

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра теорії та технології програмування

**Кваліфікаційна робота**

**на здобуття ступеня бакалавра**

за спеціальністю 122 Комп'ютерні науки


на тему:

**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ  
МОНІТОРИНГУ ЕМОЦІЙНОГО СТАНУ**


Виконала студентка 4-го курсу  
Пікуза Валентина Олегівна

  
(підпис)

Науковий керівник:  
доцент, кандидат фіз.-мат. наук  
Панченко Тарас Володимирович

  
(підпис)

Засвідчую, що в цій роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.  
Студент

  
(підпис)

Роботу розглянуто й допущено до захисту  
на засіданні кафедри теорії та технології  
програмування

« \_\_\_\_ »

2021 р.

протокол №  
Завідувач кафедри  
М. С. Нікітченко

\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

Обсяг роботи 39 сторінок, 15 ілюстрацій, 1 таблиця, 21 джерело посилання та 2 додатки.

МОНІТОРИНГ ЕМОЦІЙ, ЩОДЕННИК, XAMARIN.FORMS, SQLITE, XAML, КРОС-ПЛАТФОРМНИЙ МОБІЛЬНИЙ ДОДАТОК.

Об'єктом дослідження є емоційний стан людини, предметом дослідження є мобільний додаток «Твій емоційний помічник», завдяки якому ведеться спостереження за емоціями людини.

Мета роботи: створення крос-платформного мобільного додатку, який надавав би людям можливість вести щоденник свого самопочуття, емоцій та пов'язаних із ними подій, аналізувати записані дані шляхом зручного представлення за допомогою діаграм та графіків.

Методи розробки: розробка програмного продукту на основі аналізу існуючих засобів для моніторингу самопочуття, комп'ютерне моделювання. Інструменти розробки: технологія Xamarin.Forms, мова програмування C#.

Результати кваліфікаційної роботи: проведено аналіз емоційного стану людини та що на нього впливає, досліджено результати наукових статей про щоденникові записи; розроблено мобільний застосунок, що надасть змогу користувачам моніторити своє самопочуття.

## ЗМІСТ

Скорочення та умовні позначення	4
ВСТУП	5
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1 Наукові дослідження, що доводять корисність ведення щоденника	8
1.2 Емоційний інтелект	9
1.3 Методи зниження стресу	9
РОЗДІЛ 2 ПОПЕРЕДНІЙ ОПИС МОБІЛЬНОГО ДОДАТКУ	11
2.1 Огляд існуючих систем на ринку	11
2.2 Огляд використаних технологій	14
2.3 Вимоги до мобільного застосунку	16
РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ	18
3.1 Проектування мобільного застосунку	18
3.2 Розробка дизайну системи	20
3.3 Додавання нотаток	21
3.4 Візуалізація отриманих даних	23
3.5 Розробка вкладки «Коли емоції беруть гору»	24
3.6 Розробка панелі налаштувань	27
ВИСНОВКИ	33
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	35
ДОДАТОК А Діаграма прецедентів додатку	38
ДОДАТОК Б Підключення до Google Drive	39

## Скорочення та умовні позначення

СКБД – система керування базами даних;

XAML – eXtensible Application Markup Language;

ACID – Atomicity, Consistency, Isolation, Durability;

UWP – Universal Windows Platform;

SQL – Structured Query Language;

ANSI – American National Standards Institute;

ISO – International Organization for Standardization;

UTF – Unicode Transformation Format.

CRUD – Create, Read, Update, Delete

БД – база даних

## ВСТУП

Емоції сильно впливають на всі сфери нашого життя – це беззаперечний факт. З поширенням пандемії все більше людей почали слідкувати за своїм ментальним здоров'ям та психоемоційним станом, адже в більшості ми опинились в обмежених умовах, що в багатьох спровокувало погіршення настрою та самопочуття. Для прикладу Міністерство Охорони Здоров'я України створило «Гарячу лінію» безкоштовної психологічної допомоги під час пандемії, що доводить важливість даної теми. Психологи рекомендують вести щоденники, так як це допомагає покращити своє самопочуття. Адже якщо уникати певні проблеми, замовчувати, це може привезти не лише до простого погіршення самопочуття, але і до депресій та інших тривожних розладів.

Отже слідкувати за своїм самопочуттям важливо, постає питання як це можна зі зручністю впровадити в своє повсякденне життя? Оскільки наш час можна сміливо назвати ерою діджиталізації, програмний продукт що надасть нам цю можливість є кращим рішенням за звичайний папір та ручку. І найкомфортніше це було би робити саме із мобільним застосунком, адже смартфони в наш час у кожного під рукою.

### **Актуальність роботи та підстави для її виконання.**

На даний момент люди все більше починають приділяти увагу своєму ментальному здоров'ю та самопочуттю, що безумовно впливає як і на наш фізичне здоров'я, так і на спроможність ефективно працювати та бути в ресурсі. Тому розробка програмного рішення, що допоможе покращити дані аспекти життя є надзвичайно актуальними.

### **Мета й завдання роботи.**

Метою кваліфікаційної роботи є створення крос-платформного мобільного додатку, який надавав би людям можливість вести щоденник свого самопочуття, емоцій та пов'язаних із ними подій, аналізувати записані дані шляхом зручного представлення за допомогою діаграм та графіків, а також створення окремого

модулю всередині застосунку, що допоміг би справитись зі складними емоційним станом.

Для досягнення цієї мети поставлено такі завдання:

- Дослідити існуючі програмні рішення для щоденних записів настрою та самопочуття.
- Систематизувати та поглибити теоретичні і практичні знання у сфері мобільної розробки.
- Дослідити предметну область розробки (основні характеристики, що впливають на самопочуття людини).
- Розробити базу даних для якісного представлення в статистичному вигляді.
- Спроекувати та розробити мобільний крос-платформний додаток, який надаватиме користувачам можливість вести щоденник записів, переглядати їх, та бачити свої статистичні дані по заповнених нотатках.
- Створити розширені налаштування для користувача для більшої персоналізації мобільного застосунку.
- Розробити зрозумілий та сучасний дизайн для програмного продукту «Твій емоційний помічник».
- Спроекувати схему анімації для проведення дихальних технік.

### **Об'єкт, методи й засоби розроблення**

Об'єктом дослідження є емоційний стан людини.

Предметом дослідження є мобільний додаток «Твій емоційний помічник», завдяки якому ведеться спостереження за емоціями людини

Під час створення програмного продукту використана еволюційна модель розробки. Інструменти розроблення: технологія Xamarin.Forms, мова програмування C#.

У зв'язку з використанням потенційними користувачами різних операційних систем, було прийнято рішення розробити крос-платформний додаток, використовуючи технологію Xamarin.Forms надану безкоштовним, вільно поширюваним інтегрованим середовищем розробки Microsoft Visual Studio Community 2019 .

Для локального зберігання даних було використано СКБД SQLite.

### **Сфера застосування.**

Програмний продукт «Твій емоційний помічник» можна використовувати в особистих цілях для запису щоденних нотаток свого самопочуття, емоційного стану, для проведення подальшого самоаналізу. Також додаток можна використовувати як допоміжний засіб при психологічній терапії, адже з допомогою нього можна відслідковувати зміну емоційного стану людини.

# РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Наукові дослідження, що доводять корисність ведення щоденника

Все більше наукових організацій проводять дослідження про вплив ведення щоденних записів на наше фізичне та психологічне самопочуття.

