

Specyfikacja projektu aplikacji mobilnej do pobierania kursów walut

Autorzy:
Hubert Rędzia
Grzegorz Puczkowski

Repozytorium projektu: <https://github.com/grzepucz/currency-tracker>

1. Cel projektu:

Celem projektu jest stworzenie aplikacji mobilnej na urządzenia „Android”, która będzie pobierała kursy walut w czasie rzeczywistym oraz przedstawiała je w przyjaznej użytkownikowi formie.

2. Zakres projektu:

Użytkownik aplikacji będzie mógł:

1. Wybrać waluty, które będzie chciał śledzić.
2. Dla wybranych walut ustawiać alarmy o przekroczeniu wybranego progu kursu.
3. Ustawić powiadomienia o wysokich spadkach/wzrostach kursów
4. Za pomocą wbudowanego kalkulatora kursów przeliczać wybrane kursy
5. Sprawdzenie historycznych kursów walut.

3. Wybrane narzędzia:

Projekt napisany będzie w języku Java.

Do pobierania kursów walut wykorzystane będzie ogólnodostępne API – Fixer.io.

4. Kamienie milowe:

1. Zredagowanie specyfikacji wstępnej, która obejmuje krótki opis projektu – 20.10.2019
2. Zredagowanie specyfikacji rozszerzonej, obejmującej dokładny opis i zakres projektu – 8.11.2019
3. Implementacja początkowej wersji aplikacji ~50% - 23.11.2019
4. Uzupełnienie aplikacji o dodatkowe funkcjonalności oraz o warstwę graficzną ~90% - 09.12.2019
5. Ukończenie oraz oddanie gotowej aplikacji wraz z kompletną dokumentacją – 13.12 2019

5. Szczegółowa specyfikacja funkcjonalna projektu

Aplikacja przeznaczona będzie na urządzenia z systemem operacyjnym Android. Do działania aplikacji wymagane będzie połączenie z internetem, tak aby aplikacja responsywnie działała z API, z którego będzie pobierała dane o kursach walut. Kursy walut aktualizowane będą co 60 sekund. Aplikacja zawierać będzie funkcję obliczania kursów walut w dwóch kierunkach – normalnym i odwrotnym porządku. Będzie umożliwiała również wyświetlenie wartości kursu na wybrany dzień z historii notowań do kilku lat wstecz. Użytkownik będzie mógł śledzić wybrane kursy. Dodawanie kursów do śledzonych będzie zrealizowane za pomocą intuicyjnego przyciśnięcia przycisku, który będzie dostępny przy każdym z kursów na tabeli kursów, dostępnej w aplikacji. Zaimplementowana zostanie również funkcjonalność alertów, powiadomień informujących użytkownika o wzrostach/spadkach kursów – w zależności od upodobań, ustawień użytkownika.

6. Stos technologiczny

- Android studio - środowisko programistyczne (IDE), pozwalające wygodnie projektować, tworzyć i debugować własne programy na najpopularniejszą obecnie platformę systemową dla urządzeń mobilnych.
- API fixer.io – API zawierające dane dla 170 światowych walut, które są aktualizowane, co 60sekund. Oparte jest na module JSON, co pozwala na pobieranie danych w ciągu milisekund. Fixer API zawiera kilka punktów końcowych API, z których każdy zapewnia inną funkcjonalność:
 1. **Najnowsze kursy** - zwraca dane o kursie wymiany w czasie rzeczywistym (od 60 sekund do 10 minut w zależności od licencji) dla wszystkich

dostępnych lub określonego zestawu walut.

2. **Konwersja kursów** – punkt końcowy do przeliczania walut, którego można użyć do przeliczenia dowolnej kwoty z jednej waluty na drugą. Aby przeliczyć waluty, należy użyć punktu końcowego konwersji interfejsu API, dołączyć parametry z i do i ustaw je na preferowane kody waluty bazowej i docelowej. Możliwe jest również przeliczanie walut przy użyciu historycznych danych kursów walut.
3. **Historyczne kursy** – zwraca dane o historycznych wartościach kursów (aż do roku 1999) dla wszystkich dostępnych lub określonego zestawu walut
4. **Punkt końcowy danych zakresów czasowych** - zwraca historyczne dane o historycznym kursie wymiany między dwoma określonymi datami dla wszystkich dostępnych lub określonego zestawu walut.
5. **Punkt końcowy danych wahań** - zwraca dane dotyczące fluktuacji między dwiema określonymi datami dla wszystkich dostępnych lub określonego zestawu walut.

