

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр

спеціальність 121 „Інженерія програмного забезпечення”

(шифр і назва спеціальності)

на тему „Мобільний додаток для таксі сервісу”

Виконав: студент групи ІІЗ-20д

_____ (підпис)

Д.О. Ващук

_____ (ініціали і прізвище)

Керівник

_____ (підпис)

Д.В. Ратов

_____ (ініціали і прізвище)

Завідувач кафедри

_____ (підпис)

О.І. Захожай

_____ (ініціали і прізвище)

Рецензент В.О. Лифар

Київ – 2024

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки
Кафедра інформаційних технологій та програмування

Освітній ступінь бакалавр
спеціальність 121 „Інженерія програмного забезпечення”
(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ

Завідувач кафедри

“___” _____ Захожай О.І.
2024 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

Ващук Данило Олександрович

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка мобільного додатку для таксі сервісу

керівник роботи Ратов Денис Валентинович , д.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом університету від “06” травня 2024 року
№171/15.15-С

2. Строк подання студентом роботи 14.06.2024р.

3. Вихідні дані до роботи: Об'єктом даної роботи є процес створення мобільного додатку для таксі

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Вступ. Аналітичний огляд, з висвітленням наступних питань: що собою являють мобільні додатки для таксі та їх основні етапи розробки, розгляд додатків-аналогів. Основна частина, в якій розібрали: вибір додатків та інструментів для реалізації, проектування та реалізація проекту. Висновки. Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників) Розділ 1 діаграми статистику ринку. Розділ 2 зображення

додатків-аналогів. Розділ 3 зображення розробленого додатку для таксі сервісу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 30.03.2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Одержання завдання на виконання роботи	30.03.24	виконано
2	Укладання і погодження з керівником плану і етапів виконання роботи	06.04.24	виконано
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	20.04.24	виконано
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	22.04.24	виконано
5	Укладання та тестування програмного продукту	27.04.24	виконано
6	Укладання, оформлення та погодження пояснювальної записки з керівником	30.05.24	виконано
7	Здача готової пояснювальної записки на кафедру	14.06.24	виконано
8	Укладання доповіді і презентації	14.06.24	виконано

Студент

_____ Д.О. Ващук
підпис (ініціали і прізвище)

Керівник роботи

_____ Д.В. Ратов
підпис (ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. ПЗ-20д Ващук Д.О.

Науковий керівник:

Професор, д.т.н.

Ратов Д.В.

Оцінка наукового керівника: _____

Рецензент

ПБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

Професор кафедри ІТП

д.т.н.

підпис

Захожай О.І.

ЗМІСТ

ВСТУП.....	5
1. АНАЛІТИЧНИЙ ОГЛЯД.....	8
1.1. Основи розробки мобільних додатків.....	8
1.2. Вибір операційної системи.....	9
1.3. Загальні особливості розробки додатку для таксі.....	14
1.4. Огляд аналогічних додатків для таксі.....	16
2. ТЕХНІЧНИЙ АНАЛІЗ.....	22
2.1. Методологія розробки.....	22
2.2. Особливості розробки мобільного додатку для таксі.....	23
2.3. Функціональні особливості додатку для таксі.....	25
2.4. Збір вимог.....	26
2.5. Вибір інструментів.....	32
3. СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ.....	42
3.1. Графічний інтерфейс.....	42
3.2. Розробка програмної частини.....	48
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55

ВСТУП

Актуальність. В наш час використання нових інформаційно-комунікаційних технологій є необоротньою тенденцією світового розвитку. Сучасна епоха носить назву інформаційного віку. Цей термін обґрунтований тим, що інформація є ключовим, цінним та невід'ємним ресурсом. Володіння інформацією та вміння передавати необхідну інформацію клієнтам своєчасно, швидко та чітко вважаються основою успіху в сучасному бізнесі. Наразі Інтернет може масово та оперативно передавати текст, звук, зображення та відеоролики - практично будь-яку інформацію. З точки зору користувача, Інтернет є великим інформаційним ресурсом, де можна знайти все. Аудиторія мережі Інтернет постійно зростає, а сам Інтернет є найшвидше розвиваючимся засобом передачі інформації в історії людства. Користувачі Інтернету привертають увагу рекламодавців, оскільки серед них багато потенційних клієнтів.

Сфера послуг є однією з найбільш перспективних галузей економіки та економічної діяльності. Вона охоплює широкий спектр економічної діяльності та не є єдиним комплексом за своїм функціональним призначенням. У умовах комерціалізації економіки та враховуючи великі відстані та гостру нестачу часу, особливого значення набувають такі форми реалізації товарів і послуг, як використання послуг та реклама через Інтернет. Існує велика кількість способів комерційного впливу в мережі Інтернет. Можливе рекламування послуг, продаж товарів та їх представлення споживачеві. У сучасному бізнесі багато залежить від самопрезентації компанії, її позиціонування на ринку наданих послуг та здатності залучати нових клієнтів і ринки збуту. Наявність власного сайту в Інтернеті є одним із інструментів, що підтримують як іміджеві, так і маркетингові цілі. З точки зору бізнесу, Інтернет є сучасною рекламною платформою, яка забезпечує потік клієнтів, збільшуючи прибуток. Користування сучасними технологіями перетворює компанію в сучасний бізнес і є ключовим джерелом інформації

для потенційних клієнтів та людей, чиє висловлене мніння має велике значення.

У сучасному житті проблема переміщення має актуальність. За станом на 2021 рік в країні налічувалося близько десяти мільйонів автомобілів коли чисельність населення налічувала приблизно 41 мільйон громадян тож у більшій частині населення є вибір між користуванням громадським транспортом чи таксі.

Таксі порівняно з громадським транспортом надає більш комфортабельні умови у будь-який час що в день що вночі, за допомогою таксі можна зекономити час користуючись більш оптимальним маршрутом та пасажери можуть обирати та корегувати маршрут самотушки. Ці та інші причини роблять таксі галузею бізнесу, що буде лише розвиватись.

При виборі теми курсової роботи було важливо вибрати ту тему, що є актуальною зараз та буде актуальною в майбутньому. Наразі важко уявити життя без смартфонів з доступом до інтернету. За допомогою смартфонів можна спілкуватись, проводити вільний час чи працювати і тому знаходиться під рукою майже весь час. Використання цього в сфері таксі може сприяти суттєвим покращенням в усіх аспектах бізнесу. Наприклад, створення власного мобільного додатка для замовлення таксі дозволяє не лише зробити процес замовлення більш зручним для клієнтів, а й надає можливість компанії збирати та аналізувати дані щодо попиту, покращуючи стратегії обслуговування. Додаток допомагає знизити витрати на штат диспетчерів, телефонний зв'язок, витрати на співробітників для обліку та ведення статистики тощо, дозволяє замінити частину робітників на комп'ютерний алгоритм. За допомогою цього додатк таксі-підприємство може зберегти час і гроші, автоматизуючи багато рутинних операцій. Що допоможе зосередитися на покращенні якості обслуговування клієнтів та ефективності бізнесу.

Об'єкт дослідження: процес розробки мобільного додатку для таксі

Предмет дослідження: мобільний додаток для таксі

Мета дослідження: Метою роботи є розробка та додатку для таксі-сервісу. Цей проект передбачає інтеграцію сучасних технологій та зручних функцій для замовлення таксі, а також забезпечення інформативного та легкого у використанні інтерфейсу для користувачів.

Задачі дослідження:

1. Ознайомитись з розробкою мобільного додатку.
2. Обрати найбільш підходящий стек технологій та мов для створення додатку.
3. Розробити додаток для сервісу таксі.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Основи розробки мобільних додатків

Мобільний додаток - це програмне забезпечення, призначене для використання на мобільних пристроях, таких як смартфони, планшети, розумні годинник та інші мобільні пристрої. Ці застосунки можуть виконувати різноманітні функції, від ігор та розважальних програм до корисних інструментів для роботи, навчання або спілкування. Вони зазвичай доступні для завантаження з цифрових магазинів додатків, таких як App Store для iOS або Google Play Store для Android. Мобільні застосунки можуть бути розроблені для різних операційних систем, таких як iOS від Apple або Android від Google. Розробка мобільного додатка є процес створення програмного забезпечення, яке працює на мобільних пристроях, таких як смартфони та планшети. Цей процес включає кілька етапів: концептуалізацію ідеї, дизайн інтерфейсу користувача (UI), розробку коду, тестування та розгортання програми в магазинах додатків, таких як App Store і Google Play.

Мобільні застосунки роблять на мовах програмування високого рівня і після компілюються у машиний код операційної системи задля отримання максимальної продуктивності. Також при розробці додатку треба урахувувати особливості мобільних пристроїв: вони мають обмежені обсяги оперативної пам'яті, менші потужності процесора, батареї, можуть мати свої додаткові пристрої, як камери, гіроскопи, акселерометри, датчики освітлення задля вимурювання руху, орієнтації та інших параметрів.

Однією з основних відмінностей розробки мобільного додатка від розробки додатка для ПК є обмеженість ресурсів на мобільних пристроях. Мобільні пристрої мають менше обчислювальної потужності, оперативної пам'яті та обмежений термін роботи від батареї порівняно з настільними ПК. Це вимагає від розробників ретельнішої оптимізації коду та ефективного управління ресурсами. Також необхідно враховувати різні дозволи екранів та орієнтації, що ускладнює дизайн інтерфейсу користувача.

Крім того, процес публікації та розповсюдження мобільних додатків відрізняється від настільних програм. Для мобільних додатків необхідно пройти процес перевірки та схвалення в магазинах додатків, що може зайняти час і вимагати дотримання суворих посібників з безпеки та якості. Оновлення програм також проходять через цей процес, що може сповільнити випуск нових функцій та виправлень помилок. Важливо, що очікування користувача від мобільних додатків також відрізняються. Користувачі мобільних пристроїв очікують на миттєвий відгук, простоту використання та інтуїтивно зрозумілий інтерфейс. Це накладає додаткові вимоги на дизайн і функціональність додатків, змушуючи розробників приділяти більше уваги деталям та досвіду користувача.

Таким чином, розробка мобільних додатків є унікальним і складним завданням, яке вимагає врахування безлічі факторів, від апаратних обмежень до користувацьких очікувань і вимог платформ. Цей процес відрізняється від розробки настільних програм, але відкриває перед розробниками нові можливості для інновацій та створення програм, які можуть змінити повсякденне життя користувачів.

1.2 Вибір операційної системи

При порівнянні платформ iOS та Android для розробки програми для таксі важливо розглянути кілька ключових аспектів: поширення та аудиторія, розробка та підтримка, функціональні можливості, час та бюджет розробки. Ось докладне порівняння цих платформ щодо кожного з цих аспектів:

Поширення та аудиторія.

На загальносвітовому ринку Android значно переважає над iOS за поширеністю. Android, розроблений Google, домінує з погляду ринкової частки завдяки своїй відкритій природі та доступності на широкому спектрі пристроїв від різних виробників. Ця відкритість призвела до того, що безліч

компаній, таких як Samsung, Huawei, Xiaomi та інші, випускають пристрої на базі Android, що зробило його більш доступним і популярним у різних сегментах ринку, особливо в країнах, що розвиваються. В результаті Android займає 71.31% за квітень 2024 глобальної ринкової частки мобільних операційних систем. З іншого боку, iOS, розроблена компанією Apple, хоч і має меншу частку ринку приблизно 27.95%, зберігає значну присутність, особливо в розвинених країнах. Пристрої на базі iOS, такі як iPhone та iPad, відомі своєю якістю, стабільністю та інноваційними функціями. Apple контролює як апаратне, так і програмне забезпечення, що забезпечує тісну інтеграцію та високу продуктивність пристроїв. Однак, через більш високу ціну на пристрої, iOS менш доступна в країнах, що розвиваються, що обмежує її поширення в порівнянні з Android.

