ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПЕРЕГЛЯДУ ВІДЕО ОФЛАЙН З YOUTUBE МОВОЮ C#»**

Виконав: студент 4 курсу, групи ПД-41 спеціальності
121 Інженерії програмного забезпечення
(шифр і назва спеціальності)

Куцук В.А.
(прізвище та ініціали)
Керівник Гребенюк В.В.
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)
Нормоконтроль_____
(прізвище та ініціали)

КИЇВ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії Програмного Забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія Програмного Забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії Програмного Забезпечення

_____Негоденко О.В.

«____»_____2021 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Куцук Валерії Андріївні

(прізвище, ім‘я, по батькові)

1.Тема роботи: «Розробка мобільного додатку для перегляду відео офлайн з YouTube мовою C#»

Керівник роботи Гребенюк Віктор Вікторович, доктор філософії

затверджені наказом вищого навчального закладу від — 12.03 2021 року №65.

2. Строк подання студентом роботи 01.06.2021.

3. Вхідні дані до роботи:

3.1Розробка мобільних додатків для ОС Android

3.2 Алгоритм скачування відеофайлів без порушення прав YouTube

3.3 Тестування UI/UX користувача

3.4 Технічний функціонал Visual Studio

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

4.1 Аналіз існуючих рішень для скачування відеофайлів

4.2 Дослідження технічних документацій для створення застосунку ОС Android

4.3 Розробка функціоналу мобільного додатку для тестування

4.4 Аналіз та створення мобільного застосунку

5.Перелік графічного матеріалу :

5.1 Титульний стайд

5.2 Об’єкт, мета, предмет дослідження

5.3 Порівняльна характеристика аналогів.

5.4 Опис Visual Studio

5.5 SQLite

5.6 Макети додатку

5.7 Висновки

6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково – технічної літератури	19.04.2021 – 22.04.2021	
2	Вимоги для розробки мобільного додатку	24.04.2021 – 26.04.2021	
3	Аналіз існуючих мобільних додатків	28.04.2021 – 29.04.2021	
4	Дослідження програмних засобів	30.04.2021 – 05.05.2021	
5	Моделювання об’єкту проектування	06.05.2021 – 22.05.2021	
6	Вступ, висновки, реферат	22.05.2021 – 26.05.2021	
7	Розробка презентації мобільного застосунку	27.05.2021	
8	Здача роботи	01.06.2021	

Студентка Куцук Валерія Андріївна

Керівник роботи Гребенюк Віктор Вікторович

РЕФЕРАТ

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПЕРЕГЛЯДУ ВІДЕО ОФЛАЙН З YOUTUBE МОВОЮ C#

Об'єкт дослідження – можливість перегляду відеороликів з YouTube офлайн.

Предмет дослідження – додаток для перегляду відеороликів з YouTube офлайн.

Мета роботи – розробка відеоплеєра з можливістю перегляду відеороликів з YouTube офлайн, мовою C#.

Методи дослідження – методи розробки програмного забезпечення, обробка та аналіз інформації, дослідження інформації, методи наукового моделювання.

Наукова новизна даного додатку заключається в :

1. Безкоштовному перегляді відеороликів без реклами.
2. Використанні середовища Xamarin та мови програмування C#, які допоможуть вдало виконати поставлену мету даного застосунку.
3. Збереженні записів у базі даних за рахунок чого, їх може легко знайти та переглядати користувач.
4. Розробленні алгоритму для автоматизованої системи пошуку файлів та їх сортування, створення груп.
5. Дослідженні алгоритму скачування відеофайлів з YouTube та перегляду їх офлайн.

Основою розробленого алгоритму отримання відеофайлів, являється база даних, так як її проектування допомагає зберігати інформацію в певному вигляді таблиці, яку буде зручно обробляти користувачеві. Це допоможе йому швидко знайти потрібний файл та забезпечити перегляд у форматі офлайн.

У даній роботі виконаний аналіз конкурентів на ринку. Визначені їх переваги та недоліки. У результаті визначені основні потреби споживачів.

Середовище Xamarin дозволяє виконати всі вище описані вимоги для даного застосунку. Розроблено алгоритм передачі файлів та інтуїтивність інтерфейсу для зручності користувача. Серед інших програм проект відрізняється автоматизацією збереження відеофайлів.

Сфера використання вищезгадуваного додатку полягає у вільному доступі до файлів без застосування мережі Інтернет. Також, застосунок забезпечує підвищення ефективності процесів за рахунок ретельної структуризації його візуального наповнення.

ЗМІСТ

УМОВНІ ПОЗНАЧЕННЯ

SVN (Subversion)– вільна система керування версіями

БД – База даних

MSBuild - Microsoft Build Engine

СУБД – Система Управління Базами Даних

MVVM - (Model – View – View Model)

UML - Unified Modeling Language

XAML (eXtensible Application Markup Language)

ПК – Персональний комп’ютер

ВСТУП

Актуальність теми полягає у тому, щоб користувач міг користуватись своїми файлами, коли немає доступу до Інтернет – мережі та автоматизації процесу доступу до цих файлів.

На сьогоднішній день технології розвиваються дуже швидко, тому для користувачів важливо використовувати певні функції щоденно, що дозволяє мережа Інтернет.

Теперішні інструменти для розробки програмного забезпечення дозволяють IT – інженерам простіше та швидше реалізувати нові ідеї для різних сфер життя, а використання портативних пристроїв являється основною складовою нашого життя. Кожного року компанії : Samsung, Apple, Huawei, Xiaomi, BBK, Lenovo, LG, Sony, ASUS, Nokia випускають гаджети, вдосконалюють їх за вимогами користувачів, слідуючи новим IT - технологіям, які стрімко розвиваються. Технічні фахівці влаштовують функції, які допомагають споживачам спрощувати власне життя та насолоджуватись структуризацією інформації, яку отримують. Play Market оновлюється щоденно. Спеціалісти у галузі розробки додають на цю платформу недавно зроблені мобільні застосунки для того, щоб покращити життя купцям та отримати прибуток. Це свого роду змагання, де беруть участь експерти даної сфери, щоб вивести свій додаток у топи. Тому розробка мобільних застосунків, які допомагають займатись тайм – менеджментом чи вивчати нову сферу, є максимально актуальною. Досліджування саме мобільних додатків є важливим, оскільки ринок розробки націлений саме на мобільне середовище застосування.

На даний момент є певна кількість додатків, які дозволяють завантаження будь – яких файлів, оскільки це дозволяє отримати нові знання з тієї чи іншої області, але для цього потрібно мати вільний доступ до Інтернету. Подорожі є невід’ємною складовою успішних людей. Тільки під час поїздки не завжди є доступ до мережі Інтернет. На даний момент із швидким розвитком різноманітних додатків, користувачів стає дедалі більше, а структурувати свої файли та

насолоджуватись ними стає дедалі простіше. Це дає змогу зайняти час під час польоту чи їзди у метро.

Тому, було вирішено, що створення додатку з можливістю скачування відеофайлів та переглядати їх, коли немає доступу до Інтернет – мережі , є необхідним.

Об'єкт дослідження – можливість перегляду відеороликів з YouTube офлайн.

Предмет дослідження – додаток для перегляду відеороликів з YouTube офлайн.

Мета роботи – розробка відеоплеєра з можливістю перегляду відеороликів з YouTube офлайн на мові C#.

Основні питання дослідження :

1. Проаналізувати формати відеофайлів, які буде завантажувати користувач.
2. Проаналізувати функції, які доступні в аналогах та їх особливості.
3. Проаналізувати інструменти, які будуть використовуватись у ході даної роботи для реалізації додатку.
4. Дослідити використання алгоритму, який буде зберігати файли в базі даних.
5. Вивчити можливість використання мови C# для створення додатку.
6. Встановити попит на функції, які має виконувати створений додаток.
7. Опрацювати реалізацію та тестування програмного забезпечення для перегляду відеофайлів з YouTube офлайн.

Методи дослідження – методи розробки програмного забезпечення, обробка та аналіз інформації, дослідження інформації, методи наукового моделювання.

Наукова новизна отриманого додатку :

1. Ефективність використання розробленого алгоритму підтримки відеофайлів та постійний швидкий доступ до них з бази даних.
2. Досліджено, що зі всіма поставленими задачами буде доцільно використовувати середовище Xamarin та мову програмування C#.

3. Встановлено, що перегляд відеофайлів без доступу до Інтернет – мережі буде більш сприятливим чинником для цільової аудиторії.

Практична значущість розробки даного додатку:

Результатом виконаної роботи є додаток, який орієнтований на використання з мобільними гаджетами на платформі Android. Задача полягає саме у компактній структуризації всіх файлів програми. Розроблено алгоритм автоматизації процесу завантажування відео та подальшого зберігання цієї інформації у базі даних, що дозволяє користувачеві мати вільний доступ до застосунку без використання доступу до Інтернет – мережі.

Результати дослідження бакалаврської роботи апробовані:

1. На всеукраїнській науково-технічній конференції: «Застосування програмного забезпечення в інфокомунікаційних технологіях»
2. На всеукраїнскій науково-технічній конференції «Сучасний стан та перспективи розвитку IoT»

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика предметної області

При знайомстві з різними інформаційними застосунками, клієнт може обрати область, яка його цікавить, та успішно використовувати у повсякденному житті. Постає питання: який вид застосунків обрати для розширення кругозору чи вивчення нового матеріалу?

У повсякденному житті сучасний користувач Інтернет - простору має допоміжні інструментарії для структурування певних областей життя. Для цього, існує багато застосунків, різних категорій та сфер застосування, які допомагають керувати певними видами життєдіяльності. Це може бути : застосунок для вивчення нової мови, онлайн магазин, банківський додаток, соціальна мережа. Без певних кластерів сучасний користувач уже не може обійтись, тому що це значно спрощує функції саморозвитку чи перегляд, наприклад, текстових документів.

Спочатку дуже поширеною була розробка відеоплеєрів для ПК. Розробники прагнули створити дедалі більш зручні та функціональні особливості для цільової аудиторії, але з кожним роком екрани у смартфонах збільшуються – наслідком чого, все менше людей переглядає фільми чи медіа контент на телеекрані чи стаціонарному комп’ютері. Тому кожен з нас частіше використовує екран телефону при кожній зручній можливості, бо часто ми маємо власний гаджет під рукою у будь – який час доби, при цьому, що аудіо можна відтворювати через навушники чи динаміки телефону, а зображення через дисплей. Так, як сфера прослуховування(медіа) стає більш вимогливою, розробники стараються орієнтуватись на мобільний формат та задовольняти всі потреби користувача та вносити нові унікальні впровадження для зручності, що можливо зі швидкоплинністю часу нових технологій.

Варто засвідчити, що зараз відеоплеєри використовує кожна людина, починаючи від маленької дитини з дитсадка, закінчуючи директором школи. Варто зауважити, що медіа програвачами користуються не тільки для розваги, але й для роботи чи навчання. На роботі лектор виступає з презентацією, де наглядним прикладом тієї чи іншої побудови стає відео чи аудіо, у дитсадку чи школі вчителі також намагаються використовувати медіа файли для розвитку дітей. Насправді, мультимедіа відіграє велику роль у покращенні освіти та опановуванні нових теорій чи функцій людини, бо може як і розвивати, так і навчати. Учні і студенти використовують для розвитку в будь – якій профільній області, медики – для перегляду загальних відомостей в області анатомії, вчитель – для кращого засвоєння інформації чи відпочинку, маркетологи – для продажу нового продукту. Навіть, дивлячись телебачення, ми використовуємо відео програвач.

Можна пересвідчитись, що попит на даний вид додатку є вагомим. Бо цільову аудиторію не можна розділити на категорії чи вікові групи. Такий додаток має бути зручним та корисним для людей із різних областей.

За рахунок того, що іноді виникає проблема переглянути відео без доступу в Інтернет або завантажити декілька улюблених фільмів, або вивчити нову сферу у науці, можемо зробити висновок, що розробка мобільного відеоплеєра з можливістю перегляду відеофайлів офлайн є хорошим рішенням.

Отже, можемо пересвідчитись, що додаток має включати :

1. Перегляд відео офлайн.
2. Аналіз завантаження відео з сервісу YouTube.
3. Можливість перегляду списку завантажених відео у середовищі додатку.
4. Створення плейлістів за допомогою Бази даних.
5. Простий та інтуїтивний інтерфейс.

1.2 Аналіз та характеристика аналогів

Сьогодні створено чимало мобільних застосунків для перегляду відеофайлів. І для того, щоб почати розробляти певну систему, потрібно проаналізувати та дослідити переваги та недоліки аналогів. Це допоможе уникнути помилок та вдосконалити певні модулі власного додатку. Переглянути чого не вистачає у представлених застосунках, а що вже застаріле чи можна реалізувати краще за допомогою нових алгоритмів чи бібліотек.У ході роботи було виявлено, що на даний момент існуючі додатки не покривають всі вимоги та побажання користувачів.

У межах даної дипломної роботи досліджено чотири аналоги, які реалізовують схожі рішення даної предметної області. Перший аналог зображений на рис.1.1.

