

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської кваліфікаційної роботи  
на ступінь вищої освіти бакалавр

**на тему: «Розробка додатку для контролю парі під систему Android з  
використанням C#.net»**

Виконав: студент 4 курсу, групи ПД– 41  
спеціальності

121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

Сачик М.О.  
(прізвище та ініціали)

Керівник Дібрівний О.А.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Нормоконтроль \_\_\_\_\_  
(прізвище та ініціали)

**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

\_\_\_\_\_ О.В. Негоденко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

**ЗАВДАННЯ**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

\_\_\_\_\_ Сачик Микита Олександрович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка додатку для контролю парі під систему Android з використанням C#.net»

Керівник роботи \_\_\_\_\_ Дібрівний Олександр Андрійович, доктор філософії

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

затверджені наказом вищого навчального закладу від «12»березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вихідні дані до роботи:

3.1 Офіційна документація Microsoft.

3.2 Figma.

3.3 Visual Studio.

3.4 Azure.

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

4.1 Аналіз актуальності та проблематики розроблюваного додатку.

4.2 Аналіз та вибір інструментів для реалізації мобільного додатку.

4.3 Проектування мобільного додатку.

4.4 Висновки

## 5. Перелік графічного матеріалу

- 5.1 Титульний слайд
- 5.2 Мета, об'єкт, предмет та наукова новизна дослідження
- 5.3 Актуальність роботи
- 5.4 Аналоги
- 5.5 Технічні завдання
- 5.6 Програмні засоби реалізації
- 5.7 Зв'язок елементів додатку
- 5.8 UML діаграма бази даних мобільного додатку
- 5.9 Апробація результатів дослідження
- 5.10 Висновки
- 5.11 Кінцевий слайд

6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми бакалаврської роботи	19.04.2021	Виконано
2	Підбір науково-технічної літератури	20.04.2021 - 22.04.2021	Виконано
3	Дослідження аналогів та актуальності додатку	23.04.2021	Виконано
4	Аналіз та вибір інструментів для розробки додатку	23.04.2021 - 25.04.2021	Виконано
5	Проектування та реалізація	25.04.2021 - 20.05.2021	Виконано
6	Висновки, оформлення роботи	20.05.2021 – 23.05.2021	Виконано
7	Розробка демонстраційних матеріалів	24.05.2021	Виконано
8	Здача роботи	25.05.2021	

Студент \_\_\_\_\_  
( підпис )

Сачик М.О.  
( прізвище та ініціали )

Керівник роботи \_\_\_\_\_

Дібрівний О.А.





## РЕФЕРАТ

Текстова частина бакалаврської роботи с. 51, рис. 25, джерел 10.

*Об'єкт дослідження* – процес регулювання конфліктів.

*Предмет дослідження* – технології, за допомогою яких можливо відстежити дискусію.

*Мета дослідження* – розробка мобільного додатку для контролю парі.

У дипломній роботі був проведений аналіз аналогів що вже існують, переваг та недоліків програмних інструментів для додатка.

Виконано опис програм, які були використані у ході роботи, та середі розробки.

Додаток реалізований за допомогою мови програмування C#, з використанням фреймворку Xamarin, для кросплатформної розробки додатку на мобільний пристрій.

Ключові слова: Парі, XAML, C#, конфлікт, Android, Xamarin, Figma, .NET, Azure, GIT.

## ЗМІСТ

<b>ВСТУП</b> .....	<b>8</b>
<b>1. ТЕОРЕТИЧНА ЧАСТИНА</b> .....	<b>9</b>
1.1 Загальні уявлення про вирішення конфліктів .....	9
1.2 Методи вирішення конфліктів .....	11
1.3 Актуальність.....	13
1.4 Аналіз наявних додатків .....	18
1.5 Складання технічного завдання.....	20
<b>2. ОПИС ПРОГРАМНИХ ЗАСОБІВ</b> .....	<b>25</b>
2.1 Мова програмування C# та .NET .....	25
2.2 Xamarin.....	27
2.3 Figma .....	29
Розділ 2.4 Microsoft Azure.....	34
2.5 Photoshop .....	40
<b>3. ЗАГАЛЬНІ ВІДОМОСТІ ДОДАТКУ</b> .....	<b>43</b>
3.1 Розробка концепції.....	43
3.2 Проектування структури та створення додатку .....	44
<b>ВИСНОВКИ</b> .....	<b>57</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ</b> .....	<b>59</b>
<b>ДОДАТКИ</b> .....	<b>60</b>

## ВСТУП

*Оцінка сучасного стану об'єкта розробки.* На сьогоднішній день багато людей вирішують більшість суперечок за допомогою парі. З розповсюдженням соціальних мереж, кількість челленджів також стрімко зростає, а з ним і кількість людей що намагаються один з одним змагатися.

*Об'єкт дослідження* – процес регулювання конфліктів.

*Предмет дослідження* – технології, за допомогою яких можливо відстежити дискусію.

*Мета дослідження* – розробка мобільного додатку для контролю парі.

*Наукова новизна проекту* – створення унікального мобільного додатку для контролю дискусій з використанням системи суддів, для нагляду за змаганням, та поєднанням різних можливостей для укладання парі.

У дипломному проекті був проведений аналіз додатків-аналогів, та виявлені переваги та недоліки.

Додаток було розроблено на мові програмування C# з використанням .NET Framework та Xamarin.Forms у середовищі розробки Visual Studio 2019 для платформи Android. Інтерфейс проектував у онлайн-сервісі Figma. Оскільки я вибрав Xamarin.Forms, то інтерфейс програми був створений з допомогою мови розмітки XAML. Для елементів графічного інтерфейсу я використовував Adobe Photoshop. Проводячи аналіз зацікавленості людей до укладання парі, було з'ясовано, що багато людей не пам'ятає про більшість парі що вони уклали, і тому додаток який допоможе про них нагадати є актуальним. Також багато людей більш мотивовані виконати парі, коли за ним слідкують інші люди, що й було реалізовано.

Додаток може використовуватись як для самоконтролю, так і для вирішення багатьох суперечок з іншими.



# 1. ТЕОРЕТИЧНА ЧАСТИНА

## 1.1 Загальні уявлення про вирішення конфліктів

У кожної людини в житті були випадки конфліктів, як внутрішні, так і з іншими людьми. Так що ж таке конфлікт?

Конфлікт – це ситуація зіткнення інтересів, сторін чи поглядів, що відрізняються. Під час конфлікту неможливо досягти згоди без певних факторів, які загалом є індивідуальними. Людей, між якими розпочинається конфлікт можна назвати суперниками, опонентами.

Коли трапляється конфліктна ситуація, кожний хто бере у ньому участь прагне досягти своєї перемоги, відстоювання точки зору за допомогою аргументів, які повинні довести власну правдивість, та змусити опонента погодитись з поразкою.

Ситуацій, за яких починається конфлікт багато, але є основні фактори:

- внутрішній: фінансові проблеми, неприязнь, психологічний фактор;
- зовнішній: політична та економічна ситуація у державі, різниця у соціальному розташуванні.

Конфліктна ситуація не може існувати без:

- людини, або людей, які є учасниками конфлікту;
- певного інциденту;
- об'єкту конфлікту (причина навколо якої розгортається суперечка).

Предмет конфлікту утворюється завдяки об'єкту та учасників конфлікту, які є необхідними умовами для виникнення конфліктної ситуації. Каталізатором конфлікту є інцидент, бо він призводить до початку конфлікту. Інцидент з'являється несподівано, за необхідних передумов.

Можна виділити основні види конфліктів:

- між людиною і групою;

- між групами;
- міжособистісний;
- всередині людини;

Багато людей вважає що краще уникати конфліктів, але насправді вони можуть стати причиною розвитку людини, або групи людей.

Як визначити, що конфлікт може бути корисним:

- є стимулом до знаходження способів справедливо розв'язати та запобігти подальших конфліктів;
- за допомогою конфлікту ви здобули нових знайомих, або взаємозв'язок з вже знайомими людьми покращився;
- допомагає звернути увагу на певні ідеї та пропозиції, що потрібно ще обговорити, або допрацювати;
- допомагає визначитися з пріоритетами;
- сприяє конструктивному та безпечному прояву емоцій;
- сприяють утворенню неформальних союзів;
- допомагає розвитку у певних сферах.

Як ми переконалися, конфліктні ситуації можуть приносити користь, не лише людині яка у кінцевому результаті довела свою точку зору, а й для усіх учасників конфлікту. Важливо під час конфлікту не забувати про його мету, прислухатися до доказів опонента, та проаналізувати ситуацію, що трапилась. Не слід забувати, що під час суперечки можна дізнатись багато нового.

Як визначити, що конфлікт може бути негативним:

- може загрожувати основним інтересам сторін, що може вплинути на кардинальну зміну пріоритетів;
- дуже сильно затягується;
- впливає на погіршення відносин між людьми;

- дії у ході конфлікту можуть виконуватись швидко та необдуманно;
- може призвести до втрати цілей;
- обидві сторони не зацікавлені у конфлікті, але принципово не поступаються один одному;
- жодна зі сторін не прислухається до один одного.

## 1.2 Методи вирішення конфліктів

Психолог Дейл Карнегі казав, що єдиний спосіб перемогти у спорі або конфлікті – його не починати. Це правда, що набагато легше зробити багато корисних речей, ніж доказати деяким співбесідникам що вони дійсно не праві.

Іноді справжні баталії можуть розпочатися між давніми друзями, родичами, одруженими усього лише через невелику суперечку, бо кожен має свою власну думку, погляд на різні речі, та правда. Багато хто забуває про суть конфлікту, але відносини можуть вже бути зруйновані, не даром Сократ казав, що у суперечці народжується істина. Тому дійсно, у деяких випадках треба намагатися не конфліктувати, але що ж робити коли його не можна уникнути? Відстоювати свою позицію, чи погодитися з поразкою, не вступаючи у конфлікт?

Методи вирішення конфліктів:

- **запобігання.** Цей метод найлегший, але у багатьох ситуаціях його не можна застосувати. Ви можете спробувати просто ігнорувати конфлікт, удати що його не існує, але це не вирішить суперечку. Якщо його застосувати, то ви ризикуєте потім потрапити у ще більший конфлікт, який відбудеться через ігнорування минулого. Але він доречний коли конфлікт вам взагалі не потрібний, його просто можна уникнути без наслідків, і єдине що вам принесе – втрата часу, та зусиль. Також метод запобігання доречно використовувати коли ви точно не зможете перемогти;

- **співробітництво.** Якщо ваш опонент готовий на діалог, конструктивно вести з вами бесіду, та ви разом шукаєте як можна вирішити ситуацію найкраще, тоді рекомендовано використовувати цей метод. Це правда, що під час конфлікту з близькими вам людьми потрібно його вирішити шляхом найменших жертв, та зберегти теплі відносини. Не потрібно забувати, що під час діалогу, використовуючи цей метод, потрібно відноситись до опонента як до колеги, а не як до супротивника, та вести діалог, бо якщо висловлюється лише одна людина – це не співробітництво. За усіх умов, цей метод вирішення конфлікту – найефективніший;
- **компроміс.** При обговоренні наявних варіантів для вирішення конфлікту, супротивники намагаються знайти варіант, який зможе задовольнити всіх. Вони здатні поступитися деякими умовами та претензіями. У багатьох випадках обидві сторони у кінцевому результаті отримують не те що хотіли спочатку, але частково задоволені. Хоча існує думка, що під час компромісу жодна сторона не досягає початкової мети суперечки;
- **поступливість.** Використовується коли людина погоджується з точкою зору іншої та приймає запропоновані умови. У цьому випадку потрібно відмовитись від своїх вимог. Може трапитись через усвідомлення правильної думки супротивника, або небажанням продовжувати конфлікт, перемога у якому не обов'язкова. Також якщо є бажання не псувати відносини. Інколи поступитися сильному супротивнику не соромно, але якщо постійно використовувати метод поступливості, то ніколи не досягнете своїх цілей;
- **конкуренція.** У цьому методі основним завданням є нав'язати саме свій погляд на вирішення конфлікту, якомога вигідніше для себе закінчити суперечку;

- **парі.** Що ж робити у випадку коли жодна зі сторін впевнена щодо своєї точки зору, не хоче поступатися та прагне довести опоненту що він не правий, або не може щось зробити? У цьому випадку можна скласти парі. Парі – це угода, яка регулює конфлікт або суперечку між зацікавленими особами. Спочатку між особами обговорюються умови парі, у яких фігурує причина конфлікту, зобов'язання яке виконуватиме той, хто програв, крайній термін парі, та люди що будуть його контролювати. Цей метод дозволяє майже стовідсотково дізнатись чия точка зору була правильною.

Переглянувши методи за допомогою яких вирішуються конфлікти, можна з впевненістю сказати що для кожної ситуації потрібен власний підхід для вирішення. У багатьох випадках конфлікт може перерости у негативну суперечку та погіршення відносин між опонентами, але зазвичай цього можна уникнути за допомогою деяких порад.

Щоб вирішити суперечку мирно, потрібно:

- поважати точку зору супротивника;
- аргументувати свою точку зору за допомогою фактів;
- не позбавляти опонента права вибору;
- не переходити на особистості;
- тихо та спокійно вислухати опонента.

Проаналізувавши можливості вирішення конфліктів було виявлено що метод укладання парі найрізноманітніший.

### 1.3 Актуальність

Останнім часом люди постійно використовують соціальні мережі, наприклад Instagram, tik tok та інші. Більшість користувачів намагаються набрати велику аудиторію у цих додатках, а найшвидше це можна реалізувати виконуючи щось

божевільне, унікальне. Можна згадати як увесь світ багато разів охоплювали челенджі, через які увесь світ мав змогу побачити як мільйони людей обливають себе крижаною водою, повторюють одні й ті самі рухи під музику з банальним текстом, та багато інших. Здавалося б, що може зробити ці виклики цікавішими? Відповідь на це запитання – додаток для контролю парі.