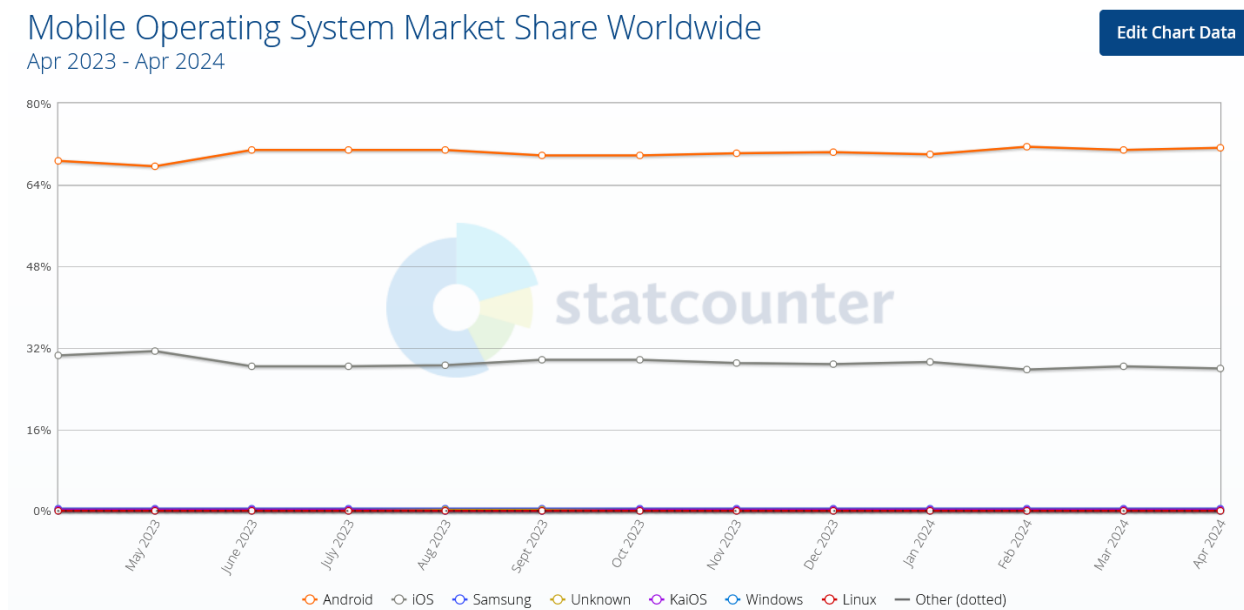


Рисунок 1 - Частина світового ринку операційних систем для мобільних пристроїв

На ринку США iOS та Android представлені як дві основні операційні системи для мобільних пристроїв, їхня частка залежить від різних факторів, включаючи переваги споживачів, доступність пристроїв та екосистеми додатків.

iOS, розроблений Apple, займає значну частку ринку у США, становлячи 56.53% за даними за квітень. Ця платформа відома своєю унікальною інтеграцією між пристроями Apple, високою якістю програмного забезпечення та акцентом на безпеці та конфіденційності даних. Користувачі iOS часто віддають перевагу цій операційній системі через її зручний та інтуїтивно зрозумілий інтерфейс користувача, високоякісні програми та широкі можливості інтеграції з іншими пристроями Apple, такими як Mac, iPad та Apple Watch.

З іншого боку, Android, розроблений Google, займає частку ринку США у розмірі 43.12%. Android відомий своєю відкритою та гнучкою платформою, яка підтримує широкий спектр пристроїв від різних виробників, таких як Samsung, Google, LG, та інші. Ця платформа пропонує великий вибір пристроїв із різними функціональними можливостями та ціновими категоріями, що робить її доступною для широкого кола споживачів. Android також відомий своєю гнучкістю та можливістю налаштування, що приваблює користувачів, які віддають перевагу різноманітності у виборі пристроїв та персоналізації інтерфейсу.

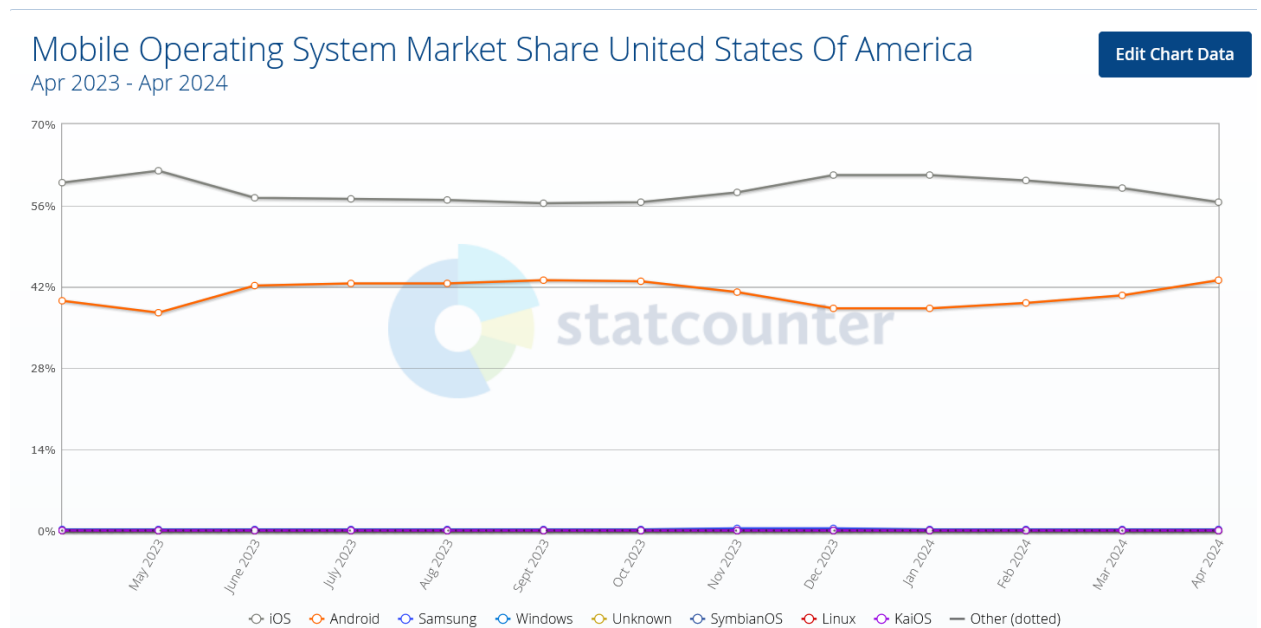


Рисунок 2 - Частина ринку операційних систем для мобільних пристроїв США

Згідно за статистикою на території України найпоширенішою мобільною операційною системою на ринку є Android, що займає 70.46% ринку за станом на квітень 2024

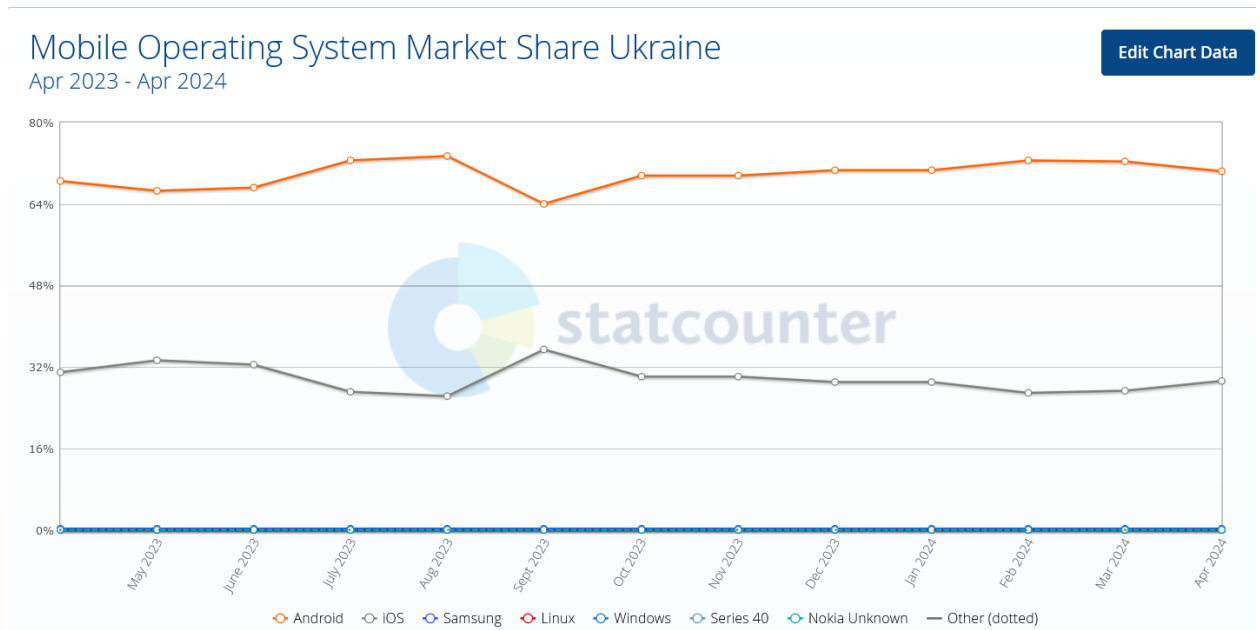


Рисунок 1 - Частина українського ринку операційних систем для мобільних пристроїв

Android: Більш широке глобальне поширення, особливо в країнах, що розвиваються. Великий вибір пристроїв різних цінових категорій, що робить Android доступним для більшої кількості користувачів.

iOS: Найбільш популярна у розвинених країнах та серед користувачів з вищими доходами. Користувачі iOS, як правило, витрачають більше на програми та внутрішньопрограмні покупки.

Розробка та підтримка.

Android: Розробка проводиться на Kotlin або Java з використанням середовища розробки Android Studio. Більш широкий вибір пристроїв та версій операційної системи потребує додаткового тестування та оптимізації для різних конфігурацій.

iOS: Розробка ведеться на Swift чи Objective-C із використанням Xcode. Екосистема пристроїв більш уніфікована, що спрощує тестування та оптимізацію.

Функціональні можливості. Android: Гнучкіша інтеграція з системними функціями та сторонніми програмами завдяки відкритому вихідному коду. Більше можливостей для налаштування інтерфейсу та функціоналу.

iOS: Суворіша політика безпеки та контролю якості. Менша фрагментація пристроїв і версій операційної системи забезпечує стабільну роботу та однаковість користувальницького досвіду.

Також при розробки мобільного додатку є три варіанти розробки додатку:

- Спеціалізація під одну ОС:

Спрямування розробки лише на одну конкретну операційну систему. Наприклад, ви можна обрати Android і розробити додаток лише для цієї платформи. При такому підході важливо усвідомити, що це може обмежити кількість потенційних користувачів, оскільки власники інших ОС не зможуть взаємодіяти з додатком. Такий вибір є розумним у випадку тестування ринку або вибору операційної системи цільовою аудиторією.

- Кросплатформенний додаток:

Розробка кросплатформенного додатка, який може працювати на різних операційних системах. Це може бути вигідним рішенням, оскільки дозволяє вам охопити більшу аудиторію та зменшити затрати на розробку. Кросплатформенні фреймворки дозволяють створювати додатки, які працюють на різних платформах і виглядають максимально однаково. Однак важливо враховувати, що через відсутність адаптації під певну операційну систему можуть виникнути труднощі з використанням деяких функцій або несправності.