Наприклад психологи з Каліфорнійського Університету запросили 20 волонтерів відвідувати лабораторію щоб просканувати їхній мозок для того щоб дослідити як впливає журналювання. Для цього вони попросили їх двадцять хвилин на день робити записи протягом декількох днів. У людей, які для записів обрали свій певний емоційний досвід, була збільшена активність у частині мозку, що називається правою вентролатеральною префронтальною корою. Це відповідно допомагало розслабити нервову діяльність пов'язану з сильними емоційними почуттями. [1, 2]

Ведення щоденника, записування своїх думок і почуттів може покращити не лише емоційний стан, але і мати певний терапевтичний ефект для фізичного здоров'я. Дослідження доводять, що ведення регулярних записів може збільшити шанси на боротьбу із такими складними хворобами як астма, ревматоїдний артрит [3], СНІД [4] та рак, а також допомагає загоюватись фізичним ранам швидше. [5]

Дослідження, проведене в 2013 році, показало, що 76% дорослих, які проводили щодня по 20 хвилин щоденно за блокнотом, три дні поспіль до запланованої медичної біопсії, повністю вилікувались через 11 днів. З іншого боку, 58% контрольної групи ще не одужали. У результаті було зроблено висновок, що лише одна година написання про страшну подію допомогла учасникам краще зрозуміти події та зменшити рівень стресу. Не менш важливим є той факт, що багато психологів застосовують введення різних типів щоденників у своїй терапії. [6]

Щоденникові дослідження - це метод дослідження, з допомогою якого збирається якісна інформація, шляхом надання учасникам записів про своє

повсякденне життя у щоденнику чи журналі про діяльність чи досвід, що вивчається. Учасники повинні це робити протягом певного періоду часу. [7]

## **1.2 Емоційний інтелект**

Регулярно відомі компанії складають списки технічних та особистісних навичок, що можуть допомогти людині успішно працювати в команді та вирішувати будь-які питання. У 2020 році соціальна мережа LinkedIn, основною метою якої є пошук і встановлення ділових контактів, опублікувала список із необхідним та важливими навичками для працівників. [8] І вперше особливої уваги приділили такому софт скілу як емоційний інтелект - здатність людини керувати емоціями та розуміти їх, усвідомлювати, що відчуває вона сама і ті, хто знаходиться поруч. [9]

Отже, для вдалого працевлаштування робітник повинен не тільки розуміти в технічних особливостях своєї роботи, але й вміти правильно себе поводити з людьми. Покращити свій емоційний інтелект людина може через пізнання себе, розуміння своїх емоцій та бажань, що відповідно допоможе краще розпізнавати ці почуття у інших людей та навчитися емпатії. Що найосновніше, це навчитись спокійно та виважено реагувати на свої відчуття, якраз в цьому і допомагає ведення особистих щоденників, з допомогою яких людина може відстежувати свої реакції.

Все це ще раз доводить актуальність та необхідність дослідження області людських емоцій та почуттів.

## **1.3 Методи зниження стресу**

Напевно у кожного траплялась така ситуація, коли негативні емоції та стрес беруть над людиною гору. В таких випадках важливим є вміння взяти себе під контроль аби якнайшвидше позбутись від тривожного стану. Для цього існують різні методики та рекомендації, наприклад прогулянка, медитація, фізичні вправи, глибоке дихання. Особливу увагу я б хотіла приділити саме дихальним технікам,

адже це спосіб який можна застосувати в будь-якій життєвій ситуації незалежно від місця свого перебування. Окрім того все більше досліджень доводять, що це є справді ефективний спосіб боротись зі стресом та безсонницею. [10]

Глибоке дихання, або іншими словами діафрагмальне дихання, «дихання животом» для багатьох людей може здаватись не природнім, адже спеціальний ритм, що використовується в різних техніках сильно відрізняється від нашого звичайного «вдишу-видишу». Так в чому тоді особливість, та як впливає цей процес на людину, її фізичний та емоційний стан?

Діафрагмальне дихання стимулює кращий кисневий обмін у тілі, між вхідним киснем та вихідним вуглекислим газом. Це в свою чергу допомагає сповільнити серцевий ритм, а також знизити та стабілізувати артеріальний тиск.

Одною із найпопулярніших технік є «4-7-8», де кожна цифра відповідає кількості секунд. [11] Процес складається з наступних етапів:

1. Вивільнити легені від всього повітря.
2. Спокійно вдихнути через ніс протягом 4 хвилин.
3. Затримати повітря на 7 секунд.
4. Видихнути через рот протягом 8 секунд.
5. Повторити мінімум 4 рази.

Тому враховуючи необхідність мати дієвий спосіб для швидкого заспокоєння, було прийнято рішення в застосунку розробити для цього окремий модуль, що допоможе в подібних критичних ситуаціях, а також зможе використовуватись як допоміжний інструмент і в регулярних тренуваннях. Процес її реалізації можна буде побачити в наступних розділах.

## РОЗДІЛ 2 ПОПЕРЕДНІЙ ОПИС МОБІЛЬНОГО ДОДАТКУ

### 2.1 Огляд існуючих систем на ринку

Оскільки сьогодні глобально по всьому світу люди все більше почали звертати та приділяти увагу своєму емоційному стану, розробка мобільних додатків з цією предметної області набирає популярності. Навіть за час, за який виконувалась кваліфікаційна робота, кількість додатків у відкритому доступі, зв'язаних з моніторингом свого самопочуття, відчутно збільшилась. Задля кращої розробки програмного продукту «Твій емоційний помічник» було проаналізовано ряд існуючих додатків із крамниці застосунків від Google PlayMarket. Розглянемо деякі із них.

На рис 2.1 зображено інтерфейс мобільного застосунку «Mood diary». До його переваг можна віднести привабливий та простий інтерфейс, мотивуючі заспокоюючі фрази та наявність паролю для застосунку. Проте він не надає користувачу можливість переглянути заповнену ним інформацію в розрізі місяцю або року, а лише записи зроблені ним в конкретний час. [12]

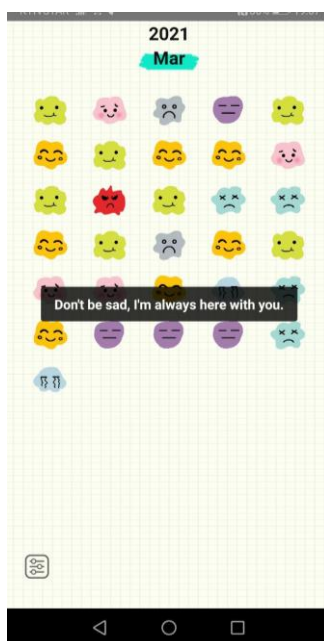


Рисунок 2.1 Основна сторінка «Mood diary»

Наступний додаток, який би я хотіла розглянути, називається «ССП – Сам Себе Психолог» та має російськомовний інтерфейс. Він надає користувачу можливість записувати свою емоції, думки у форматі спогадів, також переглядати статистику за допомогою кругової діаграми (див рис 2.2), а також встановлювати сповіщення для регулярного заповнення щоденнику. [13]

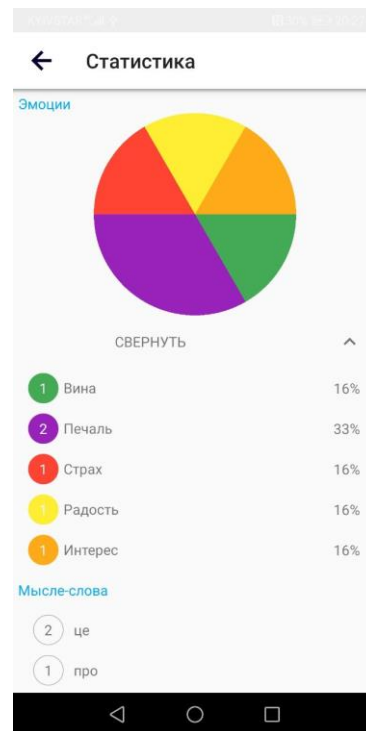


Рисунок 2.2 Сторінка статистики додатку «ССП»

Серед чудових програмних рішень наявних у вільному доступі є додаток «Feelings Diary» (див рис 2.3). Він пропонує робити записи, заповнюючи інформацію про місце, зв'язаних з подією людей, дату та відповідно емоції що відчував користувач. В ньому наявний великий перелік емоцій за категоріями. Також присутній захист інформації за допомогою пінкоду або біометричного замку. [14]