Уявіть собі, популярні люди у соціальних мережах не просто виконують якийсь челендж, а й виконують додаткове завдання, уклавши парі з іншим. У разі перемоги – зірка доказує що здатна була це виконати, й отримує винагороду. У разі поразки – вона виконує умови суперечки. Це означає, що в усіх випадках перемагають глядачі, бо отримують додаткове видовище, та ще й виконавець – більше уваги та популярності. Це не обов'язково повинна бути популярна персона, кожен при нагоді зможе скористуватись додатком, та, наприклад, зняти свою частину виконання парі, й завантажити до соціальної мережі. Оскільки суспільство дуже любить щось обговорювати, то така подія може дуже швидко розповсюдитись.

Розроблений додаток може охоплювати багато сфер та вікових категорій, але цільова аудиторія – підлітки, бо вони є найбільш конфліктними у наш час, мають багато вільного часу, вони ще не повністю сформовані, дуже люблять суперечки, та впевнені що все знають, а якщо хтось скаже що це не так – готові відстоювати свою думку до кінця. У разі суперечки, вони з легкістю зможуть її вирішити за допомогою додатку, ще й весело провести час.

У ході мого дослідження, яке було проведено у соціальній мережі instagram, було виявлено що зі 123 голосів, 77% опитаних користувачів вже уклали парі (рис 1.1). Після цього я відписав тим, хто дав позитивну відповідь, та запитав, чи забували вони колись про укладену угоду. Майже всі відповіли що забували, та не один раз, а багато разів.

Усі люди часом здатні про щось забути, здебільшого це бувають нісенітниці, але є випадки коли через це, вона може багато чого втратити, або забути про договір, який міг стати дуже результативним.

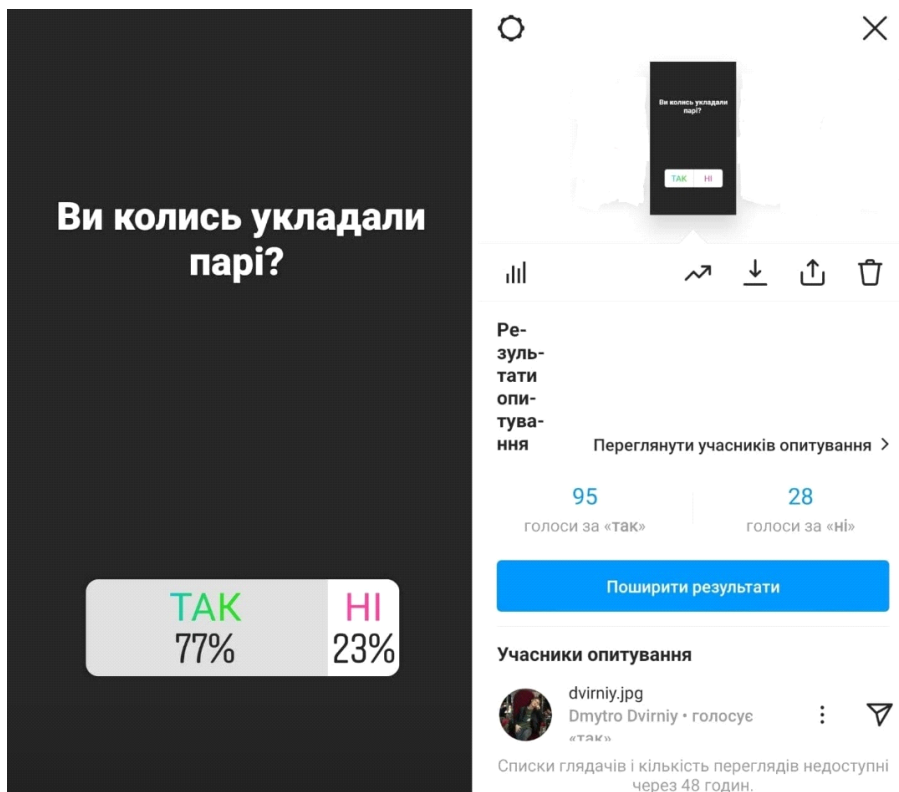


Рисунок 1.1 – Результати голосування про укладення парі

Аналізуючи отриманні результати, можна зрозуміти що велика кількість парі забувається. Це підтверджує необхідність контролю, та нагадування про складені угоди, і тому було проведено друге голосування (рис 1.2).

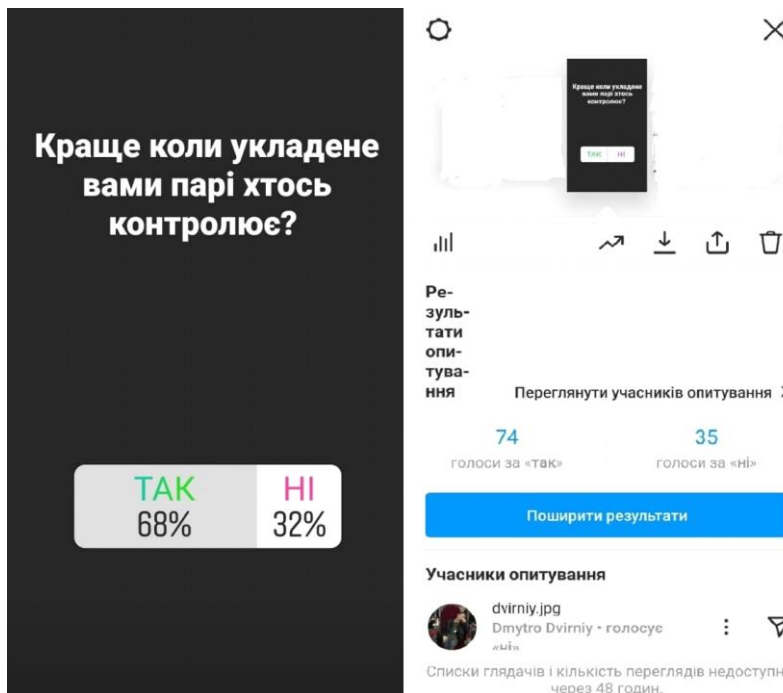


Рисунок 1.2 – Результати голосування з приводу контролю парі

У результаті другого голосування, була підтверджена гіпотеза про необхідність контролю над суперечками. Переглянувши користувачів які відповіли негативно, було встановлено що більшість із них ніколи не укладали парі, і тому вони не можуть зрозуміти необхідність контролю.

Було прийнято рішення запитати як здебільшого вдавалося вирішувати конфлікти у тих, хто проголосував що не складав парі. Здебільшого відповідали про методи поступливості, запобігання та конкуренції, рідше – компромісу. У інших я дізнався, що складаючи парі, конфлікт часто закінчувався через надання чітких фактів однією зі сторін, або доказом дій. А сторона що зазнала поразки зазвичай спокійно реагувала, бо розуміла що була не права, що й було доказано.

Підсумовуючи результати, зрозуміємо, що у більшій кількості випадків складання парі було кориснішим за інші методи.

Наступним кроком була розповідь аудиторії яка відповідала на запитання, про шість основних методів вирішення конфлікту, після чого я запропонував їм проголосувати позитивно, за умовою надання їх переваги методу складання парі. Негативна відповідь означала б їх прихильність по інших методів, виключаючи компроміс.

Метод складання парі переміг, отримавши 60% голосів, інші ж методи – лише 40%.

У вирішальному голосуванні (рис 1.3), аудиторія вирішила обрати саме метод парі, з чого випливає що він є найкращим варіантом для них вирішити суперечку. Мені також писали що він приваблює ще й з азартної точки зору, пробуджує бажання позмагатись.

Нажаль не всі зрозуміли що є дійсно діючий спосіб вирішення конфлікту, але скільки людей, стільки й думок. Мені казали що через парі, сутичка може затягнутися, й забутись, або ж ніхто нічого не зможе доказати у вирішальному результаті. Я з цим не згоден, бо про неї можна нагадати своєчасно, а для доказу є спеціальні люди, що спостерігатимуть за самим конфліктом, та результатом, що гарантує справедливе парі.



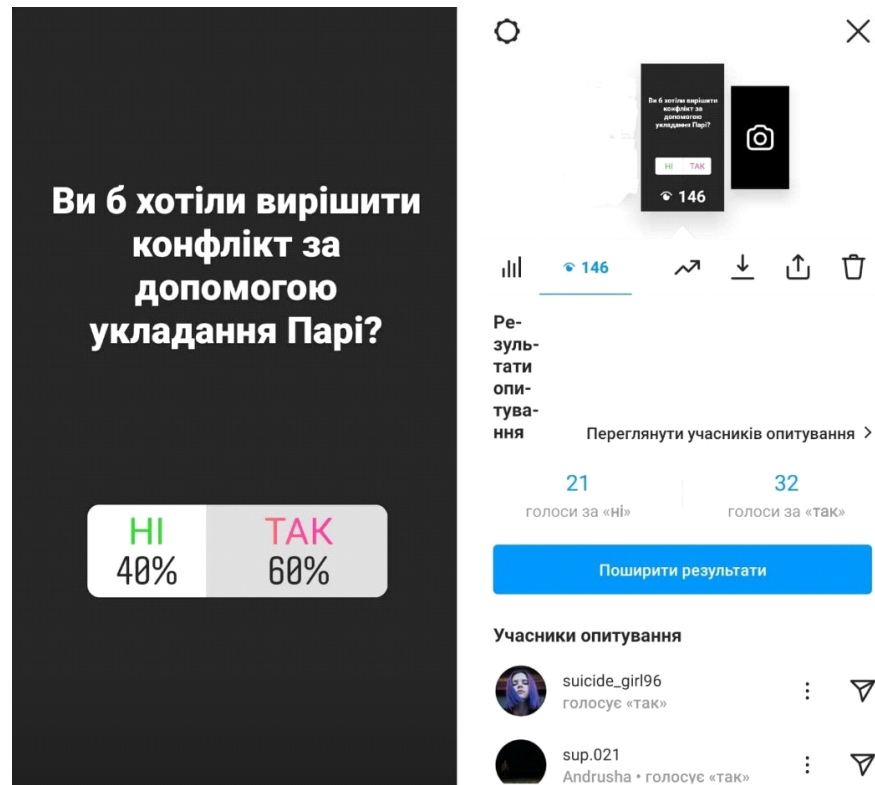


Рисунок 1.3 – Результати голосування методів вирішення конфліктів

Майже всі хто брав участь у голосуванні проживають в Україні, більшість з Києва, та віком від 18 до 24 років (рис. 1.4).



Рисунок 1.4 – Віковий діапазон та місце проживання аудиторії

За статевими ознаками різниці майже нема, на декілька відсотків було більше жінок які брали участь у голосуванні (рис 1.5).

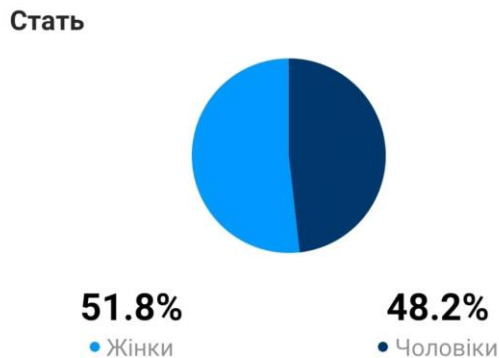


Рисунок 1.5 – Аудиторія що брала участь у голосуванні за статевими ознаками

Розробка додатку для контролю парі є досить актуальною, у чому ми впевнились виходячи з голосування. Він допоможе користувачам найсправедливішим шляхом вирішити конфлікти, допоможе збільшити аудиторію, та може бути чудовою розвагою.

#### 1.4 Аналіз наявних додатків

Ідею для розробки додатку придумав сам, і тому вважав що аналогів не існує, але я знайшов програму, яка у деяких аспектах схожа на мою, але вже не працює.

Назва першого додатку «Dare App: первое приложение для видео челленжей» (рис 1.6).

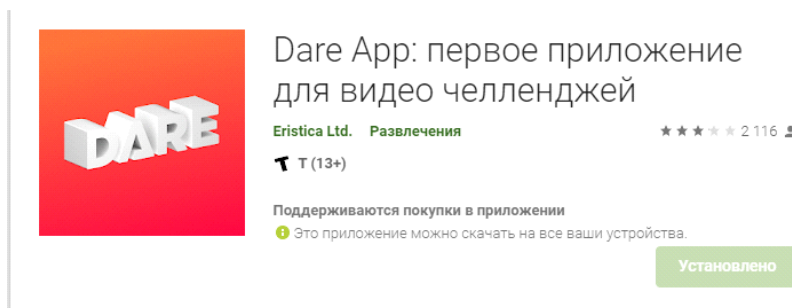


Рисунок 1.6 – Dare App у Google play

Основна ідея цього мобільного додатку була у виконанні челенджів від інших

людей, завантаження відео файлу де ви їх виконуєте, після чого користувачі оцінюють ваше виконання. Розробники кажуть що за виконання ви будете отримувати гроші, які потім зможете вивести, ще й додаткові гроші за запрошення друга, але для цього вам потрібно щось купувати за реальні гроші.

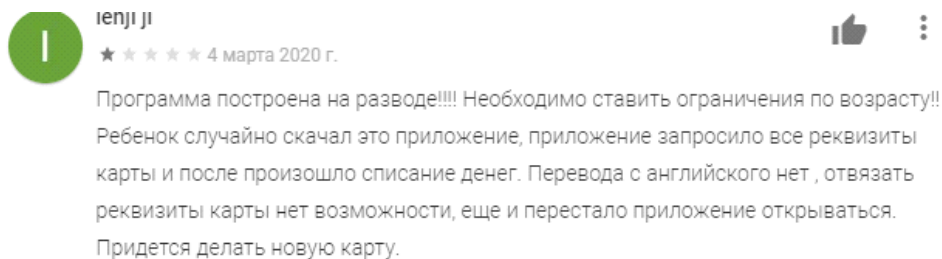


Рисунок 1.7 – Коментар до додатка Dare App

У коментарях до цієї мобільної програми (рис 1.7), було безліч відгуків про шахрайство, поповнювати гроші можливо, а ось вже вивести – ні. Деякі пишуть що у точності виконували завдання, але їм спеціально ставили «не виконано».