API Request:

```
https://data.fixer.io/api/convert
? access_key = API_KEY
& from = GBP
& to = JPY
& amount = 25
```

Rysunek 1 Przykładowy API Request

API Response:

```
{
  "success": true,
  "query": {
    "from": "GBP",
    "to": "JPY",
    "amount": 25
  },
  "info": {
    "timestamp": 1519328414,
    "rate": 148.972231
  },
  "historical": ""
  "date": "2018-02-22"
  "result": 3724.305775
}
```

Rysunek 2 Przykładowy API Response

- Junit - narzędzie służące do tworzenia powtarzalnych testów jednostkowych oprogramowania pisanego w języku Java. Głównymi zaletami tego narzędzia są między innymi: metoda najmniejszą jednostką testowania, przypadki testowe, oddzielenie testów od kodu, wiele mechanizmów uruchamiania, tworzenie raportów.
- Retrofit2 – to RESTowy klient dla Java i Androida. Ułatwia to pobieranie i przysyłanie plików JSON(lub innych plików) za pośrednictwem usługi internetowej przez REST. W Retrofitcie można skonfigurować, który konwerter jest używany do serializacji danych
Zasadniczo pozwala nam traktować wywołania API jako proste wywołania metod Java, dlatego definiujemy tylko typy parametrów żądania / odpowiedzi jako klasy Java. Cała komunikacja sieciowa oraz parsowanie JSON/XML: jest całkowicie obsługiwane przez Retrofit (np. GSON do parsowania JSON). Umożliwia wykonywanie synchronicznego lub asynchronicznego żądania http do zdalnego serwera.
- Room – Podstawowa struktura zapewnia wbudowaną obsługę pracy z surową zawartością SQL. Chociaż te interfejsy API są potężne, są dość niskiego poziomu i wymagają dużo czasu i wysiłku w użyciu: nie posiada weryfikacji nieprzetworzonych zapytań SQL w czasie kompilacji. W miarę zmiany schematu należy ręcznie zaktualizować zapytania SQL, których dotyczy problem. Ten proces może być czasochłonny i podatny na błędy. Musisz napisać dużo kodu typu „płyta podstawowa”, aby przekonwertować między zapytaniami SQL a obiektami danych Java. Room zajmuje się tymi problemami, zapewniając warstwę abstrakcji nad SQLite. Room składa się z trzech głównych elementów:

1. **Entity** reprezentuje dane dla pojedynczego wiersza tabeli, zbudowanego przy użyciu obiektu danych Java z adnotacjami. Każda jednostka jest utrwalana we własnej tabeli.
2. **DAO (Data Access Object)** definiuje metodę dostępu do bazy danych za pomocą adnotacji w celu powiązania SQL z każdą metodą.
3. **Database** to klasa posiadacza, która używa adnotacji do zdefiniowania listy encji i wersji bazy danych. Treść tej klasy określa listę DAO.

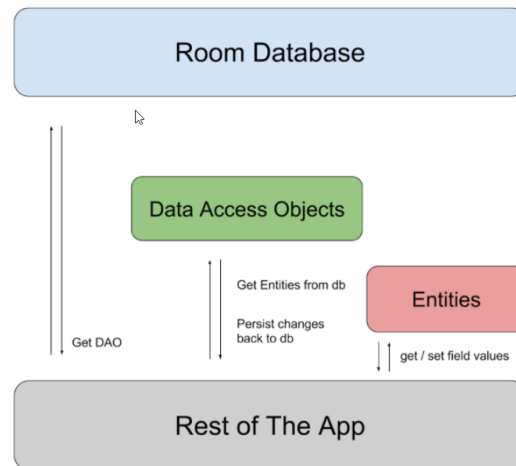
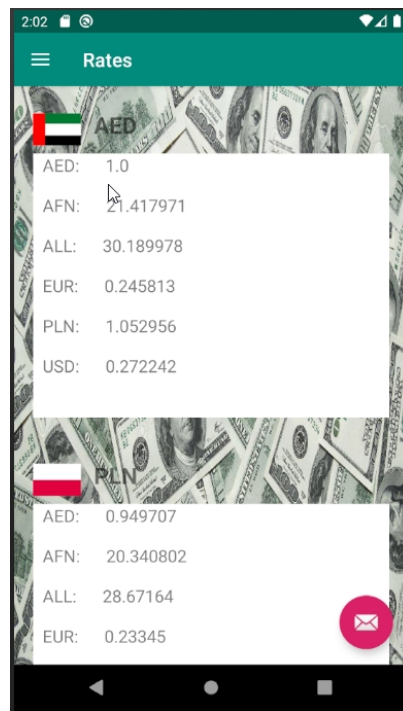