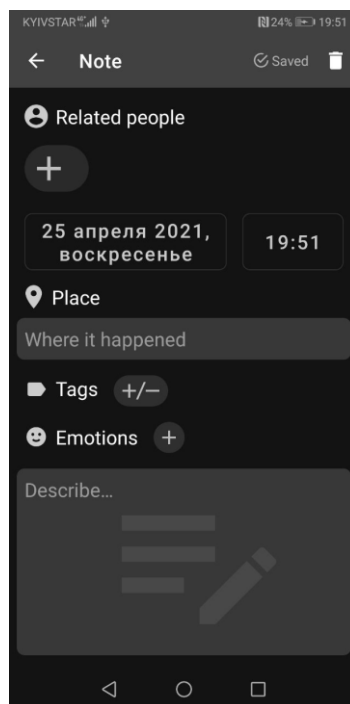


Рисунок 1.3 Сторінка додавання запису додатку «Feelings Diary»

Для кращого розуміння особливостей розроблено порівняльну таблицю для даних додатків (див табл 2.1)

Характеристика\Додаток	Mood Diary	ССП	Feelings Diary
Пароль для входу	+	-	+
Синхронізація даних	-	-	-
Візуалізація даних	-	+	-
Кастомізація зовнішнього вигляду додатку	-	-	+
Декілька записів на день	-	+	+
Колекціонування фото	+	-	-
Збирання інформації про погодні умови	+	-	-
Нагадування про запис	-	+	-

Таблиця 2.1 Порівняння додатків для моніторингу самопочуття

Аналіз існуючих рішень надав змогу краще виділити особливості потрібні для розробки додатку для програмного продукту для даної предметної області та визначить зі всім необхідним функціоналом.

## **2.2 Огляд використаних технологій**

Проект розроблений у безкоштовному інтегрованому середовищі розробки програмного забезпечення Visual Studio Community 2019 [15], за мову програмування обрано програмування С# та було використано ряд її інструментальних засобів.

Проект було створено, використовуючи технологію Xamarin.Forms. Вона надає можливість створювати крос-платформні прикладні програми, тобто такі, які можуть працювати більш ніж на одній апаратній або програмній платформі, оптимізовані під конкретні операційні системи, що забезпечує їх коректну та швидку роботу. Технологія дозволяє розробнику створювати інтерфейс користувача з набору візуальних елементів, які описуються мовою розмітки XAML. Завдяки портативним системам відображення інтерфейсу відбувається через рідні візуальні елементи відповідної операційної системи Android, iOS і UWP [16].

Технологія Xamarin.Forms підтримує асинхронність. Це дозволяє покращити процес, а саме - винести окремі задачі або певні блоки коду з основного потоку в спеціальні асинхронні методи. Це забезпечує обробку інших завдань, не чекаючи завершення попереднього. Особливо це знадобилось при створенні анімації у проекті.

XAML – декларативна мова розмітки для програми .Net Framework. Вона допомагає спростити створення користувацького інтерфейсу. Спочатку розробник можна створити видимі елементи користувацького інтерфейсу, а вже потім відокремити визначення інтерфейсу від логіки виконання, використавши при цьому файли з кодом програмної частини, які приєднані за допомогою визначень поділюваних класів. Безпосередньо для декларативної мови XAML представлення

екземплярів об'єктів, визначених у збірках, міститься в деякому наборі резервних типів. Саме в цьому полягає її основна відмінність від більшості інших мов розмітки, адже вони зазвичай являють собою інтерпретовані мови що не мають прямого зв'язку з системою резервних типів. Робочий процес мови XAML дозволяє відразу декільком учасникам, які використовують потенційно різні засоби, брати участь в розробці логіки програми та користувацького інтерфейсу [16].

Зазвичай файли XAML мають розширення `.xaml`, а уже при поданні у вигляді текстового файлу розпізнаються, як XML-файли. Завдяки такому процесу файли даного типу можна зберігати в будь якому кодуванні, що підтримується для XML, але зазвичай використовується UTF-8 [16].

Для побудови локальної бази даних було обрано СКБД SQLite, що являє собою спрощену систему керування базами даних, яка є втіленою у вигляді бібліотеки. Її було обрано з міркувань того, що вона не використовує парадигму клієнт-сервер, тобто ядро SQLite це не просто окремий процес, а воно надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, за протокол обміну використовуються виклики функцій бібліотеки SQLite. Це дозволяє зменшити час відгуку, накладні витрати та може спростити програму. Окрім того, база даних зберігається в одному локальному файлі, що знаходиться на пристрої, та легко встановлюється на операційні системи, і також не має зовнішніх залежностей, що забезпечує її автономність. SQLite зберігає всю базу даних (включаючи таблиці, індекси, дані та визначення) в єдиному стандартному файлі на тому пристрої, де виконується додаток. Простота реалізації досягається завдяки тому, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується. Проте ACID-функції досягаються шляхом створення окремого файлу-журналу.

Кілька процесів або потоків можуть одночасно без проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; в інакшому випадку спроба записати щось закінчиться невдало, і в програму повернеться код помилки. Або протягом

заданого деякого інтервалу часу відбувається автоматичне повторення спроб запису.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`. З її допомогою демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка та дозволяє звертатися до файлу бази даних на основі типових функцій операційної системи. Архітектура рушія забезпечує можливість використовувати SQLite вбудовуваних системах, а також на виділених машинах з великими гігабайтними масивами даних [17].

Проект складається з двох частин: `backend`, що відповідає за роботу з серверною частиною, а саме отримання, збереження, оновлення та видалення даних; `frontend` – котрий являє собою сам мобільний крос-платформний додаток та працює з локальною базою даних. Backend це по суті сукупність класів контролерів, завдяки яким забезпечується коректна робота з базою даних, в ній містяться дані конфігурації, які необхідні для підключення, а також контекст, який задає базу даних мобільного застосунку. У `frontend` – частині проекту є класи, котрі визначають таблиці бази даних, класи для заповнення, редагування та видалення даних таблиць і написання запитів для отримання особливих даних з локальної бази даних і змістовні сторінки зовнішнього вигляду додатку з описом функціональних можливостей візуальних об'єктів. Кожне нове вікно, яке відкривається будується з файлу типу `.haml`, який відповідає за простий та зрозумілий інтерфейс, з яким працює користувач, а також файлу типу `.haml.cs` котрий і реалізовує всі функціональні можливості конкретної сторінки контенту передбачені інтерфейсом.

### **2.3 Вимоги до мобільного застосунку**

Проаналізувавши предметну область та існуючі програмні рішення, можна сформулювати наступні вимоги до розробки.

Мобільне програмне забезпечення повинне бути зручне у використанні,

мати інтуїтивно-зрозумілий користувачеві дизайн та функціонал, надавати доступ до інформації швидко незалежно від встановленого підключення до глобальної мережі і мати просту для розуміння навігацію.

Мобільний додаток повинен бути розроблений з українським інтерфейсом.

Система повинна надавати користувачеві розширені налаштування, увімкнення/вимкнення конкретного функціоналу, додавання своєї інформації до існуючих словників в базі даних.

Система повинна мати можливість захисту даних від інших користувачів. Також повинно бути реалізовано функціонал отримання сповіщень про щоденні записи; користувач може увімкнути цю функцію та встановити час сповіщення.

## РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ

### 3.1 Проектування мобільного застосунку

Для успішного створення будь-якого проекту є важливим спершу його детально спроектувати. Тому перед самим процесом фактичної розробки необхідно було визначитись з основними функціональними можливостями мобільного застосунку. Для цього було створено use case діаграму прецедентів (див. Додаток А). З її допомогою було легше зрозуміти як повинна працювати система з боку користувача і швидше приступити до розробки.

Оскільки основна мета застосунку це запис та огляд збережених нотаток, для програми було спроектовано та розроблено базу даних, що складається із чотирьох таблиць.