Окремі версії під кожен ОС:

Третій варіант - розробка окремих версій додатка для кожної операційної системи. Це найбільш часо-ресурсозатратний варіант, оскільки вимагає розробки та підтримки кількох різних кодових баз. Проте він може бути доцільним, якщо треба надати максимально оптимізований досвід користувачам кожної операційної системи. Розробка окремих версій дозволяє максимально використовувати можливості конкретної платформи, що може позитивно позначитися на роботі додатка та задоволенні користувачів..

Висновки:

Підводячи підсумки треба зауважити що розробка мобільних додатків повинна враховувати розповсюдженість ОС. Згідно за статистикою найпоширенішею системою у світі цілком та зокрема в Україні є Android, якщо б розробка велась на ринок США то перевага у розробці була б віддана саме iOS але розробка мобільного додатку для таксі ведеться, в першу чергу, з оглядки на український ринок. У підсумку буде розумніше розробити спеціалізований під одну ОС додаток на Android, через більшу розповсюдженість та гнучкість системи.

1.3 Загальні особливості розробки додатку для таксі

Розробка програми для таксі включає безліч особливостей і ключових аспектів, які необхідно враховувати для успішного створення функціонального і зручного сервісу. Перше, що слід враховувати, - це інтерфейс користувача і досвід, оскільки додатки для таксі повинні бути інтуїтивно зрозумілими і легкими у використанні як для пасажирів, так і для водіїв. Це вимагає ретельного проектування інтерфейсів для обох типів користувачів, з урахуванням їх різних потреб та поведінки.

Функціональність є основою таких програм. Ключовими функціями для пасажирів включають можливість виклику таксі, відстеження розташування автомобіля в реальному часі, розрахунок вартості поїздки, вибір методу оплати та можливість залишати відгуки. Для водіїв важливі функції, такі як отримання замовлень, навігація до пункту призначення, управління статусом (доступний/зайнятий), а також доступ до інформації про заробіток та рейтинг. Інтеграція з картографічними сервісами, такими як Google Maps або Apple Maps, відіграє важливу роль, тому що дозволяє забезпечувати точне геолокаційне відстеження та навігацію.

Безпека та надійність системи також є важливими аспектами. Необхідно забезпечити захист користувачів, включаючи персональні та платіжні дані, а також інтегрувати функції, що підвищують безпеку поїздок, такі як можливість ділитися маршрутом з друзями або сім'єю та кнопка екстреної допомоги. Надійна серверна інфраструктура та регулярні оновлення системи допоможуть запобігти збоям та покращити продуктивність програми.

Ефективне управління платежами та інтеграція з різними платіжними системами є ще одним критично важливим аспектом. Додаток повинен підтримувати різні методи оплати, такі як кредитні картки, мобільні платежі та розрахунки готівкою. Система повинна бути надійною та забезпечувати безпеку транзакцій.

Розробка та підтримка серверної частини програми вимагають використання сучасних технологій та фреймворків, які забезпечать масштабованість та продуктивність системи. Важливо вибирати архітектурні рішення, які дозволять обробляти велику кількість запитів та забезпечать стабільну роботу сервісу за високих навантажень.

Маркетингова складова та механізм залучення користувачів також відіграють важливу роль. Додаток має пропонувати зручні механізми реєстрації та входу, а також системи заохочення, такі як знижки та бонуси за

запрошення друзів. Важливо також стежити за зворотним зв'язком користувачів та регулярно оновлювати програму, покращуючи її функціональність та усуваючи виявлені помилки.

Таким чином, розробка програми для таксі - це складний процес, що включає безліч аспектів: від проектування інтерфейсу користувача і забезпечення функціональності до інтеграції з платіжними системами і забезпечення безпеки даних. Успішний додаток має бути зручним та надійним як для пасажирів, так і для водіїв, забезпечуючи високий рівень обслуговування та безпеки.

1.4 Огляд аналогічних додатків для таксі

Було вирішено провести аналіз вже існуючих мобільних додатків для таксі Bolt(рис.1) Uklon(рис.2) і Uber(рис.3). Це дозволить оцінити вже готові, праючі рішення імплементовані у мобільний додаток. Кожен додаток буде оцінюватись за особливостями та функціями, інтерфейсом користувача а також будуть враховуватись плюси та мінуси.

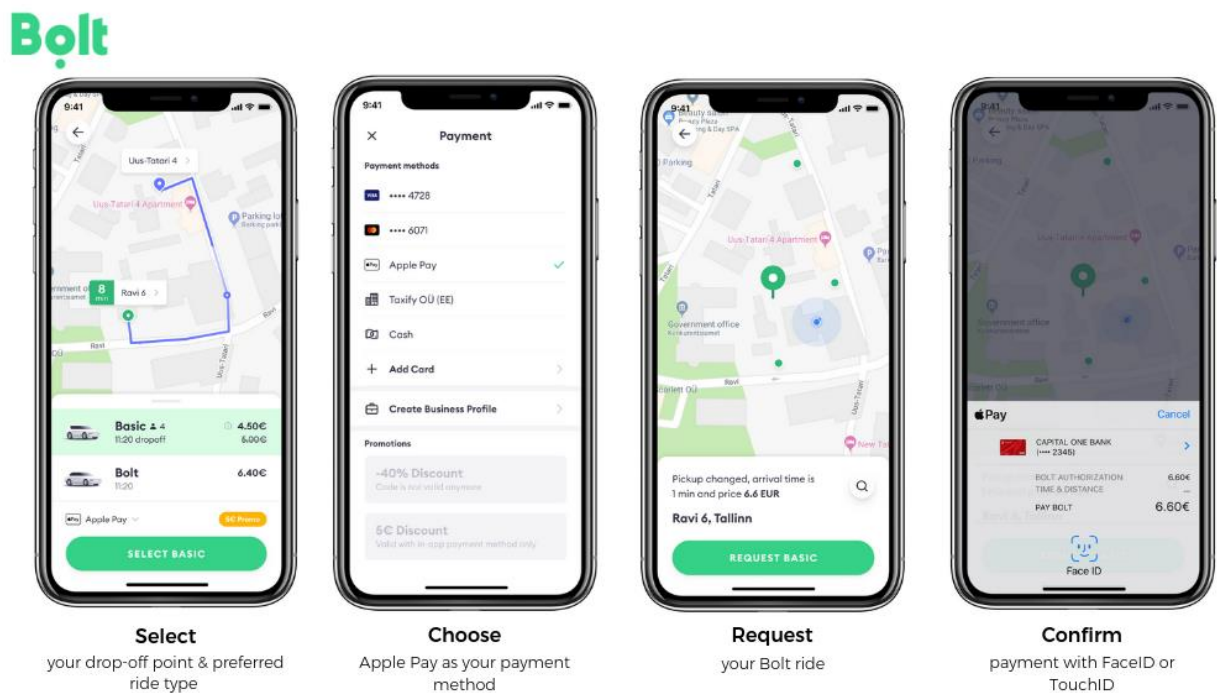


Рис.1 Bolt

Відмінності програми Volt від інших сервісів таксі та каршерингу можна виділити кілька ключових аспектів. По-перше, Volt відомий своїм акцентом на конкурентоспроможні ціни та доступні тарифи, що робить його привабливим вибором для користувачів, які цінують економічність. Друга важлива відмінність – різноманітність пропонованих транспортних засобів. Крім звичайних таксі Volt також пропонує послуги з оренди електросамокатів та автомобілів, що дає користувачам більше варіантів для вибору відповідного транспорту. Третя важлива відмінність полягає у фокусі на безпеці та якості обслуговування. Volt активно впроваджує системи перевірки водіїв та моніторингу якості обслуговування, що сприяє підвищенню рівня довіри з боку користувачів. Крім того, програма Volt часто пропонує різні знижки та акції для своїх клієнтів, що допомагає утримувати лояльність та залучати нових користувачів. Інтерфейс користувача: Volt пропонує зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко замовити таксі та вибрати бажаний тип транспорту. Додаток також має функції відстеження маршруту та оцінки водіїв.

Плюси: Конкурентні ціни, широкий вибір транспорту, зручний інтерфейс.

Мінуси: Залежність від доступності водіїв у певних регіонах.

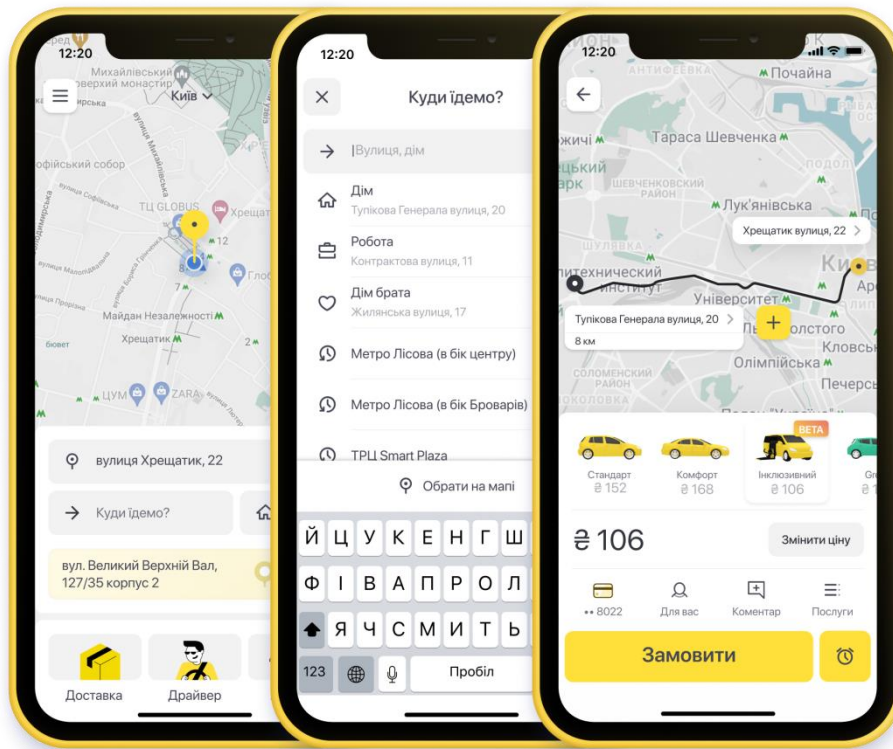


Рис.2 Uklon

Uklon, українська програма для виклику таксі, має низку унікальних особливостей, які відрізняють його від інших аналогічних сервісів, таких як Uber та Bolt. Однією з головних відмінностей Uklon є його сильна локальна прив'язка та адаптація до потреб українського ринку. Додаток активно співпрацює з місцевими таксомоторними компаніями, що дозволяє йому надавати конкурентні тарифи та широкий вибір автомобілів. Крім того, Uklon пропонує користувачам різноманітні методи оплати, включаючи готівку, що є важливим фактором на ринках, де банківські картки не дуже поширені. Важливою особливістю є гнучкість в управлінні поїздками, де пасажирів можуть самі встановлювати бажану вартість поїздки та домовлятися з водіями, що робить сервіс гнучкішим та привабливішим для користувачів. Додаток також виділяється просунутими функціями безпеки, включаючи можливість ділитися маршрутом поїздки з близькими та функцію екстреного зв'язку, що забезпечує додаткову впевненість пасажирів. Водії Uklon проходять ретельну перевірку, що сприяє підвищенню рівня безпеки та якості обслуговування. Технічно Uklon активно використовує сучасні хмарні

технології та регулярно оновлює додаток, пропонуючи нові функції та покращення. Програма лояльності та системи бонусів для постійних клієнтів стимулюють користувачів до повторних подорожей. У результаті Uklon успішно поєднує локальну адаптацію, безпеку, гнучкість та сучасні технології, що робить його популярним вибором серед користувачів в Україні.