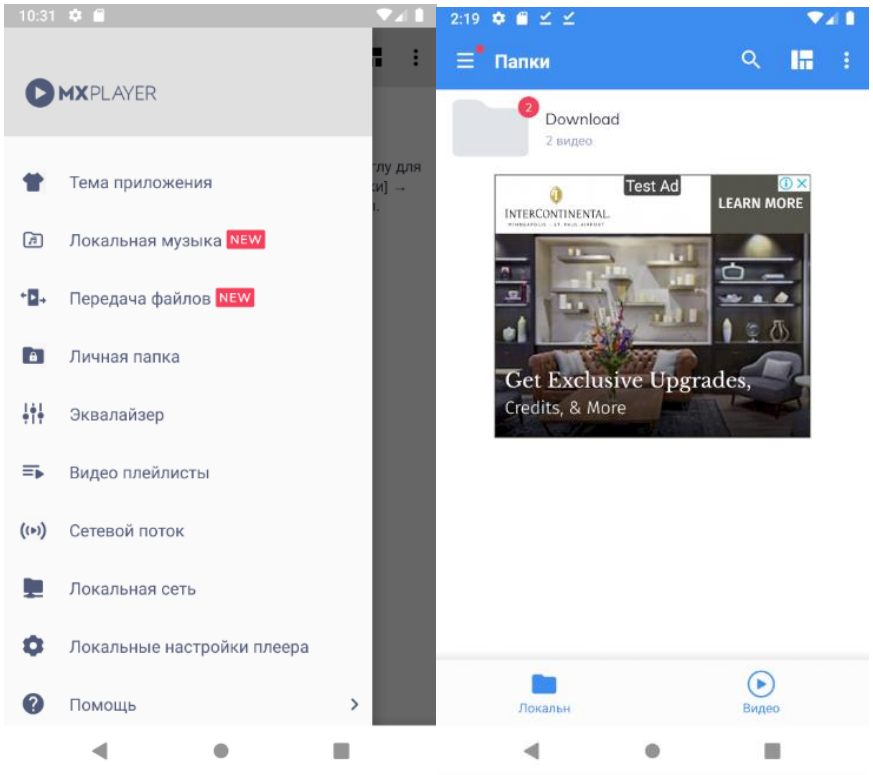


Рисунок 1.1 - Застосунок MX Player

MX Player – додаток, що дозволяє переглядати відео у широкому спектрі форматів онлайн. Логіка додатку полягає в тому, що завантажити відеофайл у самій системі неможливо, потрібно обрати його з галереї телефону для перегляду. Тобто, спочатку користувач повинен знайти сервіс, де можливе скачування цього відео, а тільки потім зайти у програму, обрати йогота переглядати. У межах системи телефону інтерфейс показує нам, що користувач скачав відео на власний телефон і водночас застосунок MX Player вказує, що з'явились нові відеофайли, які читач може переглянути. Додаток також має платну версію Pro, де пропонуються ширші можливості. У безкоштовній версії є нав'язлива реклама, яка з'являється при кожному новому кроці. Є можливість створювати папки та власні плейлісти. Додаток має меню для маршрутизації, але інтерфейс не є достатньо інтуїтивним.

XPlayer – простий інтуїтивний інтерфейс рис. 1.2.

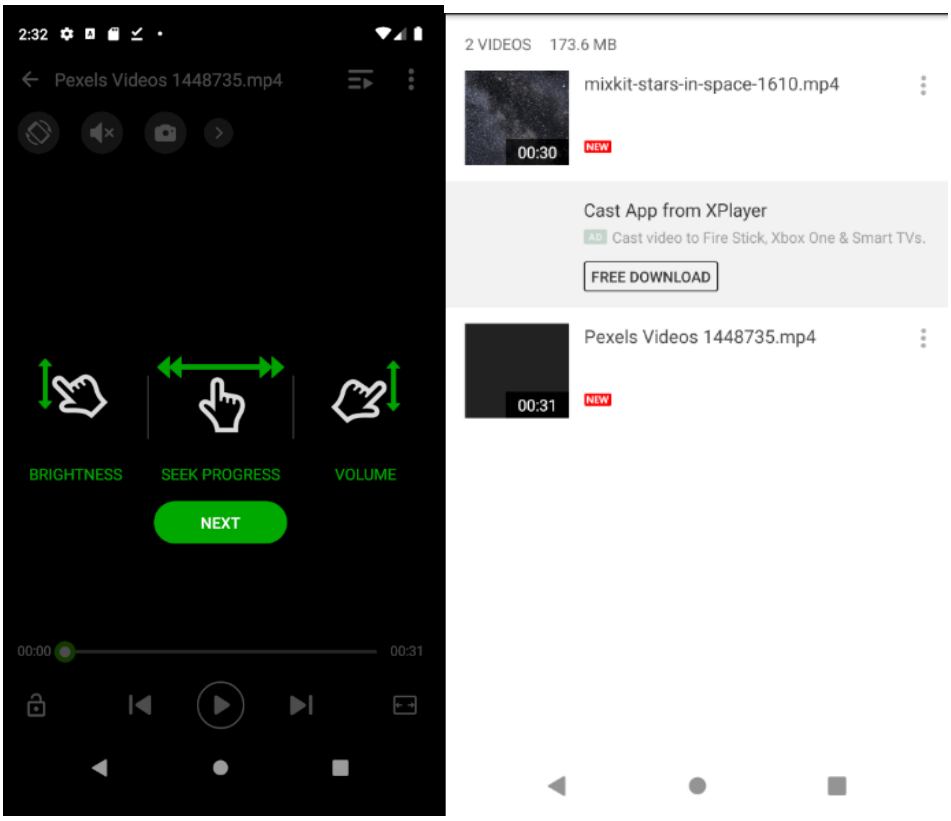


Рисунок 1.2 - Застосунок XPlayer

Переглянути відео можна з галереї телефону за допомогою даного відеоплеєру. Реклама присутня при переході на кожну нову кнопку. Перевага додатку полягає у тому, що перед початком перегляду відео відображується модальне вікно, де описано основні функції жестів, якими абонент може користуватись у ході перегляду відео. Підтримує велику кількість відеоформатів та перегляд відео онлайн.

BSPLayer – додаток має перевагу у широкому виборі функцій у меню, якими користувач може керувати при перегляді відео рис. 1.3 .

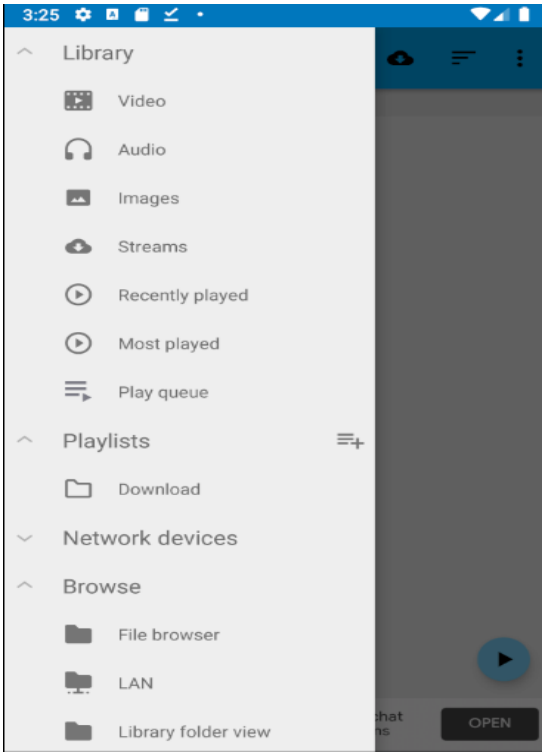


Рисунок 1.3 - Застосунок BSPlayer

Великим мінусом є підтримка реклами у безкоштовній версії. Також є налаштування відтворення відео, але не для всіх відеоформатів. Якщо користувач любить переглядяти кліпи чи просто короткі відеоролики, то для цього добре підходить функція створення плейлістів, що також є у даному дадатку. Неінтуїтивний інтерфейс. Можливо, підійде не любій цільовій аудиторії, а більш молодому поколінню.

Домашня сторінка додатку KMPLayer вказана на рис. 1.4.

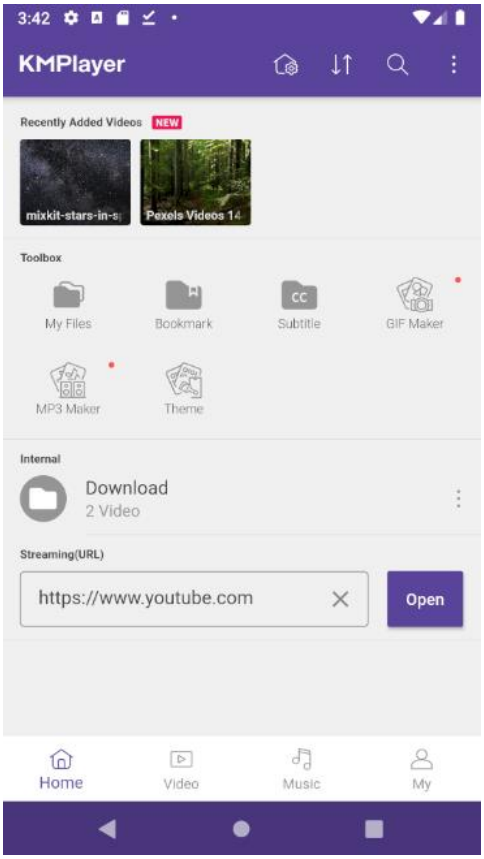


Рисунок 1.4 - Застосунок KMPLayer

KMPlayer – є одним з найбільших конкурентів за рахунок того, що не підтримує рекламу, а також має зручний, новітній дизайн. Має вдосконалену версію Pro та особистий кабінет, який можна поставити під пароль. Можна створювати плейлісти та вказувати вподобанні треки чи відео. Основним мінусом є те, що користуватись ним можна онлайн та недосконала система перегляду відео через URL – формат, що зовсім не є зручним для користувача, який обрав даний додаток для того, щоб дивитись відео з певного, конкретного ресурсу.

У даному розділі було розглянуто основні додатки – аналоги, які доступні на теперішньому ринку Play Market. За рахунок того, що з кожним роком конкурентів стає дедалі більше, а вимоги користувачів змінюються щоденно, відеоплеєр повинен покривати всі функції, що будуть зручні та інтуїтивно прості

для користувачів. Основним рішенням було розробка та проектування додатку, який буде відповідати всім вимогам та підходити користувачеві з будь – яким набором знань, сфери роботи чи навчання та без досвіду використання схожих систем, для щоденного використання.

Таблиця 1.1 - Недоліки та переваги аналогів

Назва застосунку	Переваги	Недоліки
MX Player	1. Підтримка багатьох відеоформатів. 2. Історія переглядів.	1. Підтримка реклами. 2. Онлайн використання.
XPlayer	1. Простота використання. 2. Модальне вікно з інструкцією відтворення відео.	1. Перенасичений рекламою. 2. Відсутність функціоналу перевірки знань.
BSPLayer	1. Широке меню функцій. 2. Створення різних плейлістів та вподобаних відео.	1. Немає підтримки багатьох відеоформатів.
KMPlayer	1. Новітній дизайн додатку. 2. Підтримка субтитрів.	1. Онлайн додаток. 2. Недосконала система перегляду відео через URL - формат .

Існують також інші застосунки, що схожі між собою та виконують схожі функції та потреби користувачів. Тобто : огляд відео через певний ресурс, що дозволяє переглядати його в кращій якості, іншому форматі, фоновому режимі, використовувати подвоєний звук, ознайомлюватись із субтитрами, насолоджуватись сучасним дизайном, створювати плейлісти за вподобаннями чи категоріями.

Але основним мінусом багатьох додатків є те, що потрібно переглядати в форматі онлайн або ж відео зберігається у пам’яті телефону, але немає

можливості переглянути весь список завантажених відео в самій системі. У багатьох додатках, де доступна безкоштовна версія, спостерігається активна підтримка реклами, що робить весь процес довшим, бо для того, щоб згорнути її – потрібно почекати певний проміжок часу. Тому, зазвичай, і пропонується купити оновлену версію Pro для того, щоб використовувати додаток без реклами та споглядати інші, розширені позиції.

Переваги та недоліки були враховані при розробці даного застосунку – відеоплеєр з можливістю завантажування та перегляду відео офлайн з платформи YouTube. Назва додатку – «BVYOU», що призначений для користувачів любого віку, вподобань та сфери заняття.

Основна властивість додатку «BVYOU» - це те, що використовуючи його, користувачеві необов'язково слідкувати за покриттям мережі Інтернет. Непотрібно заздалегідь думати про те, який фільм подивитись у літаку чи метро.

Основною метою застосунку являється те, щоб занурити користувача у перегляд відеофайлів без додаткових думок, що значно покращує сприйняття інформації. Система перегляду відео офлайн є хорошим рішенням і вдалою ідеєю для розробки, оскільки тема на даний момент актуальна та потрібна на теперішньому ринку.