**DayNotes** – таблиця для збереження записів користувача; містить поля

- int NoteId;
- int MoodId – id обраного настрою;
- string Emotions – містить список id обраних емоцій;
- string Activities – містить список id обраних активностей;
- string Soughts – поле для запису подій, переживань;
- Date NoteDate – дата нотатки.

**Mood** – словник настроїв

- int MoodId;
- string MoodName – назва настрою;
- string MoodColor – колір настрою для відображення у діаграмах;
- string ImagePath – шлях до зображення, що відповідає конкретному настрою.

**Activities** – таблиця, для збереження активностей, які веде користувач

- int ActivityId;
- string ActivityName – назва активності;
- int PosNegIndex – число що визначає чи є конкретна активність

позитивною чи негативною для користувача;

**Emotions** – таблиця, що зберігає різні типи емоцій

- int EmotionId;
- string EmotionName – назва емоції;
- int PosNegIndex - число що визначає чи є конкретна емоція позитивною чи негативною для користувача ( 0 – негативно забарвлення, 1 позитивне, 2 – нейтральне).

Для кожної таблиці у програмі було визначено відповідні класи, об'єкти яких будуть зберігатись в базі даних. Вони будуть виступати в якості моделі для додатку.

Також було створено клас репозиторія NotesRepository, через який проходять всі основні операції із даними. Всередині його конструктора якраз і відбувається створення підключення до БД і в загальному створення самої бази даних, якщо вона ще не існує. Як параметр передається шлях до неї із зовні. Для операцій із даними у класі SQLiteConnection визначені методи : Insert, Get, Update, Query, Execute, Table, Delete.

Оскільки підключення до БД спільне для всього застосунку воно було визначено у основному файлі App.xaml.cs (див рис 3.1)

```
public const string DATABASE_NAME = "NotesDB.db";
public static NotesRepository database;
1 reference
public App()
{
    InitializeComponent();
    if (database == null)
    {
        database = new NotesRepository(
            Path.Combine(
                Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), DATABASE_NAME));
    }
    FillDB();
    DependencyService.Register<MockDataStore>();
    MainPage = new AppShell();
}
```

Рисунок 3.1 Підключення БД

### 3.2 Розробка дизайну системи

Для мобільного застосунку було обрано Flayout Menu дизайн – користувацький інтерфейс навігації, заснований на спливаючих вікнах, оскільки це найзручніше рішення для архітектури даного мобільного застосунку, який містить різні вкладки та модулі (див рис. 3.2 ), а саме:

- Записник (сторінка, де користувач зможе додавати нові нотатки та переглянути вже створені).
- Статистика (сторінка, де можна переглянути результати регулярних записів в розрізі певного часу).
- Коли емоції беруть гору (сторінка з дихальною технікою).
- Налаштування.

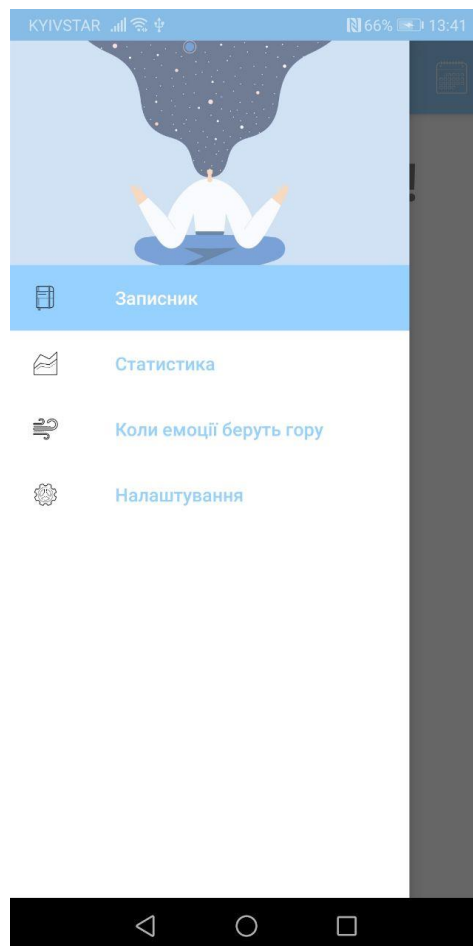


Рисунок 3.2 Flayout Menu додатку

### 3.3 Додавання нотаток

Перейшовши на вкладку «Твій записник» користувач потрапляє на сторінку NotePage, на якій відображенні поточні записи за день (див рис 3.3).



Рисунок 3.3 Зовнішній вигляд сторінки «Твій записник»

На верхній панелі цієї сторінки є кнопка із зображенням календаря, натиснувши, на яку користувач відкриє сторінку, на якій зможе знайти та переглянути всі свої записи, натиснувши на конкретну дату.

Сторінку CalendarPage розроблено з допомогою плагіну Xamarin.Plugin.Calendar. Він дозволяє відображати події через прив'язку EventCollection та відмічати окремі дні, а також має широкий спектр для кастомізації зовнішнього вигляду сторінки.

При натиску на круглу кнопку знизу, користувач перейде на сторінку AddNotePage (див рис 3.4).

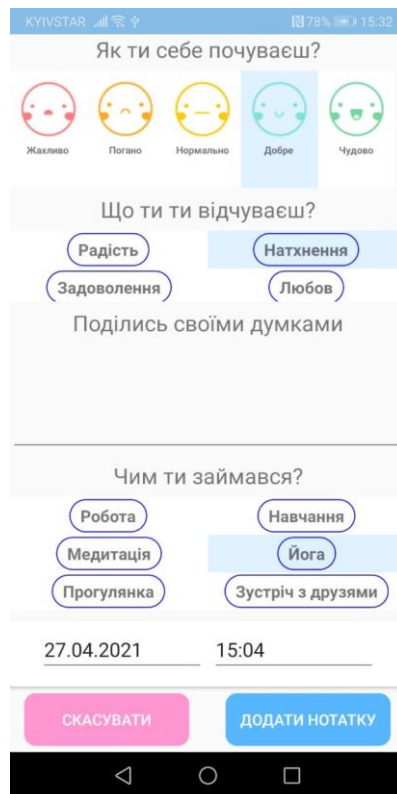


Рисунок 3.4 Зовнішній вигляд сторінки AddNotePage

На сторінці представлення поля які необхідно заповнити, відповівши на відповідні запитання:

- Як ти себе почуваєш? Користувачу необхідно обрати один із таких варіантів ( жахливо, погано, нормально, добре, чудово).
- Що ти відчуваєш? Користувачу надається вибір із емоцій заданих у базі даних, окрім того список можна поповнити новими даними в налаштуваннях. В цьому списку є можливість мультивибору.
- Поділись своїми думками. Дане поле Editor призначене для того, аби користувач міг в ньому написати свої події, пов'язані з конкретним днем або залишити важливі для нього враження, що на його думку впливають на стан та самопочуття.
- Чим ти займався? Ця графа призначена для заповнення її активністю, якою займався користувач в цей день та час. Аналогічно і з емоціями, дані можна доповнювати новими та обирати більше ніж один пункт
- Час та Дата, які викликають класи відповідно TimePicker і DatePicker

За допомогою внесених даних за цими характеристиками і будується подальша статистика.

### 3.4 Візуалізація отриманих даних

Для того щоб користувач мав змогу проаналізувати свій емоційний стан протягом певного періоду часу, було створено розділ «Статистика», в який він може перейти натиснувши на відповідний розділ у боковій панелі. Для візуалізації був використаний плагін Microcharts.Forms - проста бібліотека діаграм для широкого кола платформ.

Для аналізу було обрано такі діаграми:

- Кругова Діаграма, яка зображує у відсотковому відношенні які емоції переважають за конкретний місяць (див рис 3.5).
- Полігон частот, що відображає зміну настрою користувача відносно певного проміжку часу.
- Кругова діаграма, для візуалізації залежності емоцій від виду діяльності.