Скоріш за все розробники назбирали достатньо грошей та вирішили більше не підтримувати свою розробку (рис 1.8).

На жаль я не зміг скористуватися додатком та повністю його перевірити, але можу сказати з впевненістю – таке відношення розробників до користувачів просто жахливе.

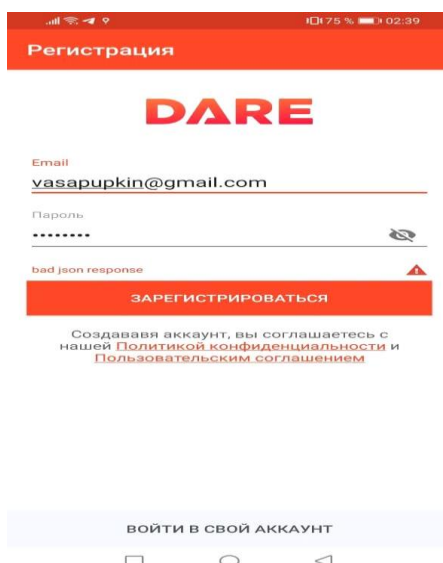


Рисунок 1.8 – Неможливо увійти до додатка Dare App

Додаток можна знайти за посиланням:

<https://play.google.com/store/apps/details?id=com.eristica.eristica&hl=ru&gl=US>

## 1.5 Складання технічного завдання

У ході дослідження було вирішено створювати мобільний додаток. В сучасному світі майже кожна людина користується мобільним телефоном, й проводить у ньому більшість часу. Можна бути впевненим, що де б не був користувач, він зможе дістати телефон, й використати програму.

Перш за все треба обрати мову програмування, за допомогою якої й буде написано мобільну програму.

Мобільний додаток можна написати за допомогою двох способів:

- кросплатформенна розробка;
- нативна розробка.

Кросплатформенний додаток розроблюється за допомогою web-сервісів, які надають змогу написання програми одразу для декількох платформ. Створення за таким сюжетом виходить набагато дешевшим, але з іншої сторони ми втрачаємо швидкість роботи, та занадто простий інтерфейс. Для того, щоб платформа змогла взаємодіяти з додатком, треба його переформувати на мову, яку розуміє платформа. Дуже доречно використовувати кросплатформенну розробку у випадку, коли треба реалізувати один додаток на кількох платформах, це ще й економить багато часу.

Нативним називається додаток, який був розроблений для платформи, яка є його рідною, тобто для визначеної платформи, визначеною мовою програмування. Його довго створювати, його розробка та підтримка багато коштує, але він досить швидко реагує на взаємодії, розроблює рідний для користувача інтерфейс, є дуже продуктивним та має змогу напряду підключатись до апаратної частини.

Нативні мови програмування:

- java;

- kotlin;
- swift;
- objective-C.

Java – найпопулярніша об'єктно-орієнтована мова програмування, більшість Android додатків написані за допомогою java. Вона ідеально підходить для роботи з платформою Android.

Код java спочатку транслюється в байт-код, після чого він виконується JVM, віртуальною машиною, а отже його код не одразу виконується інтерпретатором, та водночас Java не є повністю компільованою мовою як, наприклад, C++. Ця мова програмування підтримує автоматичну збірку сміття.

Kotlin – статично типізована мова програмування, у якої дуже простий синтаксис. Kotlin здебільшого використовується для розробки під операційну систему Android, компанія Google навіть визнала цю мову як офіційну для розробки під Android, і також додала інструменти які допомагають працювати з нею у функціонал Android Studio.

Можна вважати що Kotlin – це справжня екосистема, а не просто мова програмування, з ядром - Common Kotlin. Воно має власні бібліотеки, мову програмування, та інструменти для створення додатків.

Swift – строго типізована об'єктно-орієнтована мова програмування для розробки на OS X та IOS додатків. Має ARC – механізм автоматичного підрахунку посилань, який допомагає контролювати пам'ять. Swift є мовою програмування, що компілюється.

Створення додатку на мові Swift простий та гнучкий процес за допомогою сучасним функціям. У ньому використовуються патерни безпечного програмування.

Objective-C – це компільована об'єктно-орієнтована мова програмування, за допомогою якої створюються додатки на IOS. Багато хто думає, що Objective-C, як первинна мова розробки для IOS, вже не актуальна, бо існує Swift, якому прогнозують велике майбутнє. Але чимало додатків використовують Objective-C,

тому перехід на Swift може розтягнутись надовго.

Objective-C був створений завдяки Smalltalk та мови програмування C, з якою мова програмування повністю сумісна. У стилі Smalltalk було створено об'єктну модель, це означає що повідомлення надсилаються об'єктам. Mac OS X (Cocoa) і GNUstep, реалізації об'єктно-орієнтованого інтерфейсу OpenStep, також використовують Objective-C.

Кроссплатформенні мови програмування:

- js з використанням react native;
- dart з використанням flutter;
- c# з використанням xamarin.

React Native використовується для створення зображуваних Android та IOS додатків, та водночас є JS-фреймворком. Додатки розробляються за допомогою JavaScript з використанням CSS. Гнучкість React Native здобувається за допомогою інтерпретації коду на платформі. Інтерфейс залишається рідним, бо він створюється за допомогою JS-бібліотеки React, яку розробили у Facebook.

Хоча може здатися що вона орієнтована на браузері, насправді її призначення – мобільні платформи. Через відносно недавнє створення React Native можна помітити, у неї не вистачає документації та інструментів розробки.

Можна розробити компонент React Native, та він вже зможе працювати в Android та IOS. Це відбувається завдяки тому, що більша частина API у фреймворку кроссплатформенні.

Flutter – це фреймворк від Google, який використовує мову програмування Dart, та надає змогу створювати кроссплатформенні додатки, які можуть використовувати один єдиний код. За допомогою Flutter можна створювати графічні додатки під MacOS, Linux, Windows, різноманітні веб додатки, та мобільні програми під Android та IOS.

Більшість коду зазвичай зберігається, але невеликі фрагменти уже ж таки доведеться перероблювати під потрібну операційну систему, що сприяє економії ресурсів та часу для розробки додатку під усі зазначенні платформи. Flutter малює

на екрані смартфона, та відтворює потрібні елементи самостійно, не перероблюючи вихідний код в нативний.

Xamarin використовується для кросплатформенної розробки мобільних додатків за допомогою мови програмування C# на Windows Phone, iOS та Android. Коли ми за допомогою Xamarin пишемо якісь функції, алгоритми, то більшу частину коду можна не змінювати, вони усе одно будуть працювати як з Android, так і з IOS. Прискорити процес розробки можна завдяки Xamarin.Forms, бо за допомогою форм, ми зможемо залишати без змін ще більше частин коду.

У собі додатки Xamarin зберігають виконавчу середу Mono, та при старті роботи, воно транслює вже скомпільований C# код в команди певних операційних систем.

За допомогою платформи Xamarin, ми маємо один інструмент для легкого створення додатків одразу на усіх платформах, економлячи час та гроші. За допомогою Xamarin ми маємо прямий доступ до нативних API Android, IOS та Windows Phone.

Мені потрібно, щоб обрана мова програмування підходила до моїх вимог.

Вимоги до додатка:

- можливість створення облікового запису користувача;
- наявність функції створення виклику для самоконтролю;
- можливість пошуку інших користувачів;
- здатність кинути виклик користувачам;
- наявність функції, дозволяю переглянути надіслані користувачу парі;
- можливість відсудити парі;
- здатність переглянути парі, у яких користувачі приймають участь;
- наявність повної інформації про ухвалену угоду;
- наявність інформації про кількість перемог та поразок користувача у сутичках;

- можливість легкої адаптації до розширення функціоналу.

Аналізуючи вимоги до розробки мобільного додатку, було вирішено його створювати за допомогою кросплатформенної мови програмування, бо за її допомоги можна швидко та легко розширити функціонал, наприклад реалізувати додаток ще й на операційній системі IOS. Кращим варіантом виявилася мова програмування C# з використанням Xamarin. Спостереження, тестування та контроль версій додатку буде проводитися за допомогою git, а як веб-сервіс було обрано github.



## 2. ОПИС ПРОГРАМНИХ ЗАСОБІВ

### 2.1 Мова програмування C# та .NET

C# – об'єктно-орієнтована мова програмування, розроблена корпорацією Microsoft, метою якої є створення додатків що працюють у .NET. C# має багато спільного з C++, C, Java, бо це мова високого рівня. Якщо ви колись користувалися цими мовами, розпочати роботу над додатком буде набагато швидше й легше.

Ця мова програмування досить розповсюджена при створенні комп'ютерних та веб-додатків. Важко уявити яка мова програмування буде більш доречна при розробці комп'ютерного додатка для Windows. Дуже допомагає наявність великої кількості користувачів даної мови, бо як наслідок, збільшується кількість інформації, людей що зможуть у разі чого допомогти з труднощами які трапляються при розробці, та найголовніше – мова покращується, створюється нове програмне забезпечення та інструменти.

Можна створювати дуже об'ємні програми, які водночас будуть гнучкими, та з легкістю будуть здатними до модифікації за допомогою об'єктно-орієнтованого підходу. Ця мова програмування підтримує головні якості об'єктно-орієнтованого програмування, такі як інкапсуляція, поліморфізм та успадкування.

З новими версіями C# тільки покращується та розвивається, додаючи до свого функціоналу нові можливості, як наприклад асинхронні методи та лямбда вирази, обробка виключень та динамічне з'єднання.

Повнофункціональний веб-сайт також можна написати на C#, бо за наявності об'єктно-орієнтованого підходу, він стає легше обслуговуватись, більш ефективним, та масштабним.

За допомогою C# створюється безліч ігор, він поєднується з ігровим рушієм Unity, найпопулярнішим інструментом для розробки відеоігор. Кінцевий продукт розробки, може буди застосований як на комп'ютерних системах, так і на ігрових консолях, мобільних пристроях. Такому успіху у створенні різноманітних ігор C# сприяє об'єктно-орієнтований підхід, інтерфейси, автоматичне керування

оперативною пам'яттю комп'ютера за допомогою збирання сміття.

Головні переваги C#:

- наявність синтаксичного цукру у великій кількості. Загалом це синтаксичні можливості, які спрощують розробку, та за допомогою яких можливо перетворити велику кількість коду у більш лаконічний, читабельний та компактний. Його використання не впливає на компіляцію;
- фіксований розмір типів даних;
- підтримка переважної більшості продуктів Microsoft;
- написання додатків та програм для операційних систем IOS, Android, Linux та MacOS за допомогою Xamarin;
- індивідуальні розробники та невеликі компанії мають здатність безкоштовно користуватися багатьма інструментами, наприклад Azure, Mac Pro, Windows Server, Visual Studio;
- об'єктно-орієнтований підхід, який насправді використовується правильно;
- безпечний (збір сміття, виключення, managed);
- досить легкий для освоєння як при першому ознайомленні з програмуванням через свою зрозумілість, так і для програмістів, завдяки схожості з іншими мовами програмування.

Якщо мова йде про технології платформи .NET, наприклад про ASP.NET, Xamarin, Windows Forms, WPF, то у багатьох випадках люди мають на увазі мову програмування C#, це працює й у зворотному напрямку. Вони між собою тісно пов'язані, але не треба розглядати їх як одне й теж. Мова програмування C# була спроектована та розроблена для роботи з .NET Framework, та зміст .NET набагато ширший.

Бути середовищем, яке підтримує створення і виконання сильно розподілених компонентних додатків, ось яку мету має .NET Framework. Цей

фреймворк постачає безпеку, сумісне використання мов програмування, та загальну модель розробки під платформу Windows. Він надає можливість використовувати створені API-інтерфейси для створення комп'ютерних додатків як WPF, веб-додатків як ASP.NET, веб-апплетів як Silverlight, для роботи з базами даних як ADO.NET, та інших.

Додатки створенні за допомогою .Net є кросплатформними програмами, а фреймворк має здатність до роботи з мовами програмування COBOL, Visual Basic, C ++, як ми вже знаємо C #, та іншими.

Платформу .NET Framework заведено вважати відповіддю компанії Microsoft на іншу платформу, Java, що мала велику популярність у той час, коли була створена компанією Sun Microsystems.

Мова програмування C# була створена саме для ASP.NET, який у результаті й був написаний за допомогою цієї мови програмування. Технологія активних серверних сторінок Active Server Page, скорочено ASP.NET, стала новим інструментом платформи .NET. Вона сприяла швидкій розробці веб додатків, що взаємодіють з базами даних.

Засновник корпорації Microsoft Білл Гейтс визнав, що найкращий винахід його компанії – це .NET. Це висловлювання не було даремним, бо за допомогою потужної платформи .NET Framework створюються чудові додатки.

Найсильніші сторони .NET:

- величезна бібліотека класів;
- розмаїття технологій;
- підтримка різних мов.

## **2.2 Xamarin**

Xamarin - це програма для розробки мобільних додатків, яка має мету зробити можливим створення мобільних програм для Android та iOS за допомогою однієї мови: C #. Xamarin пропонує можливість повноцінно використовувати мобільні

платформи без необхідності вивчати мову, специфічну для кожної мобільної ОС. Таким чином, засвоївши мову C #, розробник зможе потім писати для iOS, Android або Windows mobile.

Якщо ми розробляємо мобільний додаток, що працює на 3 основних мобільних операційних системах (не маючи фреймворку, як Xamarin), ми повинні мати три різні проекти, по одному для кожної ОС. Уявіть собі, кожен раз потрібно робити правки з різними обмеженнями, й для кожної операційної системи по своєму.

