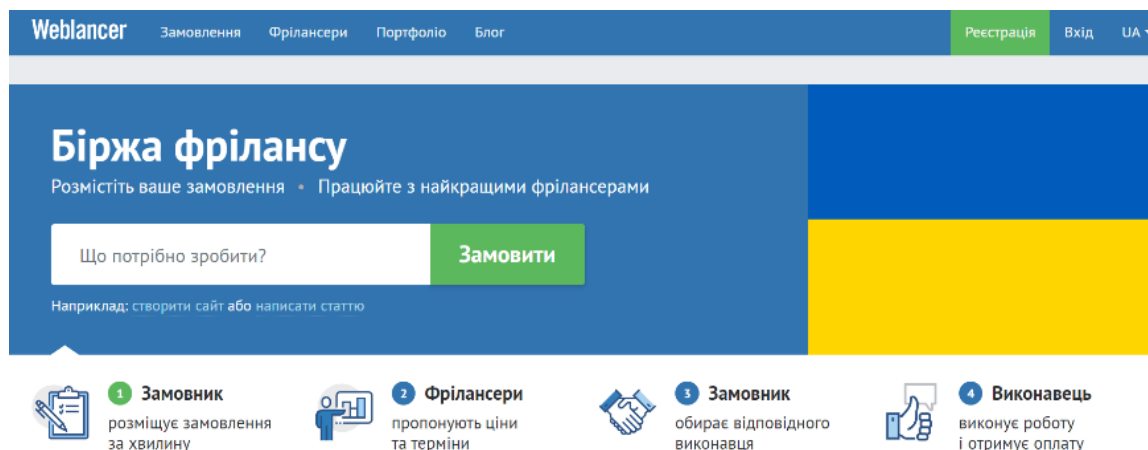


Рисунок 1.3 – Приклад інтерфейсу біржі фрілансу Weblancer

Вартість передплати залежить від обраної спеціалізації (їх може бути до 8). Якщо вибрати спеціалізацію "Створення сайтів під ключ", то щомісяця потрібно платити 5\$ за передплату. Екранну форму вікна інтерфейсу платформи показано на рисунку 1.3.

Таким чином, із аналізу випливає, що сервіси для пошуку віддаленої роботи характеризуються досить високою конкуренцією і є недосконалими для початківців, які, щоб знайти замовлення, повинні пропонувати досить низьку ціну.

1.4 Постановка задач кваліфікаційної роботи

З розвитком технологій та нестабільної ситуації в країні налагоджені форми відносин між роботодавцями і співробітниками втрачають свою актуальність й змінюються новими формами. За таких умов існує певна гнучкість, яка дозволяє віддаленій роботі швидко набирати обертів у передових сферах економіки.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналізуючи вплив COVID-19 на світовий та український фріланс, зростає реальна статистика вільного ринку праці. На фоні останніх подій мільйони людей задумалися про новий спосіб заробітку. 2020 рік став справжнім бумом у розвитку дистанційної роботи й показав зростання на 230%, в порівнянні з 2019 роком.

Вже зараз більшість організацій роблять акцент на розвитку напрямку віддаленої роботи і створення нових можливостей для дистанційних співробітників. Популярність роботи з фрілансерами в Україні суттєво зростає, виконавці можуть опублікувати свої портфоліо, а замовники – обрати фахівця.

Замовник знаходить підрядника, орієнтуючись на його спеціалізацію, рейтинг, портфоліо та відгуки клієнтів. Тоді він описує задачу, обговорює деталі, складає технічне завдання і розпочинається робота над проєктом. Після завершення розробки підрядник отримує гонорар, а клієнт – виконане замовлення.

Більшість бірж фрілансу, як свідчать результати проведеного аналізу, знаходяться на сайті без підтримки мобільного додатку або не зручні для перегляду з телефону, що ускладнює швидке реагування на пропозиції. Адже за допомогою мобільного додатку користувач може в будь-який момент перевірити подачу замовлень і обговорити замовлення в будь-якому зручному місці.

Таким чином, сформульовано мету кваліфікаційної роботи, що полягає у розробці мобільного додатку під управлінням операційної системи Android, призначеного для взаємодії користувачів додатків клієнт-виконавець. Такий продукт буде придатний для розміщення оголошень про роботу і неповний робочий день, міститиме зручні інструменти комфортного пошуку замовлень для виконавців і обговорення процедури виконання роботи. Досягнення мети можливо за рахунок виконання таких задач:

1. проаналізувати існуючі середовища проєктування мобільних додатків;
2. охарактеризувати мови програмування для ОС Android;
3. розробити прототип програмного додатка;
4. здійснити реалізацію додатка пошуку віддаленої роботи;
5. провести тестування програмного продукту;
6. розрахувати техніко-економічні показники ефективності розробки.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНИХ ДОДАТКІВ

2.1 Програмні засоби розробки мобільних додатків

Одним з найважливіших елементів процесу розробки додатків є вибір правильного середовища розробки, яке залежить не тільки від цільової платформи, але й від рівня підготовки самого розробника. Безсумнівно, розробник з широким спектром знань в області програмування зможе розробити додаток навіть в блокноті, але на відміну від розробки додатків для персональних комп'ютерів, при розробці мобільних додатків слід дотримуватися умов, які висуває сама платформа.

Система Android – це універсальна платформа, на базі якої можна створювати інноваційні додатки та ігри для мобільних пристроїв за допомогою мови Java. Android Studio – це середовище розробки Android, засноване на середовищі розробки IntelliJ IDEA, яке надає інструменти для розробки та налагодження системи. Платформа доступна для завантаження на офіційному сайті Google. Середовищі включає такі можливості [22]:

- підтримка розробки додатків на основі Gradle;
- спеціально створений під Android рефакторинг;
- інструменти для пошуку проблем із продуктивністю, зовнішнім виглядом, сумісністю версій;
- створення макетів програм на основі майстра експрес-шаблонів;
- візуальний редактор для перегляду змін в режимі реального часу, без необхідності встановлювати додаток на пристрій або запускати додаток в емуляторі.
- вбудований ADV емулятор смартфонів різних версій і виробників.

LibGDX – це фреймворк для розробки ігор і додатків, створених BadLogicGames. Основна частина фреймворку написана на Java на основі C і C++. Дозволяє розробляти кросплатформні додатки для різних типів операційних систем [19].

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

При необхідності libGDX може використовувати власний код замість коду Java, щоб розробник міг зосередитися на продуктивності. Вся функціональність залежить від Java API, саме тому немає необхідності думати про крос-компіляції рідного коду на всіх платформах. Особливості середовища LibGDX:

- підтримка Windows, Linux, Mac OS X, Android, iOS, браузерів з підтримкою WebGL;
- гнучкість у збільшенні можливостей на основі додаткових модулів: фізичного рушія Box2D для взаємодії з об'єктами та фізикою реального світу, підтримка шрифтів TrueType або робота з 3D-об'єктами;
- вбудований набір бібліотек для створення графічного інтерфейсу як для програм, так і для ігор;
- досить стабільні і високі показники розроблених додатків.