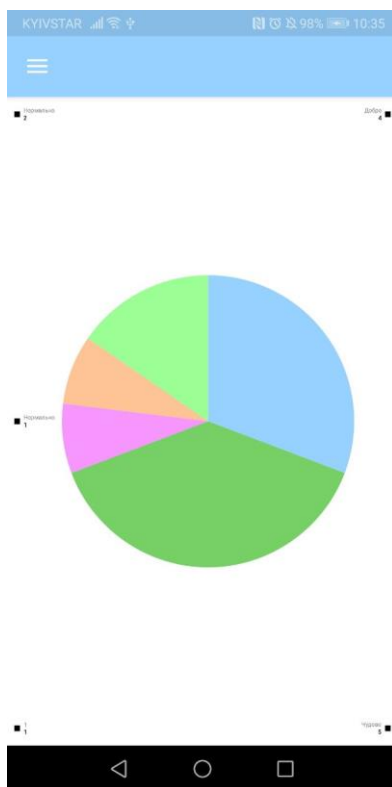


Рисунок 3.5 Сторінка статистики

Для реалізації цієї кругової діаграми необхідно було обрахувати кількість з якою зустрічається конкретний настрій. Це було реалізовано наступним чином:

```
var entries = new List<ChartEntry>();
var notes = App.database.GetNotesItems();
var MoodsList = App.database.GetMoodItems();
foreach (Mood mood in MoodsList)
{
    int count = notes.Where(x => x.curMood == mood.Id).Count();
    var ce = new ChartEntry(count);
    ce.Label = mood.MoodName;
    ce.ValueLabel = count.ToString();
    ce.Color= SKColor.Parse(mood.MoodColor);
    entries.Add(ce);
}
var chart = new PieChart { Entries = entries };
InitializeComponent();
chartView.Chart = chart;
```

Так само і для решти типів діаграм підраховується частота з якою зустрічаються окремі емоції та активності у нотатках, наявних у БД користувача

### **3.5 Розробка вкладки «Коли емоції беруть гору»**

Для виконання дихальної техніки було розроблене візуальне рішення, що допоможе швидко зорієнтуватись і правильно зробити всі кроки. Xamarin.Forms має власну інфраструктуру анімації, з допомогою якої можна змінювати властивості візуальних елементів за певний період часу.

Оскільки бувають різні техніки, було прийнято рішення надати можливість користувачу редагувати процес анімації даної сторінки в меню “Налаштування”. Він може задати такі параметри:

- BreathinSec - кількість секунд для вдиху;
- HoldSec - кількість секунд для затримки повітря після вдиху;
- BreahoutSec – кількість секунд для видиху;
- Hold2Sec - кількість секунд для затримки повітря після видиху;
- TimesRepeat – кількість повторень одного циклу дихання;

За замовчуванням на даній вкладці налаштовані опції для техніки «4-7-8», описаній в пункті 1.3.

При вході на дану вкладку користувач побачить повідомлення “Щоб розпочати вправу натисніть на кулю” (див рис 3.6). Також на панелі зверху є кнопка інформації для того щоб ознайомити користувача з даним модулем та його можливостями.

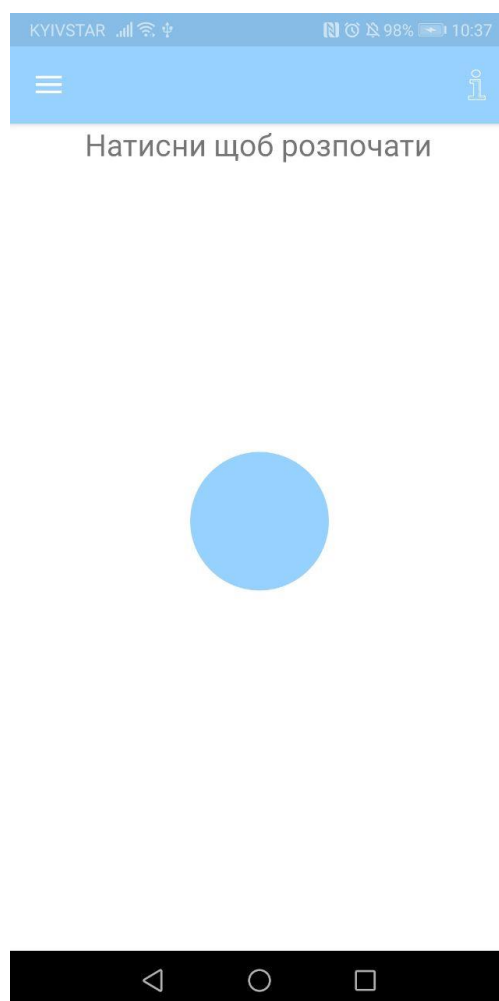


Рисунок 3.6 Початковий вигляд сторінки з анімацією

Зробивши відповідні дії, почнеться відлік часу. Перший стан “Глибоко вдихни”, під час якого починається чисельний відлік, а фігура розширюється, імітуючи сам процес вдиху, адже під час нього об’єм грудної клітки значно розширюється. На рис. 3.7 можна побачити його в дії.

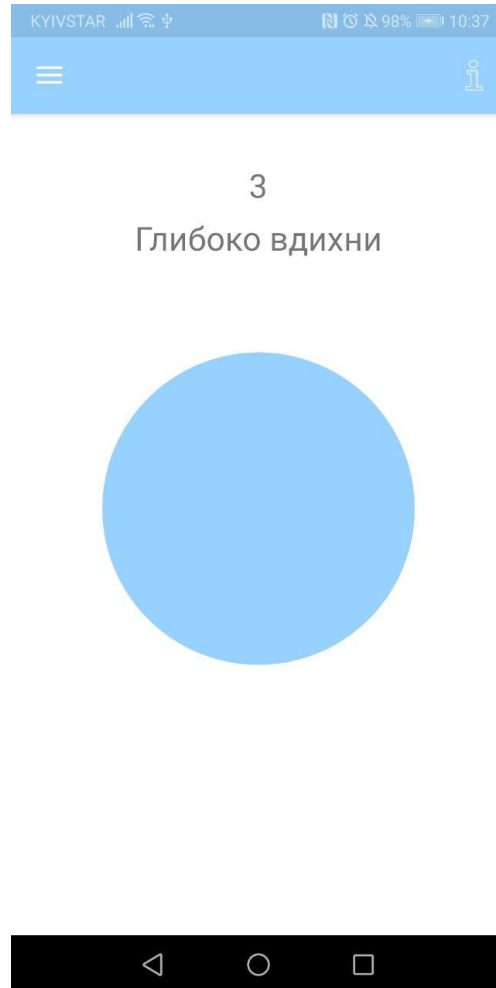


Рисунок 3.7 Сторінка анімації в дії

Після цього наступним етапом буде процес «Затримай повітря» з новим відліком під час якого об’єм фігури залишиться таким ж як і при закінченні процесу вдиху. Він зміниться на процес «Видихай», зі своїм відліком часу, під час якого фігура почне зменшуватись до своїх початкових розмірів.

Цикл буде повторюватись за заданим числом TimesRepeat або поки користувач не вирішить залишити вкладку з дихальною технікою.