Використовуючи Xamarin, відразу все стане набагато простіше! Бізнес-рівень стає загальним для 3 платформ. Економія часу стає у рази більшою, оскільки бізнес-рівень буде розроблений лише один раз, не кажучи вже про етапи налаштування, технічного обслуговування та розробки. Однак на фронтальній частині, Xamarin зберігає створення власного інтерфейсу для кожної цільової платформи.

Однією з сильних сторін Ксамаріна є чіткий розподіл проектів. Однак важливо зрозуміти різницю між двома типами проектів: портативними проектами та спільними проектами.

Портативний — це проект, в якому графічний інтерфейс та ділова частина (з мережевим рівнем та рівнем доступу до даних) спільно використовуються усіма платформами, і де, лише декілька елементів реалізовані безпосередньо.

Спільний проект — це проект, в якому спільно використовується лише ділова частина. Графічний інтерфейс та вся логіка, яка з цього випливає, реалізовані власним чином для кожної платформи.

Статистично кажучи, це можна підсумувати наступним чином:

- портативний проект має 75% загального коду та 25% специфічного коду;
- спільний проект має 50% загального коду та 50% конкретного коду.

Щоб розробити крос-платформну програму з Xamarin, ви можете використовувати Xamarin Studio або Visual Studio як IDE. Редактор Xamarin Studio є безкоштовним та доступним для Mac та Windows. Він повноцінний і дозволяє

розробляти, тестувати, генерувати, профілювати та робити графічні інтерфейси. насправді це вдосконалена версія MonoDevelop.

Однією з основних відмінностей, яку слід виділити тут, є те, що якщо ми хочемо розробити і для Windows-телефону, то у нас немає іншого вибору, крім використання Visual Studio, Xamarin Studio не має цієї можливості. Придбання ліцензії Xamarin також дозволяє повністю інтегруватися в Visual Studio.

Xamarin пропонує можливість повторного використання коду через спеціальний "магазин", який включає численні бібліотеки та компоненти. Багато з них безкоштовні, деякі платні, значна частина обмежена iOS та Android, інші також працюють на Windows Phone, деякі присвячені лише ОС. Також можна безпосередньо включити сторонні бібліотеки на Java або C, написавши посилання. Деякі вже мають свій порт у компонентах Xamarin (наприклад, бібліотеки підтримки Android).

Програми, виготовлені в Xamarin, працюють дуже продуктивно. В Android згенерований код виконується віртуальною машиною Mono, яка працює так само швидко, як машина Dalvik. Що стосується iOS, Apple забороняє роботу будь-якої віртуальної машини. Таким чином, створений код є безпосередньо власним двійковим кодом, скомпільованим за допомогою AOT.

Для використання Xamarin потрібно добре знати C#, та SDK кожної платформи. Навіть якщо розробник може досконало освоїти, наприклад C#, він усе одно може зробити проект невдалим, якщо він не знає принципів розробки кожної платформи.

## 2.3 Figma

Figma – це інструмент створення та графічного маніпулювання векторними елементами, що спеціалізується на розробці мобільних та веб інтерфейсів, вебсайтів.

За допомогою Figma можна вносити зміни у проєкт в реальному часі через веб-браузер. Цей інструмент розроблений як система дизайну, якою може

поділитися команда дизайнерів, та є потужним інструментом, який повністю відповідає сучасним тенденціям. Сьогодні не можна говорити про створення прототипів та засоби спільного проєктування, не згадуючи Figma, вона навіть входить до трійки кращих інструментів для розробки інтерфейсів, за думкою фахівців UI.

Запущений у 2016 році, цей інструмент мав приголомшливий успіх серед спільноти дизайнерів інтерфейсу користувача та фахівців з прототипування, щодо зручності використання інструменту, а також функціональних можливостей, які він може запропонувати. Він швидко стає прямим конкурентом таких інструментів, як Sketch або Adobe XD.

Figma належить до нового покоління інструментів для проєктування дематеріалізованих інтерфейсів. Він призначений для спільної роботи між різними зацікавленими сторонами в проєкті. На додаток до ресурсів та графічних елементів, які вона дозволяє експортувати, Figma пропонує можливість генерації коду, пов'язаного з різними елементами інтерфейсу.

Figma - це інструмент, який дозволяє вам ділитися та розповсюджувати систему дизайну в команді UI, UX та розробників. Він також пропонує обмін у режимі реального часу та спільні робочі функції.

Ці функції допомагають пришвидшити виробничий потік, дозволяючи співпрацю в реальному часі між різними підрозділами над одним проєктом.

Однією з головних характеристик цього інструменту є його легка доступність. Існує веб-версія, доступна через браузер, а також настільна версія для Windows та MacOS.

Попри на те, що Figma є повноцінним інструментом і пропонує безліч функцій, Figma залишається доступним інструментом для початківців. Звичайно, за умови, якщо він знає, що таке система проєктування, розуміє використання стилів і особливо логіку успадкування компонентів та об'єктів, і дотримується документації від Figma. Його дизайн був розроблений, щоб надати UX-дизайнеру

(або UI дизайнеру) інтуїтивно зрозумілий інструмент, який відповідає всім їх потребам редагування графіки та інтерфейсу.

Нові користувачі можуть легко навчитися користуватися цим інструментом та освоїти його досить швидко, щоб почати розробляти інтерактивні інтерфейси.

Інструмент також пропонує функції спільного використання моделей, що дозволяють їх переглядати в Інтернеті за простим посиланням. Ця функція є особливо цікавою, оскільки вона дозволяє отримувати відгуки або презентації під час зустрічей. Таким чином, створені моделі та прототипи доступні за цим простим веб-посиланням, що дозволяє їх переглядати та тестувати в простому веб-браузері.

Ще однією спільною функцією, яку пропонує Figma, є експорт коду. Це дозволяє дизайнерам ділитися CSS-кодом веб-макетів із розробниками інтерфейсів, щоб дозволити їм перевірити кожен елемент макету. Також є можливість експортувати код iOS та Android у мобільні додатки для кращого потоку.

Дуже важлива функція, яку ми можемо використовувати завдяки цьому інструменту – це додавання та модифікація коментарів у режимі реального часу. Ця функція особливо цікава, щоб мати швидший зворотний зв'язок та легше обробляти необхідні зміни.

Figma - це потужний інструмент, який поєднує в собі функціональність багатьох інструментів в одному додатку. Але до цього найбільш інноваційною особливістю є здатність Figma будувати систему дизайну:

1. Дизайн. Figma пропонує широкий спектр важливих інструментів для створення та керування векторними графічними елементами, таких як фігури, зображення, текстів, тощо. Отже, це перш за все інструмент графічного дизайну, який дозволяє створювати моделі веб-сайтів та мобільних додатків.
2. Презентація. Презентація моделей пропонується Figma за допомогою простого посилання. Таким чином, макети розроблені за допомогою цього інструменту, можна передавати для презентацій клієнтам або під час

різних зустрічей UX. Просте веб-посилання забезпечує доступ до моделей через браузер.

3. Прототипування. Додавання інтерактивності користувача та взаємозв'язку між різними інтерфейсами додатка також можна реалізувати у Figma. Інструмент дозволяє додавати взаємодії, кнопки що можна натискати та анімацію для створення інтерактивних прототипів. Figma дозволяє в кращому випадку динамічно прототипувати веб-сторінку, але буде важчим у використанні для настільного програмного забезпечення.

4. Експорт із різних розроблених інтерфейсів. Це може бути код CSS для веб-сайтів та веб-програм, а також код iOS та Android для мобільних додатків. Це полегшує співпрацю між розробниками UI та UX, створення інтерфейсів, адаптованих до різних пристроїв

Отже, Figma це веб-додаток, який об'єднує функції чотирьох інструментів. Це робить інструмент потужним і досить повним додатком, який відповідає всім потребам дизайнерів та розробникам інтерфейсу.

Для забезпечення співпраці між різними зацікавленими сторонами одного проекту, Figma надає змогу поділитись всім проектом, або лише одним інтерфейсом з іншою людиною. Ця функціональність корисна у випадку зворотного зв'язку із замовниками, щодо певних інтерфейсів проекту або для контролю прогресу.

Не забувайте, що ця функція спільного доступу полегшує взаємодію вашим співрозмовником, покращує ефективність та сприяє кращій співпраці.

Насправді кожна властивість компоненту обробляється індивідуально для кращої настройки стилю. Ви можете зробити це на рівні кожного екземпляра та створити власні компоненти для своїх проектів.

Також можна поміняти місцями екземпляри для кращого дизайну. Не забувайте також про функціональність "перетягування", що полегшує обмін екземплярами.

Дуже часто ігнорують те, що Figma - чудовий інструмент для роботи у



браузері. У порівнянні зі своїми конкурентами він має багато переваг. Наприклад, Sketch доступний лише як настільна програма для MacOS.

Доступ до Figma можна отримати в будь-якому веб-браузері, що має Інтернет-з'єднання. Тож незалежно від того, перебуваєте ви в дорозі, працюєте віддалено або на зустрічі з клієнтом, ваші проекти завжди будуть під рукою.

Ваші файли будуть зберігатися в хмарі для кращої доступності та миттєвої доступності. Отже, це дозволяє працювати над своїми проектами з будь-якого місця.

Навіть якщо ви не можете отримати доступ до Інтернету, ви можете використовувати версію програми для роботи з комп'ютером. Ви можете встановити її на MacOS або Windows і використовувати в автономному режимі. Документи будуть оновлені після входу в систему.

Отже, з Figma ви можете працювати де завгодно, з Інтернетом або без нього, і це величезна його перевага перед конкурентами.

Для спільного використання графічних ресурсів та забезпечення співпраці між дизайнерами однакової структури, можна створювати та обмінюватися бібліотеками Figma. Вони дозволяють створювати системи дизайну та стандартизувати певні графічні елементи.

Насправді бібліотеки Figma дозволяють спільно використовувати компоненти, а також стилі. Також можна оновити ці елементи, щоб нові версії використовувались автоматично в різних файлах. Отже, така система дизайну дозволить дизайнерам дотримуватися графічних статутів та різних стилів.

Для створення реалістичних прототипів, Figma дозволяє створювати анімації, що ілюструють різні взаємодії. Цей інструмент пропонує додавати анімацію між різними екранами програм для більш плавного використання.

Додавання анімації дозволяє створювати більш реалістичні прототипи, які занурюють користувача в UX. Отже, це дозволяє отримати більш точні та реалістичні відгуки клієнтів, оскільки вони будуть видаватися на основі інтерактивних прототипів.

Підбиваючи підсумки, Figma - це потужний інструмент для проектування

графічних інтерфейсів. Опановуючи його, можна отримати велику перевагу у кар'єрі.

## **Розділ 2.4 Microsoft Azure**

Microsoft Azure - це великий набір хмарних обчислювальних служб, що постійно розширюється, які доступні для бізнесу, розробників, державних установ та всіх, хто хоче створити додаток або запустити бізнес в інтернеті без використання апаратного чи серверного програмного забезпечення. Це найшвидше зростаючий бізнес-сегмент Microsoft за останні роки.

Головна перевага цієї хмари полягає в тому, що вона дозволяє організаціям отримувати доступ до ІТ-ресурсів без необхідності інвестувати в центри обробки даних або керувати серверами.

Як і інші постачальники хмарних послуг, Microsoft Azure дозволяє скористатися перевагами ресурсів хмарних обчислень за запитом. Це економить підприємствам значну економію, усуваючи необхідність будувати центр обробки даних на місці, підтримувати його, оновлювати, охолоджувати та платити за електроенергію.

Ця хмарна платформа також дозволяє спростити використання та адміністрування таких технологій Microsoft, як Windows Server, Active Directory та SharePoint. Тож ІТ-команди можуть зосередитись на інших проектах.

Крім того, Microsoft планує в майбутньому додати на свою платформу функціональність штучного інтелекту. В рамках конференції Microsoft Build 2018 фірма Redmond представила інтеграцію Project Brainwave (система глибокого навчання, розроблена для ШІ в режимі реального часу) з Azure.

Наприклад, RemoteApp дозволяє розгортання програм Windows на різних ОС за допомогою віртуальної машини. Подібним чином платформа об'єднує хмарні версії популярних корпоративних рішень Microsoft, таких як Active Directory або SQL Server.

Інші популярні служби Microsoft Azure включають Azure IoT Suite для

підключення та моніторингу підключених об'єктів. Ця послуга також пропонує функції телеметрії та аналізу.

Послуга HDInsight - це індивідуальне розгортання платформи Big Data Hadoop. Azure Cosmos DB - це розміщена база даних NoSQL для конкретних випадків використання. Azure Media Services надає послуги відтворення, індексації, перекодування та захисту вмісту файлу.

Загалом Microsoft Azure пропонує понад двадцять різних категорій послуг. На додаток до обчислювальних, сховища, веб та мобільних служб розробки, існують також послуги інтеграції, створення мереж, контейнеризації та аналізу даних.

Американський гігант також пропонує хмарні послуги для нових технологій, таких як штучний інтелект, блокчейн або змішана реальність.

На додаток до цих послуг, Microsoft співпрацює з постачальниками обладнання, такими як Lenovo, Dell, EMC, HP Enterprise та Huawei, пропонуючи Azure Stack – рішення, призначене для розгортання гібридних хмар. Зокрема, це дозволяє компаніям використовувати програми Azure Cloud у поєднанні з локальними даними.

Все більше компаній, які вже використовують технології Microsoft, такі як Windows Server та Active Directory, звертаються до Azure. Дійсно, підтримка Windows Server закінчилась у 2008 році, і перехід до хмари може бути вигіднішим, ніж інвестування в новий сервер та нову ліцензію Windows Server.

Це хмарне рішення особливо цікаве для малих підприємств, які не мають достатнього капіталу для обладнання та витрат, пов'язаних із розгортанням на місці або для оренди виділених серверів у традиційному центрі обробки даних. Система виставлення рахунків Azure надає їм більшу гнучкість.

