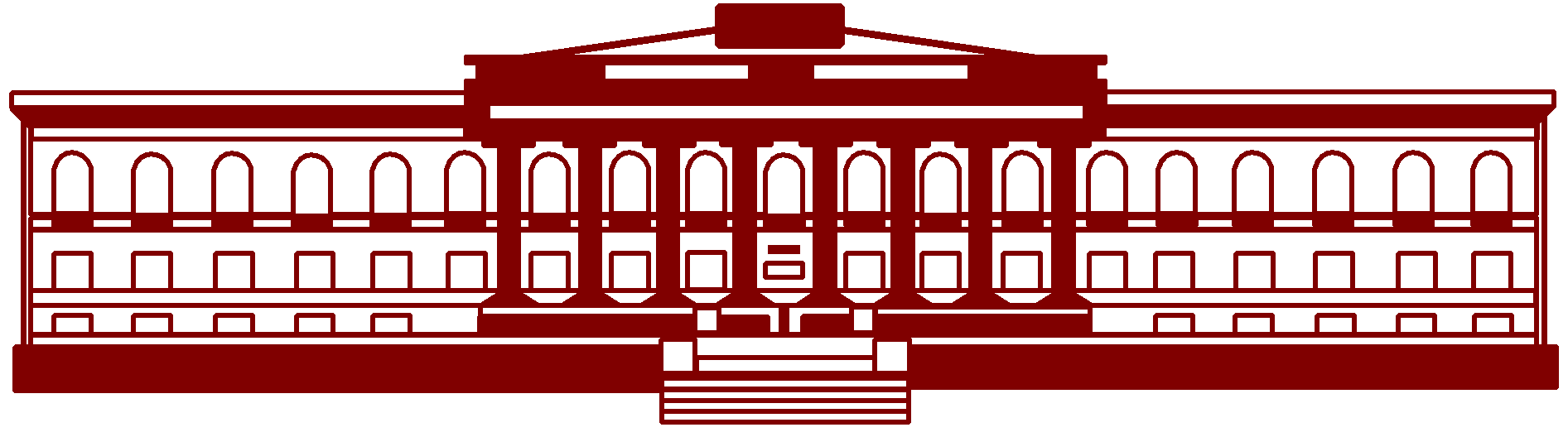
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

**Кафедра прикладних інформаційних систем**

****

**Звіт**

до виконання лабораторної роботи №1

з дисципліни «**Кросплатформне програмування**»

**Виконано:**

студ. групи ПП- 31, підгрупа 1

Вовк Олександр

**Київ – 2024**

“EasyTune”

**Вступ:**

"EasyTune" - це мультиплатформиний додаток для налагодження гітари, розроблений з використанням фреймворку Flutter, який дозволяє користувачам швидко та легко настроювати інструмент за допомогою мобільного телефону або комп'ютера.

**Огляд проекту:**

### Технічні вимоги:

* Весь додаток буде розроблений з використанням фреймворку Flutter для забезпечення мультиплатформовості.
* Використання стороннього API для аналізу аудіо та визначення частоти ноти (Web Audio API).
* Інтуїтивно зрозумілий і простий у використанні інтерфейс для запису звуку, відображення результатів аналізу та навігації користувача.
* Забезпечення сумісності з різними версіями операційних систем та різними моделями пристроїв.
* Захист конфіденційності даних користувачів, зокрема аудіофайлів, які можуть бути записані в процесі використання додатку.

### Огляд існуючих аналогів та їх переваги/недоліки:

1. GuitarTuna:
   * Переваги: Легкий у використанні, має широкий функціонал, включаючи тюнер, метроном та акорди.
   * Недоліки: Деякі додаткові функції доступні лише у платній версії, може бути відчутна затримка у відгуку.
2. BOSS Tuner:
   * Переваги: Відомий бренд, надійний тюнер, точний аналіз звуку.
   * Недоліки: Доступний тільки на платформі Android, може бути обмежений функціонал.
3. Fender Tune:
   * Переваги: Простий та ефективний інтерфейс, велика база даних з альтернативними настройками.
   * Недоліки: Деякі функції доступні тільки у платній версії, може бути відсутній підтримка для деяких пристроїв.