### 3.6 Розробка панелі налаштувань

Так як програма має різні модулі , необхідно було розробити широкий функціонал задля задоволення потреб користувача. Для зовнішнього вигляд було використано AiForms.Effects - бібліотека ефектів, яка надає значно гнучкіші функції, аніж за замовчуванням, орієнтуючись лише на операційні системи Android та iOS Xamarin.Forms проєкті. Натиснувши на відповідну вкладку на боковій панелі, користувач перейде на сторінку SettingsPage. (див рис 3.8)

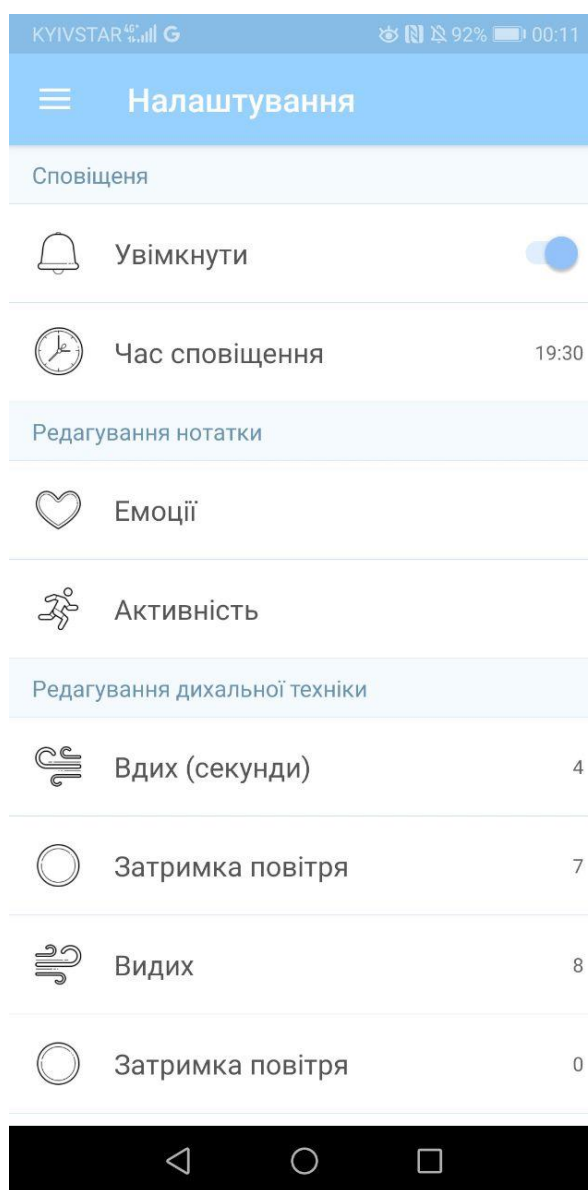


Рисунок 3.8 Сторінка налаштувань додатку

Для налаштувань було виділено наступні секції:

### **Сповіщення**

Так як основна мета додатку це моніторинг емоційного стану користувача, для кращого результату записи необхідно робити регулярно. Для цього було розроблено щоденні сповіщення (див рис 3.8), які були реалізовані за допомогою плагіна `Ham.Plugin.Notifier` – бібліотеки, що забезпечує простий крос-платформний спосіб показу локальних повідомлень. Користувач вказує в налаштуваннях час, в який він хоче їх отримувати. Також цю функцію можна вимкнути.

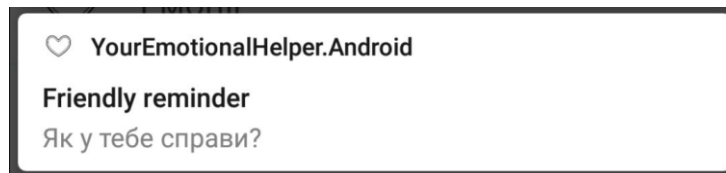


Рисунок 3.8 Нагадування для запису нотатки

### **Редагування бази даних**

Оскільки список емоцій та активності не є вичерпним, користувач надано можливість поповнювати їх новими даними. Натиснувши на список, користувач відкриє сторінку `EmotionList` або `ActivityList` відповідно. Для редагування в базі даних створені методи `CRUD`. Для списків попередньо були створені таблиці, поля із необхідними для них даними наведено у пункті 3.1.

### **Редагування дихальної техніки**

Зважаючи на те, що користувач може застосовувати різні вправи на сторінці «Коли емоції беруть гору», в налаштуваннях є параметри, що визначені як глобальні змінні, які доступні у будь-якій частині програми. І їх користувач може виставити за власним бажанням, а саме:

- Вдих – кількість секунд за які користувач має вдихнути повітря.
- Затримка повітря - кількість секунд для того аби затримати дихання.

- Видих – к-сть секунд на видих.
- Затримка повітря після видиху.
- Кількість повторень – число за яким визначається скільки разів повторюватиметься цикл.

За цими параметрами відповідно і працюватиме анімація.

## Безпека

Найважливішим налаштуванням є саме приватність, адже оскільки розробка призначена для того аби користувач в ній записував особисту інформацію про свій стан та думки. В нього є можливість цю функцію увімкнути в налаштуваннях та встановити пароль. І тоді при наступному вході в програму користувачу необхідно буде пройти автентифікацію за паролем (див рис 3.10). Процес реалізації входу за паролем здійснюється через змінну яка містить введений користувачем код.

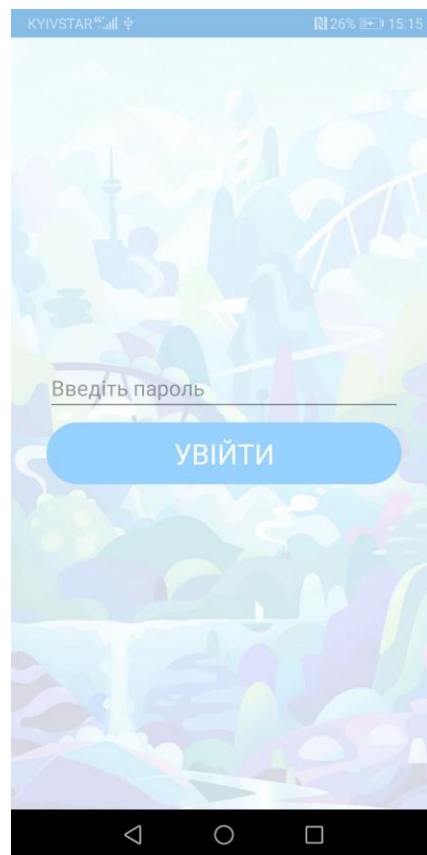


Рисунок 3.10 Сторінка автентифікації

Окрім звичного входу за паролем, ще було реалізовано вхід через біометричну автентифікацію.

### 6.6.1 Fingerprint Authentication

Для захисту даних було обрано автентифікація за допомогою датчика відбитка пальця. Для цього було встановлено NuGet плагін Plugin.FingerPrint. Оскільки кожна платформа має свої особливості додаткові бібліотеки допомагають полегшити процес розробки.

Розглянемо процес налаштування для ОС Android.

Для цього було встановлено Xamarin.AndroidX.Migration NuGet пакети, тому що даний плагін опирається на останні версії бібліотек для Xamarin. Це забезпечує автоматичне встановлення і решти необхідних залежностей.

Після встановлення необхідних бібліотек було відредаговано файл AndroidManifest та дозволено використання USE\_FINGERPRINT. Всередині OnCreate методу було додано директиву using Plugin.FingerPrint і відповідні рядки.

```
{
    CrossFingerprint.SetCurrentActivityResolver(() => this);
};
```

На рис 3.11 та 3.12 зображено реалізацію цих сторінок.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="YourEmotionalHelper.Views.MainPage">

  <StackLayout>
    <!-- Place new controls here -->
    <Button Text="Biometric authentication" x:Name="AuthButton"
      Clicked="AuthButton_Clicked"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />
  </StackLayout>

</ContentPage>
```

Рисунок 3.11 XAML-сторінка для Finger Authentication

```

private async void AuthButton_Clicked(object sender, EventArgs e)
{
    bool isFingerprintAvailable = await CrossFingerprint.Current.IsAvailableAsync(false);
    if (!isFingerprintAvailable)
    {
        await DisplayAlert("Error",
            "Biometric authentication is not available or is not configured.", "OK");
        return;
    }