Однією з головних його сильних сторін є легкість переходу до хмари для багатьох компаній, які вже використовують продукти Microsoft на місцях. Хмарні продукти тісно інтегровані з їх локальними версіями, що полегшує перехід до хмари.

Замість того, щоб встановлювати локальні сервери або орендувати фізичні

сервери в традиційних центрах обробки даних, Microsoft Azure пропонує своїм клієнтам систему виставлення рахунків, засновану на споживанні ресурсів.

Ціни варіюються в залежності від типів послуг та використовуваної місткості, а також фізичного розташування екземплярів Azure. Наприклад, ціна сховища варіюється залежно від вибраних варіантів надмірності та розподілу. Таким чином, компанії можуть заощадити гроші, платячи лише за те, що вони споживають.

Незалежно від того, що циніки нещодавно глузували з концепції хмарних обчислень, заявляючи, що "хмара — це просто чужий комп'ютер", ця критика зараз, здається, втрачає сенс. Але це спрощення описує лише невелику частину бізнесу Azure, який визначається як бізнес інфраструктура як послуга (IaaS), де служби в хмарі замінюють фізичне обладнання.

Повний набір служб Microsoft Azure охоплює набагато більше, ніж просто переміщення локальних серверів у хмару. На додаток до ресурсів IaaS, у вас є повний набір опцій Платформа як послуга (PaaS) та Програмне забезпечення як послуга (SaaS), що дозволяють вашому бізнесу отримувати доступ до хмарних служб без необхідності управління сервером. Наприклад, ви можете створити веб-сайт на основі WordPress або створити сайт на базовому вузлі JS без необхідності налаштовувати (або виправляти) базовий сервер Windows або Linux.

Крім того, розробники програм, веб-сайтів та служб можуть використовувати сховище та послуги Azure як щит, не турбуючись про безпеку або надійність базової інфраструктури.

Microsoft анонсувала Azure у 2008 році. Дебют відбувся через два роки, у січні 2010 року, як Windows Azure, а у 2014 році він був перейменований у Microsoft Azure. Зміна назви, яка далеко не зводиться до лише косметичної. Натомість це було зображенням широти хмарних служб Azure, яка виходить далеко за рамки лише пропозицій на базі Windows.

Наприкінці 2017 року Microsoft фактично повідомила, що 40% усіх віртуальних машин Azure працювали під управлінням Linux, проти з менш ніж третьою роком раніше. Глобальна інфраструктура Azure включає центри обробки

даних у 54 регіонах, що охоплюють 140 країн. Microsoft також підтримує два регіони Azure Government Secret, які знаходяться в нерозкритих місцях.

Найбільш очевидна перевага пропозицій IaaS Azure полягає в тому, що вашому бізнесу не потрібно купувати, налаштовувати, підтримувати та ремонтувати обладнання для запуску робочих навантажень у хмарі. Економія починається з вартості обладнання, але охоплює значно більшу кількість непрямих витрат, включаючи фізичний простір, необхідний для розміщення цих серверів, а також електроенергію, необхідну для їх роботи.

Оскільки ресурси на базі Azure є віртуальними, вони не можуть несподівано вийти з ладу та спричинити простої, поки ви чекаєте на ремонт чи заміну. Віртуальні апаратні ресурси можуть збільшуватися або зменшуватися так, як фізичне обладнання не може, що допомагає справлятися з раптовими стрибками трафіку на веб-сайті, що базується на Azure.

Великі компанії, яким потрібно відповідати загальним вимогам щодо конфіденційності щодо зберігання та передачі даних, можуть легко переносити дані та послуги в регіон, який вони обрали.

Для розробників Azure надає миттєвий доступ до розробки мобільних додатків, дизайну пристроїв IoT, підключення до Інтернет-сховища та ресурсів баз даних та розгортання контейнерної технології. Крім того, Microsoft інвестувала значні кошти в машинне навчання та засоби штучного інтелекту для розробників.

Одним із найосновніших випадків використання Azure IaaS є заміна фізичного сервера на віртуальний сервер, що працює в центрі обробки даних Azure, таким чином усуваючи потребу в технічному обслуговуванні.

Цей сервер може запускати будь-яку підтримувану версію робочого столу або Windows. Або ви можете вибрати довгий перелік дистрибутивів Linux, також у широкому діапазоні підтримуваних версій, включаючи Ubuntu Server, Red Hat Enterprise Linux, FreeBSD і навіть Oracle Linux. На величезному ринку Azure ви можете знайти готові до використання віртуальні сервери для майже будь-якого завдання, включаючи SQL Server, Docker, SAP Hana та (для використання з цим сервером Oracle Linux) Oracle Database.

Зрештою Azure дозволяє вам створити повну інфраструктуру віртуального робочого столу (VDI) у хмарі Azure і керувати нею за допомогою сторонніх інструментів.

Для зовсім іншого підходу звернімо увагу на Nerdio для Microsoft Azure, який дозволяє адміністраторам створювати цілу корпоративну мережу та керувати нею за допомогою сторонніх веб-порталів, які компанія виставляє як «ІТ як послугу».

На конференції Ignite, що відбулася наприкінці 2018 року, Microsoft оголосила про власну службу VDI на базі Azure під назвою Windows Virtual Desktop. Він підтримує Windows 10 у багатокористувацьких конфігураціях, доступних на будь-якому пристрої, замінюючи локальну віртуалізацію сервера.

Він також доступний на віртуальних робочих столах Windows 7, маючи одну переконливу перевагу для підприємств, які борються з планами переходу на Windows 10: три роки безкоштовних розширених оновлень безпеки після офіційного завершення підтримки Windows.

Він також включає приголомшені функції безпеки, такі як доступ до віртуальних машин у режимі реального часу, який блокує віртуальні машини на рівні мережі та блокує вхідний трафік, за винятком випадків, коли затверджуються конкретні запити на доступ.

Azure є другим за величиною постачальником хмарних обчислень, набагато відстаючи від Amazon Web Services (AWS), але значно випереджаючи своїх конкурентів. Інші конкуренти Azure включають Google Cloud Platform пропонує подібний набір інфраструктури та сервісів програм, заснованих на загальній інфраструктурі пошукового гіганта.

Інші компанії, включаючи Salesforce та Oracle, пропонують підмножину хмарних сервісів, орієнтованих переважно на наявних клієнтів цих компаній.

Спектр послуг, доступних на Azure, дуже великий. Починаючи з місткості для дуже важливих даних. Насправді, зберігання величезних обсягів даних, структурованих чи неструктурованих, є перевагою Azure.

Послуги зберігання даних Azure включають Bloke Azure, файли Azure, Таблиці Azure, та сховище NoSQL для структурованих даних.

Блоке Azure, для неструктурованих даних, містить передачу зображень, документів та відео файлів безпосередньо до браузера);

Файли Azure - це спільні файлові файли на основі хмари, доступ до яких здійснюється за допомогою протоколів стандартного блоку повідомлень (SMB);

Для переміщення баз даних, побудованих на SQL Server, існує Azure SQL Database, повністю керована служба, яка може використовуватися як керований екземпляр для перенесення локальних робочих навантажень або створення з нуля для надання бази даних SQL як послуги. Не кажучи вже про Azure Cosmos DB, дуже велику ставку Microsoft на великі дані. Також вона представлена Microsoft як повністю керована, глобально розподілена, масштабована по горизонталі, багатомодульна служба бази даних з точки зору зберігання та пропускну здатності, що підтримується угодами про рівень обслуговування (SLA).

Розробники настільних та мобільних додатків мають повний набір інструментів для створення та розгортання цих програм, починаючи із середовища розробки Visual Studio, яке поставляється в декількох версіях. На Windows Server та Windows 10 Enterprise N віртуальних машинах.

Окрім служб Visual Studio Team Services та Azure DevOps, Azure включає широкий вибір сторонніх засобів розробки для спільного використання коду, управління робочим процесом, розгортання програмного забезпечення та моніторингу продуктивності та моніторингу. Наприклад, ви можете використовувати Jenkins для створення додатків у хмарі та їх розгортання безпосередньо в Azure. Використовуйте Terraform або Ansible для налаштування інфраструктури, а потім керуйте всім цим за допомогою Chef Automate.

Контейнери — це стандартизовані, інкапсульовані середовища, що дозволяють додаткам працювати безпечно, з високою доступністю та можливістю швидкого масштабування.

Azure Marketplace дозволяє особливо легко розгортати та масштабувати зображення контейнерів. Стандарт управління робочими навантаженнями в контейнерах — це служба організації контейнерів Kubernetes, доступна в Azure як служба Azure Kubernetes (AKS).

Наскільки важливим є контейнеризація для майбутнього хмарних робочих навантажень? Можна сказати, що Microsoft повністю переробила всю свою філософію серверної системи навколо Kubernetes і найняла кількох її ключових творців. Azure пропонує на своєму ринку понад 100 зображень контейнерів, а також інструменти Docker та інших для управління цими зображеннями.

## 2.5 Photoshop

Photoshop - програмне забезпечення для обробки зображень та фотографій та ретушування, виготовлене компанією Adobe. Він використовується на комп'ютерах з ОС Windows або MacOS. Photoshop став стандартом для управління растровими зображеннями. Растрові зображення складаються з сітки точок, званих пікселями. Цікавість цих зображень полягає у відтворенні тонких градацій кольорів.

Його основними функціональними можливостями програми є:

- дизайн виділених елементів у різних режимах, що дозволяє з великою точністю вибрати частини зображення, що підлягають модифікації;
- безліч інструментів, що дозволяють обрізати зображення повністю, або частково, для редагування;
- створення різноманітних масок, що дозволяють виконувати монтаж між різними візуальними елементами;
- багато фільтрів, що дозволяють створювати спецефекти.

Це найбільш широко використовуваний програмний інструмент для редагування фотографій, маніпуляцій із зображеннями та ретуші для багатьох форматів файлів зображень та відео. Інструменти Photoshop дозволяють редагувати як окремі зображення, так і великі партії фотографій. Існує кілька версій Photoshop, включаючи Photoshop CC, Photoshop Elements та Photoshop Lightroom.

Adobe Photoshop є важливим інструментом для веб-розробників, графічних дизайнерів, фотографів та творчих професіоналів.



Також у Photoshop можна створювати макети веб-сайтів, а їх конструкції можна допрацьовувати до того, як розробники приймуть етап кодування. Окрему графіку можна створювати та експортувати для використання в інших програмах.

Photoshop CC – (Creative Cloud Photoshop) це вдосконалене програмне забезпечення для обробки зображень, яке використовується дизайнерами, веб-професіоналами, відеоредакторами, професіоналами у галузі комунікацій або рекламними агенціями, та фотографами для модифікації цифрових зображень або управління ними. Потужність його інструментів залишає повну свободу творчості вашій уяві та дозволяє вносити надзвичайно точні зміни до всіх видів знімків.

Існує кілька способів вивчити Adobe Photoshop. Найпопулярніші методи включають особисте відвідування уроків Photoshop, онлайн-уроків, навчання за допомогою онлайн-посібників Photoshop та книг. Заняття розроблені, щоб допомогти студентам скористатися як груповими навчальними заходами, так і індивідуальними інструктажами. Навчання в класі також допомагає учням долати виклики чи перешкоди за допомогою керованого навчання. Ці можливості навчання особливо корисні, коли випускаються нові функції чи інструменти.

Adobe Photoshop може зберігати або експортувати зображення та графіку у різноманітних форматах файлів.

Зображення можуть містити лише пікселі, тоді як інші можуть містити суміш пікселів та векторів. Деякі формати файлів також використовують методи стиснення для зменшення розміру зображення, а деякі варіанти стиснення навмисно видаляють дані, щоб зменшити розмір файлу. Для збереження всіх функціональних можливостей Adobe Photoshop, включаючи ефекти, маски та шари, використовується формат файлу Photoshop (PSD).

Розроблений спеціально для комп'ютерів, Photoshop доступний у 32 або 64 бітах залежно від архітектури вашої операційної системи. Він також є крос-платформним, оскільки доступний як у Windows, так і в MacOS. Інтернет-версія, доступна через будь-який веб-браузер, також існує. Це Photoshop Express.

Якщо Photoshop загалом отримує позитивні відгуки від своїх користувачів, це тому, що він має стільки переваг. Незважаючи на свій професіоналізм, він не для

всіх. Більше того, це не ідеальне програмне забезпечення, хоча з кожним новим оновленням воно виправляє свої попередні недоліки.

Завдяки багатьом параметрам, які воно пропонує, програмне забезпечення Adobe Photoshop широко використовується в мультимедіа та Інтернеті. Найвідоміший, завдяки своїй здатності виправляти фізичні недоліки. Наприклад, на обличчі можна буде видалити невеликі плями або прищі за допомогою спеціальної щітки. Завдяки вдосконаленому алгоритму Photoshop результат є цілком природним.

У професійному середовищі це інструмент, який найчастіше використовується для моделювання флаєрів, логотипів чи візиток. На веб-сайті це може бути джерелом багатьох присутніх графічних елементів, що дозволяє надати сайту індивідуальність. Неможливо процитувати всі можливі варіанти використання Photoshop, оскільки вони нескінченні.

### 3. ЗАГАЛЬНІ ВІДОМОСТІ ДОДАТКУ

#### 3.1 Розробка концепції

Аналізуючи актуальність додатку, для найефективнішого розповсюдження було прийняте рішення його розробки на мобільних пристроях, наступним етапом стане проголошення концепції.

Цільова аудиторія – підлітки та студенти, бо вони обожають доводити свою точку зору яка на їх думку є істиною, й частіше за інших втрапляють у конфліктні ситуації.

Головна ідея цього додатку також полягає в урегулюванні й контролю конфліктів, тож кожна людина не зважаючи на свій вік має змогу ним користуватися, вирішуючи свої проблеми, або весело проводячи свій час.