На рис.1.5 зображена статистика за 2020 та 2021 роки по використанню ОС Android у світі. Як показано на рис. 1.5. близько 60 – 80 % людей використовують Android. Завдяки цим даним, що вказані на ресурсі https://gs.statcounter.com/os-market-share/mobile/worldwide [1] , було вирішено розробляти додаток для ОС Android.

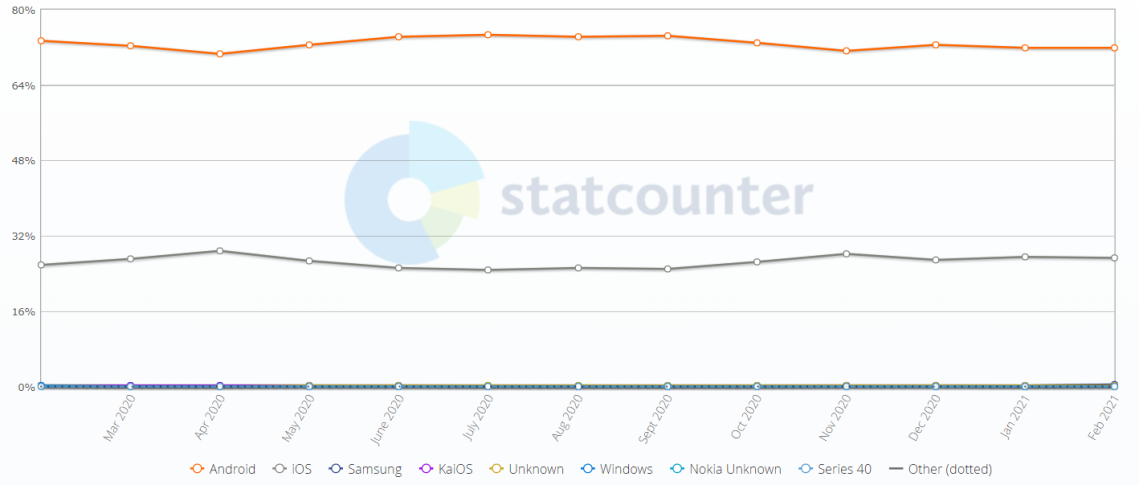


Рисунок 1.5 - Статистика використання ОС Android у світі

ОС Android – безкоштовна операційна система, що дає змогу користуватись смартфонами, планшетами, ігровими приставками, годинниками, телевізорами, електронними книгами і т.д.

Переваги використання ОС Android:

1. Пряма загрузка любих файлів із Інтернету на пам’ять телефону.
2. Великий вибір : фірми телефону, різні моделі, колір, пам’ять, функціонал, аксесуари, розмір.
3. Програми в Google Play дешеві або зовсім безкоштовні.
4. Різноманітність налаштувань дозволяє користувачу вибрати не тільки зовнішній вигляд меню, а й налаштувати глибокі поля самої системи.
5. Можна вільно завантажувати різні програми з різноманітних систем.
6. Вільний доступ для користувача до всіх файлів системи.

2 ПРОГРАМНІ ЗАСОБИ ЗАСТОСУНКУ

2.1 Вибір середовища розробки

Існує три найбільш відомих середовищ розробки : Visual Studio, Project Reader, Eclipse.

Eclipse – це один з найпопулярніших безкоштовних середовищ для розробки мобільних додатків у якому використовуються велика кількість плагінів, вимагає більш ретельного налаштування через складність підналаштувань, дозволяє швидко та гнучко керувати відладчиком.

За рахунок членів суспільства можна провести любий C# продукт по повному циклу розробки. Але великий мінус полягає у тому, що середовище вже являється застарілим, оскільки з 2016 року не підтримується Google.

Project Rider – це плагін, який з самого початку був розроблений для підвищення продуктивності Visual Studio, але тепер на його основі створена IDE. Функціональність, продуктивність та кількість опцій дозволяють швидке створення коду.

Також підтримує декілька програм, які виконуються одночасно і контроль версій дозволяє організувати роботу з Git, можна налаштувати на різні мови програмування. Середовище співпрацює з Windows, Linux, MacOS і підтримка повного циклу від створення ідеї, проектування до підтримки даного мобільного додатку. Головними недоліками вважається те, що середовище не є безкоштовним та частина функціоналу ще розробляється та є багато багів.

Visual Studio – середовище [10], що дуже добре співпрацює з C#, є безкоштовна версія Community Edition, де є все, що потрібно розробнику.

Порівняно з Eclipse та Project Rider, у середовище Visual Studio влаштовані всі потрібні функції та інструменти для розробника, що створює певний продукт. Влаштована хмара також дає перевагу даному середовищу та Xamarin, що

являється Framework та має всі сучасні функції мови C#, надійні бібліотеки, якими можна користуватись при розробці.

Також Visual Studio є безкоштовним середовищем прогрмаування.

Основним плюсом розробки на Xamarin являється те, що продуктивність Framework постійно покращується і хоче відповідати оновленим стандартам розробки [2]. Платформа пропонує комплексне рішення, щоб слідкувати за рішенням питань тестування додатку.

Отже, після огляду всіх найпопулярніших середовищ розробки можна зробити висновок, що Visual Studio буде найзручнішим, найпростішим рішенням для розробки ОС Android. Оскільки, там зібрані всі потрібні функції для розробника. Також середовище являється простим у використанні та налаштуванні.

2.2 Функціональні модулі середовища розробки Visual Studio

На рис. 2.1 зображено інтерфейс середовища розробки Visual Studio з одним із видів елементів(blank app) для розробки застосунку.

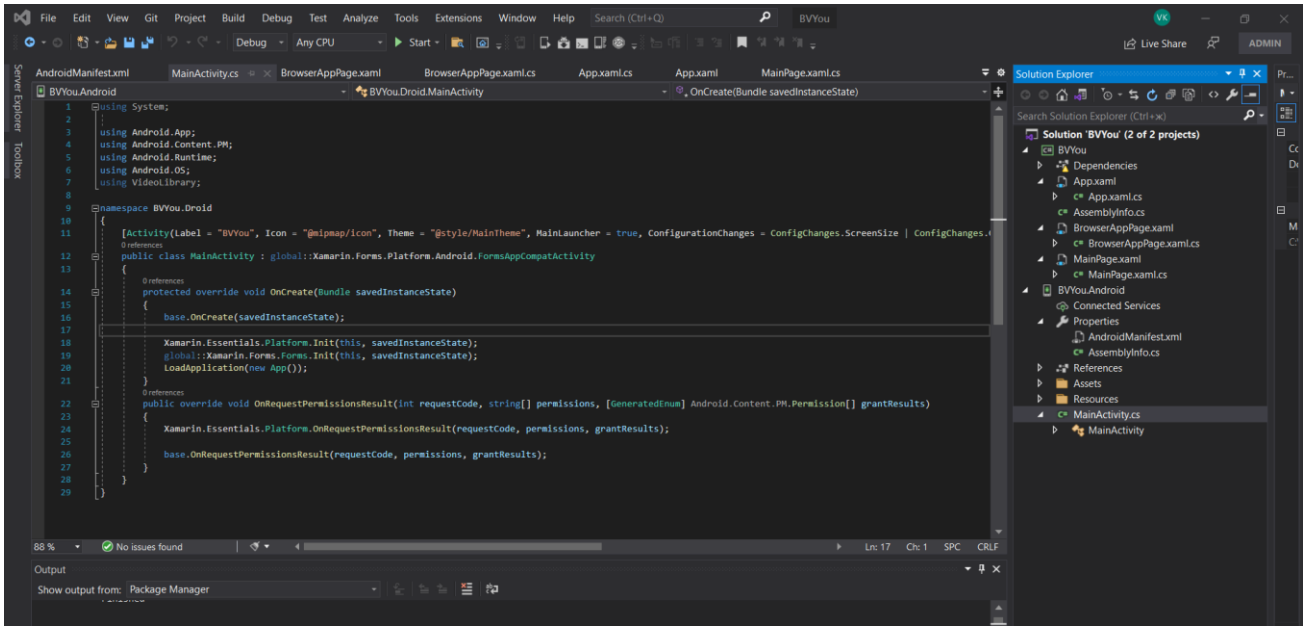


Рисунок 2.1 - Інтерфейс середовища розробки Visual Studio

Загальні модулі для розробки в середовищі Visual Studio:

1. Тестування додатку можливе без використання реального інструменту. А за допомогою емулятора, що влаштований у Visual Studio з вибором будь – якого гаджета та його версії.
2. Вибір підходу Forms або Native. Native більше використовуються для високонавантажених систем, а Forms для простіших додатків. Також розуміють на якій платформі запущений застосунок викликають необхідну реалізацію і завантажують необхідну збірку.
3. Кросплатформенність – загальний код для IOS, Android, Windows.
4. Консоль – лог файли показують хід загрузки додатку, помилки компіляції, підказки для автоматизації.
5. Швидкі жести допомагають правильно використовувати наслідуваність чи стилізацію.
6. Магазин в якому підключаються різні модулі для додаткових функцій у розробці. Дозволяє внести у проект компоненти, які написані не тільки розробниками Xamarin, але й іншими інженерами. Кількість бібліотек збільшується, вони бувають як платні, так і безкоштовні.
7. Доповнення коду, включаючи імпорт різних namespaces.
8. Зручне створення та вміст файлів в списку застосунку. Дає хорошу навігацію при створенні Content Page, вкладеність .cs файлу.
9. Механізм дебагу який включає слідування за вмістом перемінної, візуалізацію потоків.
10. Додаткова влаштована інтеграція з GIT, SVN.

Застосунок містить файл AndroidManifest.xml, де знаходиться вся важлива інформація : дозволи, версії, назви. Для запуску додатку та виконання коду, програма повинна містити коректну інформацію з файлу маніфест.

У файлі App.xaml знаходиться весь UI головної сторінки, яку може відображати емулятор. У піраміді також розрахований App.xaml.cs файл, що дозволяє запрограмувати створений раніше інтерфейс.

Описи всіх кнопок, назви додатку, знаходиться у Resources.

Xamarin використовує PCL [12](переносимі бібліотеки класів)-кросплатформенність для створення різних проектів. Використання спільного коду для різних видів додатків дає можливість створення проекту одразу на декількох платформах. Тобто, при розробці застосунку, можна обрати Xamarin.IOS, Xamarin.Android, Windows. У кожного з цих проектів буде різний набір бібліотек та файлів, але їхня підтримка буде функціональною та паралельною для написання коду.

Xamarin надає можливість підключення реального девайсу для компіляції.

Xamarin.Forms містить в собі частину WPF (Windows Presentation Foundation) – інструмент для створення хороших рішень інтерфейсу. Ці частини складаються тільки з тих компонентів, що мають аналоги на інших платформах. Xamarin.Forms має максимально кросплатформенну основу. Писати код можна під любу систему, а Xamarin сам визначає, що потрібно запускати.

Основою любого проекту в Xamarin.Forms є MVVM (Model – View – View Model). Вона відділяє зовнішній вигляд інтерфейсу, який представляється користувачеві, від логіки алгоритмів коду програми.

Переваги MVVM :

1. Механізм сповіщення змін файлів, що потрібно оновити чи описати в інтерфейсі.
2. Відбувається прив’язка View Model до функцій, що дозволяє керувати ними самостійно та забезпечує упорядкованість.
3. Використання однієї і тієї самої моделі для різних екранів представлення. Тобто, модель представлення може бути показана через любий навігатор та

розділяти екран для користувачів. Це моделювання створюється незалежно від стану і поведінки екрану, не використовуючи шаблон системи.

4. Використання View Model економить час, бо не потрібно шукати інші шляхи реалізації.
5. Будь – яка кількість моделей може характеризувати будь – яку кількість представлень.

2.3 Файл маніфесту

Кожен проект повинен містити файл AndroidManifest.xml у кореневих файлах. Він описує важливу інформацію про сам додаток, інструменти збірки Android [3].

Файл маніфесту включає :

1. Дозволи, які необхідні для доступу захищених частин системи. Також, якщо в застосунок інтегруються інші програми і вони хочуть отримати дозвіл до вмісту, то вся додаткова інформація декларується у файл.
2. Ім’я пакета програми, ідентифікатор програми, апаратні та програмні функції, що вимагає програма, які пристрої можуть встановлювати програму з Google Play.
3. Компоненти програми такі, як : <activity>, <service>, <receiver>, <provider>. Якщо описати один з цих компонентів, але не вказати його у файлі маніфесту – програма не запуститься.
4. Icon та Label – це атрибуди, що використовуються для відображення певної піктограми або тексту.
5. Бібліотеки, що використовує застосунок також вказуються у файлі.
6. Описує мінімального рівня API Android, що необхідний для роботи додатку.

Файл AndroidManifest.xml розроблюваного застосунку зображено на рис. 2.2.