Особливості та функції: Uklon - це український сервіс таксі, який пропонує широкий вибір послуг, включаючи стандартні таксі, мінівени та автомобілі класу "бізнес". Основними конкурентними перевагами Uklon є висока якість обслуговування та швидка доставка.

Інтерфейс користувача: Uklon має простий та зрозумілий інтерфейс, який дозволяє користувачам швидко та легко замовити таксі. Додаток також має функцію відстеження маршруту та оцінки водіїв.

Плюси: Широкий вибір транспорту, висока якість обслуговування.

Мінуси: Вища цінова політика порівняно з іншими додатками.

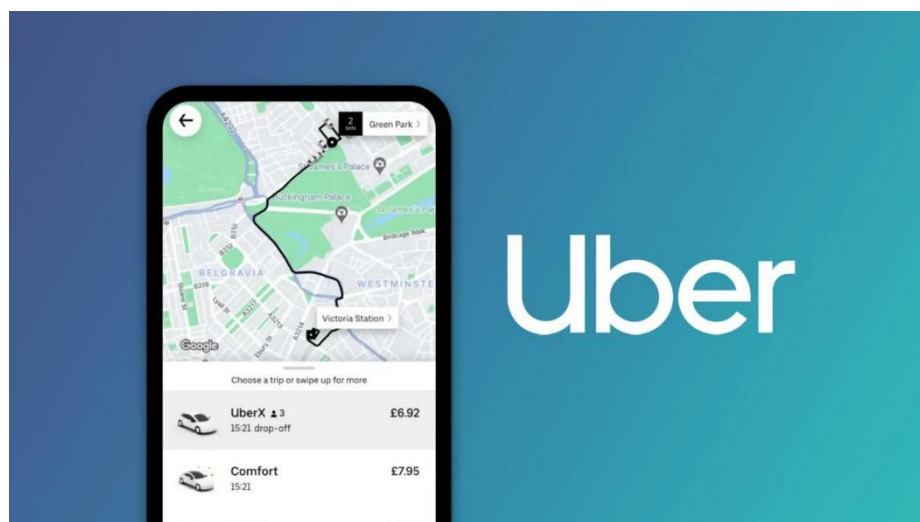


Рис.3 Uber

Uber відрізняється від інших програм для виклику таксі рядом ключових особливостей. Однією з головних відмінностей є глобальна

присутність та визнання бренду, що дозволяє Uber надавати свої послуги у сотнях міст та країн по всьому світу, забезпечуючи високий рівень стандартизації та надійності. Додаток Uber також виділяється інтуїтивно зрозумілим та зручним інтерфейсом, який дозволяє користувачам швидко та легко замовляти поїздки, відстежувати місцезнаходження автомобіля в реальному часі та отримувати інформацію про водія та транспортний засіб. Крім того, Uber пропонує різноманітність транспортних засобів, включаючи економ-класи, преміум-автомобілі, а також послуги з доставки їжі (Uber Eats) та спільні поїздки (Uber Pool). Важливою особливістю є і потужна система безпеки, що включає перевірки водіїв, систему рейтингів та відгуків, можливість поділитись маршрутом поїздки з друзями чи родиною, а також цілодобову підтримку користувачів. Uber активно впроваджує нові технології та інновації, такі як автономні автомобілі та сервіси на основі штучного інтелекту, що допомагає підтримувати високу конкурентоспроможність та адаптуватися до ринку, що швидко змінюється. Важливою відмінністю Uber є також гнучкість у методах оплати, підтримка багатьох способів оплати, включаючи кредитні картки, PayPal та мобільні платежі, що робить процес оплати зручним для користувачів по всьому світу. З точки зору водіїв, Uber пропонує порівняно вигідні умови співпраці, надаючи доступ до великої кількості замовлень та підтримуючи систему бонусів та заохочень. Всі ці особливості роблять Uber одним із лідерів ринку, пропонуючи користувачам високоякісний сервіс та широкі можливості вибору.

Особливості та функції: Uber пропонує аналогічний набір послуг, як і в інших країнах. Портфель послуг Uber включає стандартні таксі, автомобілі класу "люкс" та комфортні автомобілі. Відомий своєю глобальною присутністю та високим стандартом обслуговування.

Інтерфейс користувача: Uber пропонує знайомий та зручний інтерфейс, який вже відомий багатьом користувачам у світі. Додаток має функції відстеження маршруту, оцінки водіїв та зручні методи оплати.

Плюси: Глобальний бренд, зручний інтерфейс, висока якість обслуговування.

Мінуси: Вищі ціни порівняно з місцевими сервісами.

2. ТЕХНІЧНИЙ АНАЛІЗ

2.1 Методолгія розробки

Методологія розробки - це сукупність правил, процедур, практик та підходів, які використовуються для керування процесом розробки програмного забезпечення з метою підвищення ефективності, якості та прогнозованості розробки. Проект розробки мобільного додатку для таксі буде розроблятися за моделлю Agile.

Agile (гнучкі методології розробки) — це набір принципів та підходів до розробки програмного забезпечення, що сприяє гнучкості, адаптивності та прискореній розробці продукту. Основні принципи Agile включають ітеративність, інкрементальність, постійну взаємодію із замовником, команди, що самоорганізуються, пріоритизацію завдань і швидку реакцію на зміни в процесі розробки. Agile дозволяє швидше реагувати на зміни вимог, покращувати якість продукту та збільшувати рівень задоволеності замовника.

На рисунку 4 наведено етапи моделі Agile:

- Планування ітерації (Sprint Planning) Визначення цілей ітерації та обсягу роботи. Розподіл завдань між командою розробників. Призначення часових обмежень та визначення критеріїв завершення робіт.
- Розробка (Development): Розробка функціональності та виконання завдань згідно з планом ітерації. Регулярні зустрічі команди для обговорення прогресу та вирішення можливих проблем.
- Тестування (Testing): Виконання тестів для перевірки розроблених функцій. Виявлення та усунення помилок. Забезпечення відповідності продукту критеріям якості.

- Огляд (Review) або Демонстрація (Demo): Представлення виконаних робіт власникам продукту чи представникам бізнесу. Оцінка результатів ітерації та збір відгуків.
- Ретроспектива (Retrospective): Аналіз результатів ітерації та процесу роботи команди. Виявлення сильних і слабких сторін процесу та шляхів для його поліпшення.

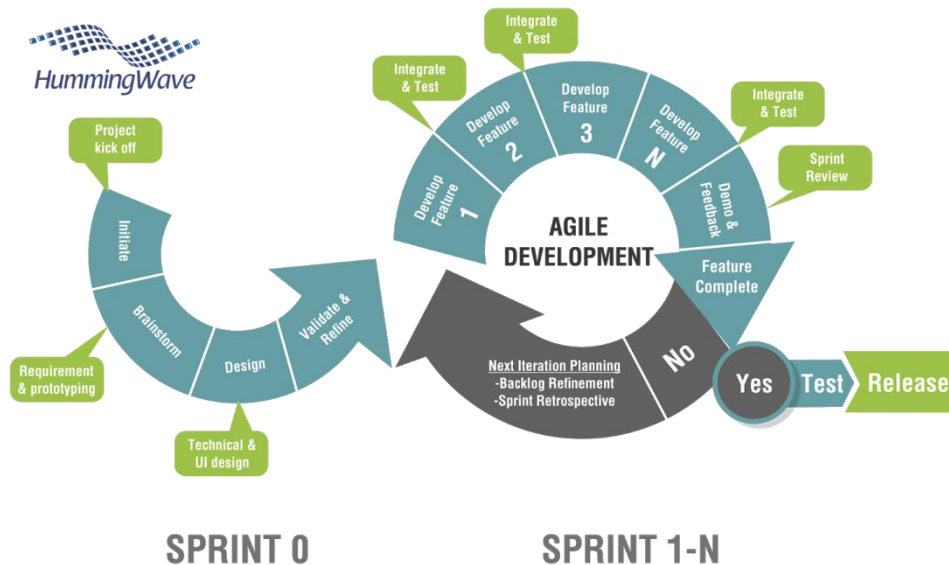


Рис.4 Модель Agile

Ці етапи повторюються протягом всього процесу розробки, що дозволяє команді швидко адаптуватися до змін і непередбачених обставин, а також забезпечує постійне вдосконалення якості продукту. На останній фазі визначається чи потрібно починати доробку чи реліз. У цій методології розробки після кожної ітерації замовник може спостерігати результат і розуміти, чи задовольняє він його чи ні. Це одна з переваг гнучкої моделі. До її недоліків відносять те, що через відсутність конкретних формулювань результатів складно оцінити трудомісткість і вартість, потрібні на розробку.

2.2 Особливості розробки додатку для таксі

Розробка мобільного додатку для таксі потребує уваги до специфічних потреб цієї галузі. Основні аспекти, які потрібно враховувати, включають геолокацію та навігацію, систему замовлення, оплату через додаток,

підтримку рейтингів та відгуків, систему сповіщень, контактну підтримку, забезпечення безпеки, адаптацію до місцевих ринків, аналітику та відстеження даних.

Геолокаційний сервіс необхідний для точного визначення місцезнаходження користувачів і доступних автомобілів, а також для інтеграції з системами навігації для оптимального маршруту. Простий та ефективний процес замовлення таксі, включаючи вибір типу автомобіля та адреси, є важливим для зручності користувачів. Можливість безготівкової оплати поїздки через додаток дозволяє забезпечити зручність та безпеку. Підтримка рейтингів та відгуків створює можливість користувачам оцінювати якість обслуговування та сприяє покращенню сервісу. Система сповіщень про статус замовлення та інші події допомагає користувачам бути в курсі процесу поїздки. Можливість швидкого зв'язку зі службою підтримки є важливою для вирішення проблем. Функції безпеки гарантують надійність та безпеку поїздки.

Варто розуміти, що коли ми говоримо про додаток таксі, то йдеться не лише про «клієнтську частину» але додаток для таксі складається з 3 частин:

1. Мобільний додаток для клієнтстві

Це частина структури, яка відповідає за взаємодію пасажирів з компанією. Важливо зосередитися на дизайні та інтерфейсі системи і впровадити необхідну кількість функцій для зручного використання додатка. Пасажир повинен встановити додаток, після чого може користуватися послугами компанії.

2. Мобільний додаток для водіїв

Додаток для таксистів також повинен бути зрозумілим настільки, щоб людина могла користуватися ним, не відволікаючись від дороги. Також

необхідно впровадити певні функції: Додаток для водіїв це робочий інструмент, який дозволяє відстежувати свій рейтинг, отримувати замовлення та вести статистику по виконаній роботі. Доступ до додатку водій отримує від адміністратора системи.

3. Адміністративна частина

Це веб-або мобільний інтерфейс, який використовується адміністраторами таксі-компанії для управління системою в цілому. Це включає у себе прийом замовлень, розподіл їх між водіями, встановлення цін та тарифів, контроль якості та зв'язок з клієнтами і водіями.

Ці три типи додатків працюють разом, утворюючи комплексну систему таксі-сервісу, яка задовольняє потреби як пасажирів, так і водіїв, а також дозволяє ефективно керувати бізнесом компанії.