Короткий опис застосування – цей мобільний додаток націлений на контроль суперечки між користувачами що хочуть закласти парі, та на самоконтроль. За його допомогою ви маєте можливість кинути виклик іншому користувачу, описавши сам виклик, вказавши на час що виклик буде актуальним, додати користувачів що будуть стежити за вашою дуеллю, та й звісно, написати що буде у разі вашої перемоги, або поразки. Також ви маєте можливість посісти місце судді, що буде наглядати за протистоянням інших користувачів. Одним із найважливіших функцій цього додатку є змога до власної мотивації, створюючи особливе парі, за допомогою якого ви маєте змогу викликати себе, також накладаючи певні рамки та угоди.

Головні вимоги додатку:

- профіль, у якому зображується кількість виконаних, та не виконаних парі;
- список друзів;
- створення виклику на змагання іншого користувача;
- список укладених парі;
- можливість кинути виклик собі;

- інформування користувачів з приводу закінчення дуелі;
- здатність відсудити парі інших користувачів.

Унікальність додатку:

- система суддів, за допомогою якої користувачі можуть не хвилюватися щодо контролю їх угоди;
- поєднання різних можливостей для створення парі, з собою або користувачем, також з судьями, або без них;
- єдиний додаток для контролю парі.

### **3.2 Проектування структури та створення додатку**

Для розробки проекту мобільного додатку, було проаналізовано багато інструментів, як наслідок було вирішено використовувати Figma, що є потужним інструментом для проектування графічних інтерфейсів.

Перш за все нам потрібно визначитись із основними елементами інтерфейсу, щоб потім про них не забути.

Головні елементи інтерфейсу додатку:

1. Кнопка для надання інформації про додаток.
2. Логотип додатку.
3. Текстове поле для логіну.
4. Текстове поле для паролю.
5. Кнопка реєстрації.
6. Кнопка входження до профілю.
7. Текстове поле для дублювання паролю.
8. Кнопка головної сторінки.
9. Кнопка додання.
10. Кнопка відмови.
11. Текстове поле пошуку користувачів.
12. Кнопка складених парі.
13. Кнопка перегляду детальної інформації.

14. Кнопка повернення.
15. Кнопка профілю.
16. Кнопка створення парі для себе.
17. Кнопка вибору друга.
18. Кнопка перегляду парі.
19. Кнопка для відсудження парі.
20. Кнопка для вибору переможця.
21. Календар для вибрання дати.
22. Текстове поле виклику.
23. Текстове поле у разі перемоги.
24. Текстове поле у разі поразки.
25. Кнопка укладення парі.
26. Кнопка запрошення користувача як суддю.

Визначившись із головними елементами інтерфейсу можна починати працювати у Figma. Для початку роботи з цим інструментом, потрібно перейти на його головний сайт, зареєструватися та створити новий файл дизайну.



Рисунок 3.1 – Прототип початкової сторінки додатку

На початковій сторінці має бути можливість увійти користувачу що вже реєструвався, зареєструватися новому користувачу, та при натисканні на кнопку зі знаком запитання, отримати інформацію про додаток, що й було реалізовано (рис 3.1).

При натисканні на кнопку реєстрації, повинні переходити на нове вікно, де ми й будемо вводити свій логін, пошту та пароль, тож нам потрібно поєднати нову сторінку з початковою. Це можна реалізувати у Figma за допомогою обрання кнопки реєстрації, та перетаскуванні стрілки до потрібної нам сторінки. Так ми створили посилання з кнопки на іншу сторінку. Щоб переглянути усі посилання, змініть вікно дизайн на прототип у Figma.

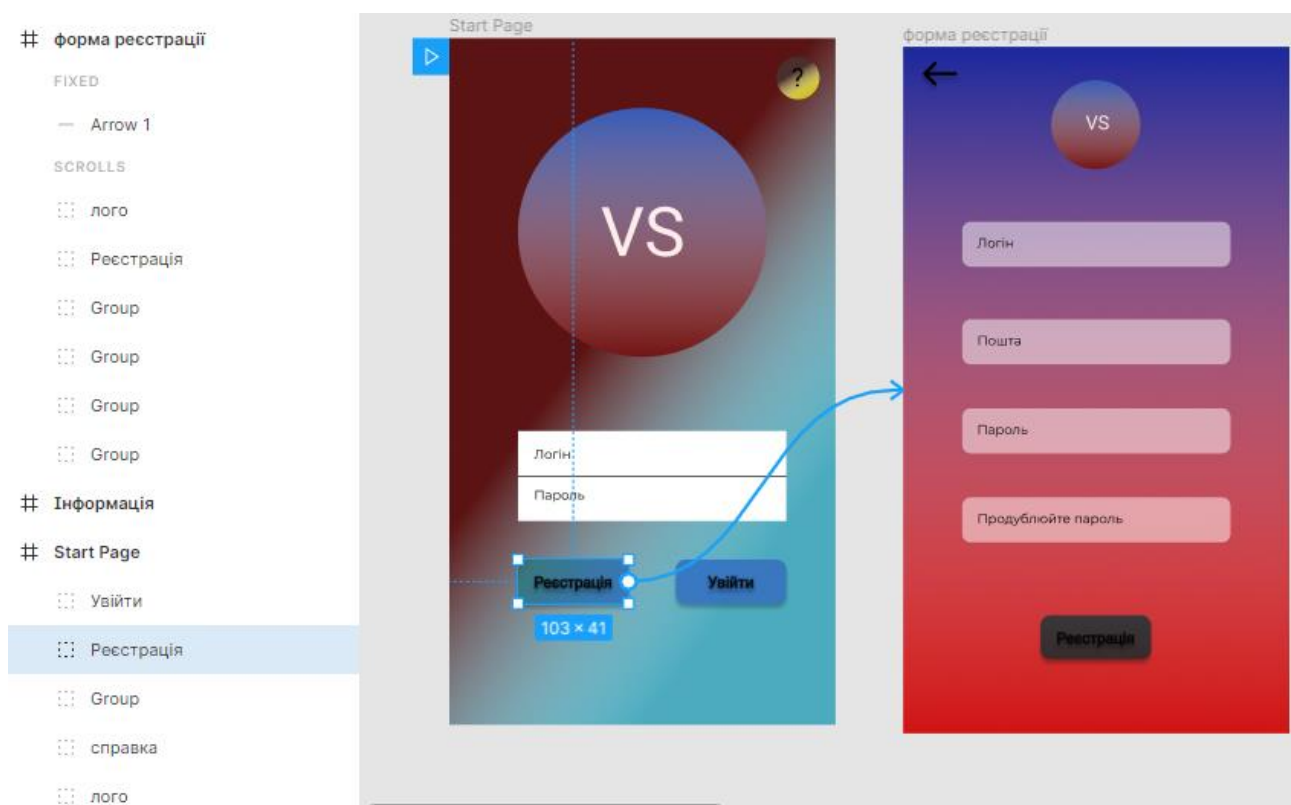


Рисунок 3.2 – Зображення посилання кнопки реєстрації на початкову сторінку

Створення головної сторінки – це найвідповідальніша частина, бо інтерфейс повинен бути інтуїтивно зрозумілим для нового користувача, щоб він одразу не вийшов із додатку.

Головна сторінка повинна бути багатофункціональною, але у той же час простою, із графічної точки зору. Оскільки це перша сторінка, не враховуючи автентифікацію, то ми повинні створити кнопки для швидкого переходу між основними можливостями додатку. Ці кнопки краще за все розташувати у нижній частині екрану, бо так користувачу буде легше на них натискати пальцями рис (3.3).

Основними елементами інтерфейсу головної сторінки будуть 4 кнопки:

- парі для себе – створення угоди із самим собою;
- кинути виклик другу, що дозволяє обрати користувача що є у вас в друзях і кинути йому запрошення на парі;
- переглянути виклики, які інші користувачі вам відправляють;
- відсудити парі, тобто обрати переможця у суперечці, де вас вибрали у ролі судді.

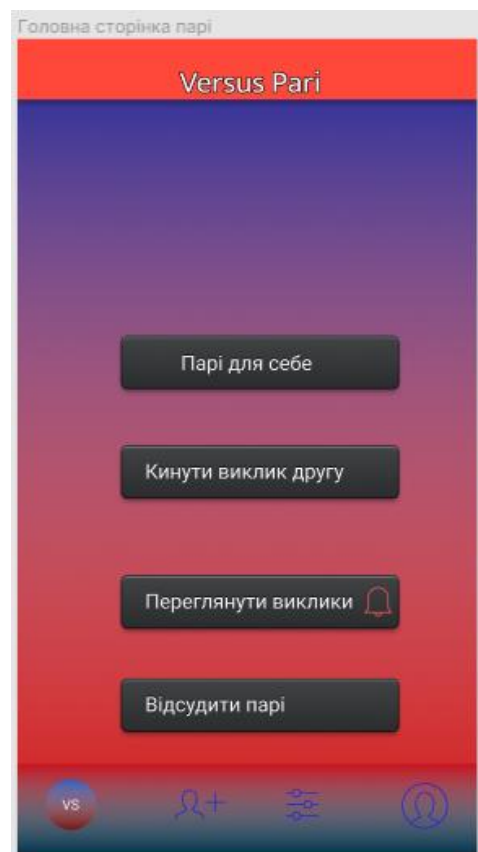


Рисунок 3.3 – Прототип головної сторінки додатку

Якщо користувач зайшов перший раз, то він має додати інших користувачів, бо без цього він не буде здатний викликати гравця на парі, але зможе скласти парі для себе (рис 3.4.).

Парі для себе

Versus Parí

Виклик

Дедлайн

Jan 2020

Mo	Tu	We	Th	Fr	Sa	Su
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Що відбудеться у разі перемоги

Що відбудеться у разі програшу

Укласти парі

Рисунок 3.4 – Вигляд форми для заповнення парі для себе

Цій формі ми спочатку заповнюємо виклик, тобто що саме ви хочете зробити. Далі обираємо дедлайн, тобто скільки у нас є на це часу. У наступні два текстових поля ви прописуєте винагороду за виграш, або покарання за програш. По закінченню заповнювання форми, вам прийде повідомлення про закінчення реєстрації, та запропонує вам переглянути їх. При переході до



ваших парі, ви побачите прийняті вами угоди, а натиснувши на кнопку перегляду, у вас відкриється сторінка з інформацією про суперечку (рис 3.5).

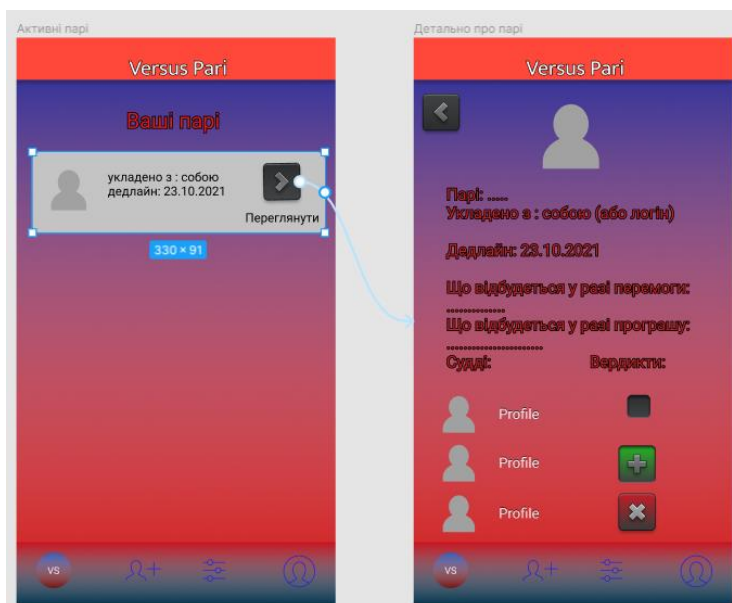


Рисунок 3.5 – Перехід до сторінки з описом деталей ухваленого парі

Для додавання ж у друзі, потрібно натиснути на другу кнопку, що знаходиться у інтерактивній панелі, написати логін друга та натиснути на пошук. Там з'явиться його профіль, потрібно буде лише відправити запрошення у друзі. Він з'явиться у вашому списку друзів, одразу після прийняття запиту у друзі (рис 3.6).

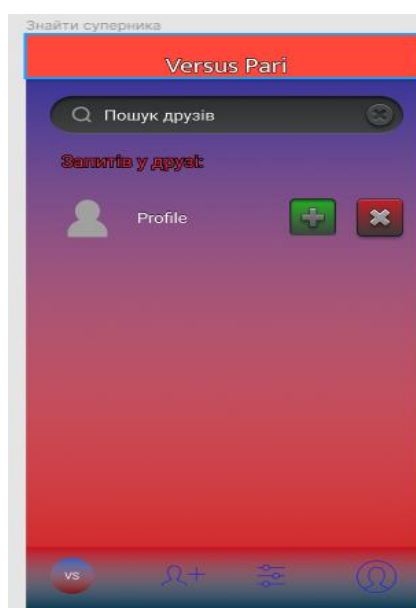


Рисунок 3.6 – Перехід до сторінки з описом деталей ухваленого парі

Список ваших друзей буде відображатися у вашому профілі, разом із фотографією, логіном, та рахівником ваших перемог та поразок (рис 3.7).

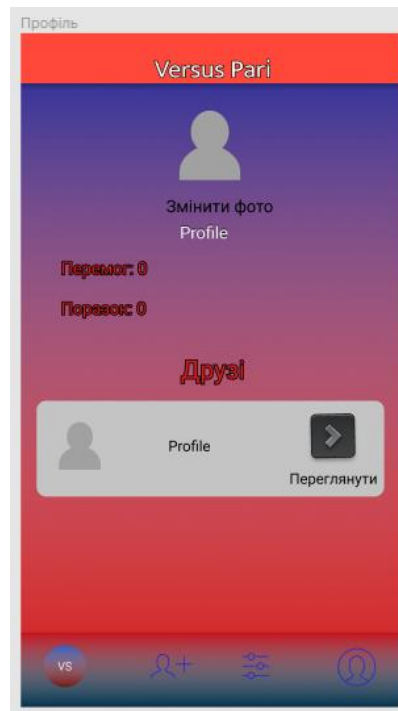


Рисунок 3.7 – Профіль користувача

При виклику на парі іншого користувача, у вас буде лише одна відмінність від форми для заповнення парі для себе (рис 3.4), це наявність можливості додання суддів (рис 3.8).