## **Вимоги до функціональності:**

### Функціональність:

* Запис звуку: Користувач може записувати звук гітари за допомогою мікрофону свого пристрою (комп'ютера або мобільного телефону).
* Аналіз звуку: Додаток використовує стороннє API для аналізу записаного звуку та визначення частоти ноти, яку генерує струна гітари.
* Візуалізація результатів: Результати аналізу відображаються на інтерфейсі додатку у вигляді графіку або числових значень, що вказують на поточну частоту.
* Настройка гітари: Користувач отримує відповідні рекомендації щодо необхідної частоти для кожної струни гітари, щоб вони були настроєні правильно.
* Мультиплатформовість: Додаток доступний як для операційних систем Windows, так і для Android.

## **Вимоги до інтерфейсу**

### Опис інтерфейсу користувача:

* + - Головний екран: На головному екрані розміщується кнопка для запуску запису звуку гітари. Після запису звуку відображається інформація про частоту ноти, яка була визначена.
    - Результати аналізу: Результати аналізу відображаються у вигляді числових значень частоти ноти та назви ноти. Інформація про кожну струну гітари відображається окремо з відповідним кольором або піктограмою.
    - Настройка гітари: Для кожної струни гітари вказується необхідна частота та нота для настройки. Користувач може переглянути додаткову інформацію та поради щодо настройки гітари.

### Вимоги до взаємодії з користувачем (UX):

* + - Простота використання: Інтерфейс повинен бути простим та зрозумілим для користувачів будь-якого рівня. Кнопки та елементи керування повинні бути легко доступними та зрозумілими.
    - Налагодження гітари: Користувач повинен отримати чіткі інструкції та рекомендації щодо настройки кожної струни гітари після аналізу звуку.
    - Візуалізація результатів: Результати аналізу повинні бути візуально привабливими та легко зрозумілими. Кольорова позначка частоти ноти допоможе користувачеві швидко розпізнати, чи настроєна струна гітари правильно.

## **Архітектура системи**

### Визначення загальної архітектури системи:

Система "EasyTune" буде розроблена з використанням архітектурного підходу Model-View-Controller, що дозволить легко розділити її на компоненти та забезпечити високу модульність і розширюваність.

### Компоненти системи та їх функції:

* + - Model: Відповідає за обробку даних та бізнес-логіку. Включає в себе логіку аналізу звуку, визначення частоти ноти та рекомендації щодо настройки гітари.
    - View: Відповідає за відображення інтерфейсу користувача. Включає в себе графічні елементи та взаємодію з користувачем.
    - Controller: Керує взаємодією між моделлю та представленням. Обробляє дії користувача та виконує відповідні дії, активуючи відповідні методи моделі.

## **Вимоги до продуктивності та масштабованості**

### Опис очікуваної продуктивності застосунку:

* + - Швидкість відгуку: Застосунок повинен реагувати на дії користувача миттєво, без помітних затримок.
    - Ефективне використання ресурсів: Додаток повинен ефективно використовувати ресурси пристрою (пам'ять, процесор) для забезпечення плавної роботи.
    - Стабільність: Застосунок повинен бути стабільним та надійним у різних умовах експлуатації.

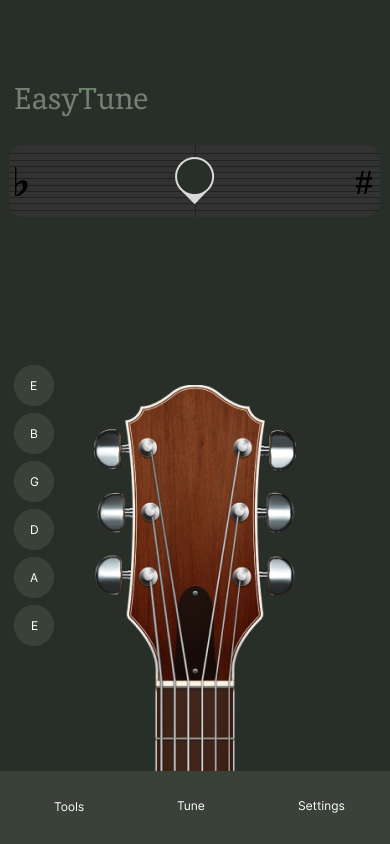
### Вимоги до швидкодії та обробки даних:

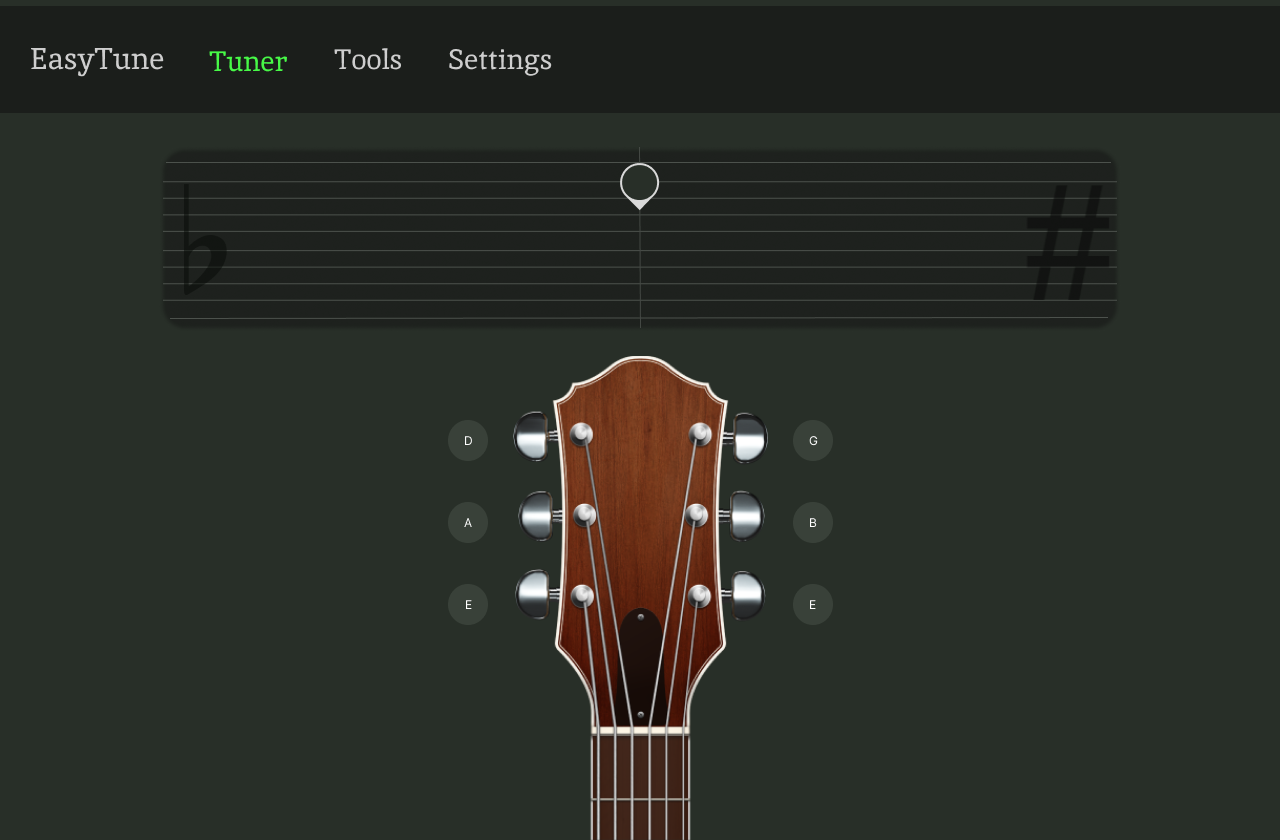
1. Швидкодія аналізу звуку: Система повинна бути здатна проводити аналіз звуку та визначати частоту ноти швидко та точно.
2. Оптимізація роботи з даними: Забезпечити ефективну роботу з великими обсягами аудіо-даних для забезпечення плавної роботи додатку.