У контексті випускної кваліфікаційної роботи розробка мобільного додатку для таксі обмежується лише додатком для клієнтів.

2.3 Функціональні особливості додатку для таксі

Завдяки огляду аналогічних додатків було виявлені основні функції кожного виду додатків для таксі. Треба виділити обов'язкові функції додатку для таксі для 3 частин додатку:

Клієнтська

- Онлайн-карта, яка дозволяє відстежувати рух транспорту в реальному часі;
- Можливість пошуку найближчого автомобіля;
- Вибір автомобіля за параметрами;
- Функція додаткових вимог - кількість місць у салоні, наявність багажника, кріплення для велосипеда, корзини для тварин, сидіння для дітей та інші умови перевезення;
- Таксометр, що показує вартість поїздки;

- Дані про водіїв, включаючи рейтинг, відгуки;
- Варіанти оплати та зв'язок з клієнтом;
- Зв'язок з адміністрацією.

Водійська

- Таблиця замовлень та сповіщення про надходження нових замовлень в реальному часі;
- Навігатор;
- Електронна карта з деталізацією поїздок;
- Таксометр;
- Коментарі до замовлення від пасажирів, в яких вказані особливості поїздки;
- Робоча статистика водія, в якій зазначений його рейтинг, відгуки, фінансовий звіт, бонуси та інша інформація.

Адміністративна

- Інтерактивна карта, що показує онлайн замовлення;
- Функції модерації заявок;
- Управління роботою водіїв, перевірка та допуск нових працівників, відключення від системи;
- Обробка скарг;
- Графіки та таблиці з інформацією про кількість замовлень, автомобілів на зміні, інформацію про відстані та інші дані.

Було вирішено вести розробку додатку клієнтської частини.

2.4 Збір вимог

Збір вимог, за допомогою якого інженерні ідеї передаються вимогам програмної системи. Ці ідеї спочатку дають лише концептуальну інформацію на високому рівні абстракції, але пізніші уточнення створюють форми, подібні до текстів у мовах програмування. Для проектування треба розглянути принципи та функціонал додатку, а саме:

Принципи роботи таксі - в мобільному додатку.

- Клиент повинен встановити спеціальне приложеніе на свій смартфон або планшет. Клієнт має можливість оформити замовлення за допомогою цього додатку; він також має можливість вибрати тип транспортного засобу, а також вказати адрес поставки та адреси прибуття.
- Різна кількість пасажирів. Для замовлення відповідного автомобіля клієнт має можливість вибрати необхідне кількість людей у додатку. Наприклад, ви можете вибрати машину з великим кількістю сидінь, якщо вам потрібно замовити такси на кілька людей. Вибір автомобіля. Відповідно до ваших потреб у додатку такси ви можете вибрати різні види транспортних засобів. Наприклад, ви можете вибрати автомобіль преміум-класу або економ-класу для конкретного випадку.
- Статус замовлення. Клієнт має можливість в режимі реального часу переглядати статус свого заказа в приложеніі після його оформлення. Коли ви точно знаєте, скільки часу залишилося до прибуття машини, це зручно.
- Оцінка керівника Клієнт може оцінити роботу водителя в додатку після завершення поїздки. Це дозволяє іншим клієнтам вибирати найбільш надійних і кваліфікованих водіїв, а також покращує сервіс.
- Бізнес. Такси, які працюють через мобільні пристрої, часто включають різні функції безпеки, такі як оцінка водіїв і можливість прослідувати маршрут автомобіля перед посадкою. Це створює відчуття безпеки для клієнтів і дозволяє їм більше довірити такси, яке доступне в мобільному додатку.
- В цілому, порівняно з традиційними службами такси, використання мобільного додатку допомагає замовити такси. Клієнти отримують більш прозорі послуги, економлять час і сили та відчують більшу впевненість у перевозці з цим варіантом.

Регистрація та підготовка заказу

- Для використання такси через додаток займає всього кілька хвилин. Коли програма завантажується на ваш смартфон, вам потрібно буде заповнити персональні дані та податкову інформацію.
- Є можливість зробити замовлення такси після завершення реєстрації. Для цього вам лише потрібно відкрити програму та ввести адрес, з якого почалася ваша подорож і куди ви їдете. Ви можете вказати своє місцеположення вручну, але система визначить його автоматично. Крім того, ви можете вибрати тип транспортного засобу, додаткові послуги або задати додаткові параметри вашої подорожі.
- У процесі оформлення заказа ви можете побачити інформацію про вартість поїздки та приблизне час прибуття автомобіля. Крім того, ви можете переглядати відгуки та коментарі інших людей про водія, щоб вибрати найкращого фахівця.
- Після підтвердження замовлення про водія та машину надходить повідомлення, а також інформація про передбачуваний час приїзду. Ви можете відстежувати рух автомобіля на карті та дізнатися його точне місцезнаходження.
- Після завершення поїздки ви можете оцінити роботу водія та залишити коментар про якість обслуговування. Також ви отримаєте чек на вказаний платіжний метод, а сума буде автоматично списана.

Автоматичний пошук водія

- Коли клієнт робить замовлення через програму, прохання про поїздку та його поточне місце відправляються на сервер, де відбувається

автоматичний пошук доступних водіїв. За допомогою спеціальних алгоритмів визначається найближчий водій, який може виконати замовлення.

- Клієнту відображається інформація про водія, включаючи його ім'я, рейтинг, автомобіль та номер ліцензії. Якщо клієнт згоден, водієві надсилається повідомлення про замовлення, і він може прийняти або відхилити його.
- Автоматичний пошук водія дозволяє значно зменшити час очікування клієнта. У той же час така система забезпечує більш ефективне використання ресурсів таксопарку, оскільки замовлення автоматично розподіляються між доступними водіями. Це допомагає збільшити дохід кожного водія та скоротити простой машин.
- Розширені функції автоматичного пошуку водія можуть включати облік переваг клієнта, таких як модель автомобіля, додаткові послуги і т.д. Також можна використовувати алгоритми, що враховують пробки, дорожні роботи та інші фактори, які можуть впливати на час доставки водія до клієнта.
- Підсумовуючи все перераховане вище, автоматичний пошук водія є однією з ключових особливостей таксі за додатком, яка забезпечує більш швидке, зручне та ефективне замовлення таксі для клієнта.

Відстеження маршруту та оплата

- Після того, як водій прийняв ваше замовлення, ви зможете бачити його поточне місцезнаходження на карті прямо в додатку. Це дозволяє вам

контролювати, куди їде ваше таксі і наскільки точно він слідує заданому маршруту.

- Відстеження маршруту також допомагає вам визначити оптимальний час приїзду таксі та дозволяє планувати свій час. Наприклад, ви можете підготуватися до виходу на вулицю заздалегідь, щоб зменшити час очікування таксі.
- Крім того, таксі за додатком пропонує зручні варіанти оплати. Ви можете зв'язати свою банківську картку або електронний гаманець з програмою та оплачувати поїздки одним натисканням кнопки. Це зручно, безпечно і дозволяє уникнути незручних ситуацій із готівкою.

Ці вимоги створюють основу для розробки клієнтського додатку для таксі, який буде забезпечувати зручний та ефективний сервіс для користувачів.

Візуальне оформлення

Візуальне оформлення будь-якого додатку є важливою річчю, гарне візуальне оформлення буде притягувати користувачів, а погане оформлення навпаки відштовхувати та заплутувати. У світі, що постійно розвивається, розробки мобільних додатків дизайн відіграє ключову роль у створенні успішних і зручних для користувача додатків. Дизайн мобільного додатка включає в себе не тільки візуальну естетику, але також аспекти взаємодії з користувачем (UX) та інтерфейсу користувача (UI). Добре продуманий мобільний додаток не тільки приваблює користувачів, а й підвищує їхню залученість і задоволеність.

Дизайн - це більше, ніж просто зробити додаток візуально привабливим; мова йде про створення інтуїтивно зрозумілого та зручного користувацького досвіду. Добре продуманий дизайн може вплинути на сприйняття, поведінку та задоволеність користувачів. Для перших вражень.

Дизайн - це перший аспект, з яким стикаються користувачі при взаємодії з програмою. Візуально привабливий та добре продуманий дизайн створює позитивне перше враження та спонукає користувачів до подальшого вивчення. Добре продуманий мобільний додаток підвищує залучення користувачів і спонукає їх проводити більше часу в додатку. Інтуїтивно зрозуміла навігація, естетично приємні візуальні ефекти та інтерактивні елементи — все це сприяє захоплюючому досвіду користувача. Також дизайн впливає на утримання користувачів, гарантуючи, що додаток буде простим у використанні, приємним та відповідним потребам користувачів. Добре спроектована програма створює приємний і приємний досвід, збільшуючи ймовірність того, що користувачі повернуться і порекомендують програму іншим. Дизайн відіграє ключову роль у розробці програми, впливаючи на його візуальне сприйняття, зручність використання та загальне враження користувача. Це не тільки естетичний аспект, а й функціональний, що визначає, наскільки легко та приємно користувачеві взаємодіяти з додатком. Насамперед дизайн має бути інтуїтивно зрозумілим. Користувачі очікують, що функції будуть легко доступні і знаходитися в логічних місцях. Хороший дизайн враховує психологію користувачів та їх очікування, мінімізуючи кількість кроків для досягнення мети. Дизайн також сприяє створенню єдиного бренду. Це стосується не лише логотипу та колірної схеми, а й загального стилю елементів інтерфейсу. Узгоджений дизайн допомагає посилити впізнаваність та довіру до додатку. Для розробки мобільних програм важливо враховувати адаптивний дизайн, який забезпечує оптимальне відображення та зручність використання на різних пристроях та дозволах екранів. Це включає адаптацію елементів інтерфейсу, таких як кнопки і текст, для максимального комфорту користувачів на всіх типах пристроїв, від смартфонів до планшетів. Крім того, дизайн має пряму вплив на зручність використання програми. Чим логічніше та інтуїтивніше розташовані елементи інтерфейсу, тим менше користувачеві доводиться думати, що сприяє підвищенню задоволеності користувачів та рівня їхньої залученості

до додатку. За основу дизайну мобільного додатку буде взятий саме Uber через його зручність, мінімалістичність, прототу та інтуїтивність для користувача.

Структуруючи вище сказане:

1. Оцінка та відгуки. Можливість залишення відгуків та оцінок після завершення поїздки.
2. Онлайн-замовлення та відстеження таксі. Користувач може легко та швидко замовити таксі через додаток. Можливість відстеження руху транспорту в реальному часі на карті.
3. Зручність та інтерфейс. Інтуїтивно зрозумілий та зручний інтерфейс. Можливість зберігання адрес клієнтів для швидкого замовлення.
4. Дизайн. Простий та ненав'язливий.
5. Вибір та налаштування поїздки. Можливість вибору типу автомобіля (стандарт, комфорт, мінівен тощо). Передперегляд вартості поїздки та розрахунок вартості.
6. Управління замовленнями. Можливість перегляду історії замовлень. Редагування або скасування активних замовлень.
7. Платіжні опції. Підтримка різних методів оплати: кредитні картки, електронні гаманці.
8. Сповіщення та комунікація. Сповіщення про статус замовлення та іншу важливу інформацію. Можливість зв'язку з водієм або підтримкою додатку. Відображення інформації про водіїв, їх рейтинг та відгуки.
9. Оцінка та відгуки. Можливість залишення відгуків та оцінок після завершення поїздки.