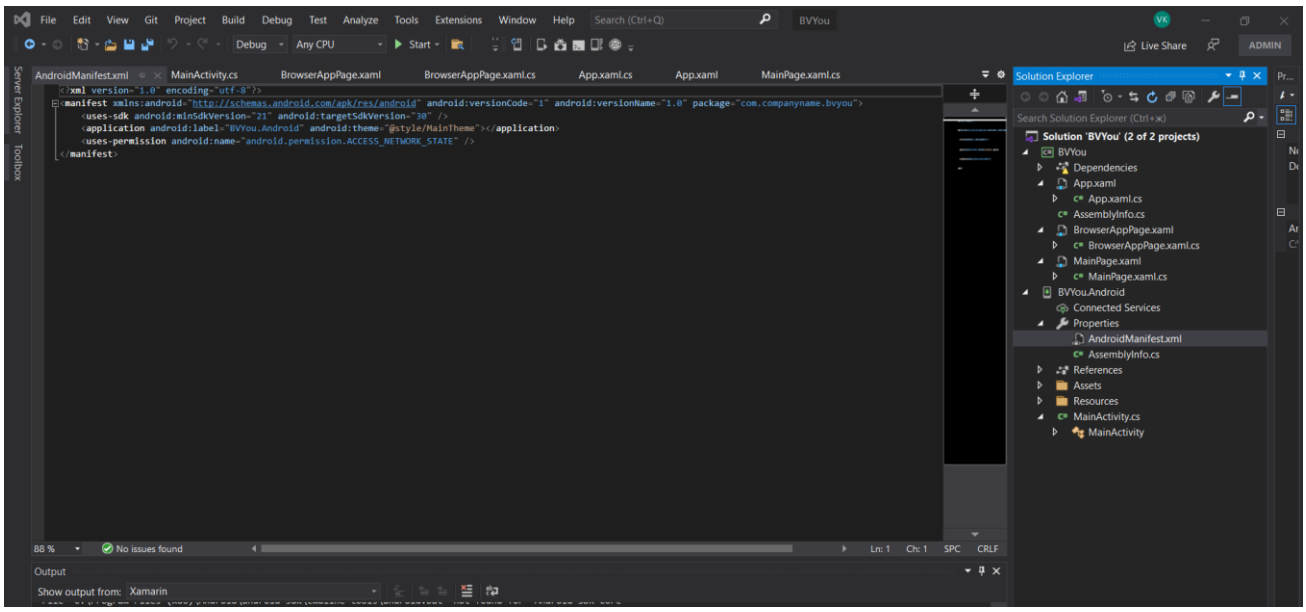


Рисунок 2.2 - Файл AndroidManifest.xml розроблюваного застосунку

2.4 Платформа автоматичної збірки проекту MSBuild

MSBuild(Microsoft Build Engine) – система автоматичної збірки додатків[4], надає змогу збірки для файлу XML способи обробки додатку. Visual Studio використовує MSBuild, але він від неї не залежить.

Основні принципи роботи MSBuild :

1. Якщо викликати файл msbuild.exe для проекту, то можна створювати, керувати, використовувати продукт у середі MSBuild без встановлювання середовища розробки Visual Studio.
2. Якщо Visual Studio використовується, то у проекті є файли з розширенням .csproj, .vbproj і т.д., то вони будуть містити у собі файли XML, що виконується при створенні проекту.
3. Також MSBuild можна викликати за допомогою комадного рядку чи консолі всередині Visual Studio.
4. Якщо код написаний в середовищі програмування Visual Studio, то збірку чи інтеграцію коду можна виконувати за допомогою платформи. Зазвичай це

характерно для великих проектів з декількома розробниками, що дає комфортне перенесення коду у власне середовище від іншого фахівця.

5. MSBuild має відкритий формат файлів проекту - це дозволяє багато разів використовувати певні правила збірки, які можна розкласти на інші файли для того, щоб збірки могли виконуватись в одному форматі.
6. Для того, щоб запустити збірку проекту на платформі потрібно в терміналі середовища програмування виконати наступне рис.2.3:

```
% SystemRoot% \ Microsoft.NET \ Framework \ v2.0.41115 \ MSBuild.exe
```

Рисунок 2.3 - Збірка проекту за допомогою платформи MSBuild

2.5 Вибір мови програмування

C# - об’єктно – орієнтована мова програмування [6], розроблена корпорацією Microsoft[5]. Дана мова програмування тісно співпрацює з Xamarin. А, як відомо з попередніх розділів, дане середовище програмування надає можливість створення додатків як для IOS, так і для Android.

Головною відмінністю виконування компіляції додатку являється машина Dalvik для Android та Ahead of time для IOS.

Dalvik – віртуальна Java машина, що записує файли додатку в байт код, що генерується в команди процесору уже при виконуванні застосунку, на відміну від Ahead of time, що компілюється перед виконуванням програми. Середовище розробки Xamarin добре бачить дану відмінність, тому надає різні компілятори для даних платформ.

Якщо зрівняти дві мови програмування для IOS та Android рис. 2.4, стає очевидним, що C# виглядає набагато сприйнятливіше, простіше та займає менше рядків коду.

Рисунок 2.4 – Порівняння мови Objective – C та C#

Також мова програмування C# була обрана за наступними характеристиками [9]:

1. Об’єктно – орієнтована мова програмування, що дозволяє спрощувати написання коду за рахунок багатої бібліотеки.
2. Сучасність – сучасна мова програмування на якій можна писати великі, потужні, надійні проекти.
3. Покращує безпеку програми. Код, що має доступ тільки до місця пам’яті програми та має дозвіл на її виконання.
4. Хороша сумісність, оскільки, C# код можна пов’язувати з іншими мовами програмування в межах розробки одного додатку – це дозволяє зменшити повторюваність коду, що збільшує ефективність процесу створення застосунку.
5. Структуризація полягає у тому, що мова програмування C# вимагає високий рівень логічної структури програми для кращого розуміння програмного коду.
6. Автоматичний збір сміття. C# автоматично збирає та стирає сміття за рахунок ефективної системи, що вже влаштована в програму.
7. Проблема витоку пам’яті відсутня, бо основною перевагою є сильне та велика резервне копіювання .

8. Підтримка Microsoft.
9. C# може розробляти різні програмі для IOS, Android, Windows Phone за допомогою Xamarin.
10. Методи в даній мові програмування за замовчуванням не є віртуальними.

2.6 XAML у середовищі Xamarin

XAML(Extensible Application Markup Language) – декларативна мова розмітки додатків на основі XML, що є частиною Microsoft. За допомогою певної структури та функцій відображає та структурує інформацію. Ціль створення даної мови є те, щоб зробити її універсальною та охопити функції і можливості, які рідко використовуються.

При створенні XAML файлу в середовищі Xamarin рис.2.5 завжди відтворюється файл з розширенням .xaml та файл C# з розширенням .xaml.cs. C# файл створюється для того, щоб опрацьовувати все, що містить файл XAML та бути до нього підкріпленим.

Після створення .xaml файлу, C# файл успадковується від Content Page. Клас, що знаходиться у цьому файлі має такий компонент, як InitializeComponent(), що викликається з конструктора. Без його ініціалізації код програми може працювати несправно, оскільки він бере на себе все, що було створено в xaml документі.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="BVYou.Droid.Page1">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Welcome to Xamarin.Forms!"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Рисунок 2.5 – Приклад XAML файлу за замовчуванням в середовищі програмування Visual Studio

Відомості про XAML :

1. В XAML класах створювання об’єктів відбувається за допомогою XML об’єктних елементів або атрибутів. Іноді потрібно присвоїти властивості тип об’єкту, який неможливо використати при звичайному структуруванні. XAML допомагає використати тег типу елемент – властивість, що дуже схоже на C# - ім’я класу та звертання через крапку до ім’я властивості.
2. XAML документ завжди повинен мати посилання на інший XAML файл, що використовується у програмі, або на Content Page.
3. XAML файл обов’язково повинен містити кореневий компонент - Content Page, бо він зберігає в собі простір імен, що використовується для XML та елементи і артибути, що вбудовані для реалізації мови XML, та простори імен та ім’я класу, які успадковуються від Content Page рис. 2.5.
4. Основним плюсом XAML являється те, що можна розділити інтерфейс та логіку сторінки, що є зручним у використанні, особливо, при розробці об’ємних мобільних застосунків.

2.7 Використання системи контролю версій Git та GitHub

Система контролю версій дозволяє розробникам чи студентам зберігати чи змінювати назви або сам код в систему з постійним доступом до неї. Наприклад, якщо виникла якась проблема, можна повернути робочу версію коду і не витрачати час на дебаг построково. Також система дає можливість доступу одразу декільком людям для зміни коду чи запиту, що є доволі зручною функцією, оскільки ті, хто мають доступ до даного проекту, отримають сповіщення на

пошту про новий коміт і одразу розуміють, що отримати відповідь або новий елемент у програмі.

Існує декілька типів системи контролю версій:

1. Локальні системи контролю версій. Можна копіювати файли в окрему директорію, використовуючи цю функцію як метод. Кожна зміна у файлі зберігається датовано і висока швидкість відновлення, за рахунок цієї особливості можна відновлювати файли, якщо попередній вихідний код програми перестав працювати вірно.
2. Централізовані системи контролю версій були створенні через проблеми зв'язку з іншими розробниками, які працювали в одному проекті. Системи мають один спільний сервер, який має всі версії та вітки репозиторіїв. При копіюванні репозиторію, який був видалений, відбувається повне копіювання даних. Також система дає можливість для розробки та збереження файлів для всієї команди. Також ваговим плюсом є швидкість роботи в ході роботи.
3. Розподіленні системи контролю версії. Клієнти використовують не просто скачування знімку файлу, а весь його репозиторій зі всіма комітами. Тобто, у кожного клієнта є повна копія всіх файлів і внесених до них змін. Великою перевагою являється те, що використовуючи розподіленні системи контролю версії можна співпрацювати із декількома віддаленими репозиторіями за рахунок чого, розробники однієї програми можуть працювати над декількома проектами без утрати файлів.

Git – розпроділенна система контролю версій, що дає IT – фахівцям відслідковувати зміни у файлах, бачити свою історію змін чи історію змін проекту над яким працюють також й інші розробники.

Переваги Git :

1. Open – source, який є безкоштовним для використання.
2. Швидкий для коміту та простий в експлуатації.
3. Резервне копіювання є єфективним для зберігання файлів.
4. Історія змін файлів доступна завжди та доступна для перегляду.

Сервіс репозиторіїв GitHub являється багатофункціональним в перегляді контролю версій, управлінням власного коду, задачами, багами. Також є Wiki для кожного окремо створеного проекту.

Отримання доступу Git – репозиторію, завантажений на GitHub, можна отримати за рахунок командного рядку чи команд.

Вітку можна використовувати, коли потрібно розділити блоки проекту чи створити інший закодований кусок роботи, що відрізняється від іншої вітки. Їх можна змінювати, видаляти, додавати, і також додавати їхні версії. Це робиться для того, щоб візуально розуміти, яка версія актуальна іншому розробнику на даний момент та швидко орієнтуватись рис. 2.6.

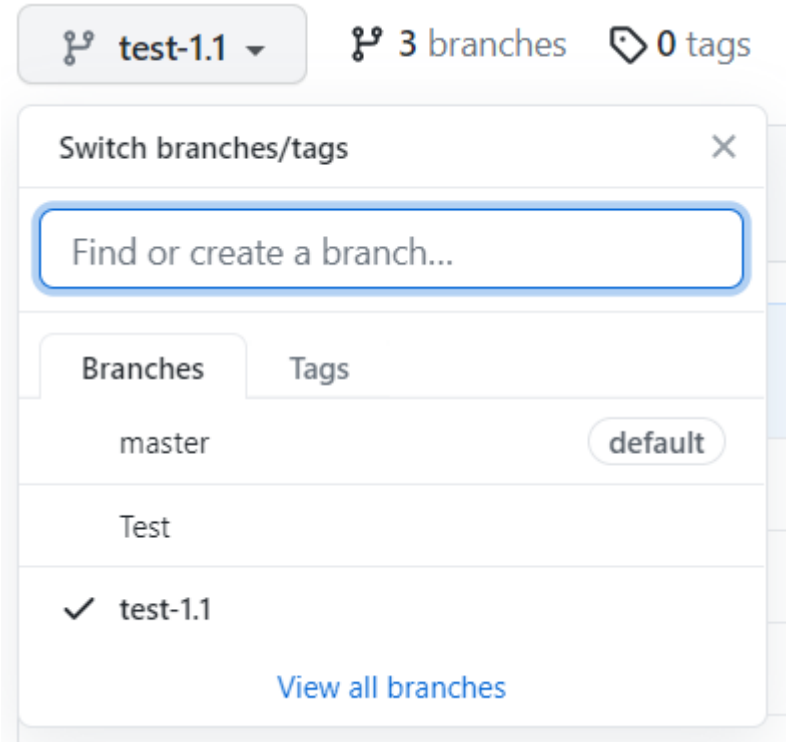


Рисунок 2.6 – Приклад використання віток в GitHub

GitHub має такі функції :

1. Документація.
2. Pull requests.
3. Історія комітів.
4. Повідомлення за допомогою email.
5. Графіки
6. Інтеграція багатьох сервісів
7. Встановлення версій
8. У сховище можна зберігати як і файли кодів, так і відео, зображення, аудіо та все, що пов’язано з проектом.

2.8 Використання платформи Lucid для створення макету дизайну додатку

У ході розробки застосунку також важливо використовувати хороший інтуїтивний інтерфейс, щоб заохотити користувача, а не тільки алгоритм рішення застосунку. Для цього, був обраний веб – сервіс Lucid, який широко використовується для будь – якої сфери застосування як і професійними фахівцями, так і звичайними користувачами. Приклад дизайну для застосунку «BVYou» зображений на рис. 2.7.

Рисунок 2.7 – Макет дизайну розроблюваного застосунку

Lucid – безкоштовний онлайн – сервіс для особистого використання. За допомогою цієї платформи можна створювати архітектуру, проектування, маршрутизацію, схематизувати робочі процеси, створювати діаграми, макети для застосунку.

3 МОДЕЛЮВАННЯ ТА РОЗРОБКА ЗАСТОСУНКУ

3.1 Постановка задачі та опис можливостей застосунку

На даний момент сучасні користувачі десктопів та гаджетів хочуть використовувати свій час весело чи навпаки задіяти його з метою науки. Часто ми опиняємось в непередбачуваних ситуаціях, коли немає мережі Інтернет, наприклад : поїзд, літак, ліфт. І кожну хвилину свого часу ми бажаємо провести з користю. Відео та аудіо файли допомагають нам у цьому. Кожного разу, як ми зустрічаємось з друзями чи колегами по роботі, обговорюємо фільми, серіали, музику – це завжди є хорошою та сприйнятливою темою для обговорення.

Завдання, яке має покривати застосунок полягає у тому, щоб надати можливість перегляду відеофайлів офлайн, скачавши їх з певної платформи.

Важливо розробити алгоритм, який буде реалізовувати алгоритм, що надає можливість відтворення відео у додатку та інтуїтивний інтерфейс, щоб швидко найти відео з назвою у списку та переглянути його.

Правильна, коротка та повномірна маршрутизація повинна конструювати інформацію так, щоб користувач не витрачав багато часу для того, щоб розібратись із функціоналом застосунку. Інтерфейс має бути націлений на цільову аудиторію любого віку.

Основні можливості функціоналу розроблюваного застосунку : обробка алгоритму скачування відеофайлів з сервісу YouTube, перегляд відео офлайн, відтворюваний список завантажених відео, інтуїтивний, сучасний інтерфейс без багатопотокового меню та його сторінок.

Можливості розроблюваного застосунку :

1. Користувач заходить в програму та одразу завантажується сервіс Youtube, де можна обрати відео шляхом прогортування сторінок чи пошук відео за допомогою сторінки пошуку.

2. Після того, як споживач вибрав відео з переліку запропонованих, натискає на файл відео, яке завантажується. Далі, потрібно натиснути кнопку «Зберегти», що дає можливість скачування відеофайлу на сторінку «Downloads».
3. Після того, як користувач натиснув кнопку завантажених відео, відеофайл, що він завантажував раніше, з’являється на цій сторінці. Таким чином, формується список всіх скачуваних відеофайлів користувача.
4. Також сторінка «Downloads» надає можливість перегляду відеофайлу одразу після того, як користувач натисне відео. Файл відображається у верхній частині екрану з відтворенням звуку та картинки відео.
5. Натискаючи на кнопку «PlayLists», споживач може сворювати власні плейлісти з обраними відео, які теж обираються із списку всіх завантажених ним відео.
6. Якщо користувач хотітиме повернутись на платформу скачування відеофайлів, потрібно натиснути кнопку «YouTube» та сервіс відкриється у веб відтворюванні.

Головна особливість додатку – відео з сайту завантажуються без авторизації та затримок за рахунок асинхронних процесів. Користуючись BVYou, кожен користувач має змогу додати бажане відео в пам'ять свого телефону, та одразу переглянути його без підключення до мережі інтернет. Оптимізація методів та швидкості завантаження відео, дає можливість за декілька секунд почати перегляд відео без реклами. Влаштований плейліст дає змогу відібрати улюблені файли та завжди мати до них швидкий доступ. Додаток звільняє користувача від обов'язкових фінансових витрат при стандартному процесі збереження відео з сайту, при цьому є найбільш швидким рішенням серед аналогів.

3.2 Розробка моделей застосунку

3.2.1 UML діаграма діяльності

UML (Unified Modeling Language) [7] – уніфікована мова програмування для побудови будь – яких бізнес – процесів і розробки додатків [7]. Використовується на кожному процесі життєвого циклу. UML підтримує багато видів діаграм, такі як:

1. Class Diagram.
2. Package Diagram.
3. Component Diagram.
4. Deployment Diagram.
5. Collaboration Diagram.
6. Object Diagram.
7. Composite Structural Diagram.

Мова UML являється простим концептуальним рішенням для розробки різних застосунків. Процес об’єктно – орієнтованого аналізу з використанням всіх можливостей включає одну важливу особливість : включати в діаграму потрібно тільки ті окремі моделі, які є об’ємними та важливими у порівнянні з тими, що також існують у проекті, тільки вважаються другорядними, щоб не навантажувати процес аналізу.

Діаграма діяльності – це певний набір правил для опису основної частини логіки функціоналу застосунку рис. 3.1. Діаграма показує послідовність подій, які реалізовують загалом певну програму. Це своєрідна піраміда подій, що базуються на відкритті основної концепції застосунку. Діаграма діяльності відноситься до логічної моделі, оскільки візуалізує особливості реалізації класів та при необхідності показати алгоритми їхнього виконання.

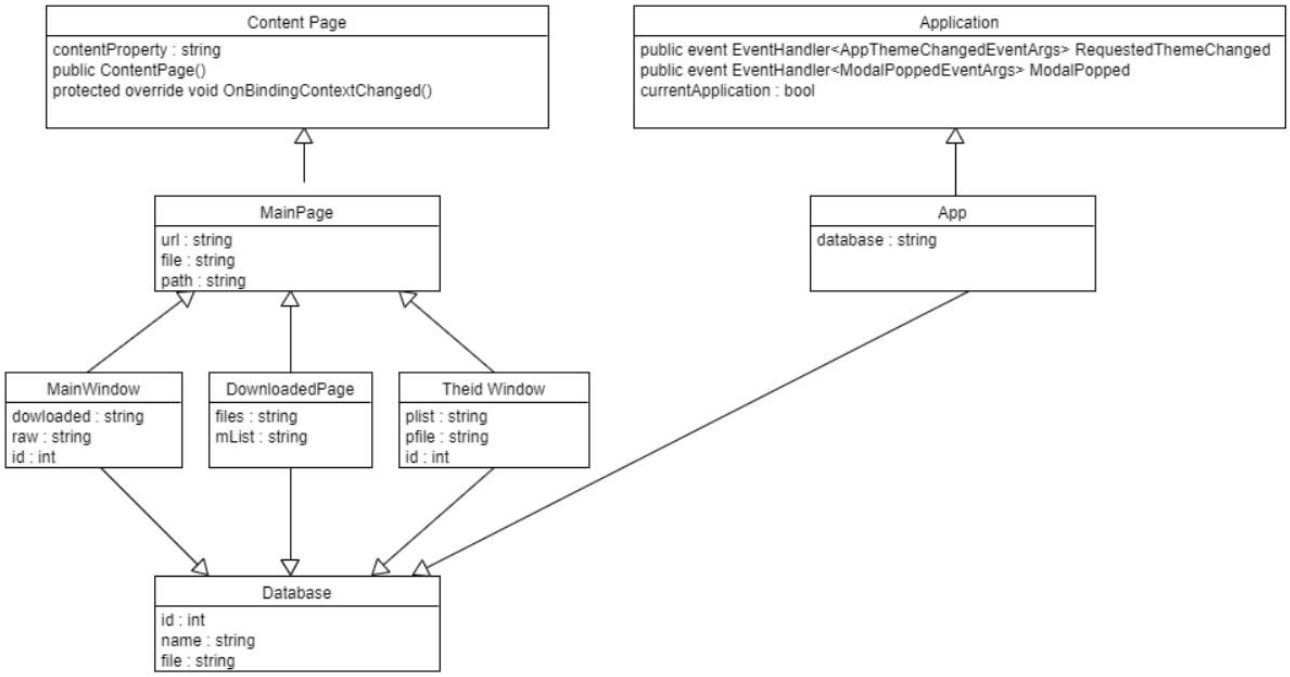


Рисунок 3.1 - UML діаграма діяльності

3.2.2 UML діаграма використання

Діаграма використання (Use case або прецедентів) створюється для того, щоб відобразити функціонал додатку зі сторони користувача. Це певний аналіз функціональних дій з його сторони. З боку головної особи, функції поділені на групи чи підгрупи, що дозволяють швидко орієнтуватись при використанні додатку. Кожна з цих груп чи сторінок повинна дати користувачеві певний результат після взаємодії з ними. Дизайн застосунку роблений так, щоб було інтуїтивно зрозуміло для людини будь – якого віку. У діаграмі використання має бути зображений актор та функціональні вимоги.UML діаграма використання зображена на рис. 3.2.

UML діаграма використання грунтується на таких правилах :

1. Абстрагування : у діаграму потрібно включати тільки ті елементи функціоналу програми, яке мають до неї безпосереднє відношення, тобто, виконують певний відрізок складу застосунку.

2. Багатомодельність : одна діаграма не може точно описати весь функціонал програми, який має користувач, тому можна зображувати його однією залежністю, яка включає в себе декілька уявлень, кожен з яких відображає певну поведінку додатку.
3. Ієрархічність : описуючи певний застосунок з боку користувача, можна використовувати різний рівень деталізації та об’єму. За рахунок цього, перші рівні діаграми описують загальні характеристики її принципів, в той час, як наступні рівні розкривають функціональні рівні та результати, які бачить та використовує споживач.

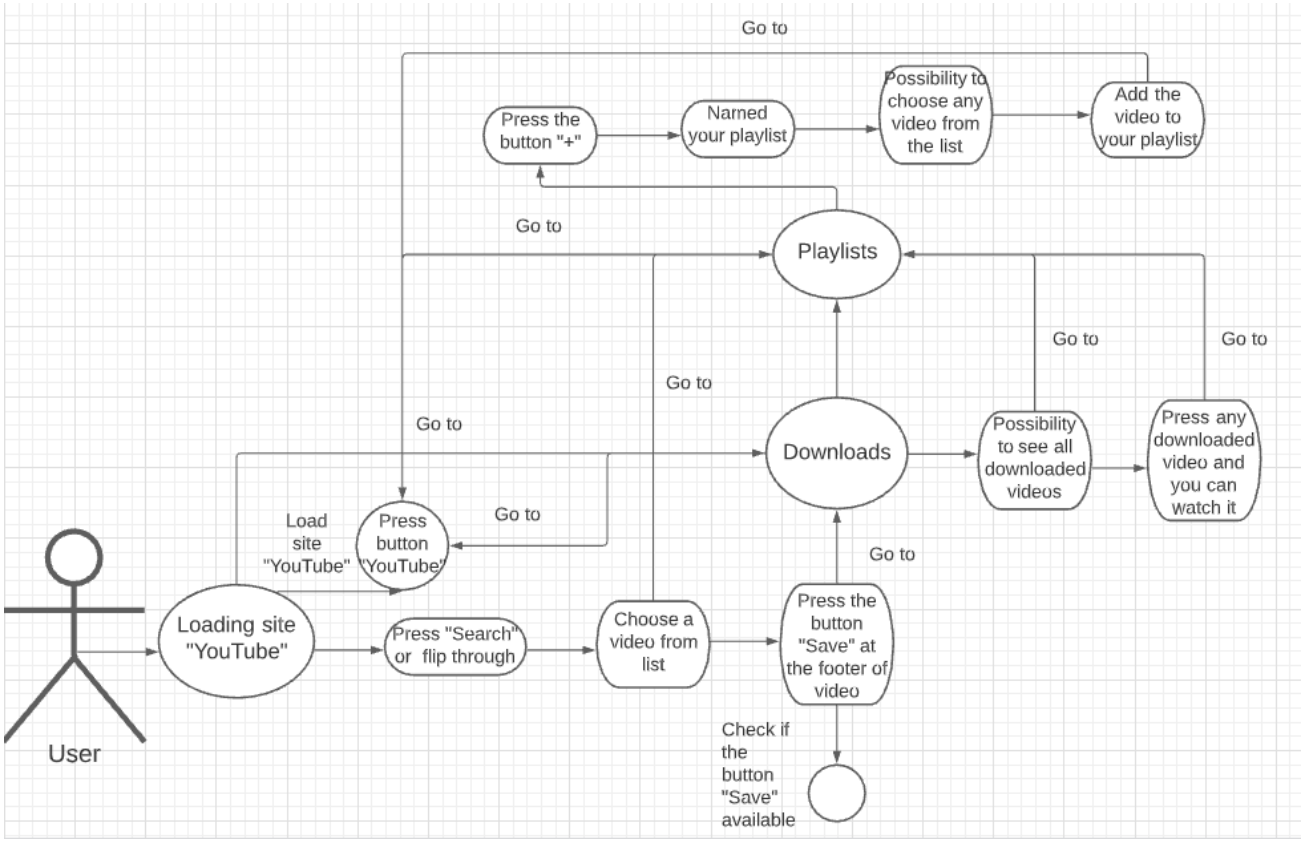


Рисунок 3.2 - UML діаграма використання

3.3 База даних SQLite

SQLite – це влаштована бібліотека, яка написана на мові C, яку можна підключати до своєї програми[8].