Figure 1. Room architecture diagram

7. Instrukcja użytkowania aplikacji

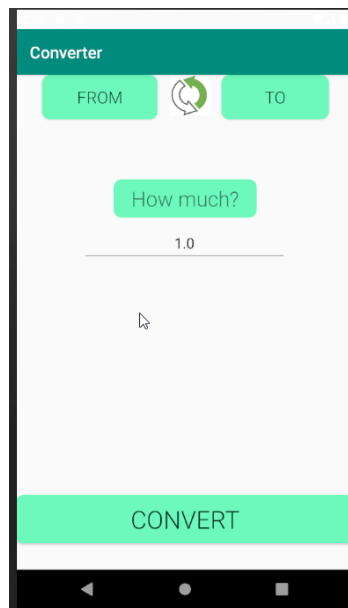
Aplikacja dzieli się na 4 moduły:

- Zakładka „Rates”, w której znajdują się śledzone przez użytkownika kursy. Po wciśnięciu w daną walutę wyświetlą się wszystkie kursy po przeliczeniu.

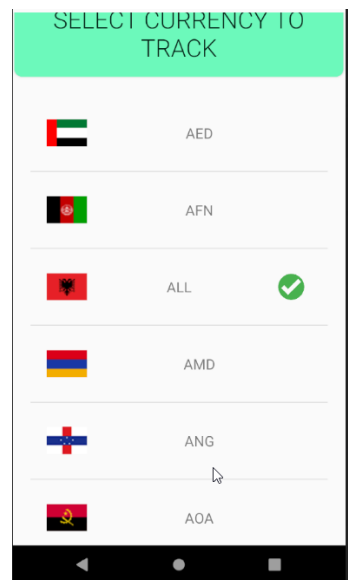


Rysunek 3 Lista ulubionych walut oraz ich kursy

- Zakładka „Convert”, w której znajduje się narzędzie do przeliczania kursów wybranych walut. Za pomocą przycisków „from” oraz „to” można wybrać z jakiej waluty do jakiej chcemy przekonwertować wpisaną kwotę w edytowalnym polu tuż poniżej etykiety „how much”



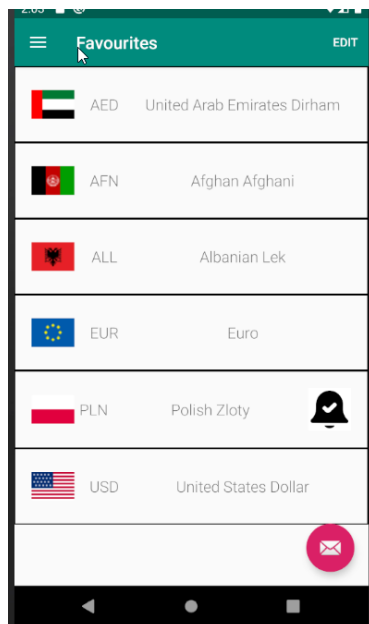
Rysunek 4 Widok do wyboru walut i kwoty konwersji



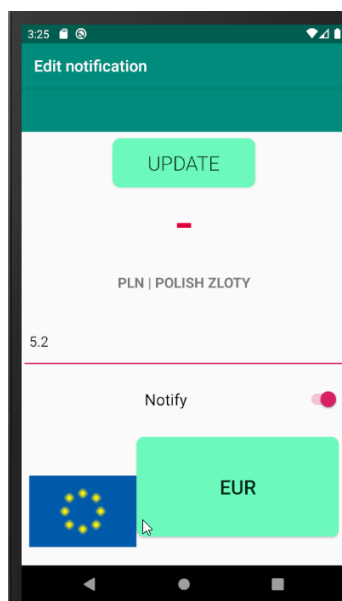
Rysunek 5 Widok wyboru waluty konwersji

- Zakładka „Favourites”, z której można edytować śledzone kursy. W każdym

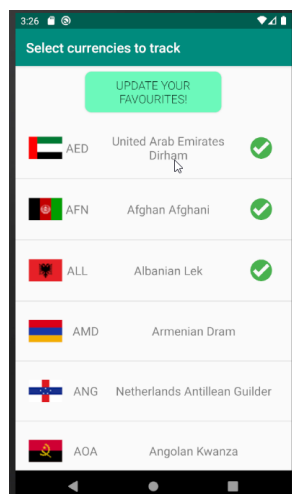
momencie możliwym jest usunięcie lub dodanie naszych ulubionych walut. Z tego widoku można również ustawić powiadomienia dla wybranej waluty. Po kliknięciu otworzy się nowy widok z możliwością ustawienia progu przekroczenia kursu.