2.5 Вибір інструментів для розробки

Так як мобільний додаток для таксі ділиться на 3 частини: пасажирську, водійську та адміністративну(серверну) було вирішено у

випускній кваліфікаційній роботі робити пасажирську та водійську роботи на Android studio, а адміністративну(серверну) частину замінити на API.

1. Android studio

Android Studio - це офіційне інтегроване середовище розробки (IDE) для створення програмного забезпечення під платформу Android. Воно розроблено компанією Google і є основним інструментом для розробки Android-додатків. Android Studio надає розробникам різноманітні зручності та інструменти для створення високоякісних інтерфейсів користувача, функціональних додатків та ігор для мобільних пристроїв, таких як смартфони та планшети.

Однією з ключових переваг Android Studio є його інтеграція з Android SDK (Software Development Kit), що дозволяє розробникам легко доступатися до всіх необхідних інструментів, бібліотек та API для розробки Android-додатків. Крім того, Android Studio має вбудовані засоби для створення інтерфейсів користувача за допомогою редактора макетів, розробки логіки програми за допомогою редактора коду, налагодження програмного забезпечення за допомогою вбудованих засобів відлагодження та тестування додатків на різних пристроях і віртуальних пристроях. Однією зі значущих особливостей Android Studio є підтримка Kotlin - сучасної мови програмування, яка стала офіційною мовою для розробки Android-додатків поряд з Java. Kotlin пропонує більш ефективний та безпечний спосіб написання коду для Android-платформи, і Android Studio забезпечує повну підтримку цієї мови, що дозволяє розробникам використовувати її для створення додатків. Крім того, Android Studio постійно оновлюється та покращується командою розробників Google, що дозволяє розробникам користуватися новітніми функціями та технологіями для створення інноваційних додатків для Android. Інтеграція з іншими сервісами Google,

такими як Firebase, також робить Android Studio потужним інструментом для розробки сучасних мобільних додатків з розширеним функціоналом, таким як аналітика, бази даних в реальному часі, автентифікація користувачів та інше.

Узагальнюючи, Android Studio є незамінним інструментом для розробників, які прагнуть створювати високоякісні та інноваційні мобільні додатки для платформи Android. Його потужність, зручність у використанні та постійне оновлення роблять його першим вибором для багатьох розробників по всьому світу.

2. Java

Java - це потужна та універсальна мова програмування, яка була розроблена компанією Sun Microsystems (пізніше придбаною компанією Oracle) на початку 1990-х років. Java була створена з урахуванням принципів "Write Once, Run Anywhere" (Напишіть один раз, запустіть скрізь), що означає, що програми, написані на Java, можуть бути запущені на будь-якій платформі, яка підтримує віртуальну машину Java (JVM).

Однією з ключових особливостей Java є її портативність. Код, написаний на Java, може виконуватися на різних операційних системах, таких як Windows, macOS та різні дистрибутиви Linux, без необхідності зміни вихідного коду. Це досягається завдяки тому, що Java компілюється в проміжний байт-код, який потім виконується на JVM.

Java також відома своєю простотою в використанні та розумінні. Вона використовує синтаксис, подібний до синтаксису мови C ++, що робить її більш доступною для програмістів, знайомих з іншими мовами програмування. Багата стандартна бібліотека Java надає розробникам широкий спектр інструментів для розробки різних типів додатків, включаючи веб-додатки, мобільні додатки, настільні додатки та багато іншого. Java також широко використовується в корпоративному середовищі

завдяки своїй надійності, масштабованості та безпеці. Багато великих корпорацій використовують Java для створення високонавантажених систем, таких як сервери додатків, веб-сервіси та бази даних.

Важливо відзначити, що Java - це не тільки мова програмування, але й платформа розробки програмного забезпечення. Окрім мови програмування Java включає в себе різні інструменти розробки, такі як компілятори, відлагоджувачі та середовища розробки, такі як Eclipse та IntelliJ IDEA, які полегшують процес створення програм на Java. Незважаючи на те, що Java була розроблена понад 25 років тому, вона залишається однією з найпопулярніших та широко використовуваних мов програмування в світі. Її постійний розвиток та обширна спільнота розробників роблять Java важливим інструментом для створення сучасного програмного забезпечення.

3. Firebase

Firebase — це платформа для розробки мобільних та веб-додатків, створена компанією Firebase Inc. і пізніше придбано Google у 2014 році. Firebase пропонує безліч інструментів та сервісів, які допомагають розробникам створювати високоякісні додатки, покращувати їхню взаємодію з користувачами та збільшувати дохід, це потужна платформа для розробки додатків, яка пропонує широкий спектр інструментів та сервісів, необхідних для створення, управління та масштабування мобільних та веб-додатків. . Інтеграція різних компонентів Firebase дозволяє розробникам зосередитися на створенні якісного досвіду користувача, не турбуючись про серверну інфраструктуру та безпеку даних.

Основні компоненти Firebase

1. Firebase Authentication

Firebase Authentication надає методи реєстрації та аутентифікації користувачів з використанням різних провайдерів, таких як:

- Електронна пошта та пароль
- Телефонні номери
- Google, Facebook, Twitter та інші соціальні мережі

Простота інтеграції та підтримка кількох провайдерів аутентифікації роблять цей сервіс особливо зручним для розробників.

2. Firebase Realtime Database

Firebase Realtime Database – це хмарна NoSQL база даних, яка дозволяє зберігати та синхронізувати дані між користувачами у реальному часі.

Основні особливості включають:

- Зберігання даних у форматі JSON
- Синхронізація даних у реальному часі між клієнтами та базою даних
- Офлайн доступність даних та автоматична синхронізація після відновлення з'єднання

3. Cloud Firestore

Cloud Firestore – це сучасна хмарна база даних, яка також підтримує синхронізацію даних у реальному часі. На відміну від Realtime Database, Firestore пропонує складнішу структуру даних, такі як колекції та документи, що дозволяє краще організувати дані.

4. Firebase Cloud Storage

Firebase Cloud Storage призначений для зберігання та надання доступу до медіафайлів, таких як фотографії та відео. Цей сервіс легко інтегрується з Firebase Authentication та забезпечує безпеку даних за допомогою правил доступу на рівні файлів.

5. Firebase Cloud Messaging (FCM)

FCM дозволяє надсилати повідомлення та повідомлення на пристрої користувачів. Основні можливості включають:

- Надсилання повідомлень на iOS, Android та веб-додатки
- Гнучкі налаштування доставки повідомлень
- Висока надійність та масштабованість

6. Firebase Hosting

Firebase Hosting надає швидкий та надійний хостинг для статичних файлів, таких як HTML, CSS, JavaScript, а також динамічних веб-додатків. Основні можливості включають:

- Безкоштовний SSL сертифікат
- Швидка доставка контенту через CDN
- Просте розгортання за допомогою командного рядка

7. Firebase Analytics

Firebase Analytics пропонує потужні інструменти для аналізу поведінки користувачів. Основні функції включають:

- Збір даних про користувачів та події
- Створення звітів користувача та сегментів
- Інтеграція з Google Ads та іншими рекламними платформами

8. Firebase Performance Monitoring

Цей сервіс дозволяє розробникам відстежувати продуктивність їх додатків у часі. Firebase Performance Monitoring допомагає виявляти проблеми з продуктивністю та знаходити їх докорінні причини.

9. Firebase Crashlytics

Crashlytics – це потужний інструмент для відстеження та керування збоями у додатку. Основні можливості включають:

- Збір та аналіз даних про збої

- Повідомлення про критичні помилки
- Деталізовані звіти про причини збоїв

10. Firebase Remote Config

Firebase Remote Config дозволяє змінювати поведінку та зовнішній вигляд програми без необхідності оновлення її в магазинах програм. Розробники можуть задавати параметри та умови для зміни конфігурації програми у реальному часі.

Переваги використання Firebase:

- Інтеграція з іншими сервісами Google: Firebase легко інтегрується з такими сервісами, як Google Analytics, Google Ads та BigQuery, що дозволяє створювати потужні аналітичні та рекламні рішення.
- Масштабованість: Firebase розроблений для масштабування і може підтримувати програми з мільйонами користувачів.
- Безпека: Всі дані в Firebase захищені за допомогою правил безпеки та SSL-шифрування.
- Крос-платформність: Firebase підтримує розробку програм для iOS, Android та веб-платформ, що робить його універсальним інструментом для різних типів проєктів.

Недоліки Firebase:

- Закриті вихідні джерела. Firebase є комерційним проєктом Google, тому компанія не публікує вихідний код. Змінити платформу під себе та переписати модулі – не вдасться.
- Залежність від Google. Якщо в якийсь момент ви захочете перейти на інше рішення, то багато доведеться створювати з нуля і вирішувати проблеми сумісності сторонніх модулів. Крім того, Firebase прив'язана до Google Cloud. Розробники не можуть вибрати іншого хмарного провайдера.

- Обмежений тип бази даних. Платформа пропонує тільки бази даних NoSQL. Це може суттєво ускладнити міграцію проекту на інші рішення.

З його допомогою можна швидко розгорнути бекенд зі своєю серверною логікою, підключити бази даних та налаштувати авторизацію користувачів. Часто платформу використовують для створення MVP та розробки за обмежених ресурсів. За допомогою Firebase веб-/мобільний додаток можна розгорнути без запуску власного внутрішнього сервера. Однак Firebase також може співіснувати з сервером програм.

Якщо програмі потрібен лише код на стороні клієнта або не потрібна складна обробка, тоді всі потреби серверної частини можуть обслуговуватися Firebase. Якщо програмі потрібна складніша обробка або інтеграція зі сторонніми API, тоді Firebase може знаходитися між клієнтами та сервером програми. Клієнти отримують доступ до даних Firebase, а сервер прослуховує зміни в даних. Сервер маніпулює даними у Firebase, про що сповіщаються клієнти. В іншій архітектурі розгортання Firebase логічно розташовується поруч із сервером програм. Клієнти спілкуються з обома. Ця архітектура ідеально підходить для великих застарілих додатків, де деякі частини (наприклад, функції реального часу) обслуговуються Firebase, а старіші частини обслуговуються застарілим кодом сервера.

Розробники Firebase дуже піклуються про безпеку. Всі центри даних мають необхідні сертифікати, дані шифруються, є управління доступом на основі ролей (RBAC), і Firebase дозволяє відстежувати, хто і коли отримав доступ до даних програми.

Спочатку сервіс був розроблений для мобільної розробки, тому він легко працює з програмами на Android і iOS. Крім того, прогресивні (PWA) та інші веб-застосунки також підтримуються. Також можна використовувати Firebase під час розробки в Unity.

4.API

API (Application Programming Interface) – це набір правил і механізмів, які дозволяють різним програмним додаткам взаємодіяти один з одним. API визначає, яким чином один компонент програми може запросити дані або послуги іншого компонента. API можуть надавати доступ до функціональності операційної системи, бібліотек, інших програм або зовнішніх сервісів через чітко визначені методи та протоколи.

Основні типи API

Web API: Це інтерфейси, що надають доступ до веб-служб через Інтернет. Зазвичай вони використовують протоколи HTTP чи HTTPS.

Library API: Надають доступ до функцій та методів бібліотек програмування.

Operating System API: Дозволяють програмам взаємодіяти з функціями операційної системи.

Застосування API у проекті таксі

Для створення повноцінної програми для таксі, необхідно використовувати різні API для забезпечення взаємодії між різними компонентами системи, такими як фронтенд, бекенд та зовнішні сервіси.