СУБД SQLite – це БД, яка може напряму зв’язуватись з файлами програми, оскільки вона налаштована так, що самостійно не потрібно збирати її в власній системі.

Особливості SQLite :

1. БД працює без окремого процесу серверу.
2. Підтримує більшість функцій мови SQLite.
3. Повна база даних зберігається на одному диску.
4. Являється автономною, тобто , без без зовнішніх залежностей.
5. Не потребує налаштувань у власній системі, оскільки надається з нульовою конфігурацією.
6. Підтримує тільки чотири тип даних : integer, real, text, blob.

Головне вікно бази даних SQLite зображено на рис. 3.3.

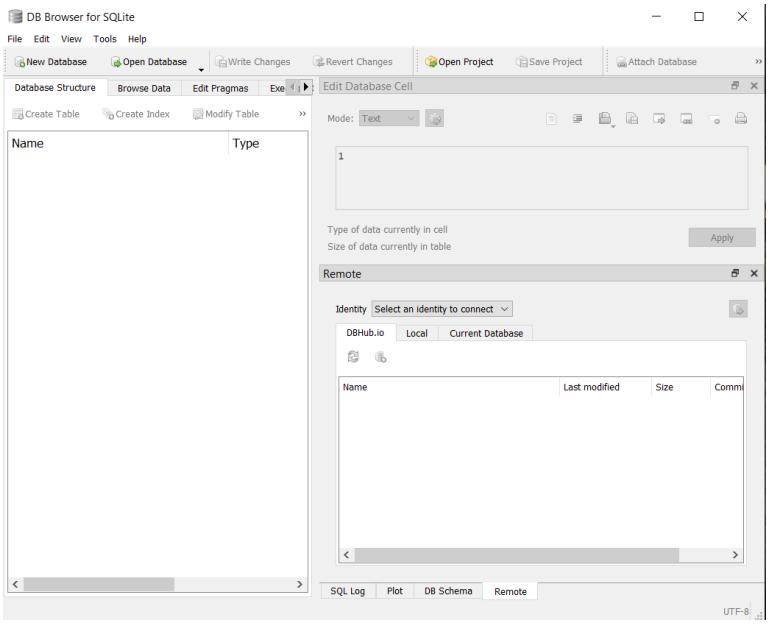


Рисунок 3.3 – Головне вікно СУБД SQLite

Для того, щоб завантажити пакети для БД SQLite у свій застосунок, потрібно в середовищі розробки Xamarin зайти в Tools > NuGet Package Manager > Manage NuGet Packages for Solutions > Browse та інсталювати бібліотеки для роботи з SQLite sqlite – net – pcl та system.data.sqlite. Після чого, потрібно завантажити програму DB Brower for SqLite, яка дозволить створити базу даних SQLite та завантажити її у середовище розробки Visual Studio [14]. При відкритті програми DB Brower for SqLite, база даних SqLite створилась сама. Дальше потрібно дати ім’я базі даних, зберегти її та створити таблиці рис. 3.4., в яких будуть зберігатись відеофайли та всі дані, що потрібні для них.

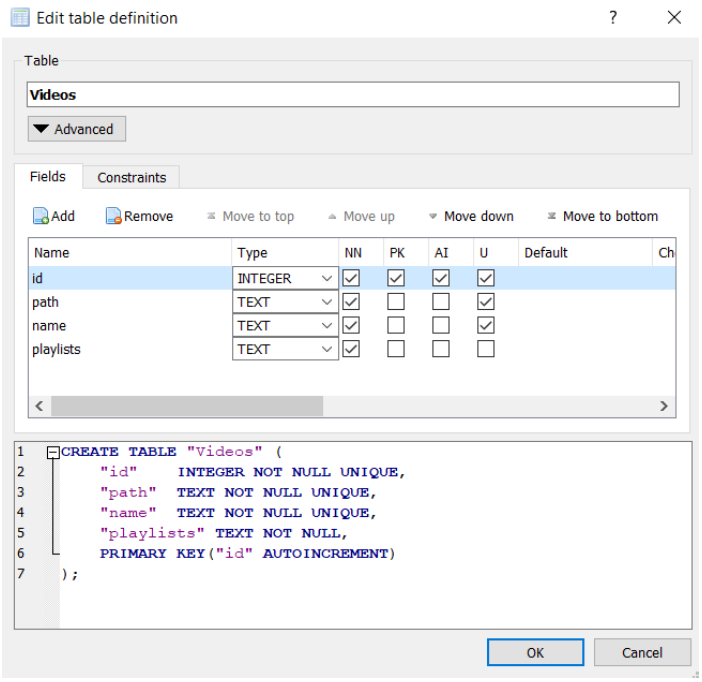


Рисунок 3.4 – Таблиця Videos

Створення таблиць відбувається наступним чином : Create Table > Name of Table. Перша таблиця має назву Videos. Для того, щоб додати поля в таблицю потрібно натиснути у полі Fields кнопку «Add» та заповнити поля : ім’я поля, тип даних, primary key, автоінкремент та унікальність. У таблиці videos є такі поля : id, path, name, playlists.

У таблиці Playlists є такі поля : id, name, videos, для яких також потрібно встановити ім’я поля, тип даних, primary key, автоінкремент та унікальність. У таблиці Videos та таблиці Playlists полю id належить primary key, тому що поле id являється унікальним для двох таблиць, бо кожне відео, яке буде завантажувати користувач має власний унікальний номер. Таблиця Playlists зображена на рис. 3.5.

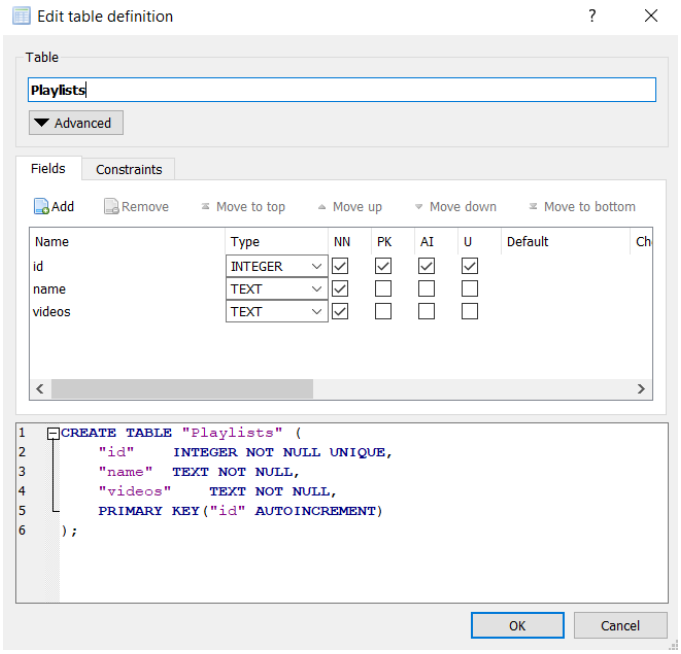


Рисунок 3.5 – Таблиця Playlists

Далі потрібно підключити БД до розроблюваної системи рис. 3.6.Для того, щоб відеофайли, які завантажує користувач, зберігались у базі даних, потрібно створити окремий клас, у моєму випадку - SampleData, який буде говорити програмі в яке поле та з якими даними зберегти відео та додаватиме його у список в базу даних SQLite.

```
public class SampleData
{
    private SQLiteAsyncConnection database;
    private int video_counter=0;
    private int playlist_counter = 0;
    1 reference
    public SampleData(string dPath)
    {
        database = new SQLiteAsyncConnection(dPath);
        database.CreateTableAsync<Video>().Wait();
        database.CreateTableAsync<PlayList>().Wait();
    }
```

Рисунок 3.6 – Клас SampleData

У класі Video описано всі властивості, які передаються в БД SQLite у таблицю Videos рис. 3.7.

```
public class Video
{
    11 references
    [SQLite.PrimaryKeyAttribute, SQLite.AutoIncrement] public int Id { get; set; }

    7 references
    public string Name { get; set; }
    4 references
    public string Path{ get; set; }
    6 references
    public string Playlists { get; set; }
}
```

Рисунок 3.7 – Клас Video

У класі SampleData створена функція додавання плейліста в БД рис. 3.8.

```
public async Task<List<PlayList>> AddNewPlayList(string name)
{
    await database.InsertAllAsync(new PlayList[]
        {
            new PlayList{Id=playlist_counter, Name = name, Videos = ":"}
        });
    playlist_counter++;

    return await database.Table<PlayList>().ToListAsync();
}
```

Рисунок 3.8 – Функція AddNewPlaylist, що записує плейліст в БД

За допомогою функції AddVideoToPlayList(), йде звертання до двох таблиць : відео та плейлісти, далі за допомогою функції GetObject(), обираються конкретні об’єкти відео та плейліста по id, після чого редагуються їхні зв’язки та вносяться зміни до БД.

```
public async Task<List<PlayList>> AddVideoToPlayList(string video_id,string playlist_id)
{
    var videos = await database.Table<Video>().ToListAsync();
    var playlists = await database.Table<PlayList>().ToListAsync();

    Video video = GetObject(videos, video_id);
    PlayList playlist = GetObject(playlists, playlist_id);

    video.Playlists += playlist_id + ":";
    playlist.Videos += video_id + ":";

    await database.UpdateAsync(video);
    await database.UpdateAsync(playlist);
    return playlists;
}
```

Рисунок 3.9 – Функція додавання списку відео в плейліст в БД

3.4 Розробка дизайну застосунку за допомогою розширюваної мови розмітки XAML

XAML (eXtensible Application Markup Language) – мова розмітки, що дозволяє розділи логіку коду та візуальний вигляд сторінки.

У середовищі Visual Studio створено 2 файли – MainPage.xaml рис.3.10. та App.xaml, які в піраміді мають ще файли .xaml.cs, в яких описана логіка роботи цих сторінок. У файлі MainPage.xaml написаний Grid (макет, що дозволяє розділити сторінку та вертикальні та горизонтальні лінії), щоб розділити головне вікно додатку для перегляду відео. Також підключений простір імен для того, щоб використовувати елементи управління та бібліотека для відтворення відео, опис кнопок(ширина, висота і т.д.)

Рисунок 3.10 – Файл MainPage.xaml

MainPage.xaml наслідується від ContentPage (візуальний елемент, який відображає певне представлення на сторінці). Кожен файл .xaml використовує свій візуальний елемент.

У файлі App.xml вказано простір імен, фоновий малюнок, його кольори, тобто, все, що належить основним функціям застосунку рис. 3.11.

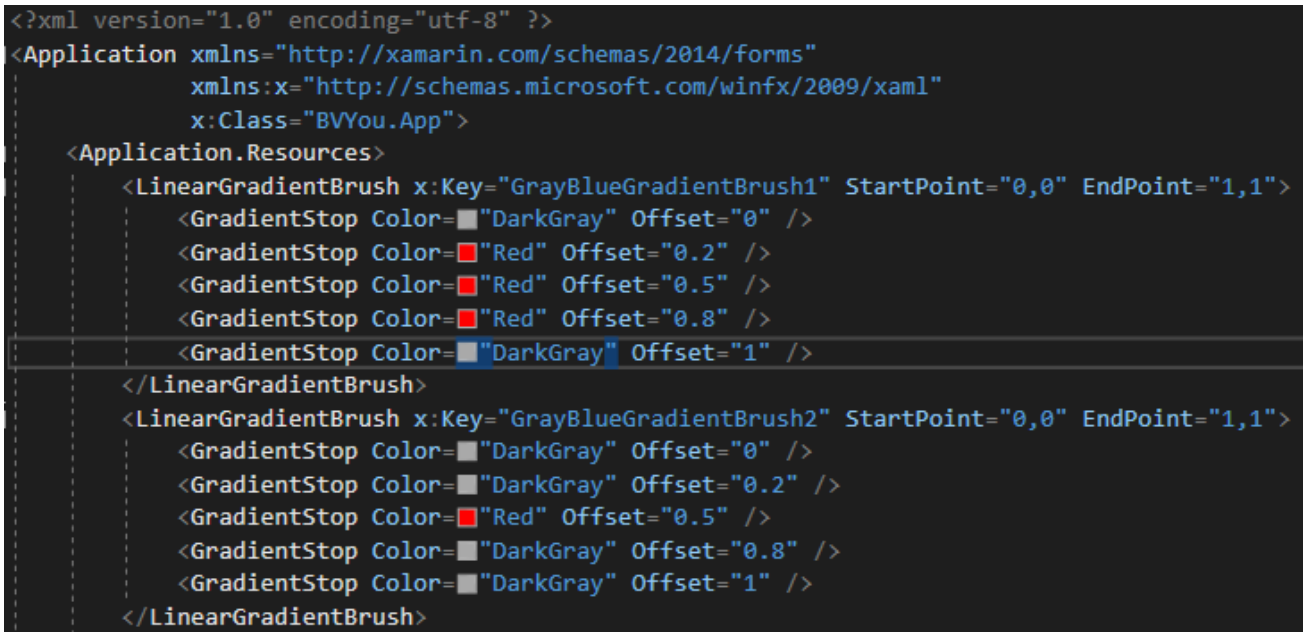


Рисунок 3.11 – Файл App.xml

3.5 Компілювання програми за допомогою емулятора середовища програмування та застосунку Vysor

У Visual Studio можна компілювати додаток через емулятор рис. 3.12.

