Narzędzia i strony wykorzystane w pracy:

**Android Studio**

Android Studio to najpopularniejsze IDE wykorzystywane w procesie tworzenia aplikacji mobilnych w technologii Android. Narzędzie te w procesie tworzenia aktywności oraz fragmentów proponuje użytkownikowi wiele szablonów. Decydując się na stworzenie aktywności bądź fragmentu z dostępnych szablonów generuje się kod, który jest otoczony potrzebnymi zdarzeniami w samym kodzie Javy lub Kotlina (w zależności jaki język wybraliśmy podczas tworzenia projektu) oraz powiązane pliki xml.

Do tworzenia layoutów wykorzystuje się język XML, jednak Android Studio oferuje tworzenie poprzez graficzny interfejs. Po lewej stronie posiada spis wszystkich atrybutów, co znacznie ułatwia prace nad wyglądem aplikacji.

Jako, że Android Studio oparty jest na IDE od JetBrains posiada takie same funkcjonalności jak inne IDE od wydawcy. Nagodna uwagę funkcjonalność jest łatwe zarządzanie repozytorium zdalnym od GitHuba. Możemy łatwo utworzyć repozytorium, commitować i pushować zmiany, ale też możemy cofać wersje do odpowiedniego commita.

Android Studio dostarcza też możliwość otworzenia aplikacji na emulatorze. Aby to zrobić najpierw tworzymy urządzenie o wybranych przez nas parametrach, wybieramy mu system operacyjny. Następnie możemy uruchomić bildowanie i uruchomienie aplikacji poprzez naciśnięcie przycisku „Run” zaraz koło listy rozwijanej z dostępnymi urządzeniami.

Kolejną funkcjonalnością wartą uwagi jest asystent Firebasa. Z jego pomocą utworzymy projekt na Firebase, nawiążemy połączenie z istniejącym projektem, dodamy odpowiednie zależności do pliku gradle i dostaniemy przykładowy kod na łączenie się z funkcjonalnościami z Firebasa.

**Gradle**

**ADB**

Ten interfejs pozwala na wiele operacji na urządzeniu mobilnym. Jednak najważniejszą rzeczą, do której najczęściej wykorzystuje się ADB to możliwość instalowania aplikacji bezpośrednio na urządzenie mobilne. Wystarczy włączyć na urządzeniu tryb debugowania i uruchomić w oknie poleceń komendę adb install <ścieżka do pliku apk>. Jednak Android Studio pozwala na uruchomienie komendy poprzez jeden przycisk „Run” zaraz koło listy rozwijanej z dostępnymi urządzeniami.

**Biblioteka TarsosDSP**

Jest to biblioteka typu opensource do przetwarzania dźwięku. Cała biblioteka, jeśli było to możliwe, opiera się na czystej Javie. Na repozytorium tej biblioteki możemy uzyskać lik do przykładowych programów, stworzonych na podstawie tej biblioteki. Możemy zobaczyć takie aplikacje jak: detektor głośności dźwięku, spektrogram oraz gra polegająca na zaśpiewaniu dźwięku najlepiej jak to możliwe.

**GIMP**

Jest to darmowa aplikacja służąca do obróbki plików graficznych. W projekcie GIMP został użyty do stworzenia schematów akordów gitarowych.

**Firebase**

Jest to platforma wspierana posiadająca wiele funkcjonalności, które wspomagają pracę nad aplikacjami mobilnymi oraz webowymi. Łatwość zarządzania danymi przez intuicyjny graficzny interfejs sprawia, że do tej platformy interesują się coraz więcej programistów. W aplikacji użyłam takie funkcje jak: autentyfikacja (pozwala na łatwe zarządzanie zarejestrowanymi użytkownikami), „Cloud Firestoire” (jest to baza danych, która pozwala na ładowanie danych w czasie rzeczywistym, jest to baza NoSQL) oraz „FirebaseStorage” (jest to miejsce, w którym możemy przechowywać pliki, które później możemy pobierać w naszej aplikacji) .

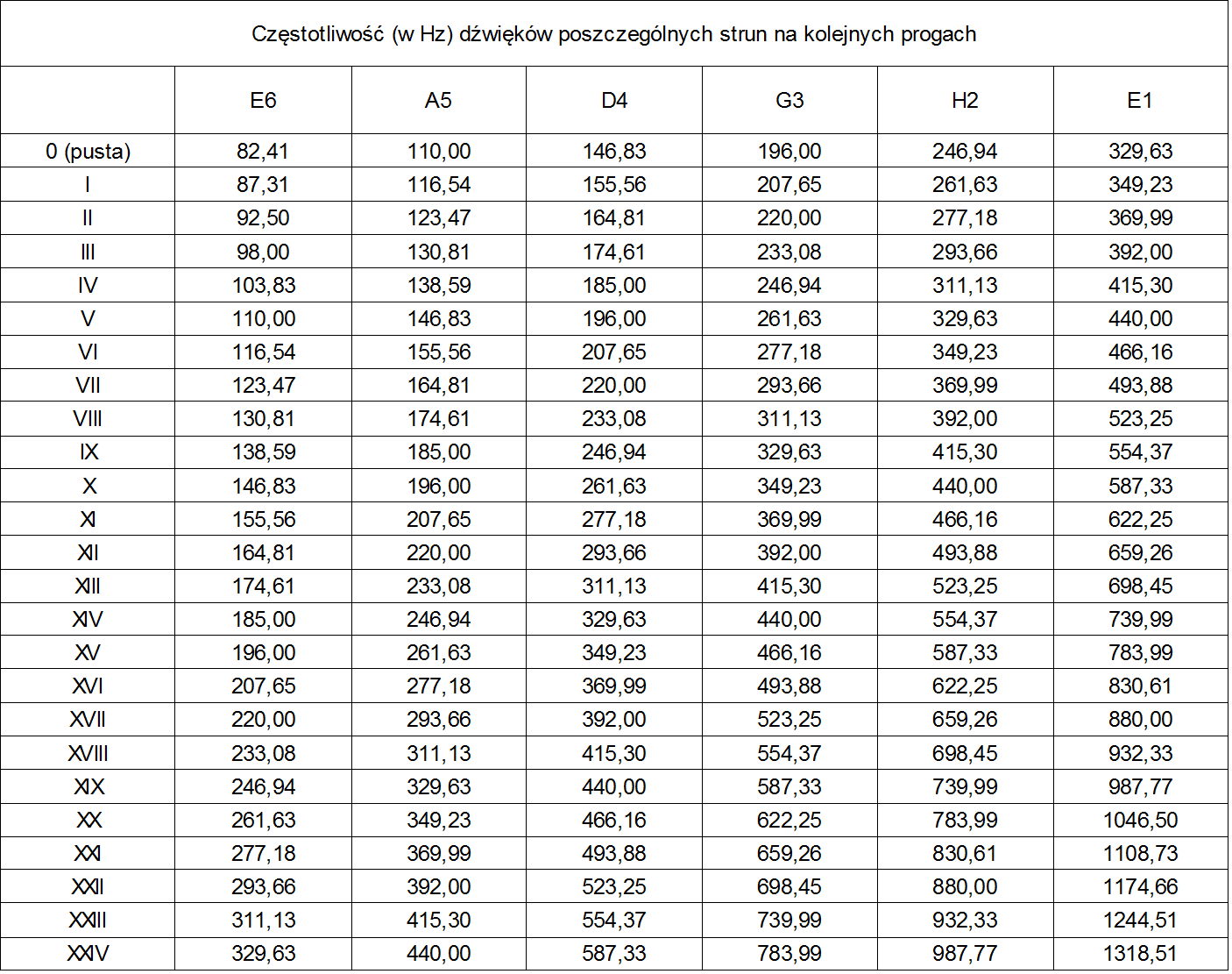
**Istniejące rozwiązania**:

Najbardziej popularną aplikacją dostępną w sklepie aplikacji dl systemu Android jest „GuitarTuna” od „Yousician Ltd.”. Korzystałam z niej dość duży okres i mogę stwierdzić, że do tej pory jest ona najlepsza wśród dostępnych aplikacji. Posiada ona stroik gitarowy, metronom ze zmiennymi parametrami oraz naukę akordów. Wszystkie funkcjonalności są bardzo dobrze zrobione. Aby nastroić gitarę nie trzeba zmieniać strun, nauka akordów wskazuje, która struna brzmi inaczej niż powinna. Do pewnego momentu można korzystać z tej aplikacji za darmo, jednak, jak większość aplikacji dostępnych w sklepie Play, aby dostać dostęp do wszystkich możliwości trzeba zapłacić. To sprawiło, że postanowiłam zrobić własną aplikacje, w której nie będę musiała płacić za dostęp do wszystkich funkcji.

Kolejną aplikacją wartą uwagi jest „Ultimate Tuner” od „Tabs4Acoustic – Free guitar tools”. Ta aplikacja skupia się wyłącznie na strojeniu. Mamy do wyboru większość strojeń gitarowych z przeróżnych typów muzycznych oraz różne gitary od gitary klasycznej do hawajskiej włącznie). Po kliknięciu w struny dostajemy dźwięk nastrojonej gitary i tak możemy nastroić ją ze słuchu.

Wiele innych aplikacji nie oferują strojenia poprzez mikrofon przykładem może być aplikacja „Gutar Tuner” od „appsmz”. Działa ona na zasadzie puszczania dźwięku po kliknięciu w daną strunę. Dźwięk można zapętlić poprzez kliknięcie w przycisk w górnej części ekranu. Dla początkujących jest to duże utrudnienie , ponieważ nie są w stanie stwierdzić czy dźwięk puszczany z aplikacji jest taki sam jak dźwięk uzyskany po szarpnięciu za strunę gitary, którą trzyma w ręku.

**Budowa gitary**

Gitara składa się z trzech istotnych części. Pierwszą z nich jest pudło rezonansowe zawierające duszę, otwór rezonansowy, podstrunnik oraz mostek, o który zaczepia się struny. Druga część zawiera gryf a na nim oznaczone są progi. Ostatnia część łączy się z gryfem poprzez siodełko i zawiera klucze na które nawijane są struny gitary. Po przyciśnięciu struny na odpowiednim progu oraz szarpnięciu struny otrzymujemy dźwięk o danej częstotliwości. I tak znając wymaganą częstotliwość na poszczególnych progach możemy nastroić gitarę. Wymagane częstotliwości przedstawia poniższa tabela:

Źródło: <http://skalba.com/wp-content/uploads/2016/06/Tabela1v1.png>

Biblioteka **TarsosDSP** pozwala nam na zbadanie częstotliwość dźwięku. Wystarczy utworzyć dispatcher, który odpowiada za ustalenie z jakiego źródła chcemy pobrać dźwięk, poziom czystości dźwięku dla odpowiednego urządzenia (w tym przpadku ustalamy wartość na 22050, wartość zalecana to 44100, jednak przy tej wartości szum był zbyt duży) oraz określić minimalną wartość bufora, która potrzebna jest do stworzenia nowej instancji klasy odpowiedzialnej za nagrywanie dźwięku (w tym przypadku 1024). Następnie przy użyciu stworzonego w bibliotece Handlera tworzymy nowy wątek (aby nagrywanie dźwięku nie blokowało głównego działania aplikacji oraz aby działanie aplikacji było płynne) odpowiedzialny za przetwarzania dźwięku. Po przepełnieniu naszego, wcześniej określonego, bufora dostajemy informacje o wyniku działania funkcji odpowiedzialnej za sprawdzenie częstotliwości dźwięku. Wynik działania przekazujemy dalej do metod sprawdzających różnice pomiędzy otrzymaną częstotliwością a częstotliwością docelową. Kolejno tworzymy obiekt klasy AudioProcessor i w konstruktorze określamy jaki algorytm chcemy użyć do przetwarzania dźwięku, Handler oraz dwie wartości typu int opisanych wcześniej wartości (poziom czystości dźwięku oraz wartość bufora). Następnie do dispathera dodajemy obiekt odpowiedzialny za przetwarzanie dźwięku oraz uruchamiamy wątek, który będzie działał aż do momentu przejścia do innej aktywności lub zmiany trybu strojenia.