Основні API, необхідні у проекті таксі

1. Firebase API
2. Google Maps API
3. Stripe API

1. Firebase API

Firebase надає набір API для різних сервісів, необхідних для створення та керування мобільними та веб-додатками. У проекті таксі Firebase API використовуватиметься для наступних завдань:

Firebase Authentication: API для аутентифікації користувачів (реєстрація, вхід, аутентифікація через соціальні мережі).

Firebase Realtime Database або Cloud Firestore: API для зберігання та синхронізації даних (користувачі, подорожі, транспортні засоби).

Firebase Cloud Messaging (FCM): API для надсилання повідомлень користувачам (статус подорожі, оновлення).

Firebase Storage: API для зберігання та вилучення медіафайлів (фото профілю, документи водія).

Приклади використання:

Реєстрація та вхід користувачів за допомогою Firebase Authentication API.

Збереження даних про подорожі до Firebase Realtime Database або Cloud Firestore.

Надсилання повідомлень водіям та пасажиром про статус поїздки за допомогою Firebase Cloud Messaging.

2. Google Maps API

Google Maps API надає потужні інструменти для роботи з картами та геолокацією, що є ключовим елементом для програми таксі. Основні функції Google Maps API, які будуть використовуватись у проекті:

- Maps SDK for Android: Для відображення карт та розташування користувачів та водіїв.
- Places API: Для автозаповнення адрес та пошуку місць.

- Directions API: Для побудови маршрутів між пунктами відправлення та призначення.
- Geocoding API: Для перетворення адрес в координати і навпаки.

Приклади використання:

Відображення карти з поточним розташуванням користувача.

Автозаповнення адрес при введенні пунктів відправлення та призначення.

Побудова та відображення маршруту від пасажирів до водія та до пункту призначення.

3. Stripe API

Stripe API буде використовуватися для обробки платежів, що є важливою частиною функціональності програми таксі. Основні можливості Stripe API:

- Payment Intents API: Для створення та підтвердження платежів.
- Customer API: Для керування інформацією про клієнтів та збереження платіжних даних.
- Payouts API: Для переказу грошей водіям.

Приклади використання:

- Створення платіжного наміру та опрацювання платежу при завершенні поїздки.
- Збереження та управління платіжною інформацією пасажирів.

Виплати водіям за виконані поїздки.

3. СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ

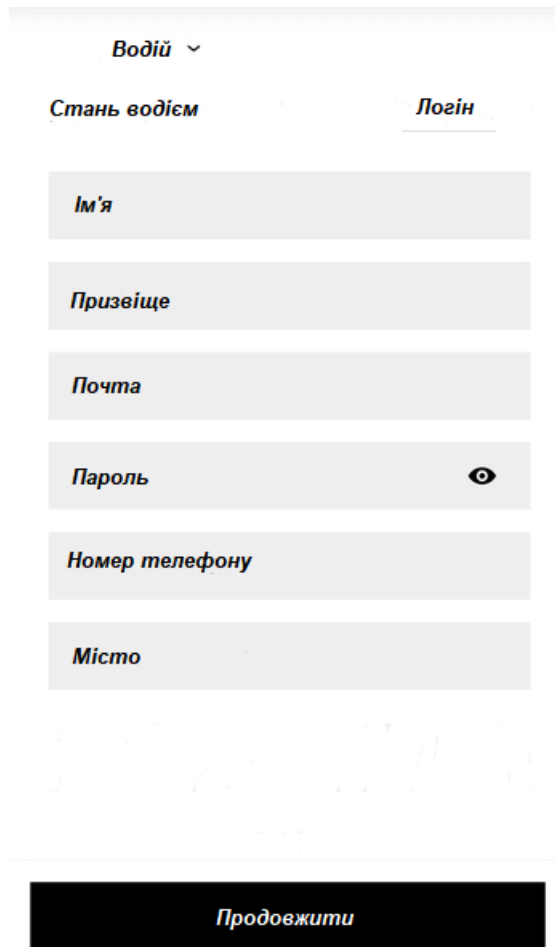
3.1 Графічний інтерфейс

Розробка графічних інтерфейсів слід виконувати за допомогою програмних засобів, які найкраще підходять для конкретного типу проекту та мають функції, необхідні для виконання завдань. Задля випукної кваліфікаційної роботи був обраний сервіс Adobe XD

Adobe XD (Experience Design) - це інструмент, розроблений Adobe для створення прототипів і дизайну інтерфейсів користувача. Він дозволяє дизайнерам та розробникам створювати інтерактивні макети та прототипи додатків та веб-сайтів, оптимізуючи процес проектування.

Для реалізації програми для замовлення таксі на платформі Android з використанням Android Studio, Java та Firebase, Adobe XD може бути дуже корисним інструментом. З його допомогою дизайнери можуть швидко створювати макети екранів програми, визначати зовнішній вигляд та розташування елементів інтерфейсу. Це дозволяє команді краще візуалізувати концепції програми, оптимізувати досвід користувача і затверджувати дизайн до початку розробки. Adobe XD також підтримує створення інтерактивних прототипів, що дозволяє моделювати користувацькі сценарії та потоки взаємодії з програмою. Це особливо важливо для додатків замовлення таксі, де інтерфейс користувача повинен бути інтуїтивно зрозумілим і ефективним, спрощує спільну роботу між дизайнерами та розробниками, дозволяючи експортувати графічні ресурси та створювати специфікації дизайну. Це сприяє більш ефективній інтеграції дизайну у

розробку програми на Android Studio з використанням Firebase, прискорюючи процес та підвищуючи якість кінцевого продукту.



Водій ▾

Стань водієм Логін

Ім'я

Прізвище

Почта

Пароль

Номер телефону

Місто

Продовжити

Рис.3.1 Екран реєстрації користувача

На рисунку можна побачити як виглядає один з екранів мобільного додатку для таксі сервісу після переносу інтерфейсу з Adobe XD до Android studio. Розуміючи під перенесенням реалізування програмним засобом екран до Android studio. Це робиться через макети layouts за допомогою xml файлу, де інтерфейс з Adobe XD додається до дотаку.

Щоб перенести дизайн з Adobe XD в Android Studio для реалізації програми замовлення таксі, дизайнери спочатку створюють макети та прототипи в Adobe XD, визначаючи зовнішній вигляд та взаємодію всіх екранів програми. Після затвердження дизайну експортуються графічні елементи, такі як іконки, кнопки та зображення у форматі, підтримуваному

Android (наприклад, PNG або SVG). Ці ресурси зберігаються у відповідних папках проекту Android Studio, таких як `res/drawable` або `res/mipmap`. Дизайнер також створює специфікації дизайну, включаючи розміри елементів, кольори, шрифти та інтервали, що допомагає розробникам відтворити точний інтерфейс Android Studio. Розробники використовують ці ресурси та специфікації для створення XML-макетів в Android Studio, які описують користувацький інтерфейс програми. Кожен екран, створений у Adobe XD, відтворюється у вигляді окремих XML-файлів, де визначаються положення, розміри та стилі елементів інтерфейсу. Java або Kotlin використовуються для написання логіки програми, включаючи обробку взаємодій користувачів та інтеграцію з Firebase для автентифікації та зберігання даних.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      android:background="@color/white"
7      android:gravity="center"
8      android:orientation="vertical"
9      android:padding="40dp"
10     android:id="@+id/layout">
11
12     <com.google.android.material.textfield.TextInputLayout
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:textColorHint="@color/black">
16         <EditText
17             android:layout_width="match_parent"
18             android:layout_height="wrap_content"
19             android:drawablePadding="5sp"
20             android:drawableStart="@drawable/ic_email_black_24dp"
21             android:hint="@string/email"
22             android:id="@+id/email"
23             android:inputType="textEmailAddress"
24             android:textColor="@color/black"
25             android:textSize="20sp" />
26     </com.google.android.material.textfield.TextInputLayout>
27
28     <com.google.android.material.textfield.TextInputLayout
29         android:layout_marginBottom="30sp"
30         android:layout_width="match_parent"
31         android:layout_height="wrap_content"
32         android:textColorHint="@color/black">
33         <EditText
34             android:layout_width="match_parent"
35             android:layout_height="wrap_content"
36             android:backgroundTint="@color/black"
37             android:drawablePadding="5sp"
38             android:drawableStart="@drawable/ic_vpn_key_black_24dp"
39             android:hint="@string/password"
40             android:id="@+id/password"
41             android:inputType="textPassword"
42             android:textColor="@color/black"
43             android:textSize="20sp" />
44     </com.google.android.material.textfield.TextInputLayout>
45
46     <Button
47         android:layout_width="match_parent"
48         android:layout_height="wrap_content"
49         android:layout_marginTop="30dp"
50         android:background="@color/colorAccent"
51         android:text="@string/register"
52         android:id="@+id/register"
53         android:textColor="@android:color/white" />
54
55 </LinearLayout>

```

Рис.3.2 Екран реєстрації користувача у Android studio

Таким чином, дизайн, створений в Adobe XD, перетворюється на функціональну програму, що підтримує взаємодію з користувачами і забезпечує всі необхідні функції замовлення таксі. Цей процес вимагає тісної співпраці між дизайнерами та розробниками, щоб гарантувати, що кінцевий продукт відповідає оригінальному баченню та забезпечує чудовий користувацький досвід.



Рис.3.3 Сторінка історії замовлень таксі

```

19
20
21 <androidx.cardview.widget.CardView
22     android:layout_width="match_parent"
23     android:layout_height="wrap_content"
24     android:visibility="gone">
25     <LinearLayout
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
28         android:orientation="vertical"
29         android:layout_marginBottom="10sp">
30         <TextView
31             android:background="@color/lightGrey"
32             android:padding="20sp"
33             android:textColor="@color/black"
34             android:layout_width="match_parent"
35             android:layout_height="wrap_content"
36             android:text="@string/balance" />
37     </LinearLayout>
38 </androidx.cardview.widget.CardView>

```

Рис.4 Реалізація інтерфейсу замовлень