Rysunek 6 Lista ulubionych walut



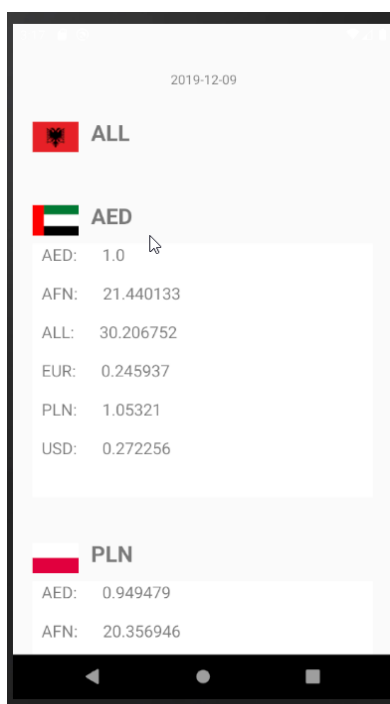
Rysunek 7 Okno edycji powiadomień



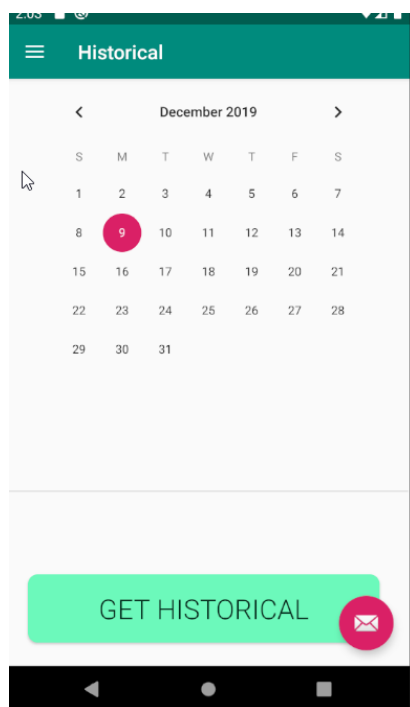
Rysunek 8 Widok wyboru ulubionych walut

- Zakładka „Historical”, w której można sprawdzić historyczne wartości kursów wybranych walut.

Po wejściu w widok wyświetli nam się kalendarz z aktualną datą. Za pomocą niego możemy wybrać datę dla której chcemy sprawdzić kurs, który z kolei możemy wybrać w następnym kroku. Po wciśnięciu na dowolną śledzoną walutę wyświetli się nam kurs.



Rysunek 9 Wyświetlanie wartości kursów walut



Rysunek 10 Wybór daty dla historycznych wartości kursów

8. Opis metodyki – Kanban

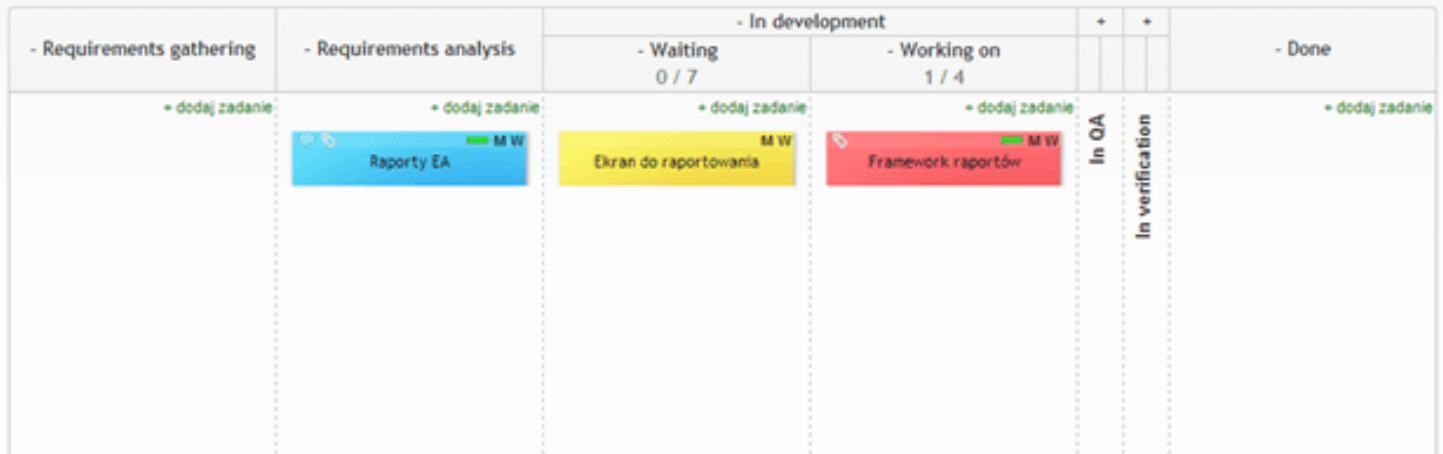
Metoda Kanban opiera się na wizualizacji procesu, podczas którego stosowana może być na przykład tak zwana tablica kanbanowa, która może mieć postać fizyczną lub wirtualną. Najprostszy układ takiej tablicy to podział na czynności „do zrobienia”, „w trakcie” oraz „zrobione”, które mogą być opisane na różnym poziomie szczegółowości. Stosowany system powinien być uzależniony od potrzeb w związku z którymi powstała tablica kanbanowa. Umożliwia ona przedstawienie zarówno procesów bardzo prostych, jak również wyjątkowo złożonych, dotyczących wielu zespołów. Tablica może łączyć różne zespoły skupione na jednym zadaniu.

Kanban pomaga wygładzić przepływ pracy w celu zmaksymalizowania „przepustowości” i osiągnięcia produktów o wysokiej jakości. W odróżnieniu do wielu metodologii wprowadzających przełomowe zmiany w procesach organizacji, Kanban jest ewolucyjnym systemem preferującym stopniowe ulepszanie istniejących procesów organizacji. To sprawia, że wdrożenie Kanban jest znacznie łatwiejsze od innych rozwiązań, co czyni go coraz bardziej popularnym narzędziem do zarządzania wszelkiego rodzaju pracami, w tym programowania zwinnego.

W świecie produkcji, przepływ i etapy Kanban są widoczne wg linii produkcyjnych i określonych stanowisk pracy w jednostce produkcyjnej. Wizualne wskaźniki dostępności zdolności do pracy mogą być oznaczone kartami fizycznymi lub innymi znakami na stanowiskach pracy lub za pomocą innych środków elektronicznych. W świecie usług i oprogramowania, gdzie nie ma fizycznej linii „produkcji”, przepływ pracy może być przedstawiony na fizycznej lub elektronicznej tablicy.

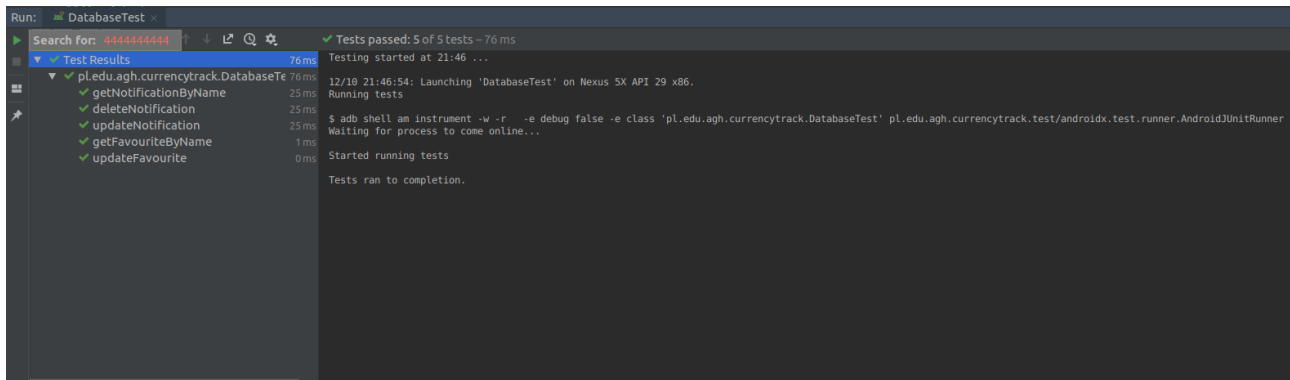
Tablica Kanban umożliwia zespołom wizualizację przepływu pracy lub strumienia wartości, a za pomocą „kart” lub „nalepek” – fizycznych lub elektronicznych „kart Kanban” – zapewnia wszystkim uczestnikom wizualny feedback odnośnie sposobu przepływu pracy w całym projekcie lub funkcji.

Dla mnie Kanban jest mega intuicyjny. W kolejnych postach przybliżę wybrane zasady jakie towarzyszą tej metodzie.



9. Testy aplikacji

9a. Testy bazy danych



9b. Testy API