Для переваг середовища розробки включають:

- широкий асортимент плагінів;
- кросплатформеність;
- збільшення швидкості розробки за рахунок вже створених методів;
- велика кількість документації.

Недоліки LibGDX:

- платформа призначена для розробки ігор, а не додатків;
- функціональність залежить від Java API, саме тому досить складно доопрацювати будь-які методи.

React – це платформа і, одночасно, JavaScript фреймворк для розробки рідною мовою React. Його особливість полягає в тому, що можна розробляти гібридні мобільні додатки, які можуть бути встановлені на різних операційних системах. При цьому додатки, створені на цьому фреймворку, практично не будуть відрізнятися від рідних [30, 31]. Варто сказати, що нативний додаток – це додаток, код для кожної платформи якого пишеться окремо. На відміну від цього, гібридний код програми записується один раз для всіх платформ. Причини використання React.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

1. React Native програми написані мовою Java Script, яка є однією з найпопулярніших мов програмування. Основну частину коду розробник пише на Javascript, яка є поширеною мовою програмування для всіх платформ, потім цей код взаємодіє з рідними компонентами ОС. В результаті розробник отримує додаток, який працює на платформах: iOS, Android, Universal Windows Platform [31].

2. Щоб почати розробку програми, потрібен досвід написання коду на Javascript і знання React. До речі, всі платформи мають подібні особливості, через що ризик плутанини в процесі розробки досить високий. Іншими словами, React Native простий і зручний, якщо розробник розуміє, що потрібно зробити і отримати в результаті розробки.

3. Скорочення витрат часу за рахунок кросплатформної розробки.

4. Близькість до рідних додатків за зовнішнім виглядом, через що користувач, який звик до появи додатків на певній ОС, не буде відчувати дискомфорту.

5. Автоматичні оновлення.

Таким чином, можна сформулювати переваги React:

- кросплатформний;
- простота і зручність розробки;
- великий обсяг документації.

Існує також велика кількість мов, за допомогою яких можна створити якісний додаток. Деякі мають вбудоване середовище розробки, що значно полегшує процес створення програми. Іншим потрібні додаткові інструменти, але їх функціональність не менш широка. Тому далі охарактеризуємо, на яких мовах відбувається програмування Android.

Java є одним з найстаріших і найбільш перевірених методів розробки якісного додатку. Це популярна мова програмування для Android, яка в 2019 році увійшла в ТОП-5 кращих і функціональних мов. Однак при цьому вона досить складна, розібратися в ній новачку важко, оскільки мова працює по типу об'єктно-орієнтованого програмування.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Основною перевагою Java є наявність власного середовища розробки під назвою Android Studio. Процес розробки програм спрощується за рахунок візуального UI-редактора, функції автодоповнення коду та інших можливостей.

З точки зору функціональності, Java є однією з найпотужніших мов. Вона дозволяє реалізовувати найскладніші задачі і проекти. Тут використовуються не тільки класи Java, але і XML-файли.

Kotlin – відносно молода мова, яка з'явилася в 2017 році. Kotlin включає кращі якості інших мов. Він має масу переваг, включаючи автоматичне виявлення типів даних, функції-розширення, простий у розумінні, освоєнні та доступний для кожного [28, 29]. Створено Kotlin на основі Java, тому переходити з Java дуже просто.

Мова легко інтегрується з багатьма фреймворками, легко вивчається, її код відкритий. За допомогою чіткого синтаксису підвищується продуктивність під час розробки додатків, а послідовність легко дотримується. Стислість, легке читання і лаконічність – основні якості Kotlin.

Але в той же час мова містить недоліки. Швидкість компіляції коду нестабільна, вона може проходити як швидко, так і зі значними затримками. Мова нова, тому навчального матеріалу не так багато, спільнота ще не досконально розроблена, тому в ході труднощів або питань розробникам часто доводиться самотійно шукати рішення.

C/C++ є однією з найстаріших мов, яка використовується для створення власних додатків з потужною завантаженою функціональністю. Це можуть бути ігри або технічно складні програми. Часто для розробки додатків використовується не тільки C/C++. На ній пишуться тільки певні частини або модулі, які виконують складні операції і задачі. Основною перевагою мови є запуск коду, який виробляється не через віртуальну машину Java, а безпосередньо через пристрій. Це дозволяє отримати більше контролю над елементами системи пристрою [19].

C/C++ є універсальною мовою, здатною підтримувати різні стилі програмування. Це дає багато свободи розробнику, при цьому не накладаючи

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

навантаження на програму. Підходить для роботи в будь-якому середовищі програмування. Це одна з основних мов, за допомогою якої здійснюється не тільки програмування на Android, але і на будь-яких інших системах.

Мова Python сприймається таким чином, що програмування на Android нею неможливе. Проте шанувальники цієї мови створили потужні інструменти, які допомагають розробляти додатки і компілювати код. Незважаючи на те, що далеко не найзручніша мова для мобільної розробки, вона все ж Python користується великим попитом [3].

Python досить простий в освоєнні та вивченні, зрозумілий і має зручний синтаксис. Працює з різними середовищами розробки, що робить його універсальним і доступним для всіх. Швидко зростає, набирає популярність, активно розвивається спільнотою і підтримкою. На основі додаткових інструментів, Python також використовується для розробки мобільних додатків.

Таким чином, проаналізувавши найпопулярніші середовища розробки додатків, для реалізації мобільного додатку пошуку віддаленої роботи обрано середовище Android Studio, що містить велику кількість модулів, які корисні при розробці мобільних додатків, наприклад, вбудований вигляд виду створених вікон, інструмент емуляції мобільних пристроїв для налагодження програми.

На основі аналізу найпопулярніших мов програмування для ОС Android, для реалізації мобільного додатку обрано мову Kotlin, яка є доступною та легкою для розуміння кожному.

2.2 Алгоритм розробки мобільного додатку

Для розробки мобільного додатку необхідно виконати певні кроки, починаючи із вивчення предметної області та аналізу вимог до системи [1, 2, 13, 20]. Етапи проектування мобільних додатків наведено на схемі у КР.КІ.9499965.00.00.000.ПЗ.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Аудиторія додатку в цілому – це, як правило, молодь і люди середнього віку, які активно шукають віддалену роботу або неповний робочий день. Послугою можуть користуватися тільки люди, які досягли 16-річного віку. Таким чином, у додатку повинні бути реалізовані такі функції:

- надання функціоналу для введення вхідних даних у вигляді додавання нового порядку в існуючу базу даних;
- надання користувачу можливості вибору категорії;
- формування реакції системи в разі відсутності підключення до сервера;
- зберігання даних і надання доступу до даних для представлення переліку замовлень;
- навігаційне меню у всіх фрагментах програми;
- можливість реєстрації користувачів;
- пошук по розділах.

До проектування системи висуваються також функціональні вимоги вимоги.

1. Інтерфейс системи повинен бути максимально гнучким, щоб користувач не витрачав багато часу на освоєння сервісу.
2. Дизайн програми не повинен бути громіздким.
3. Вікно авторизації не повинно вимагати від користувача величезної кількості вхідних даних.
4. Для роботи програми потрібен доступ до інтернету, весь процес роботи з сервісом повинен здійснюватися в онлайн-режимі.
5. Додавання нових пропозицій – елементів, має відбуватися автоматично.
6. Додавання нових елементів не повинно порушувати архітектуру та цілісність програми.
7. Забезпечення безпеки, в тому числі при роботі з персональними даними користувачів.

Опишемо алгоритм розробки програмного додатку. Найпершим етапом розробки програмного забезпечення є процедура проведення комплексного аналізу вимог до створеного програмного забезпечення з метою визначення ключових цілей і задач кінцевого продукту. Результатом аналізу є формування

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

основних нормативних актів, на які підрядник буде спиратися в своїй роботі – технічного завдання з розробки програмного забезпечення. Технічне завдання повністю описує поставлені перед розробником задачі і характеризує кінцеву мету проекту в розумінні замовника.

Наступним ключовим етапом є проектування, тобто моделювання теоретичної основи програмного продукту. Найсучасніші інструменти програмування дозволяють частково поєднувати етапи проектування і кодування, тобто технічну реалізацію проекту, виходячи з об'єктно-орієнтованого підходу, але повноцінне планування вимагає більш ретельного і скрупульозного моделювання.

Якісний аналіз перспектив і можливостей створеного продукту є основою для його повноцінного функціонування і виконання всього комплексу задач, покладених на програмне забезпечення. Одним із складових етапу проектування, є вибір інструментів і ОС.

Таким чином алгоритм розробки мобільного додатку пошуку віддаленої роботи, містить три кроки і підкроки.

Крок 1. Підготовчий етап:

- генерація ідей;
- дослідження подібних розробок з виявленням їх сильних і слабких сторін;
- дизайн інтерфейсу, створення ескізів;
- розробка імен;
- вивчення технологій розробки мобільних додатків на ОС Android.

Крок 2. Основний етап:

- вибір кольорів і стилів шрифтів;
- створення макетів екрану за допомогою графічного редактора зображень;
- розробка додатків в студії Android;
- створення стилів і шаблонів головних вікон;
- реалізація пошуку;

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

- здійснення авторизації (підрядник, замовник);
- реалізація шаблону анкети;
- реалізація шаблону замовлення;
- тестування продукту на смартфонах.

Крок 3. Завершальний етап:

- виправлення виявлених недоліків;
- створення програмних значків.

На основі вищевикладеного можна перейти до процедури проєктування прототипу програмного додатку, що опишемо у наступному підрозділі.

2.3 Прототип мобільного додатку

Прототипи стали незамінним етапом професійної розробки програмного забезпечення, який візуалізує кінцевий продукт і допомагає виявити серйозні проблеми на ранніх етапах, допомагаючи їх уникнути на пізніх стадіях розробки. При проєктуванні додатку пошуку віддаленої роботи необхідно розробити інтуїтивно зрозумілий для користувачів графічний додатковий інтерфейс. Таким чином, основні структури під час створення інтерфейсу:

- діяльності (Activities);
- ресурси (файли XML) – включають компонування, рядки, зображення, стилі;
- файл AndroidManifest.xml – містить інформацію про налаштування конфігурації;
- клас View – базовий клас для всіх компонентів інтерфейсу користувача. Відповідає за промальовування зображення та обробку подій.
- айл R.java – відповідає за ідентифікацію ресурсів.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Варто пам'ятати, що програма може містити кілька діяльностей (екранних форм), одна з яких визначається, як головна і відображається при першому запуску програми. Також діяльності однієї програми незалежні один від одної.

Для розробки віджетів існує 2 способи:

1. XML-файл компонування, що містить опис графічних елементів та їх параметрів;
2. динамічне створення у програмному кодї класу екранної форми.

Програмування графічного інтерфейсу користувача управляється подіями, які реагують на ініційовані для користувачів події, такі як торкання екрану. Крім безпосереднього написання коду програм, можна скористатися середовищем розробки Android Studio, що дозволяє збирати графічний інтерфейс з готових об'єктів, таких як кнопки та текстові поля, перетягуючи їх у певні місця екрану, додаючи підписи та змінюючи їх розміри.

Компоненти графічного інтерфейсу в Android називаються уявленнями (views) [21]. Вертикальне уявлення LinearLayout використовується для розміщення тексту та графіки так, щоб кожне представлення займало половину вертикального простору LinearLayout. Компонент LinearLayout також дозволяє розміщувати уявлення по горизонталі. Для виведення тексту в програмі використовується компонент TextView, а графіка відображається у компоненті ImageView.

Графічний інтерфейс, створений для програми за замовчуванням, містить компонент TextView. Різні параметри цього компонента – текст, розмір шрифту, колір тексту, розмір відносно компонента ImageView в LinearLayout – налаштовуються у вікні якостей середовища розробки.

Мова XML (eXtensible Markup Language, тобто мова розширення розмітки) є природним способом опису графічних інтерфейсів [12]. Розмітка XML добре читається як людиною, так і комп'ютером. У контексті Android вона використовується для опису макетів компонентів та їх атрибутів: розміру, позиції, кольору, розміру тексту, полів та відступів. Android Studio розбирає розмітку XML, щоб відобразити макет у макетному редакторі та згенерувати код Java,

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

формує графічний інтерфейс на стадії виконання. Також файли XML використовуються для зберігання ресурсів програми: рядків, чисел, кольорів.

У кожного додатка існує тема, що визначає оформлення стандартних компонентів. Тема вказується у файлі AndroidManifest.xml програми. Можна налаштувати різні аспекти теми, визначаючи ресурси у файлі styles.xml, що знаходиться в папці res/values.

Android tools включає емулятор "Android Virtual Device" (AVD) [21]. Емулятор AVD дозволяє тестувати програми на віртуальному мобільному пристрої з ОС Android. Він створює кілька віртуальних пристроїв з різними конфігураціями.

Проект – це група пов'язаних файлів (наприклад, файли коду, ресурси та графічні файли), що утворюють додаток. Робота над додатком починається із створення проекту. Щоб створити додаток у середовищі Android потрібно перейти до пункту меню File -> New-> New Project. Після цього з'явиться діалогове вікно розробки нового проекту. Потрібно вибрати Activity, яка буде використовуватися за замовчуванням для головного екрана програми: (див. рис. 2.1).

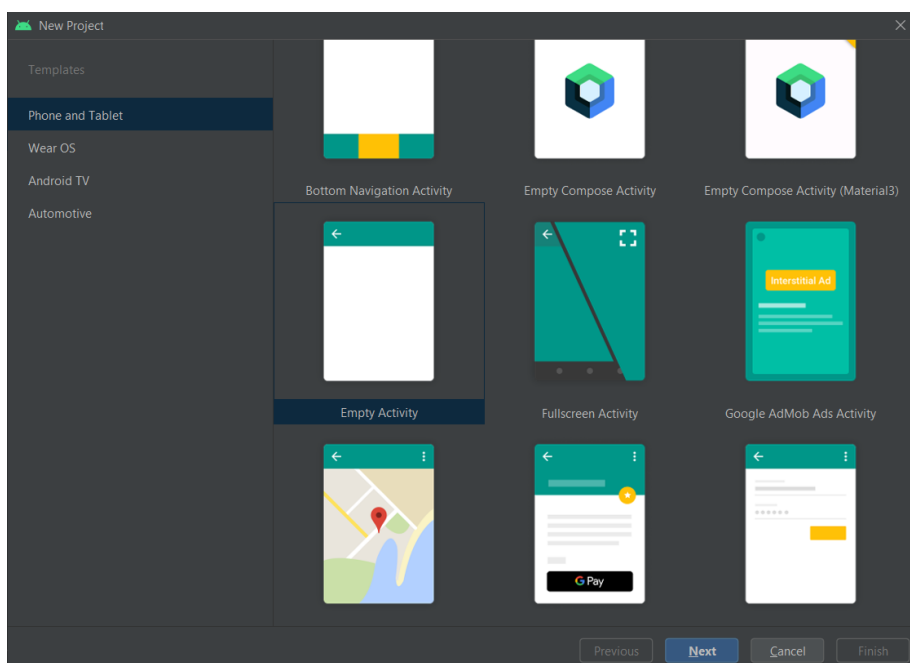


Рисунок 2.1 – Приклад вибору Activity

На кроці New Project (див. рис. 2.2) майстра Create New Project вводимо таку інформацію:

- Name: – ім'я програми;
- Package Name: – ім'я пакета Java для вихідного коду програми;
- Save Location: – шлях до папки на комп'ютері, в якій зберігатиметься проєкт. Типово Android Studio розміщує папки нових проєктів у вкладеній папці AndroidStudioProjects каталог облікового запису користувача.
- Language – вибір мови програмування;
- Minimum SDK – встановлення мінімальної версії проєкту, яка підтримується. За замовчуванням встановлюється версія Android 5.0, що покриває 98,6% пристроїв Android. Залишимо по замовчуванні і натискаємо кнопку Finish.

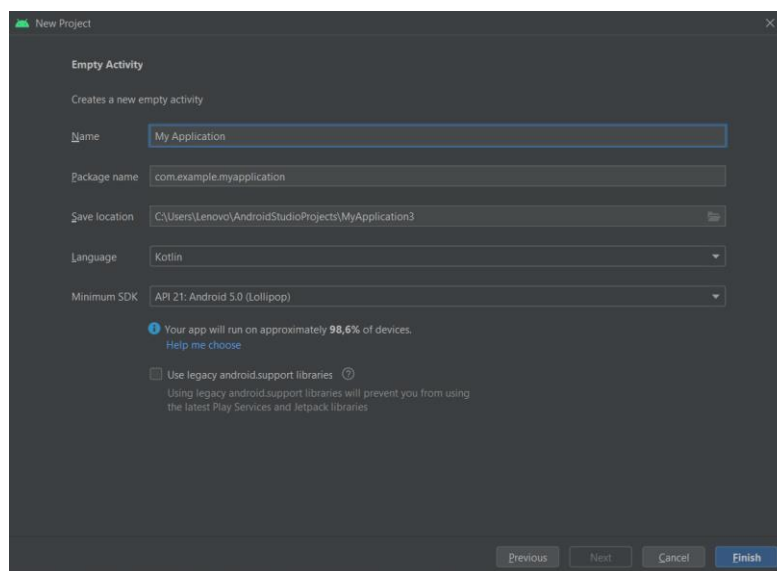


Рисунок 2.2 – Екранна форма вікна створення нового проєкту

Розглянемо структуру проєкту програми під ОС Android, яка створюється за замовчуванням. Проєкт може містити різні модулі й всі вони описуються файлом setting.gradle. Весь значущий код, файли інтерфейсу, класи kotlin за замовчуванням знаходяться у папці (модулі) app (див. рис. 2.3) [29].

Файл build.gradle містить інформацію, яка використовується при побудові проєкту. Кожен модуль має власний файл build.gradle, який визначає конфігурацію побудови проєкту, специфічну для даного модуля. Кожен проєкт містить один

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

модуль – app. У цьому модулі можна побачити кілька папок і файлів, найважливішими із яких є:

- каталог libs – призначений для зберігання бібліотек, які використовуються додатком;
- каталог src – призначений для зберігання вихідного коду. Він містить низку підкаталогів. Вихідні коди розміщуються в папці java.

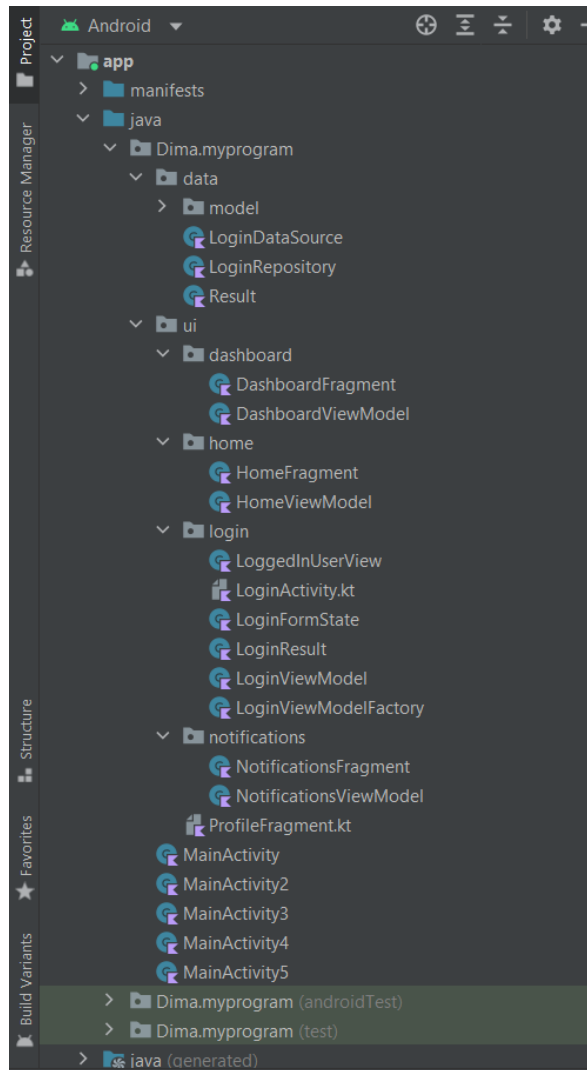


Рисунок 2.3 – Структура Android проекту

На відміну від багатьох програм Kotlin, програми Android не містять методу main. Натомість, у них використовуються чотири типи виконуваних компонентів – активності (activities), служби (services), провайдери контенту та ширококомвні приймачі (broadcast receivers). Додаток може мати багато активностей, одна з яких – перше, що ми бачимо під час запуску програми. Користувачі взаємодіють з

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

активності через уявлення (views) – компоненти GUI, що успадковують від класу View (пакет android.view).

Активність може керувати декількома фрагментами (fragments). На телефоні кожен фрагмент, зазвичай, займає цілий екран, а активність перемикається між фрагментами, виходячи з взаємодій користувача. На планшетах активності часто відображають кілька фрагментів на екран, щоб більш ефективно використовувати доступний простір.

Протягом свого існування активність може перебувати в одному з кількох станів – активному (тобто виконуватися), призупиненому або зупиненому. Переходи активностей між цими станами відбуваються у відповідь на різні події. «Активна активність» відображається на екрані і «має фокус» – тобто взаємодіє із користувачем. Призупинену активність видно на екрані, але не має фокусу (наприклад, на час відображення діалогового вікна із повідомленням). Користувач не може взаємодіяти з призупиненою активністю, доки вона знову не стане активною. Зупинена активність не відображається на екрані і, ймовірно, буде знищена системою, коли потрібно звільнити пам'ять, яку вона займає.

Активність зупиняється, коли інша активність перетворюється на активний стан. Наприклад, коли відповідаємо на телефонний дзвінок, програма, що управляє дзвінками, стає активною, а попередня програма зупиняється. При переходах активності між цими станами виконавче середовище Android викликає різні методи життєвого циклу (всі ці методи визначаються у класі Activity з пакета android.app). У програмах кожна активність буде перевизначати метод на створення. Цей метод викликається виконавчою системою Android при запуску активності – тобто коли її графічний інтерфейс готовий відобразитися, щоб користувач міг взаємодіяти з активністю. Також у активностей існують інші методи життєвого циклу: onStart, onPause, onRestart, onResume, onStop і onDestroy.

Для будь-якої Activity викликається метод onCreate, який відповідає за створення вікна. Для того, щоб можна було відобразити вид вікна, викликається метод setContentView, до якого передається числове значення ідентифікатора

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

ресурсу відображення, що отримується від класу нащадка. Тут також встановлюється ActionBar, який є у всіх Activity програми.

Після розгляду загальних питань побудови проєкту можна приступити до самої реалізації програми. Створюємо графічний інтерфейс авторизації користувача для програми. Макетний редактор (Layout Editor) дозволяє розробити графічний інтерфейс користувача шляхом перетягування у вікно додатки компонентів GUI, таких як Button, TextView, ImageView та ін. [21].

За замовчуванням опис макета для шаблону Empty App зберігається у файлі XML з ім'ям activity_main.xml у папці res/layout. Скористаємося макетним редактором та вікном Component Tree для побудови програми. Розмітка XML у файлі activity_main.xml буде відредагована лише для того, щоб змінити спосіб розміщення компонентів TextView, EditText та Button, що використовуються в програмі.

Оскільки екрани пристроїв Android мають різні розміри, роздільну здатність та щільність пікселів (DPI, Dot Per Inch), розробник, зазвичай, надає зображення з різними дозволами, а ОС вибирає графік, виходячи з щільності пікселів пристрою. Через це папка res проєкту містить кілька вкладених папок, імена яких починаються з префіксу drawable.

Для реалізації авторизації користувача реалізуємо клас MainActivity:

```
class MainActivity : AppCompatActivity() {  
    private lateinit var textView: TextView  
    private lateinit var editText2: TextView  
    private lateinit var mBind: ActivityMainBinding  
    private lateinit var mAuth: FirebaseAuth  
    private lateinit var button1: Button  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        textView = findViewById(R.id.textView)  
        editText2 = findViewById(R.id.EditText2)  
        mBind = ActivityMainBinding.inflate(layoutInflater)
```

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

setContentView(mBind.root)

button1 = findViewById(R.id.button1)

button1.setOnClickListener { inicFun() }
    }

private fun inicFun() {
    mAuth = FirebaseAuth.getInstance()
    if (mAuth.currentUser!=null){
        val reg = Intent(this, MainActivity3::class.java)
        startActivity(reg)
    }else{
        val noreg: Toast = Toast.makeText(this, "Перевірте правильність вводу
данних або зареєструйтесь", Toast.LENGTH_SHORT)
        noreg.show()
    }
}

fun buttonFirst(view: View) {
    val firstButton: Toast = Toast.makeText(this, "Hallo", Toast.LENGTH_SHORT)
    firstButton.show()
}

fun buttonTwo(view: View){
    val three = Intent(this, MainActivity2::class.java)
    startActivity(three)
}
}

```

Для реалізації реєстрації користувача реалізуємо клас MainActivity2, код якого наведено у додатку Б. Реалізуємо ряд перевірок для коректної реєстрації користувача.

Створимо графічний інтерфейс основного Activity програми, в якому відбувається взаємодія користувачів на основі інтеграції фрагментів. Код розмітку XML поміщено у додатку В.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Для реалізації Activity програми в якому відбувається взаємодія користувачів реалізуємо клас MainActivity3. Підключимо меню та фрагменти (див додаток Г).

Таким чином, розроблено простий графічний інтерфейс у макетному редакторі та налаштовано властивості компонентів у вікні властивостей. У класах задіяно багато засобів об'єктно-орієнтованого програмування мовою Kotlin, включаючи класи, об'єкти, інтерфейси, анонімні внутрішні класи та успадкування. Також наведено концепцію заповнення графічного інтерфейсу, тобто перетворення вмісту файлу XML на його екранне представлення.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ДОДАТКУ ПОШУКУ ВІДДАЛЕНОЇ РОБОТИ

3.1 Архітектура програмного додатку

Програмне застосування для ОС Android складається з набору активностей, кожній з яких відповідає екран програми. Кожна активність представлена у проекті класом, реалізованому мовою Kotlin, що зберігається в однойменному файлі з розширенням .kt [21, 27]. Кожна активність відповідає xml файл-опису. У xml-файлі описано у вигляді xml-коду розташування об'єктів, що візуалізуються. При запуску активності система Android автоматично розпізнає розмір екрана мобільного пристрою і приводить контент, що виводиться, у відповідність до розмітки, описаної в xml-файлі. Таким чином, одна і та ж сама активність буде виглядати однаково незалежно від діагоналі пристрою. Також для кожної програми Android повинен існувати xml-файл, в якому у вигляді xml-коду будуть прописані мінімальні вимоги до системи, а також активність, що викликається при запуску програми.

Діаграма, представлена на рисунку 3.1, відображає дії користувача при вході в програму.

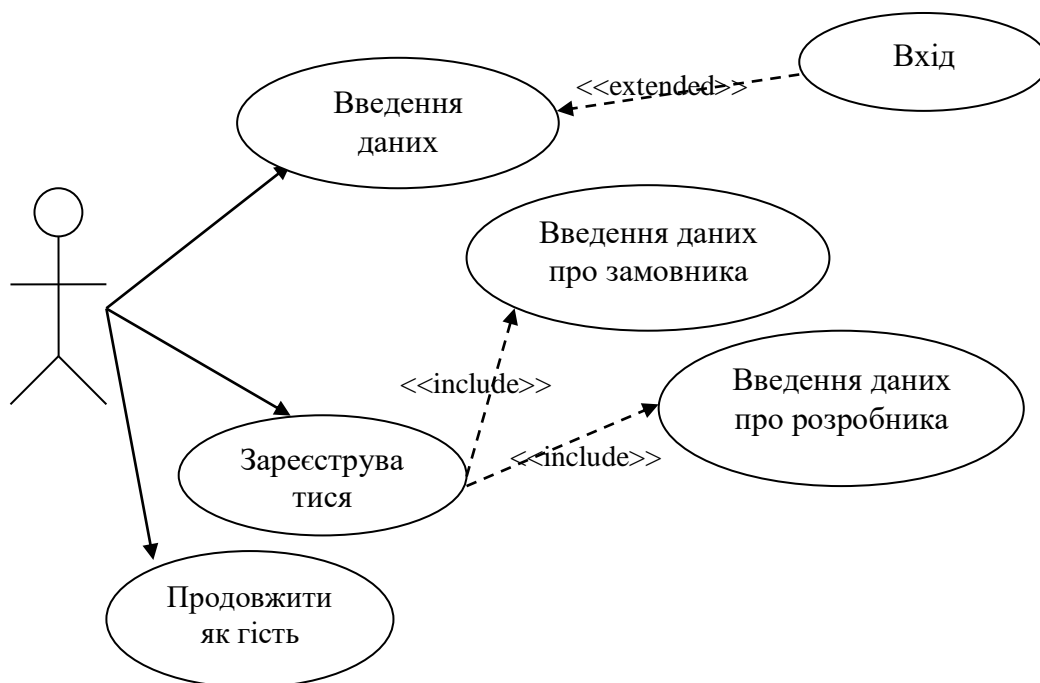


Рисунок 3.1 – Діаграма варіантів дій користувача при користуванні додатком

						КР.КІ.9499965.00.00.000.ПЗ	Арк. 40
Змн.	Арк.	№ докум.	Підпис	Дата			

При відсутності облікового запису користувач повинен зареєструватися у системі. Діаграма дій користувача при реєстрації представлено на рисунку 3.2.



Рисунок 3.2 – Діаграма дій користувача при вході реєстрації

Користувач має можливість створення та редагування свого профілю. Для зміни профілю доступні параметри, як показано вказані на рисунку 3.3 для виконавця та рисунку 3.4. для замовника.



Рисунок 3.3 – Варіанти редагування профілю виконавця



Рисунок 3.4 – Варіанти редагування профілю замовника

Користувач має змогу переглядати профілі інших користувачів. При цьому доступні такі варіанти дій, як зображено на діаграмі рисунка 3.5.

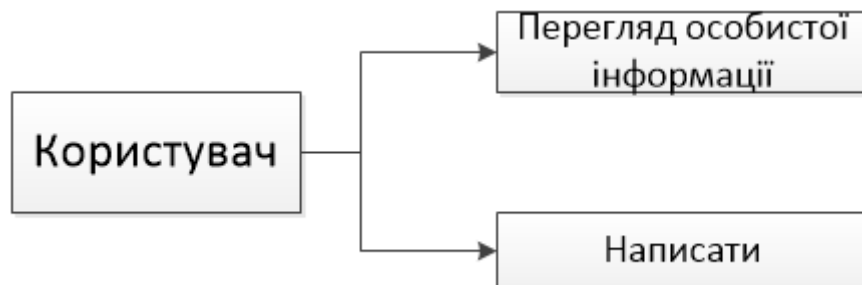


Рисунок 3.5 – Варіанти перегляду профілю інших користувачів

Зареєстрований користувач має змогу переглядати стрічку замовлення. При цьому доступні такі дії, як показано на схемі рисунка 3.6.



Рисунок 3.6 – Варіанти дій користувача зі стрічкою замовлення

Коли замовник створює своє замовлення, для цього необхідно вказати такі параметри: назва, категорія, місце, дата, короткий опис, вказати місце роботи на карті (див. КР.КІ.9499965.00.00.000.С1).

3.2 Інтерфейс користувача

Для входу в обліковий запис використовується зв'язка email і пароль, вказані при реєстрації [12]. Якщо реєстрацію ще не було здійснено, можна її здійснити, натиснувши кнопку з відповідною назвою (див. рис 3.7).

Реєстрація – це вже давно звичний для всіх процес, проте вона повинна бути швидкою та не втомлювати нудним заповненням інформації про себе. Тому користувачу запропоновано ввести тільки основну інформацію – email та пароль (див. рис. 3.8). Поле email є унікальним, отже не можна зареєструвати пошту, яка вже є у системі.

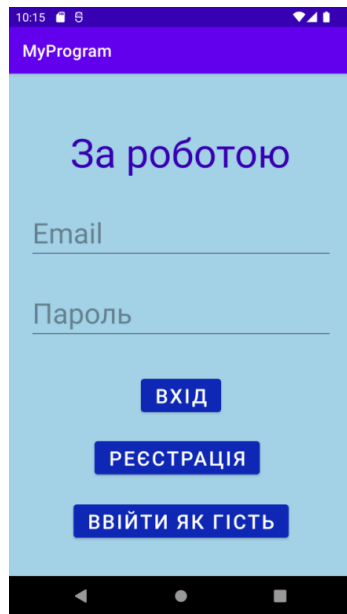


Рисунок 3.7 – Вхід в обліковий запис

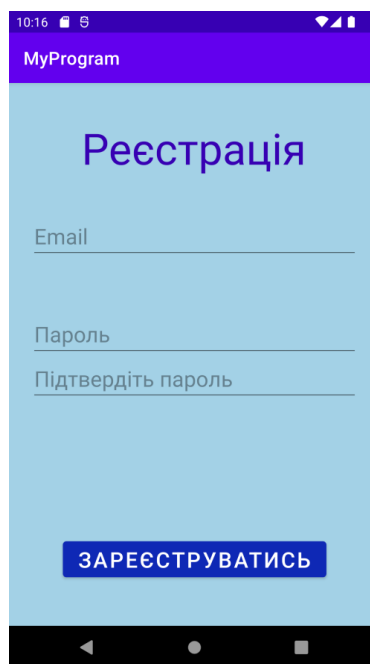


Рисунок 3.8 – Екранна форма вікна реєстрації

Для реєстрації нового користувача потрібно спочатку провести перевірку. Email повинен містити символ «@», при відсутності якого виводиться повідомлення на екран (див. рис. 3.9).

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.9 – Екранна форма вікна із некоректною реєстрацією

Також потрібно провести перевірку пароля, який повинен містити більше 6 символів та співпадати в двох стовпчиках: «Пароль» та «Підтвердіть пароль». При невиконанні вказаних вимог на екран виводиться відповідне повідомлення (див. рис. 3.10).

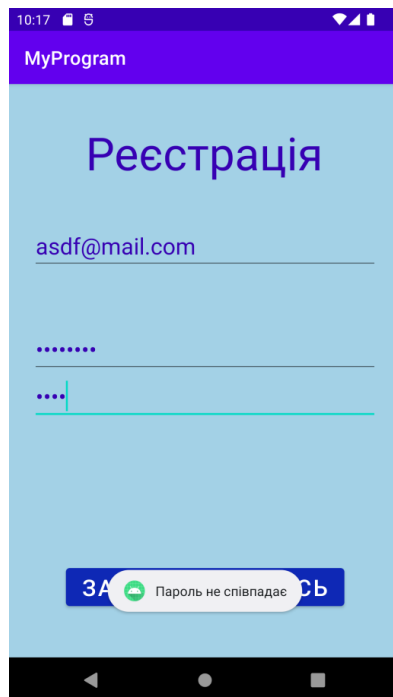