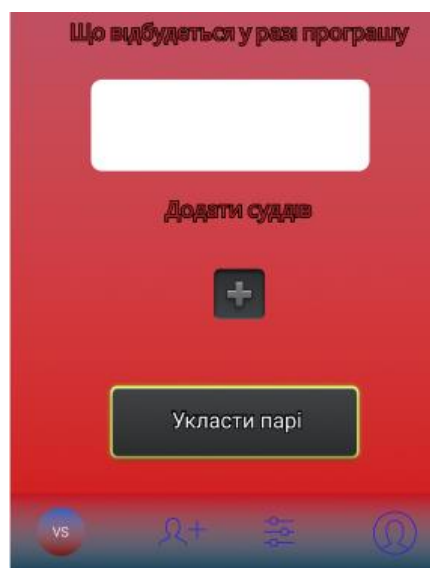


Рисунок 3.8 – Реалізація можливості додати суддів

Якщо користувача запросили наглядати за певним парі, то у разі погоди з запрошенням, у нього з'являється нова форма парі (рис. 3.9).

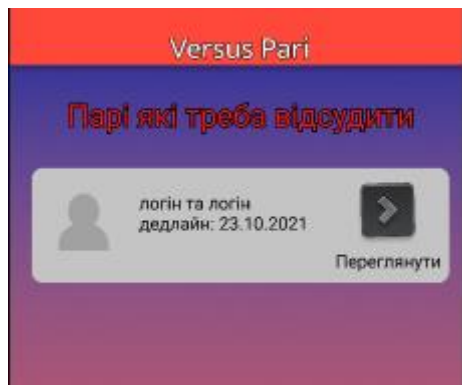


Рисунок 3.9 – Перелік парі, які потрібно переглянути

При натисканні на кнопку перегляду, він має вирішити хто з супротивників виконав завдання, та переміг у сутичці (рис. 3.10), проаналізувавши парі.



Рисунок 3.10 – Форма парі, яку треба відсудити

Була реалізована змога перегляду парі що надійшли вам від друзів. Якщо на головній сторінці (рис. 3.3) натиснути на перегляд парі, то побачимо логін користувача що викликав нас. Після цього, переглядаємо його виклик (рис. 3.11).



Рисунок 3.11 – Виклик на парі, який надійшов від іншого користувача

Можна побачити логін нашого друга, термін парі, умови та суддів. Ми можемо або погодитися, й виклик потрапить у список ваших парі, або просто від нього відмовитись.

Таким чином, ми створили повноцінний прототип, за допомогою якого буде набагато легше реалізувати інтерфейс створюваного додатку, та прослідкувати зв'язки між сторінками.

Після створення прототипу розпочинаємо розробку додатку з встановлення обраного середовища розробки Microsoft Visual Studio 2019 Community з офіційного сайту Microsoft.

Після завантаження, потрібно створити новий проект, у який й буде нашим додатком після завершення розробки.

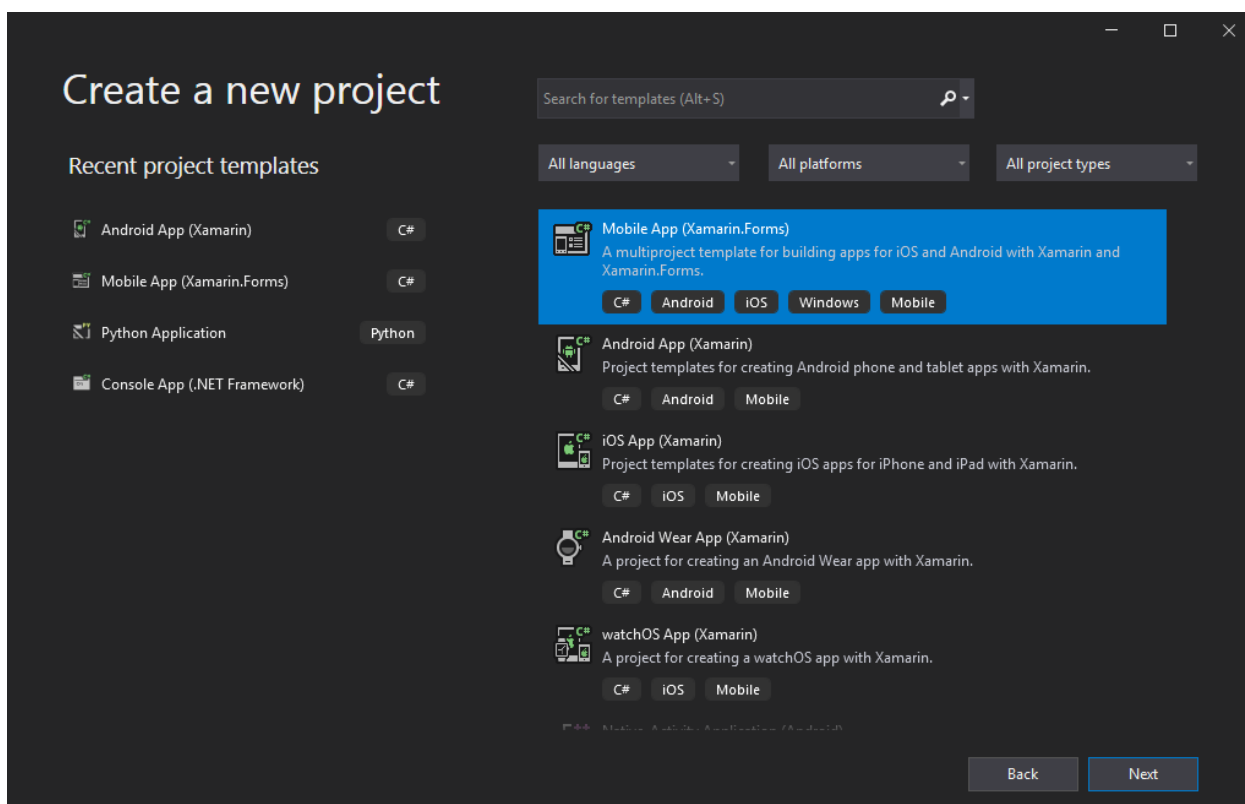


Рисунок 3.12 – Обираємо шаблон для створення додатку

Потрібно обрати Mobile App (Xamarin.Forms) (рис. 3.12), бо у майбутньому, це дасть нам змогу дуже легко розробити додатки й на IOS та Windows phone.

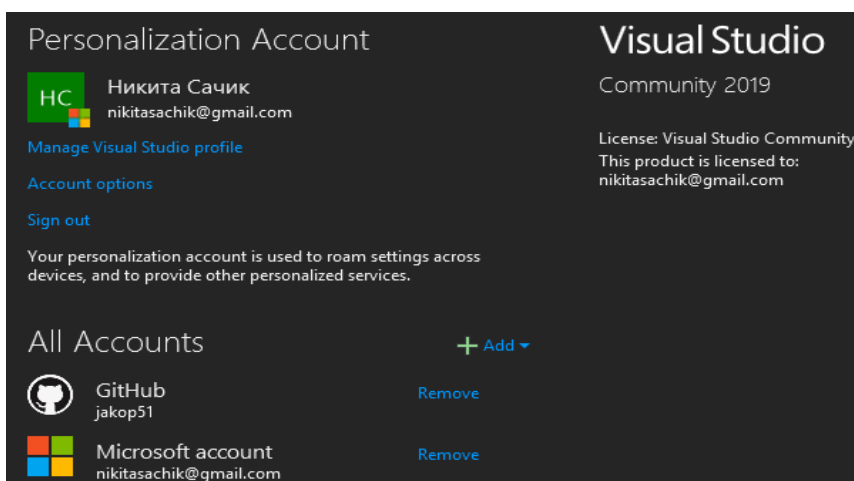


Рисунок 3.13 – Результат вдалого підключення до GitHub

Далі створюємо обліковий запис у GitHub, після його потрібно перейти до Visual Studio, натиснути на свій профіль, обрати налаштування, та натиснути на кнопку додавання профілю. Після цього приєднуємося до свого GitHub профілю, й він у нас має відобразитись у списку підключених профілів (рис 3.13).

На нашому профілі GitHub, створюємо новій репозиторій, та копіюємо посилання на нього. Повертаємося до коду, й створюємо нове вирішення, після чого вставляємо посилання на наш репозиторій, та публікуємо у GitHub.

Переходимо до GitHub, та у нашому репозиторії бачимо, що наш проект вдало завантажився (рис 3.14), і надалі ми зможемо слідкувати за його змінами.

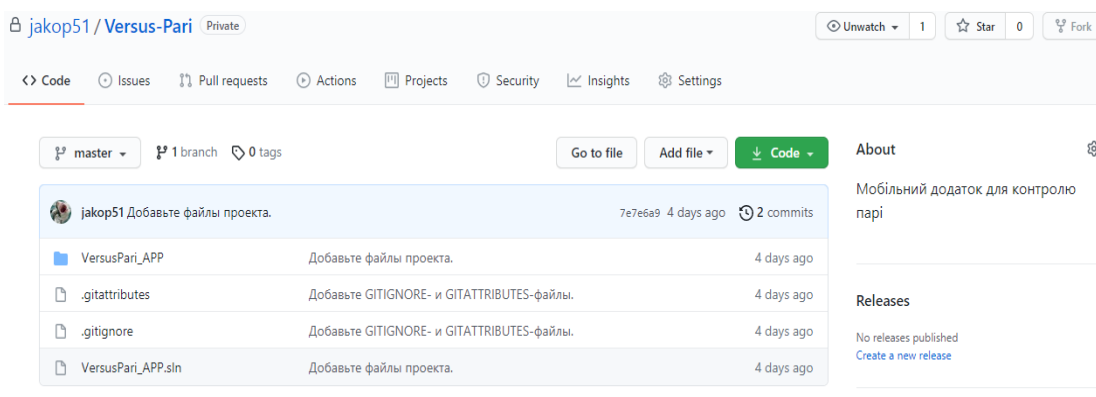


Рисунок 3.14 – Результат завантаження проекту до GitHub

Далі потрібно створити елементи інтерфейсу. Це буде дуже легко, оскільки ми їх вже прописали у прототипі, але треба їх покращити, зробити дійсно гарними. Після їх створення, нам залишиться лише додати їх у проект, із чим нам допоможе впоратись XAML.

Для створення елементів інтерфейсу, ми будемо використовувати Photoshop, бо він має безліч можливостей для створення та редагування фотографій.

Відкриваємо Photoshop, створюємо новий проект, вибираючи його

розмір, та починаймо його робити. Першим я створив логотип (рис. 3.15).

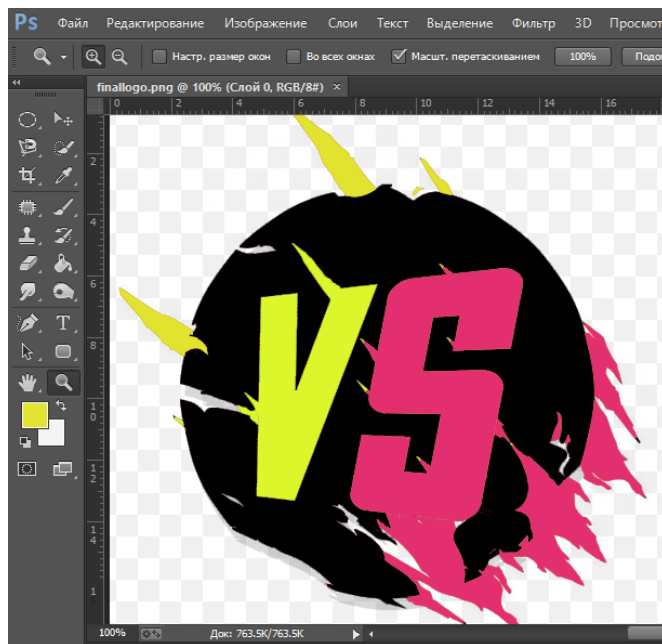


Рисунок 3.15 – Результат створення логотипу за допомогою Photoshop

Далі відповідно до прототипу початкової сторінки, прописуємо її за допомогою XAML (поля та кнопки для підключення та реєстрації реалізуємо після підключення до бази даних), підключаємо телефон як емулятор, та дивимось на результат (рис. 3.16).

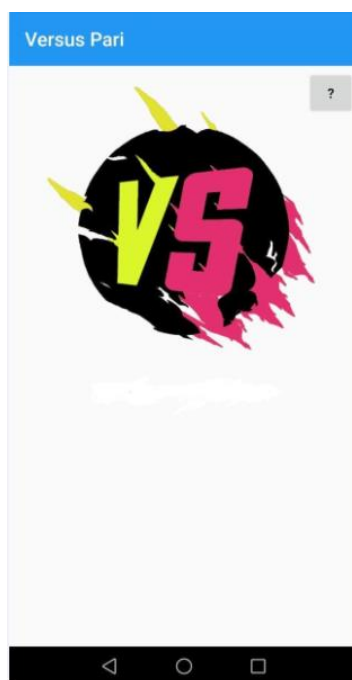


Рисунок 3.16 – Початкова сторінка без реєстрації та входу

Нам потрібно розробити базу даних, отже створюємо UML діаграму бази даних нашого мобільного додатку (рис. 3.17).