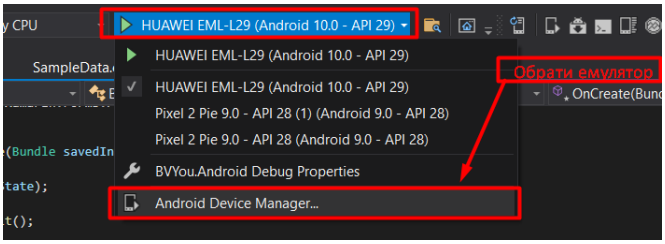


Рисунок 3.12 – Обрати емулятор у середовищі Visual Studio

І також через реальний телефон. Для того, щоб запускати застосунок через емулятор, потрібно відкрити поле емулятору на натиснути Android Device Manager.

Після чого, обрати емулятор, який пропонується чи додати новий з вибором характеристик, розширень та версій та інсталювати його у середовище програмування. На рис. 3.13. зображено компілювання додатку в емуляторі Pixel 2 Pie 9.0.

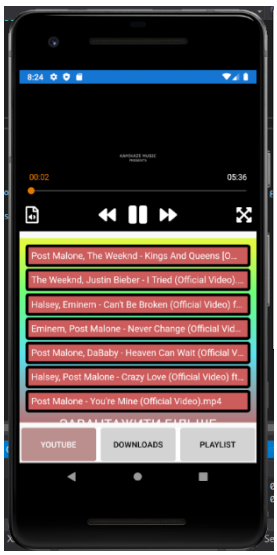


Рисунок 3.13 – Сторінка Downloads застосунку «BVYou»

Для того, щоб запустити додаток через реальний телефон потрібно завантажити програму Vysor (програма для тестування застосунків). Після підключення смартфону до ПК, потрібно підключити його в додатку, та нажати кнопку «Play». Після чого, відбудеться відкриття нового модального вікна з інтерфейсом реального смартфону рис. 3.14.

Переваги Vysor :

1. Має влаштований функціонал запису екрану та скріншот.
2. Простий в налаштуванні та використанні.
3. Можливість керувати девайсом, що підключений, з комп’ютера.
4. Стабільність роботи додатку.
5. Можна використовувати повноекранний режим з власного комп’ютера.

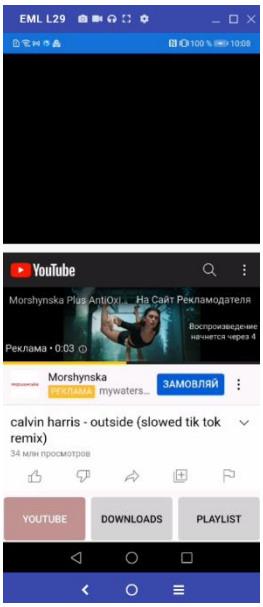


Рисунок 3.14 – Вигляд застосунку «BVYou» через реальний сматфон

Перед тим, як компілювати програму потрібно обрати власний компілятор телефону рис.3.15.

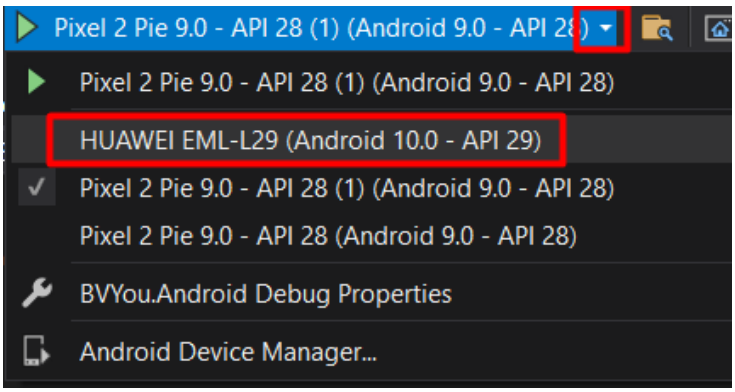


Рисунок 3.15 – Вибір власного девайсу для компіляції програми

Після того, як програма компілюється відбувається документація логів у середовищі розробки Visual Studio автоматично у вкладці Debug. Туди записується повний поетапний процес компілювання застосунку, натискання кнопок та все, що робить користувач чи завантажує програма, чи, навпаки, помилки, які були оброблені під час запуску додатку. На рис. 3.16. зображено вкладку Debug та процес компілювання додатку.

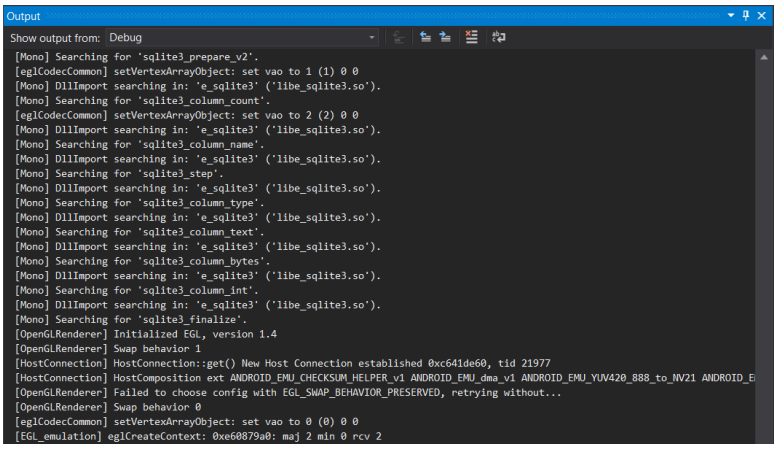


Рисунок 3.16 – Компіляція застосунку «BVYou»

3.6 Тестування застосунку

Test Case – це документ, який засвідчує використання певний дій, які необхідні для того, щоб тестувати застосунок. Цей документ складається з трьох

аспектів : умова, опис тесту, очікуваний результат [15]. У даних таблицях представлено сторінки застосунку, що будуть тестуватись, які умови вони повинні використовувати, також представлені описи тестів, при яких користувач має виконувати крок за кроком, як вказано в тесті. Очікуваний результат означає, що програма при виконуванні умов та описі кроків переходів по програмі, повинен виконувати дані умови тесту, що передбачує логіка програми.

Тестування застосунку являється важливим спектром роботи з програмою, оскільки вона повинна виконувати всі умови, які передбачені логікою алгоритмів додатку. Застосунок, повинен працювати коректно та без збоїв для того, щоб користувач міг насолоджуватись користуванням програмою та зупинився свій вибір саме на цьому додатку, а не на аналозі. Для цього і створюються тест кейси, де можна перевірити розроблюваний функціонал програми та зрозуміти зі сторони користувача, як він буде це бачити та чи буде користуватись даним застосунком.

У таблиці 3.1 описано дві сторінки : YouTube(головна сторінка застосунку), та Downloads(там, де з’являються відео після завантаження їх користувачем). Також перелічений результат після того, як споживач буде користуватись певною функцією додатку.

Таблиця 3.1 – Функціонал завантаження відео

Передумови : завантажити застосунок

Умова	Опис тесту	Очікуваний результат
Відеофайл з’явився на сторінці «Downloads» .	1. На сторінці «YouTube» обрати відео через пошук чи скрол. 2. Натиснути на кнопку «Зберегти». 3. Сторінка «YouTube» перезавантажилась.	Відеофайл, який обрав користувач з’явилось на сторінці «Downloads».

Кнопка «YouTube» клікабельна.	1. Натиснути на кнопку «Downloads». 2. Перейти на сторінку «YouTube»	Сервіс «YouTube» відкрився при натисканні на кнопку «YouTube».
Відео відтворюється.	1. Завантажити відео або вибрати із списку на сторінці «Downloads». 2. Натиснути на відеофайл.	Відеофайл відтворюється в першій половині екрану на сторінці «Downloads».
Перехід на сторінку «YouTube».	1. На сторінці «Downloads» після завантаження одного чи багатьох відеофайлів .користувачем з'являється кнопка «Завантажити більше». 2. Натиснути на кнопку «Завантажити більше».	Кнопка «Завантажити більше» зробила перехід на сторінку «YouTube».

У таблиці 3.2 описано сторінку Playlists, її можливості та те, як буде реагувати програма після певних дій користувача. Що він може бачити на сторінці та як користуватись певним функціоналом, що там зображений.

Таблиця 3.2 – Тестування функціоналу сторінки «Playlists»

Плейліст створений	1. Після завантаження відео користувачем перейти на сторінку «Playlists». 2. Натиснути на кнопку «+». 3. Дати ім'я плейлісту. 4. Обрати зі списку відео, відеофайл, який потрібно додати у плейліст. 5. Натиснути на плейліст та видалити відео з плейліста.	1. Плейліст сворений. 2. Ім'я плейліста відображається на сторінці «Playlists». 3. Відеофайл додався у плейліст. 4. Відео з плейліста видалено.

4 МАРШРУТИЗАЦІЯ КОРИСТУВАЧА

Для того, щоб почати роботу із застосунком необхідно скачати його з PlayMarket. Після завантаження додатку відкривається сторінка з сервісом YouTube рис.4.1.

Рисунок 4.1 – Головна сторінка додатку «BVYou»

Для того, щоб завантажити відео, користувачу потрібно обрати будь – яке відео із запропонованих чи через пошук, натиснути на нього. Після чого користувач бачить кнопки : лайк, дізлайк, поширити, зберегти, поскаржитись.

Коли користувач натисне на кнопку «Зберегти» - сторінка «YouTube» оновлюється, відеофайл відтворюється заново. Після чого, відео, що завантажував користувач з’явиться на сторінці «Downloads» рис. 4.2.

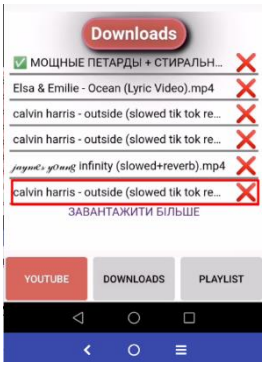


Рисунок 4.2 – Відеофайл, що завантажив користувач

Далі, якщо натиснути на кнопку «Завантажити більше» - йде перехід на сторінку «YouTube». Також перехід можна зробити, натиснувши на кнопку «YouTube» і застосунок автоматично завантажить нову сторінку.

Для відтворення відеофайлу потрібно натиснути на назву відео рис.4.3. і воно відтвориться у верхній частині екрану. Також, якщо натиснути на кнопки «Завантажити більше», «Downloads», «Youtube», «Playlists», відео також буде відтворюватись у верхній частині екрану рис.4.4.

Це дає можливість користувачу переключатись між сторінками в той час, як відтворюється відео, що він обрав раніше.

Також, в той час, поки він дивиться відео, споживач може обирати наступне відео для перегляду чи скачування.

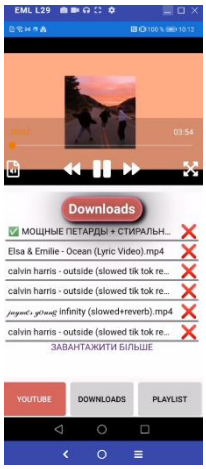


Рисунок 4.3 – Відтворення відеофайлу, що обрав користувач із списку

Приклад відтворення відео на інших сторінках додатку на рис.4.4.

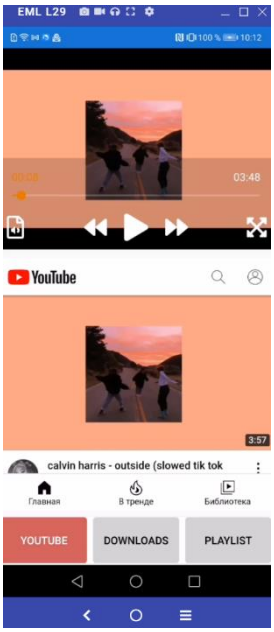


Рисунок 4.4 – Відтворення відео на інших сторінках додатку

Функціонал сторінки «Playlists» зображений на рис.4.5.