    AuthenticationRequestConfiguration conf =
        new AuthenticationRequestConfiguration("Authentication",
            "Authenticate access to your personal data");

    var authResult = await CrossFingerprint.Current.AuthenticateAsync(conf);
    if (authResult.Authenticated)
    {
        //Success
        await DisplayAlert("Success", "Authentication succeeded", "OK");
    }
    else
    {
        await DisplayAlert("Error", "Authentication failed", "OK");
    }
}
}

```

Рисунок 3.12 XAML-сторінка для Finger Authentication

### Синхронізація даних

Оскільки для того щоб моніторинг емоційного стану був справді корисним, користувачу необхідно аби він завжди мав доступ до своїх даних, адже можна змінити телефон і якщо записана інформація зберігається лише локально на смартфоні, при зміні телефону усі записи буде просто втрачено. Виходячи із цих міркувань, важливо було розробити можливість зберегти дані на хмарному сховищі. Для цього було обрано Google Drive.

### 3.6.2 Використання Google Drive API

Google Drive API дозволяє створити додатки в яких можна буде зберігати дані на цьому хмарному сховищі.

Для цього спершу необхідно було зареєструвати мобільний додаток в консолі Google Developer API із відповідно увімкненим Google Drive API.

Наступним кроком було створення `GoogleApiClient` із Google Drive API в області дії. І після його підключення необхідно зробити наступні кроки:

1. Для запису вмісту файлу витягнути сам вміст диску з допомогою `DriveContentsResult`
2. Після отримання результату записати дані на Google диск
3. Встановити метадані для файлу
4. Створити файл на основі диску із вмістом гугл диска.

Процес реалізації підключення до Google Drive для Android наведено в Додатку Б (файл `AndroidManifest`).

## ВИСНОВКИ

Отже, в результаті виконання кваліфікаційної роботи розроблено крос-платформний мобільний додаток, основною метою якого є покращення якості життя користувача та його емоційного стану, шляхом ведення щоденних нотаток щодо свого настрою, емоцій та активності за день.

У ході виконання було досліджено предметну область, наукові статті про те який вплив на психічне та фізичне здоров'я має ведення особистого журналу, а також опрацьовано інформацію про важливість емоційного інтелекту для працевлаштування у сучасному світі.

Було досліджено різні способи боротьби зі стресом та розроблено в застосунку окремий модуль з дихальною технікою.

Було проаналізовано існуючі програмні рішення для моніторингу емоційного стану людини, зведено таблицю, яка визначає основні їхні переваги та недоліки, що краще допомогло зрозуміти який функціонал необхідний для застосунку для даної предметної області.

У ході виконання кваліфікаційної роботи було систематизовано та поглиблено набуті протягом навчання знання у сферах мобільної розробки під різні операційні системи, а також освоєно навички крос-платформної розробки засобами Xamarin.Forms, та створення бази даних мобільною розробки з допомогою SQLite. Поглиблено та покращено знання мови C# та її інструментальних засобів. Також набуто нові навички в розробці анімації використовуючи внутрішні функції Xamarin.Forms.

Також було розглянуто та застосовано багато додаткових бібліотек (Xam.Plugin.Notifier, AiForms.Effect, Microcharts.Forms, Xamarin.Plugin.Calendar і ін.) для Xamarin.Forms, що надало можливість розробити зручний та зрозумілий інтерфейс для користувача. Розглянуто біометричну автентифікацію та застосовано Google Drive Api.

Оскільки дана предметна область набуває популярності, надалі планується покращити розроблений додаток, доповнити його додатковим функціоналом: розробити можливість користувачу налаштовувати кольорову тему застосунку, зробити можливість завантаження звіту по заповнених даних у вигляді pdf-файлу, доповнити застосунок корисним контентом та інформацією із даної предметної області. Наступним етапом планується розмістити застосунок на онлайн-крамницях, таких як Google Play та App Store та надати користувачам можливість вільно користуватись даним застосунком.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Emotional and physical health benefits of expressive writing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cambridge.org/core/journals/advances-in-psychiatric-treatment/article/emotional-and-physical-health-benefits-of-expressive-writing/ED2976A61F5DE56B46F07A1CE9EA9F9F#B3>.
2. Expressive Writing and Wound Healing in Older Adults [Електронний ресурс] – Режим доступу до ресурсу: [https://journals.lww.com/psychosomaticmedicine/Abstract/2013/07000/Expressive\\_Writing\\_and\\_Wound\\_Healing\\_in\\_Older.10.aspx](https://journals.lww.com/psychosomaticmedicine/Abstract/2013/07000/Expressive_Writing_and_Wound_Healing_in_Older.10.aspx).
3. Effects of Writing About Stressful Experiences on Symptom Reduction in Patients With Asthma or Rheumatoid Arthritis [Електронний ресурс] – Режим доступу до ресурсу: <https://jamanetwork.com/journals/jama/fullarticle/189437>.
4. Effect of Written Emotional Expression on Immune Function in Patients With Human Immunodeficiency Virus Infection: A Randomized Trial [Електронний ресурс] – Режим доступу до ресурсу: [https://www.fmhs.auckland.ac.nz/assets/fmhs/som/psychmed/petrie/docs/2004\\_HIV\\_writing\\_trial.pdf](https://www.fmhs.auckland.ac.nz/assets/fmhs/som/psychmed/petrie/docs/2004_HIV_writing_trial.pdf).
5. Writing Can Help Injuries Heal Faster [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scientificamerican.com/article/writing-can-help-injuries-heal-faster/>.
6. Using Diary Methods in Psychological Research [Електронний ресурс] – Режим доступу до ресурсу: <http://www.columbia.edu/~nb2229/docs/Iida,%20Shrout,%20Laurenceau%20&%20Bolger%20%282012%29.pdf>.
7. Diary studies [Електронний ресурс] – Режим доступу до ресурсу:

[https://en.wikipedia.org/wiki/Diary\\_studies](https://en.wikipedia.org/wiki/Diary_studies).

8. The Most In-Demand Hard and Soft Skills of 2020 [Электронный ресурс] – Режим доступа до ресурсу:  
<https://business.linkedin.com/talent-solutions/blog/trends-and-research/2020/most-in-demand-hard-and-soft-skills>.
9. Emotional intelligence [Электронный ресурс] – Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/Emotional\\_intelligence](https://en.wikipedia.org/wiki/Emotional_intelligence).
10. Relaxation techniques: Breath control helps quell errant stress response [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.health.harvard.edu/mind-and-mood/relaxation-techniques-breath-control-helps-quell-errant-stress-response>.
11. How to use 4-7-8 breathing for anxiety [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.medicalnewstoday.com/articles/324417>.
12. Mood Diary [Электронный ресурс] – Режим доступа до ресурсу:  
<https://play.google.com/store/apps/details?id=com.RapStar.MoodDiary>.
13. Сам Себе Психолог [Электронный ресурс] – Режим доступа до ресурсу:  
<https://play.google.com/store/apps/details?id=com.bk2rl.sssp>.
14. Feelings Diary [Электронный ресурс] – Режим доступа до ресурсу:  
<https://play.google.com/store/apps/details?id=io.sergiolabs.feelingsdiary>.
15. Visual Studio Community 2019 [Электронный ресурс] – Режим доступа до ресурсу:  
<https://visualstudio.microsoft.com>
16. Hermes D. Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals / Dan Hermes., 2015.
17. SQLite [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.sqlite.org/about.html>
18. Google Drive API [Электронный ресурс] – Режим доступа до ресурсу:  
<https://developers.google.com/drive/api/v3/about-sdk>.

**19.** Публикация в Google Play [Электронный ресурс] – Режим доступа до ресурсу:

<https://docs.microsoft.com/ru-ru/xamarin/android/deploy-test/publishing/publishing-to-google-play/?tabs=windows>.

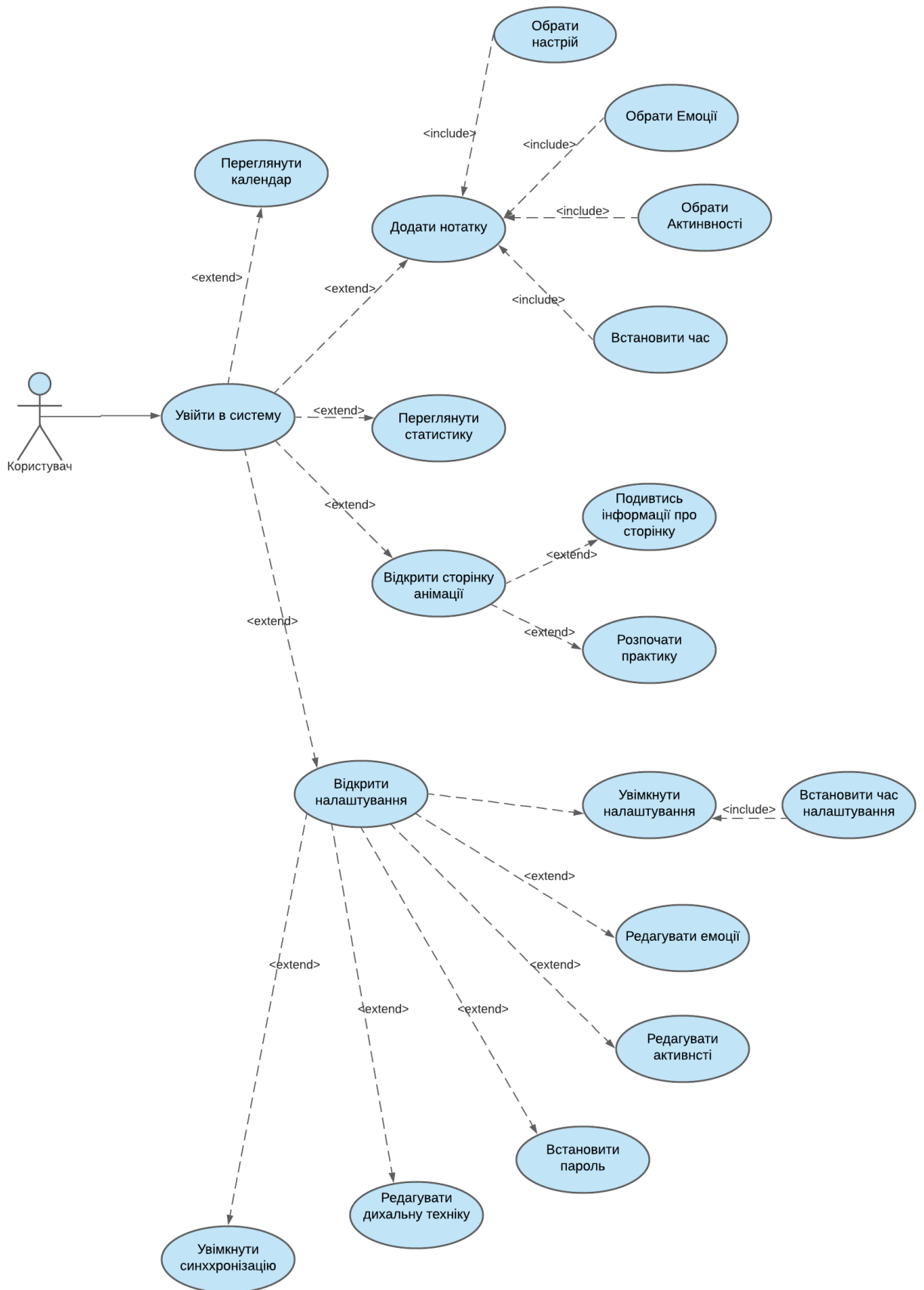
**20.** Introduction to the Google Drive Android API [Электронный ресурс] – Режим доступа до ресурсу:

<https://developers.google.com/drive/android/create-filecreate-file>.

**21.** Fingerprint Authentication [Электронный ресурс] – Режим доступа до ресурсу:

<https://docs.microsoft.com/en-us/xamarin/android/platform/fingerprint-authentication/>

## ДОДАТОК А Діаграма прецедентів додатку



## ДОДАТОК Б Підключення до Google Drive