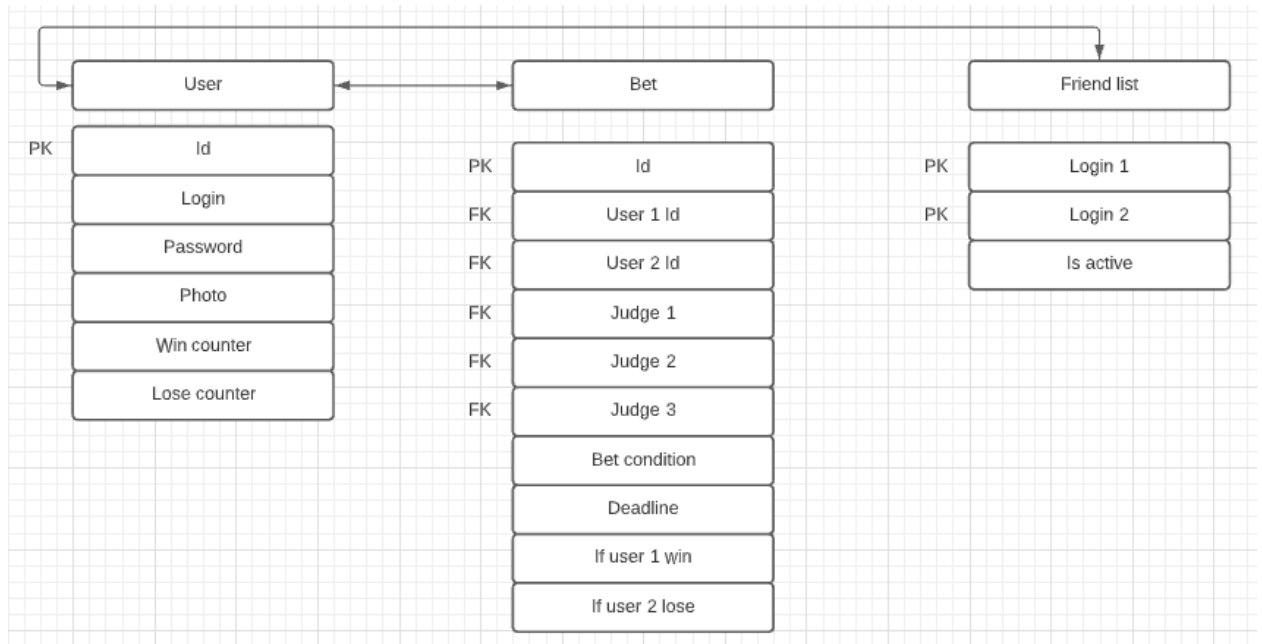


Рисунок 3.17 – UML діаграма бази даних мобільного додатку

Нам залишається лише зареєструватися в Azure, створили там базу даних за нашою діаграмою, й підключити її до Visual Studio.

Далі було розроблено додаток, крок за кроком повторюючи прототип, прописуючи логіку, використовуючи створені елементи дизайну та розташовуючи їх за допомогою XAML.



## ВИСНОВКИ

У результаті виконання даної дипломної роботи було розроблено мобільний додаток, за допомогою якого можна створювати та контролювати пари. У роботі описується шлях створення мобільного додатку від ідеї, її аналізу, до її втілення у життя.

1. Під час роботи над дипломним проектом, було досліджено конфлікти, знайдено найкращі шляхи їх вирішення, проаналізовано актуальність цієї теми, та визначено її наукову новизну. Було зроблене рішення створити саме мобільний додаток, через його розповсюдження, найкращу відповідність вимогам мого додатку й найбільшим потенціалом успіху в майбутньому. Для покращення розуміння можливостей та функцій, які реалізуватиме мій додаток, був проведений аналіз конкурентів.

2. Було проведено аналіз інструментів та програмних засобів реалізації, які найліпше впораються зі створенням графічного інтерфейсу та усього функціоналу додатку. У ході аналізу було обрано використання C# як мову програмування, Figma для створення прототипу, Microsoft Visual Studio 2019 як середовища розробки, Xamarin для кросплатформної розробки мобільного додатку, Photoshop для створення елементів графічного дизайну, Microsoft Azure як хмарну обчислювальну службу з розташуванням бази даних, та GIT для контролю версій.

3. Визначення концепції, вимог та головних елементів інтерфейсу мобільного додатку. Далі був розроблений детальний прототип, який дозволяє переглянути увесь функціонал та графічний інтерфейс. Було створено UML діаграму бази даних мобільного додатку.

4. Розробка графічного інтерфейсу за допомогою XAML і Photoshop, підключення проекту до GIT, зв'язок з Azure, та створення там бази даних, написання логіки додатку за допомогою мови програмування C# у

середовищі розробки Microsoft Visual Studio 2019 з використанням Xamarin.Forms.

Розроблений додаток має сприяє створенню та врегулюванні конфліктів, своєчасному нагадуванні про закінчення парі, інформування про їх результати, та здатність сприяти мотивації користувача.

Створений мобільний додаток має значні перспективи подальших досліджень. За допомогою Xamarin можна легко адаптувати додаток під платформи IOS та Windows phone, що значно збільшить кількість аудиторії. Можна буде доповнити функціонал, додавши можливість складати парі з випадковими користувачами, рейтинг користувача, можливість завантажувати файли з аудіо та відео змістом для підтвердження виконання умов парі.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Ed Snider Mastering Xamarin.Forms – Second Edition /- Packt Publishing – c. 56 – 62.
2. John Bach Xamarin: The Ultimate Beginner’s Guide to Learn Xamarin Step by Step /- mEm Inc – c. 14 – 26.
3. Can Bilgin Mobile Development with .NET: Build cross-platform mobile applications with Xamarin.Forms 5 and ASP.NET Core 5 – 2nd Edition /- Packt Publishing – c. 348 – 360.
4. Johan Karlsson, Daniel Hindrikes, Xamarin.Forms Projects: Build seven real-world cross-platform mobile apps whith C# and Xamarin.Forms /- Packt Publishing – c. 47 – 59.
5. Adam Nathan Universal Windows Apps with XAML and C# Unleashed /- Sams Publishing – c. 104 – 130.
6. Dan Hermes, Dr. Nima Mazloumi, Building Xamarin.Forms Mobile Apps Using XAML: Mobile Cross-Platform XAML and Xamarin.Forms Fundamentals /- Apress – c. 116 – 125.
7. Dmitry Anoshin, Dmitry Foshin, Roman Storchak, Xenia Ireton, Azure Data Factory Cookbook: Build and manage ETL and ELT pipelines with Microsoft Azure's serverless data integration service /- Packt Publishing – c. 255- 283.
8. Andrew Troelsen, Phil Japikse, Pro C# 7: With .NET and .NET Core /- Apress – c. 403 – 420.
9. Mohamed Wali Learn Microsoft Azure: Build, manage, and scale cloud applications using the Azure ecosystem /- Packt Publishing – c. 281 – 289.
10. Daniel Hindrikes, Johan Karlsson, Xamarin.Forms Projects: Build multiplatform mobile apps and a game from scratch using C# and Visual Studio 2019 /- Packt Publishing – c. 329 – 334.

## ДОДАТКИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
 ТЕХНОЛОГІЙ  
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### «Розробка додатку для контролю парі під систему Android з використанням C#.net»

Виконав студент 4 курсу  
 групи ПД-41  
 Сачик Микита Олександрович  
 Керівник роботи  
 Доктор філософії  
 Дібрівний Олесь Андрійович

Київ – 2021

#### МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ТА НАУКОВА НОВИЗНА ДОСЛІДЖЕННЯ

**Мета роботи** – розробка мобільного додатку для контролю парі.

**Об'єкт дослідження** – процес регулювання конфліктів.

**Предмет дослідження** – технології, за допомогою яких можливо відстежити дискусію.

**Наукова новизна проекту** – створення унікального мобільного додатку для контролю дискусій, з використанням системи суддів, які наглядатимуть за змаганням, та поєднанням різних можливостей для укладання парі.

## АКТУАЛЬНІСТЬ РОБОТИ

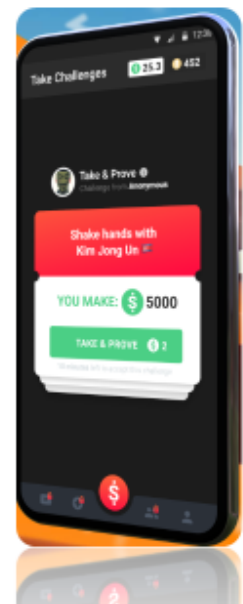
- соціальні мережі;
- мотивація;
- регулювання конфліктів

3

## АНАЛОГИ

### Dare App

Переваги	Недоліки
Гарний інтерфейс	Погана реалізація контролю завдань
Можливість завантажувати відео	Хоча додаток описаний як безкоштовний, за основний функціонал потрібно платити
	Більшість заявлених функцій не працюють, або працюють інакше
	Розробники не виплачують обіцяні гроші
	Не працює



4

## ТЕХНІЧНІ ЗАВДАННЯ

- можливість створення облікового запису користувача;
- наявність функції створення виклику для самоконтролю;
- можливість пошуку інших користувачів, та додавати їх до списку друзів;
- здатність викликати користувачів, наявних у списку друзів;
- наявність функції, дозволяю переглянути надіслані користувачу парі;
- можливість відсудити парі;
- здатність переглянути парі, у яких користувачі приймають участь;
- наявність повної інформації про ухвалену угоду;
- наявність інформації про результати парі користувача;
- можливість легкої адаптації до розширення функціоналу.

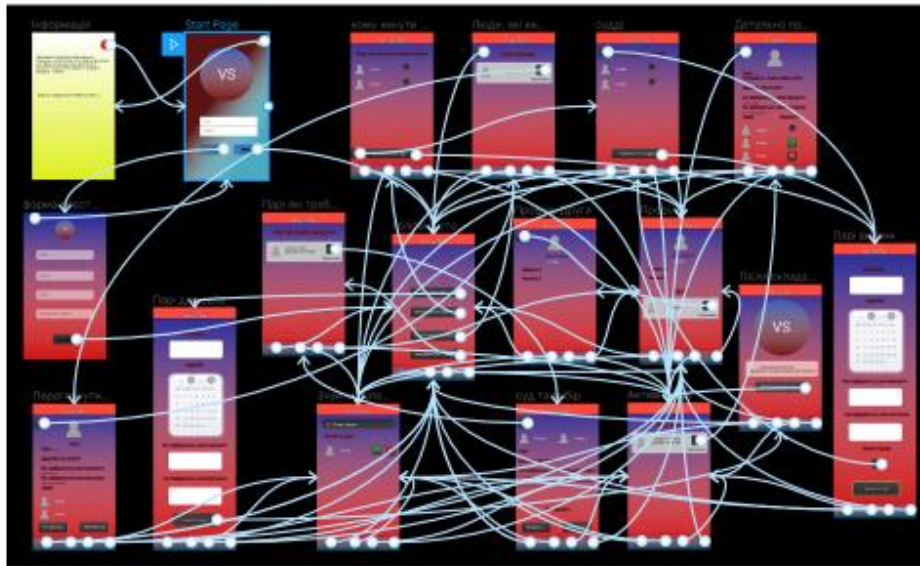
5

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



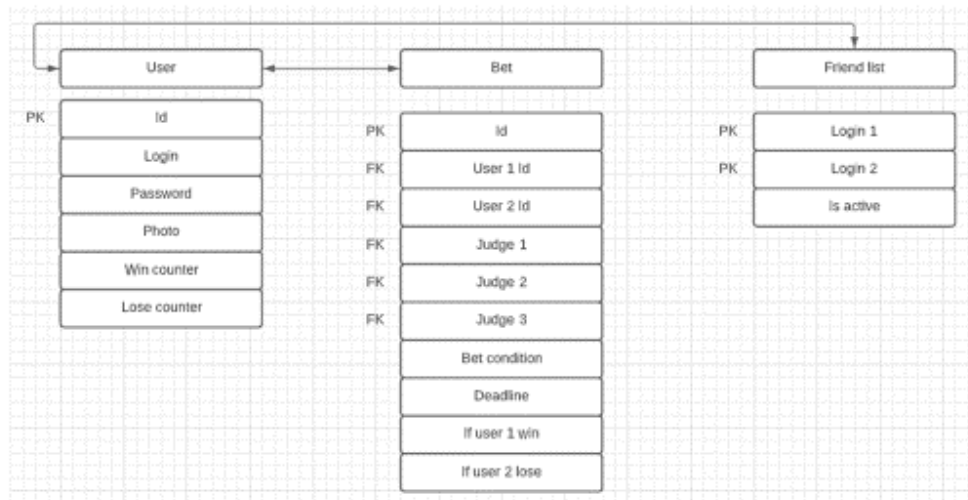
6

## Зв'язок елементів додатку



7

## UML діаграма бази даних мобільного додатку



8

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Сачик М.О. науковий семінар факультету інформаційних технологій навчально-наукового інституту телекомунікацій та інформатизації «Перспективи інформаційних технологій найближчого майбутнього: Інтернет речей, роботизація, блокчейн», 11.04.2018р., ДУТ, м. Київ.

9

### ВИСНОВКИ

1. Було досліджено конфлікти, знайдено найкращі шляхи їх вирішення, проаналізовано актуальність цієї теми, та визначено її наукову новизну.
2. Проведено аналіз інструментів та програмних засобів реалізації, які найліпше впораються зі створенням графічного інтерфейсу та усього функціоналу додатку.
3. Визначено концепцію, вимоги та головні елементи інтерфейсу мобільного додатку, розроблений детальний прототип, що демонструє функціонал та графічний інтерфейс, створено UML діаграму бази даних мобільного додатку.
4. Реалізовано графічний інтерфейс та функціонал мобільного додатку.

Перспективи подальших досліджень:

1. Легка адаптація додатку під ОС IOS та Windows phone.
2. Рейтинг користувачів
3. Можливість складення парі з випадковими користувачами
4. Можливість завантаження файлів для підтвердження виконання умов

10



ДЯКУЮ ЗА УВАГУ!