Рисунок 3.10 – Екранна форма вікна перевірки даних

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

При успішній реєстрації користувач переходить на активіті входу. Після коректного вводу логіна та пароля користувач переходить до активіті «Стрічка робіт» (див. рис. 3.11). В такому вікні будуть відображені всі оголошення користувачів. Для більш зручного користування додано фільтри та пошук по ключових словах.



Рисунок 3.11 – Екранна форма вікна оголошень

Для формування нового оголошення потрібно скористатися конструктором (див. рис. 3.12).

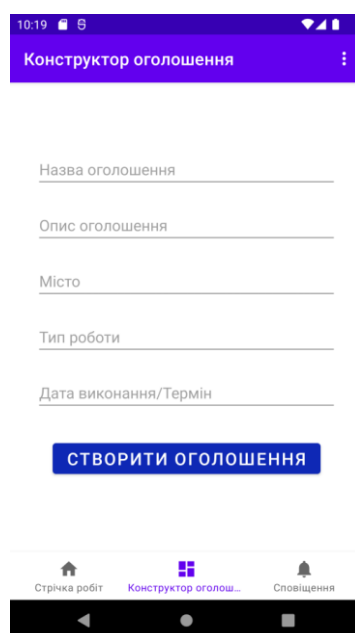


Рисунок 3.12 – Приклад конструктора формування оголошення

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Отриманні повідомлення, що стосуються оголошень або системних, заносяться в активіті «Сповіщення», як показано на рисунку 3.13.

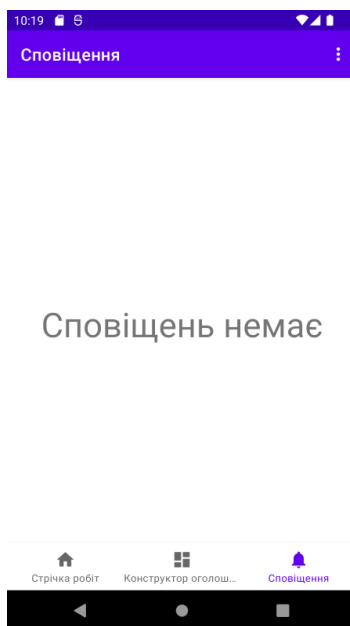


Рисунок 3.13 – Екранна форма вікна сповіщень

В додатку знаходиться випадаюче меню в правому верхньому куті (див. рис. 3.14), яке виконує задачу редагування особистої інформації користувача (див. рис. 3.15) та налаштування профілю (див. рис. 3.16). За допомогою такого меню користувач має можливість вийти зі свого облікового запису.

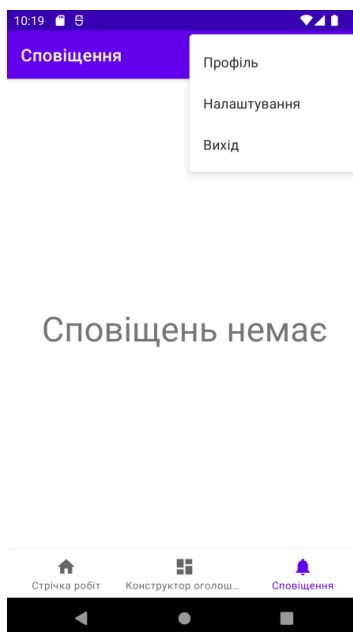


Рисунок 3.14 – Приклад випадаючого меню

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

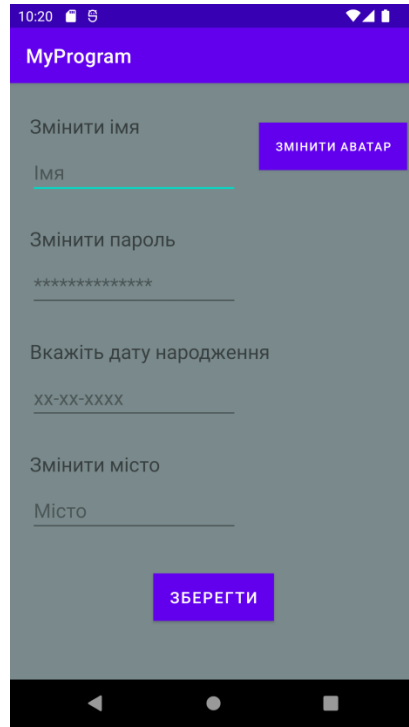


Рисунок 3.15 – Приклад профілю користувача

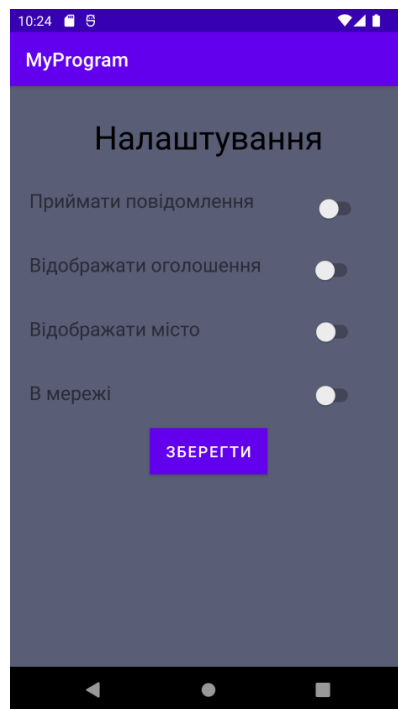


Рисунок 3.16 – Екранна форма вікна налаштування профілю

Таким чином, як видно із представлених рисунків, додаток достатньо простий в користуванні, зручний та зрозумілий для пересічного користувача. Назва додатку «За роботою» впливає із його основної функції.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3. Тестування мобільного додатку

У процесі розробки програми проводилося поетапне тестування з метою виявлення програмних помилок. Для цього були створені емулятори смартфона і планшета з різними діагоналями екрана для різних версій Android. Тестований програмний продукт послідовно запускався на цих емуляторах, його поведінка аналізувалася і при необхідності за результатами аналізу вносилися зміни у код [15].

У результаті експериментальних досліджень суттєвих недоліків не виявлено. Функціонал додатку відповідає поставленим вимогам та може бути застосований для пошуку віддаленої роботи. Відлагоджено систему комунікації між замовником та виконавцем.

Як перспектива подальших досліджень, шляхами розвитку мобільного додатку пошуку дистанційної роботи можуть бути:

- інтеграція додатків зі сторонніми базами даних;
- реалізація інтерфейсу на різних мовах;
- використання додатку з підтримкою віддаленого сервера;
- додавання різноманітних фільтрів;
- можливість виконувати віддалену роботу через сервіс додатку;
- можливість продивлятися геодані оголошення.

Зважаючи на вищезазначене, обране середовище для проєктування мобільних додатків Android Studio та мова програмування Kotlin уможливили досягти поставленої задачі. Розроблений мобільний додаток буде корисним для тих, хто працює віддалено на інших платформах, а також для новачків, які прагнуть заробляти, не виходячи з дому.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У роботі розв'язано задачу проектування мобільного додатку пошуку віддаленої роботи й при цьому отримано практичні результати.

1. Проведено аналіз існуючих мобільних додатків, виокремлено їх функціональні характеристики, наведено класифікацію, їх переваги та недоліки.

2. Проаналізовано сучасні ОС для розробки мобільних додатків, зазначено їх особливості, функціональні можливості, приклади застосування та порівняння на світовому ринку.

3. Охарактеризовано сучасні фріланс-біржі для віддаленої роботи й зазначено, що в основному платформи призначені для відкриття у браузерях й не супроводжуються програмними додатками для мобільних телефонів чи інших пристроїв.

4. Проведено аналіз програмних засобів розробки мобільних додатків, виокремлено їх переваги, недоліки та особливості застосування. Обґрунтовано доцільність використання середовища Android Studio та мови Kotlin для проектування мобільного додатку пошуку віддаленої роботи.

5. У результаті детального вивчення предметної області проведено аналіз вимог до програмної системи, сформульовано технічне завдання та наведено алгоритм проектування мобільного додатку.

6. Розроблено прототип додатку пошуку віддаленої роботи, описано основні структури та функції. На цій основі створено архітектуру мобільного додатку, описано інтерфейс користувача та основні задачі. Результатами тестування підтверджено його працездатність та відповідність вимогам, що свідчить про перспективність застосування таких продуктів для розв'язування типових задач.