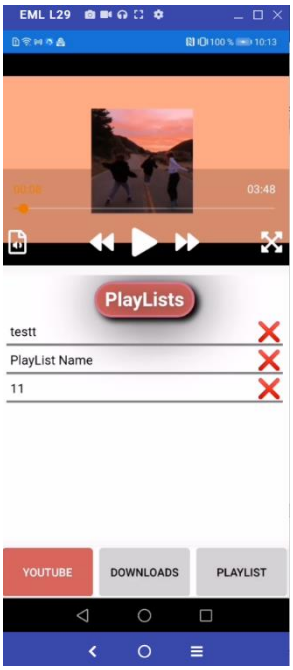


Рисунок 4.5 - Сторінка «Playlists»

На рис. 4.6 зображено можливості сторінки «Playlists» : редагування плейліста : відео. Користувач може додати відео зі списку всіх відео до плейліста, видалити плейліст чи додати його. Також видалити відео зі списку плейліста чи обрати інше відео (всі дані зберігається до БД : додавання плейліста, додавання відео зі списку, видалення плейліста та видалення відео з плейліста).

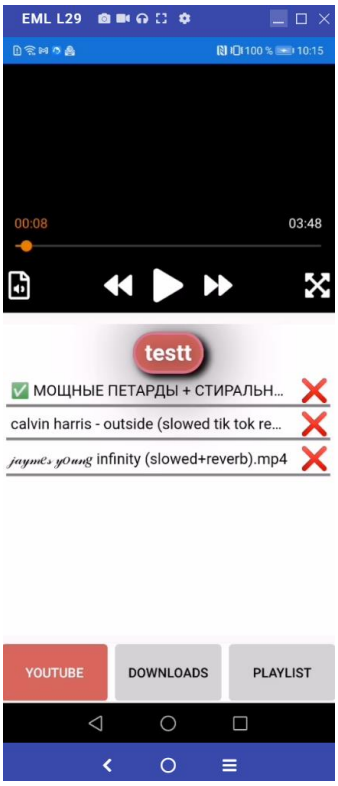


Рисунок 4.6 – Редагування плейліста

ВИСНОВКИ

У дипломній роботі розроблявся алгоритм для завантажування відеофайлів з сервісу Youtube та їх перегляд офлайн.

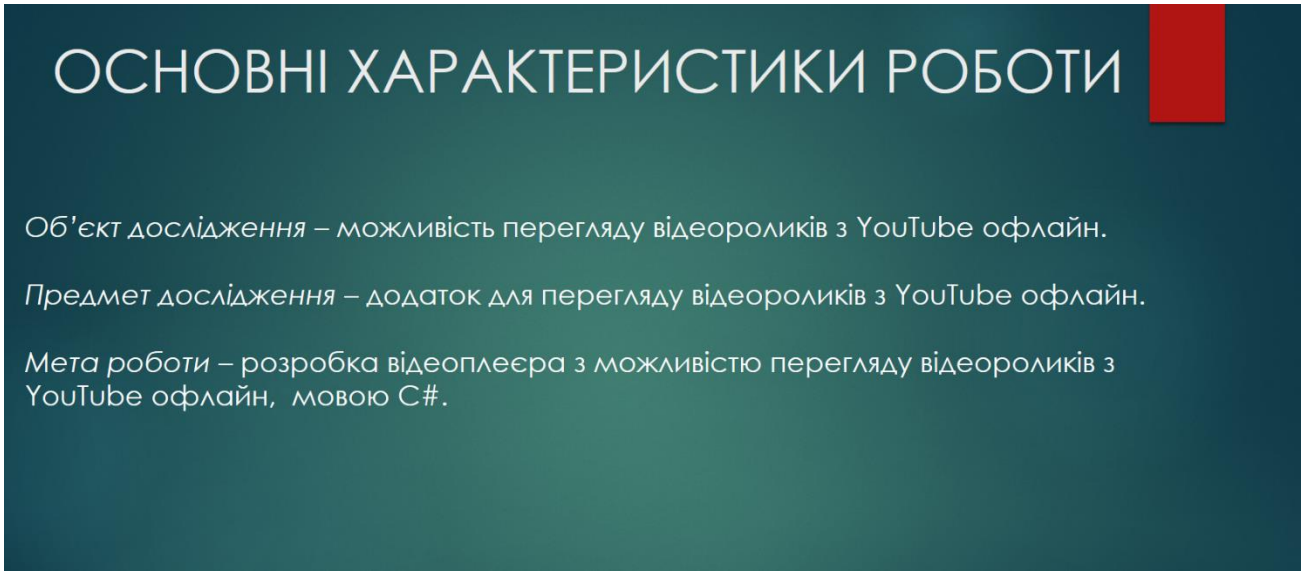
1. Проаналізувавши актуальність проблеми та попит цільової аудиторії на даний вид застосунків, зроблений висновок про необхідність розробки застосунку «BVYou».
2. Досліджено аналоги – їх недоліки та переваги, побудувавши порівняльну таблицю, визначено вимоги до платформи, маршрутизацію додатку, дизайн.
3. Враховані формати файлів, які буде чи захоче завантажувати користувач, вид та вигляд ресурсу, аналіз того чи потрібно використовувати авторизацію.
4. Після аналізу інструментів : БД, мови програмування, середовища розробки проведено огляд доцільності використання саме цих платформ. Досліджено, що реалізація на таких платформах : СУБД – SQLite, мова програмування – C#, середовище розробки – Visual Studio, вміщують комплект всіх переваг та не поступаються швидкістю, автоматизацією та функціональністю інших сучасних рішень.
5. Встановлено попит функцій, що виконує застосунок. Головною перевагою якого вважається мінімальна маршрутизація додатку з максимальною інтуїтивністю, що доступна користувачу.
6. Виконано тестування застосунку на емуляторі Visual Studio та на реальному телефоні. Визначено, що додаток відповідає всім вимогам, що були встановлені раніше та відтворює логіку та модель елементів програми на різних девайсах.

СПИСОК ЛІТЕРАТУРИ

1. Gs.statcounter.com Mobile Operating System Market Share Wordwide [Електронний ресурс] : [Інтернет портал]. - Електронні дані. - [Mobile Operating System Market Share Wordwide] – Режим доступу : https://gs.statcounter.com/os-market-share/mobile/worldwide.
2. Xamarin The good and bad of Xamarin Mobile Development [Електронний ресурс] : [Інтернет портал]. – Електронні дані. – [Altex Soft] – Режим доступу : https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/.
3. David Britch. «Enterprise Application Patterns using Xamarin.Forms», 2017. - С. 36 – 44.
4. Manifest App Manifest Overview[Електронний ресурс] : [Інтернет портал]. - Електронні дані. - [Google Developers] – Режим доступу : https://developer.android.com/guide/topics/manifest/manifest-intro.
5. Sayed Ibrahim Hashimi. «Inside the Microsoft Build Engine: Using MSBuild and Team Foundation Build», 2009. – С. 3 – 10.
6. Naveed Zaman, Sam Hobbs. «Objected Oriented Programming Using C#», 2013. – С. 13-18.
7. Booch, Rambaugh, Jacobson, Fowler, Brown. «Static Modeling using the Unified Modeling Language (UML)», 1999. – С. 66.
8. Sibsankar Haldar. «SQLite Database System. Design and Implementation», 2015. – С. 74 – 83.
9. Marino Posadas. «C# and .Net Framework», 2016. - С. 127 – 156.
10. Ockert J. du Preez. «Visual Studio 2019 in Depth», 2019. - С. 18-24.
11. Wesley Sweetser, Jr. «Visual Basic for the Approved WorkMan», 2018. – С. 17 – 23.
12. Snider Ed. «Mastering Xamarin.Forms», 2016. – С. 25 – 58.

Додаток А


Дипломна робота
«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПЕРЕГЛЯДУ ВІДЕО ОФЛАЙН З YOUTUBE МОВОЮ C#»
Виконав:
Куцук В.А.
Керівник:
Гребенюк В.В.
Київ 2021


ОСНОВНІ ХАРАКТЕРИСТИКИ РОБОТИ
Об'єкт дослідження – можливість перегляду відеороликів з YouTube офлайн.
Предмет дослідження – додаток для перегляду відеороликів з YouTube офлайн.
Мета роботи – розробка відеоплеєра з можливістю перегляду відеороликів з YouTube офлайн, мовою C#.

Порівняльна характеристика аналогів

Назва застосунку	Переваги	Недоліки
MX Player	1. Підтримка багатьох відеоформатів. 2. Історія переглядів.	1. Підтримка реклами. 2. Онлайн використання.
XPlayer	1. Простота використання. 2. Модальне вікно з інструкцією відтворення відео.	1. Перенасичений рекламою. 2. Відсутність функціоналу перевірки знань.
BSPLayer	1. Широке меню функцій. 2. Створення різних плейлістів та вподобаних відео.	1. Немає підтримки багатьох відеоформатів.
KMPlayer	1. Новітній дизайн додатку. 2. Підтримка субтитрів.	1. Онлайн додаток. 2. Недосконала система перегляду відео через URL - формат .

Опис Visual Studio

Visual Studio – середовище, що дуже добре співпрацює з C#, є безкоштовна версія Community Edition, де є все, що потрібно розробнику.
Порівняно з Eclipse та Project Rider, у середовище Visual Studio влаштовані всі потрібні функції та інструменти для розробника, що створює певний продукт. Влаштована хмара також дає перевагу даному середовищу та Xamarin, що являється Framework та має всі сучасні функції мови C#, надійні бібліотеки, якими можна користуватись при розробці.
Також Visual Studio є безкоштовним середовищем прогрмаування.
Основним плюсом розробки на Xamarin являється те, що продуктивність Framework постійно покращується і хоче відповідати оновленим стандартам розробки [2]. Платформа пропонує комплексне рішення, щоб слідкувати за рішенням питань тестування додатку.

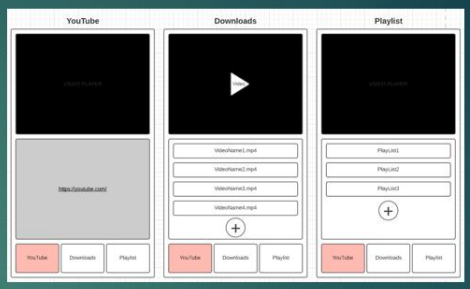
Переваги СУБД SQLIte:

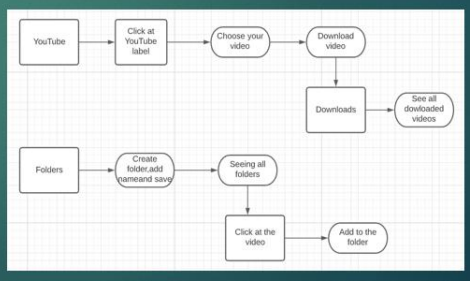
SQLite - це вбудована кроссплатформенная БД, яка підтримує досить повний набір команд SQL і доступна у вихідних кодах (на мові С).

- ✓ SQLite можна скомпілювати самому
- ✓ Тексти програм SQLite знаходяться в public domain, тобто взагалі ніяких обмежень на використання.
- ✓ Доступна консольна утиліта для роботи з базами (sqlite3.exe, «a command-line shell for accessing and modifying SQLite databases»).
- ✓ Легко переносити: за замовчуванням, БД - це один файл (в кроссплатформенную форматі)
- ✓ Тип стовпчика не визначає тип значення, що зберігається в цьому полі записи, тобто в будь-який стовпець можна занести будь-яке значення;

Макети застосунку

Макети застосунку створені за допомогою декларативної мови розмітки – XAML. Вона допомагає відтворити певні розділи програми на вигляд та логіку, що допомагає структурувати файли у самій програмі. Файли .xaml наслідуються від Content Page, що надає можливість використовувати певні атрибути, елементи, простори імен класів та імена класів, що використовуються в XML. На рисунку зображено основну маршрутизацію додатку : Page 1, Page 2, Page 3 та Use Case до макетів





Висновки

1.Проаналізувавши актуальність проблеми та попит цільової аудиторії на даний вид застосунків, зроблений висновок про необхідність розробки застосунку «BVYou».
2.Досліджено аналоги – їх недоліки та переваги, побудувавши порівняльну таблицю, визначено вимоги до платформи, маршрутизацію додатку, дизайн.
3.Враховані формати файлів, які буде чи захоче завантажувати користувач, вид та вигляд ресурсу, аналіз того чи потрібно використовувати авторизацію.
4.Після аналізу інструментів : БД, мови програмування, середовища розробки проведено огляд доцільності використання саме цих платформ. Досліджено, що реалізація на таких платформах : СУБД – SQLite, мова програмування – C#, середовище розробки – Visual Studio, вміщують комплект всіх переваг та не поступаються швидкістю, автоматизацією та функціональністю інших сучасних рішень.
5.Встановлено попит функцій, що виконує застосунок. Головною перевагою якого вважається мінімальна маршрутизація додатку з максимальною інтуїтивністю, що доступна користувачу.
6.Виконано тестування застосунку на емуляторі Visual Studio та на реальному телефоні.
7.Визначено, що додаток відповідає всім вимогам, що були встановлені раніше та відтворює логіку та модель елементів програми на різних девайсах.