```
38
39 <LinearLayout
40     android:id="@+id/empty_layout"
41     android:layout_width="match_parent"
42     android:layout_height="match_parent"
43     android:gravity="center">
44     <TextView
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:text="@string/Немає ("
48         android:drawablePadding="10sp"
49         android:drawableEnd="@drawable/ic_sentiment_dissatisfied_black_24dp"
50         android:textColor="@color/black"
51         android:textStyle="bold"
52         android:textSize="20sp"/>
53 </LinearLayout>
54
```

Рис.5 Реалізація відсутності історії замовлень

3.2 Розробка програмної частини

Для прикладу програмної реалізації додатку для сервісу таксі більш раціональним буде розбір деяких основних функцій та важливих деталей. Крім того, детально опишемо елементи, на які необхідно звернути особливу увагу під час розробки, тестування та впровадження. Цей підхід дозволить глибше зрозуміти структуру, функціональність та критичні моменти.

Оглянемо реєстрацію облікового запису користувача додатку для таксі.

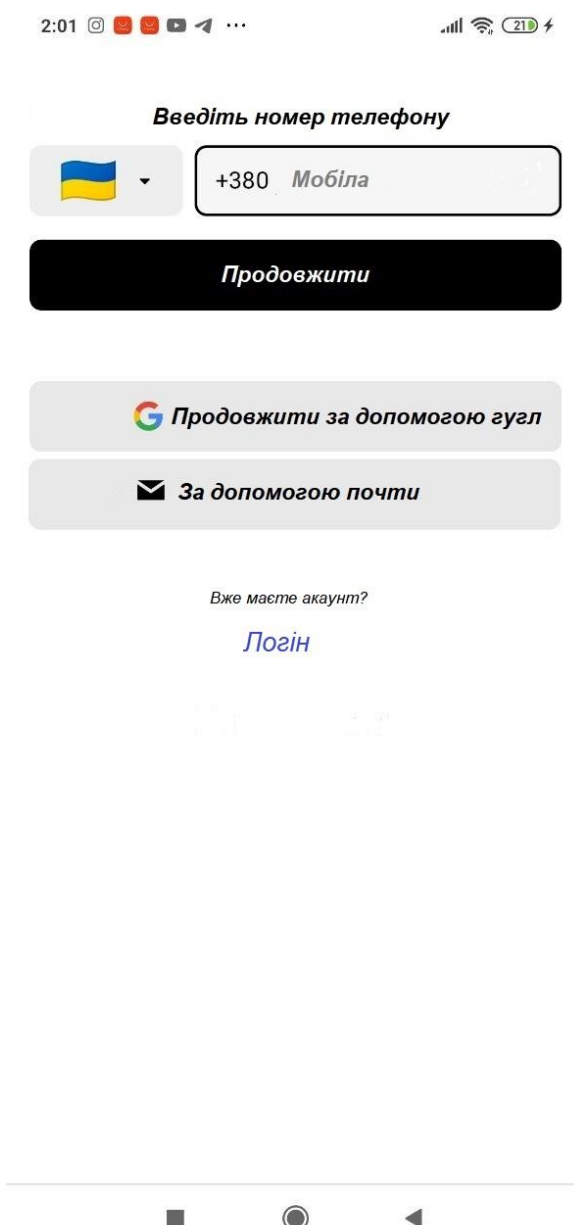


Рис.3.6 Реєстрація користувача

Реєстрація користувача потребує запис номеру, з подальшим підтвердженням за допомогою СМС, або використати реєстрацію за допомогою гугл акаунту або електронної пошти. При введенні номеру телефону чи адреси пошти за допомогою API перевіряється кількість цифр телефону та вірність адреси. Якщо після перевірки введених даних все пройшло добре то далі йде верифікація за допомогою СМС чи коду що прийшло до пошти. Зараз розглянемо чому так відбувається:

```

public class MenuFragment extends Fragment implements View.OnClickListener, GoogleApiClient.OnConnectionFailedListene

    private GoogleSignInClient mGoogleSignInClient;

    private static final int RC_SIGN_IN = 1;
    String name, email;
    String idToken;
    private FirebaseAuth firebaseAuth;
    private View view;

    private boolean started = false;

```

Рис.3.7 Фрагмент дає користувачу вибір між логіном та реєстрацією

```

private void handleSignInResult(GoogleSignInResult result) {
    if (result.isSuccess()) {
        GoogleSignInAccount account = result.getSignInAccount();
        idToken = account.getIdToken();
        name = account.getDisplayName();
        email = account.getEmail();
        // you can store user data to SharedPreferences
        AuthCredential credential = GoogleAuthProvider.getCredential(idToken, null);
        firebaseAuthWithGoogle(credential);
    } else {
        // Google Sign In failed, update UI appropriately
        Toast.makeText(getActivity(), "Login Unsuccessful", Toast.LENGTH_SHORT).show();
    }
}
}

```

Рис.3.8 Фрагмент за допомогою API отримує дані з результату виклику та перевіряє їх.

```

private void firebaseAuthWithGoogle(AuthCredential credential) {
    firebaseAuth.signInWithCredential(credential)
        .addOnCompleteListener(getActivity(), task -> {
            if (task.isSuccessful()) {
                Toast.makeText(getActivity(), "Login successful", Toast.LENGTH_SHORT).show();
            } else {
                task.getException().printStackTrace();
                Toast.makeText(getActivity(), "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
            }
        });
}
}

```

Рис.3.9 Логін/реєстрація за допомогою API Google, якщо при активації аккаунт не існує то він буде автоматично створений.

```

public class RegisterFragment extends Fragment implements View.OnClickListener {

    private EditText    mEmail,
                      mPassword;

    private View view;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(@NotNull LayoutInflater inflater, ViewGroup container,
                            Bundle savedInstanceState) {
        if (view == null)
            view = inflater.inflate(R.layout.fragment_registration, container, false);
        else
            container.removeView(view);
    }
}

```

Рис.3. 10Фрагмент, що відповідає за реєстрацію користувача.

```

private void register(){
    if(mEmail.getText().length()==0) {
        mEmail.setError("please fill this field");
        return;
    }
    if(mPassword.getText().length()==0) {
        mPassword.setError("please fill this field");
        return;
    }
    if(mPassword.getText().length()< 6) {
        mPassword.setError("password must have at least 6 characters");
        return;
    }
}

```

Рис.3.11 Фрагмент, що перевіряє коректність кожного поля, а після реєструє та створює запис в базі даних.

```

final String email = mEmail.getText().toString();
final String password = mPassword.getText().toString();

FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, password).addOnCompleteListener(getActivity(), task -> {
    if(!task.isSuccessful()){
        Snackbar.make(view.findViewById(R.id.layout), "sign up error", Snackbar.LENGTH_SHORT).show();
    }
});

```

Рис.3.12 Firebase створює користувача.

```
/**
 * Initializes the design Elements and calls clickListeners for them
 */
private void initializeObjects() {
    mEmail = view.findViewById(R.id.email);
    mPassword = view.findViewById(R.id.password);
    Button mRegister = view.findViewById(R.id.register);

    mRegister.setOnClickListener(this);
}
@Override
public void onClick(View view) {
    if (view.getId() == R.id.register) {
        register();
    }
}
}
```

Рис.3.13 Елемент що закінчує реєстрацію і переводить користувача до головного екрану

Головне меню мобільного додатку для таксі є центральною частиною інтерфейсі коирстувача. На ньому користувачі взаємодують з основними функціональними можливостями додатку.

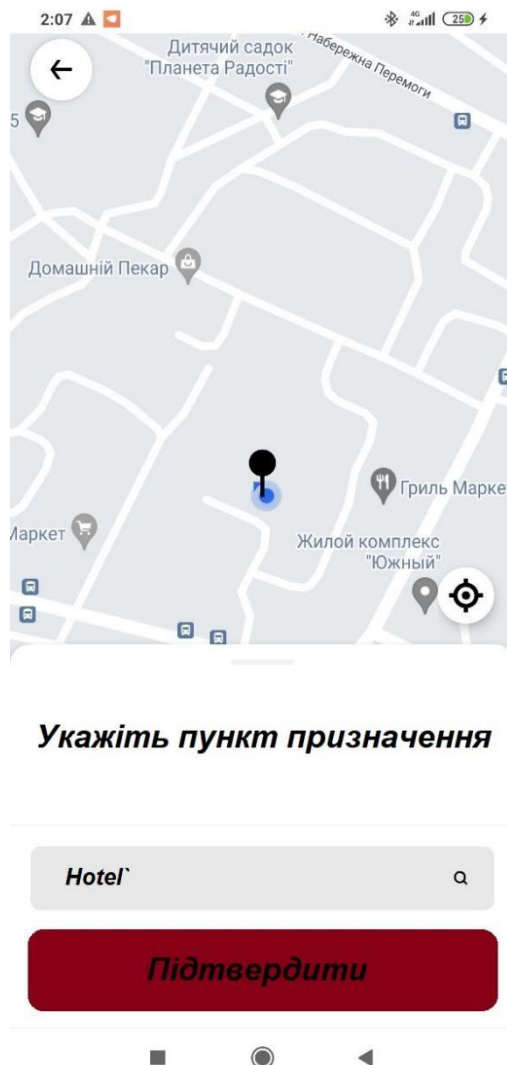


Рис.3.14 Екран замовлення таксі як частина головного екрану

На головному екрані додатку користувач може бачити карту і своє місцеположення на ній, чи є поблизу вільні водії, тут можна вибрати місце доставки або тип авто. Для роботи з геолокацією та мапою використовується Google Maps API

Після того як користувач нажав кнопку замовлення таксі на сторони додатку користувач відправляє запит до Firebase де у базі Firebase Realtime Database додається замовлення. У цій роботі обробка замовлень йде через Firebase Cloud Function, де йде оповіщення водіїв про нове замовлення через Firebase Cloud Messaging та , наприклад, при прийнятті замовлення воно зникає у базі.

Сервіс оплати має значну роль в сервісу для таксі. У контексті роботи програми для замовлення таксі взаємодія з платіжним шлюзом відбувається

через кілька кроків, забезпечуючи безпечну та зручну обробку платежів. Спочатку програма збирає дані про платіж від користувача. Це може включати інформацію про кредитну або дебетову карту, такі як номер картки, термін дії та CVV код або використання методів. Ці дані вводяться користувачем у програмі через захищену форму. Далі програма надсилає цю платіжну інформацію на сервер, який обробляє запит. Серверний додаток, взаємодіючи з платіжним шлюзом, надсилає запит на авторизацію платежу. Запит містить зашифровані дані картки та суму, що підлягає сплаті. Платіжний шлюз, отримавши запит, перевіряє дані картки та здійснює авторизацію. Він зв'язується з банком-емітентом картки для перевірки доступності коштів та справжності даних. Якщо картку підтверджено

ВИСНОВКИ

У роботі з дослідження та розробки мобільного додатку для таксі було проведено глибокий аналіз сучасних вимог та потреб користувачів у сфері пасажирських перевезень. Застосування принципів Agile методології дозволило ефективно керувати процесом розробки, забезпечуючи гнучкість та відкритий зворотній зв'язок з клієнтами. Основні функціональні можливості додатку були ретельно пророблені та впроваджені з метою забезпечення максимальної зручності та задоволення потреб кінцевих користувачів. Під час дослідження виявлено високий потенціал успішності та впровадження додатку на ринку, що підкреслює актуальність даної кваліфікаційної роботи. Результати дослідження можуть служити основою для подальшого вдосконалення та розвитку мобільних технологій у сфері пасажирських перевезень.

У ході виконання дипломної роботи було розроблено мобільний додаток для таксі, призначений для оптимізації процесів замовлення та надання таксі послуг. У процесі аналізу існуючих рішень було виявлено основні проблеми та недоліки, які вдалося врахувати під час проектування та розробки нашого додатку. Основні характеристики розробленого додатку включають зручний інтерфейс, інтуїтивно зрозумілу навігацію, високу швидкість обробки запитів і надійну систему оплати. Впровадження функції відстеження розташування автомобіля та можливості оцінки якості обслуговування дозволяють покращити взаємодію між клієнтами та водіями. Тестування програми показало його високу стабільність та надійність у різних умовах експлуатації. Користувацькі відгуки та результати опитувань підтверджують позитивне сприйняття користувачів та готовність до використання розробленого рішення на ринку, розроблений мобільний додаток для таксі є ефективним інструментом для оптимізації та вдосконалення сервісу таксі, забезпечуючи високий рівень задоволеності клієнтів та підвищення конкурентоспроможності підприємства на ринку пасажирських перевезень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Android. Полное руководство" - Билл Филлипс, Крис Стюарт, Кристин Марсикано
2. Программирование для профессионалов. 4-е издание. — СПб.: Питер, 2021. — 704 с.: ил. — (Серия «Для профессионалов»). ISBN 978-5-4461-1657-7
3. Head First. Java Берт Бейтс Кети Сьерра ФБ901956У 9786175220337
4. Taylor, James (2004). Managing Information Technology Projects.
5. Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles Paperback – 28 Aug. 2016