7. Проведено розрахунок показників ефективності проектування мобільного додатку пошуку віддаленої роботи, результатами підтверджено доцільність розробки.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Етапи розробки мобільного додатку: веб-сайт. URL: <https://brander.ua/blog/etapy-rozrobky-mobilnoho-dodatku> (дата звернення 19.04.2024.).

2. Етапи створення мобільних додатків: веб-сайт. URL: <https://webcase.com.ua/uk/blog/etapi-stvorennja-mobilnogo-dodatku/> (дата звернення 23.04.2024.).

3. Ічанська Н. В., Улько С. І. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки // Системи управління, навігації та зв'язку. 2020. Вип. 1 (59). С. 74-78.

4. Заякін Д.К. Мобільний додаток для організації зустрічей: дипломна робота. Київ: НАУ, 2021. 91с.

5. Заяц К.В. Мобільний додаток для формування стартап – команди: дипломна робота. Київ: КПІ ім. І. Сікорського, 2021. 84 с.

6. Кабанчик: веб-сайт. URL: <https://kabanchik.ua/ua/freelance> (дата звернення 25.04.2024.).

7. Маліш Б.Б. Проєктування мобільного додатку “Карта укриттів”: дипломна робота. Львів: ЛНУ ім. І. Франка, 2023. 34 с.

8. Мельник К.О. Мобільний додаток Інтернет-магазину: дипломна робота. Київ: НАУ, 2021. 93 с.

9. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп’ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько, О.Й. Піцун / Під ред. Л.О.Дубчак. Тернопіль: ЗУНУ, 2024. 54 с.

10. Посібник із дизайну мобільних додатків під iOS та Android. Основні відмінності платформ: веб-сайт. URL: <https://ux.pub/editorial/posibnik-iz-dizainu-mobilnikh-dodatkiv-pid-ios-ta-android-osnovni-vidminnosti-platform-1a5d> (дата звернення 04.05. 2024 р.).

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

11. Проектування мобільних застосунків: ключові етапи та підводні камені: веб-сайт. URL: <https://wezom.com.ua/ua/blog/proektirovanie-mobilnogo-prilozheniya> (дата звернення 11.05.2024 р.).

12. Розробка мобільних додатків від А до Я: повний гайд: веб-сайт. URL: <https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatktiv-vid-a-do-ja-povnij-gajd/> (дата звернення 11.05.2024 р.).

13. Тестування мобільних додатків – що це таке, типи, процеси, підходи, інструменти та інше!: веб-сайт. URL: <https://www.zaptest.com/uk/тестування-мобільних-додатків-що-це-т> (дата звернення 11.04. 2024 р.).

14. Типи мобільних додатків: веб-сайт. URL: <https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types> (дата звернення 07.05.2024 р.).

15. Топ 7 українських та міжнародних фріланс-бірж: огляд: веб-сайт. URL: <https://kosht.media/top-7-ukrainskykh-ta-mizhnarodnykh-frilans-birzh-ohliad/> (дата звернення 14.05.2024 р.).

16. Фріланс біржа №1 в Україні: веб-сайт. URL: <https://freelance.ua> (дата звернення 07.05.2024 р.).

17. Шматко О. В., Поляков А.О., Федорченко В.М. Аналіз методів і технологій розробки мобільних додатків для платформи Android : навч. посіб. Харків : НТУ «ХПІ», 2018. 284 с.

18. Android Studio: веб-сайт. URL: <https://developer.android.com/studio> (дата звернення 17.05.2024 р.).

19. 8 Best Android Frameworks for App Development in 2023: веб-сайт. URL: <https://www.intellectsoft.net/blog/best-android-frameworks/> (дата звернення 18.05.2024 р.).

20. Cross-platform vs. Native Mobile App Development: Choosing the Right Dev Tools for Your App Project: веб-сайт. URL: <https://medium.com/all-technology-feeds/cross-platform-vs-native-mobile-app-development-choosing-the-right-dev-tools-for-your-app-project-47d0abafee81> (дата звернення 14.05.2024 р.).

21. Cross-platform vs native app development?: веб-сайт. URL: <https://www.brightec.co.uk/ideas/cross-platform-vs-native-app-development> (дата

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

звернення 11.05.2024 р.).

22. Freelancehunt: веб-сайт. URL: <https://freelancehunt.com/ua> (дата звернення 10.05.2024 р.).

23. Guide to app architecture: веб-сайт. URL: <https://developer.android.com/topic/architecture> (дата звернення 13.05.2024 р.).

24. Kotlin: веб-сайт. URL: <https://kotlinlang.org/docs/faq.html#is-kotlin-an-object-oriented-language-or-afunctional-one> (дата звернення 18.05.2024 р.).

25. Kotlin Programming Language: веб-сайт. URL: <https://kotlinlang.org> (дата звернення 16.05.2024 р.).

26. React Native: веб-сайт. URL: https://en.wikipedia.org/wiki/React_Native (дата звернення 06.05.2024 р.).

27. React Native. Build native mobile apps using JavaScript and React: веб-сайт. URL: <https://facebook.github.io/react-native/> (дата звернення 08.05.2024 р.).

28. Savka N.Ya., Krynytskyi O.V., Zhigalo Ya., Pohorilets A.V Modeling web projects in the style of minimalism // Proceeding of International Scientific Conference “Organization of scientific research in modern conditions” March, 2024. USA: Washington, 2024. С. 37-39.

					КР.КІ.9499965.00.00.000.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		