```
namespace YourEmotionalHelper.Droid
{
    [Activity(Label = "YourEmotionalHelper", Icon = "@mipmap/icon", Theme = "@style/MainTheme", MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation)]
    public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
    {
        IDriveApiDriveContentsResult
        {
            const string TAG = "GDriveExample";
            const int REQUEST_CODE_RESOLUTION = 3;

            GoogleApiClient _googleApiClient;

            protected override void OnCreate(Bundle savedInstanceState)
            {
                base.OnCreate(savedInstanceState);
                Rg.Plugins.Popup.Popup.Init(this);
                Xamarin.Essentials.Platform.Init(this, savedInstanceState);
                global::Xamarin.Forms.Forms.Init(this, savedInstanceState);
                // Xamarin.Forms.Controls < PLATFORM > .Calendar.Init();
                LocalNotificationsImplementation.NotificationIconId = Resource.Drawable.emotion;
                AIForms.Renderers.Droid.SettingsViewInit.Init(); // need to write here
                PinItemViewRenderer.Init();
                LoadApplication(new App());
                base.OnCreate(savedInstanceState);

                SetContentView(Resource.Layout.Main);

                Button button = FindViewById<Button>(Resource.Id.myButton);
                button.Click += delegate
                {
                    if (_googleApiClient == null)
                    {
                        _googleApiClient = new GoogleApiClient.Builder(this)
                            .AddApi(DriveClass.API)
                            .AddScope(DriveClass.ScopeFile)
                            .AddConnectionCallbacks(this)
                            .AddOnConnectionFailedListener(onConnectionFailed)
                            .Build();
                    }
                    if (!_googleApiClient.IsConnected)
                        _googleApiClient.Connect();
                };
            }

            protected void onConnectionFailed(ConnectionResult result)
            {
                Log.Info(TAG, "GoogleApiClient connection failed: " + result);
                if (!result.HasResolution)
                {
                    GoogleApiAvailability.Instance.GetErrorDialog(this, result.ErrorCode, 0).Show();
                    return;
                }
                try
                {
                    result.StartResolutionForResult(this, REQUEST_CODE_RESOLUTION);
                }
                catch (IntentSender.SendIntentException e)
                {
                    Log.Error(TAG, "Exception while starting resolution activity", e);
                }
            }

            public void OnConnected(Bundle connectionHint)
            {
                Log.Info(TAG, "Client connected.");
                DriveClass.DriveApi.NewDriveContents(_googleApiClient).SetResultCallback(this);
            }

            protected override void OnActivityResult(int requestCode, Result resultCode, Intent data)
            {
                base.OnActivityResult(requestCode, resultCode, data);
                if (requestCode == REQUEST_CODE_RESOLUTION)
                {
                    switch (resultCode)
                    {
                        case Result.Ok:
                            _googleApiClient.Connect();
                            break;
                        case Result.Canceled:
                            Log.Error(TAG, "Unable to sign in, is app registered for Drive access in Google Dev Console?");
                            break;
                        case Result.FirstUser:
                            Log.Error(TAG, "Unable to sign in: RESULT_FIRST_USER");
                            break;
                        default:
                            Log.Error(TAG, "Should never be here: " + resultCode);
                            return;
                    }
                }
            }

            void IResultCallback.OnResult(Java.Lang.Object result)
            {
                var contentResults = (result).JavaCast<IDriveApiDriveContentsResult>();
                if (!contentResults.Status.IsSuccess) // handle the error
                    return;
                Task.Run(() =>
                {
                    var writer = new StreamWriter(contentResults.DriveContents.OutputStream);
                    writer.Write("Stack Overflow");
                    writer.Close();
                    MetadataChangeSet changeSet = new MetadataChangeSet.Builder()
                        .SetTitle("New Text File")
                        .SetMimeType("text/plain")
                        .Build();
                    DriveClass.DriveApi
                        .GetRootFolder(_googleApiClient)
                        .CreateFile(_googleApiClient, changeSet, contentResults.DriveContents);
                });
            }
        }
    }
}
```