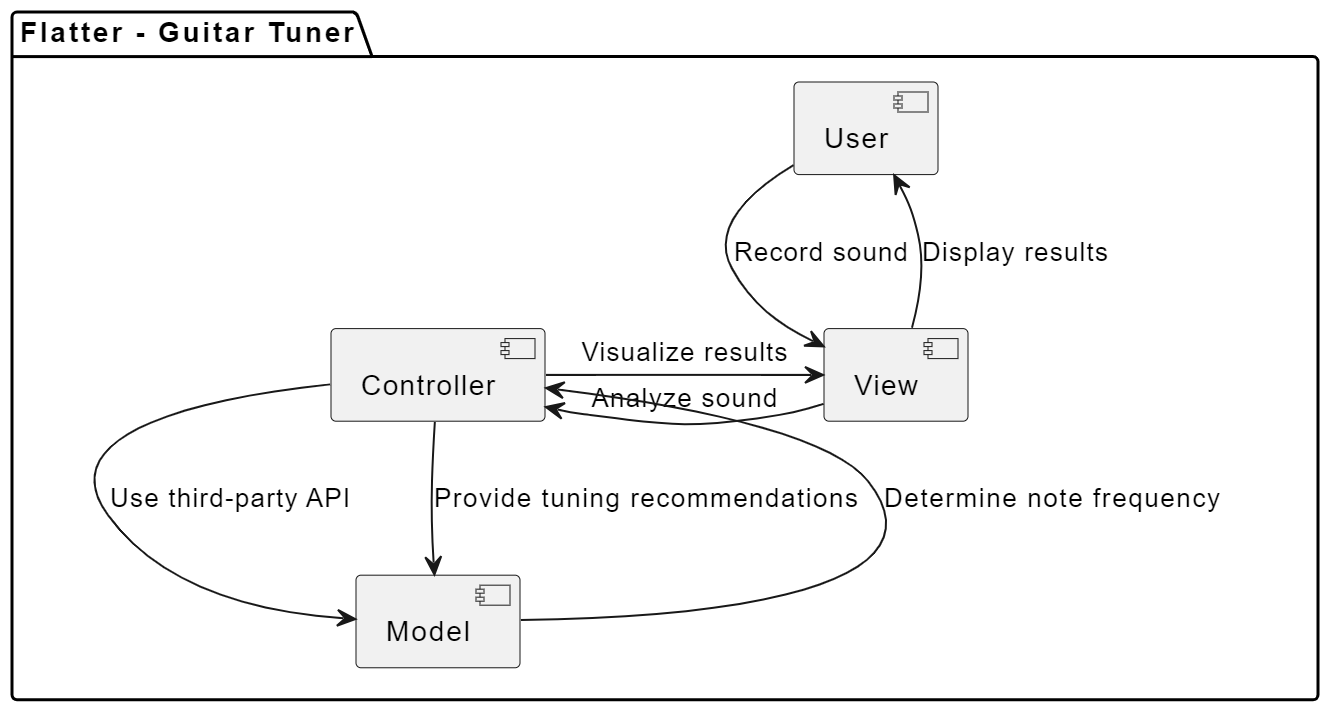
### Плани масштабованості для майбутнього розвитку:

1. Масштабування серверних компонентів: Якщо використовується серверна частина (наприклад, для обробки даних на більш потужних серверах), планується масштабування цих компонентів для забезпечення високої продуктивності при збільшенні обсягу користувачів.
2. Оптимізація алгоритмів: Планується постійне вдосконалення алгоритмів аналізу звуку та обробки даних для поліпшення швидкодії та ефективності додатку.

## **Додатки**

Макет мобільного застосунку:

Макет desktop застосунку:

Діаграми роботи застосунку: