



Aplikacja wspierająca produktywność

Jakub Józefowicz, Aleksandra Mysiak, Adam Rozenek, Krystian Sztenderski



Założenia

- Połączenie techniki Pomodoro z listą zadań do zrobienia
- Podział czasu pracy na jednostki - sesje
- Umożliwienie zarządzania czasem pracy i przerwami
- Odliczanie czasu pracy w automatyczny sposób przydzielające zadania na daną sesję pracy
- Możliwość grupowania zadań w kategorie
- Możliwość definiowania tygodniowego planu zadania



Wymagania funkcjonalne, cz. 1

- definiowanie zadań z nazwą, przewidywaną liczbą sesji pomodoro
- wyświetlanie dodanych zadań
- edycja liczby sesji pomodoro dla danego zadania
- ustawianie długości sesji pomodoro i przerw, liczby sesji pomodoro w cyklu
- tworzenie i edycja kategorii zadań
- przypisanie i zmiana kategorii zadania
- definiowanie dla zadań tygodniowego, powtarzalnego planu sesji
- edycja tygodniowego planu sesji dla danego zadania
- usuwanie zakończonej sesji z kolejki



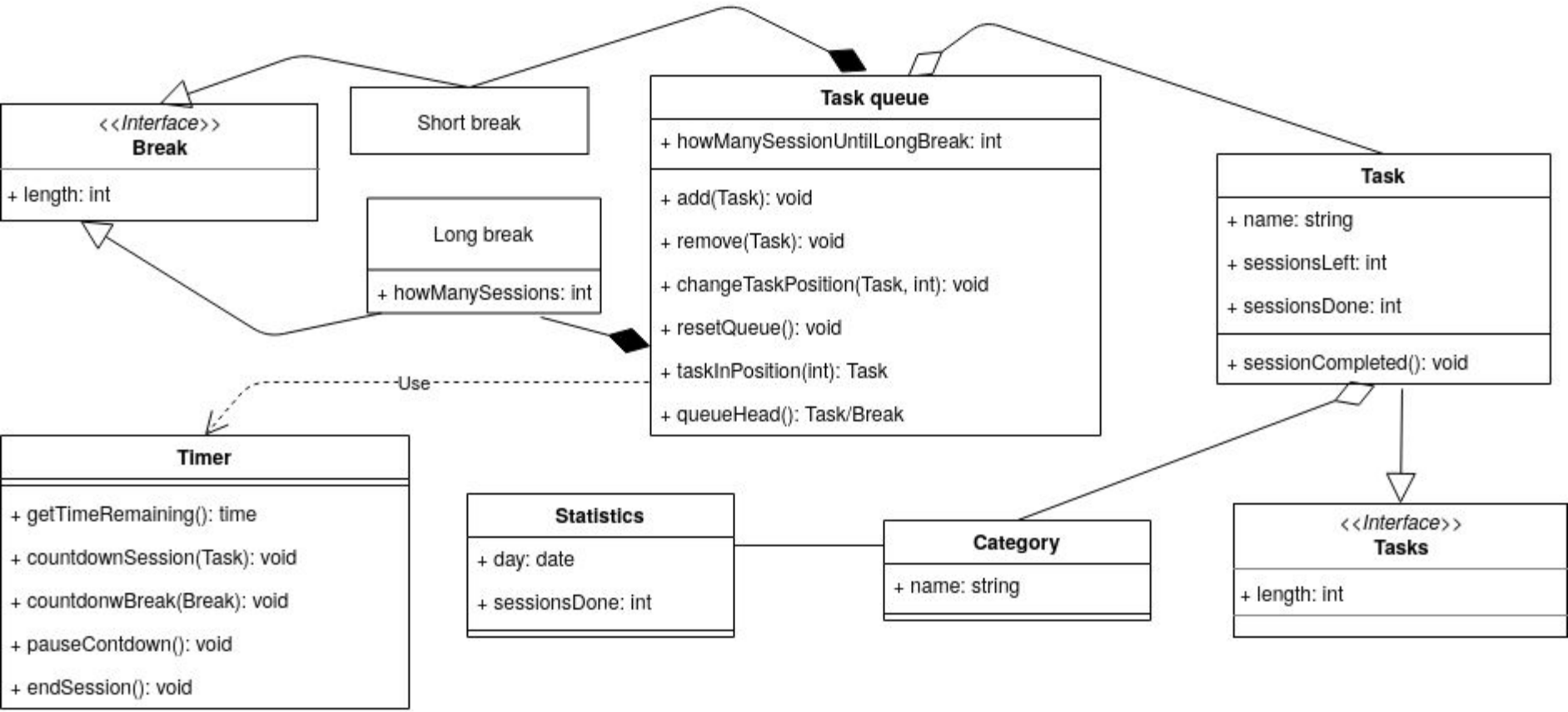
Wymagania funkcjonalne, cz. 2

- ręczne dodawanie do kolejki sesji dla dodanych wcześniej zadań
- zmiana przez użytkownika kolejności sesji w kolejce, czyszczenie kolejki
- odliczanie czasu do zakończenia kolejnej sesji z kolejki
- odliczanie czasu przerwy pomiędzy zakończoną sesją a kolejną sesją z kolejki
- zliczanie wykonanych sesji dla danego zadania i wyświetlanie ich liczby
- możliwość wcześniejszego zakończenia sesji przez użytkownika
- możliwość wcześniejszego zakończenia przerwy przez użytkownika
- wyświetlanie powiadomień o upływie czasu sesji/przerwy
- wyświetlanie statystyk na temat wykonanych sesji



Architektura, cz. 1

Czyli jak sobie wyobrażaliśmy model logiczny aplikacji...





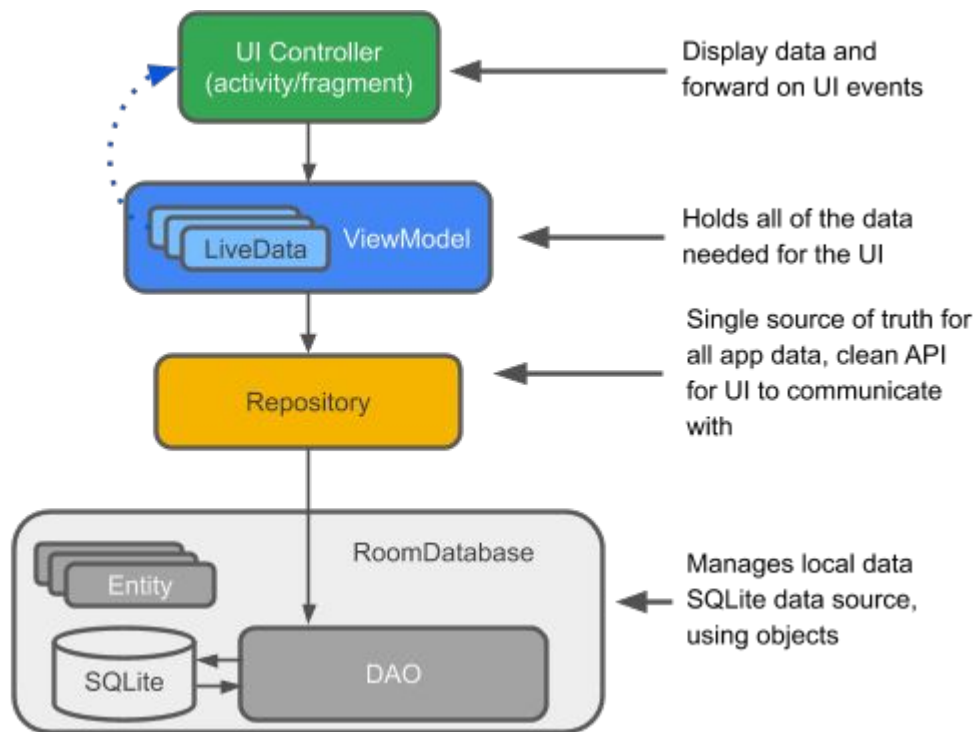
Architektura, cz. 2

...i co z tego wyszło.

Architektura

w zrozumiałym skrócie

- Fragmenty, odpowiedzialne za UI, komunikują się z ViewModelami
- Komunikacja między fragmentami również za pośrednictwem VM
- LiveData umożliwia monitorowanie zmian w bazie danych
- Repozytorium zbiera dane z różnych źródeł, rozdziela operacje między wątki
- Data Access Object zapewnia metody dostępu do bazy danych
- Właściwa baza danych komunikuje się jedynie z DAO



Organizacja software development process

Podział pracy na sprinty

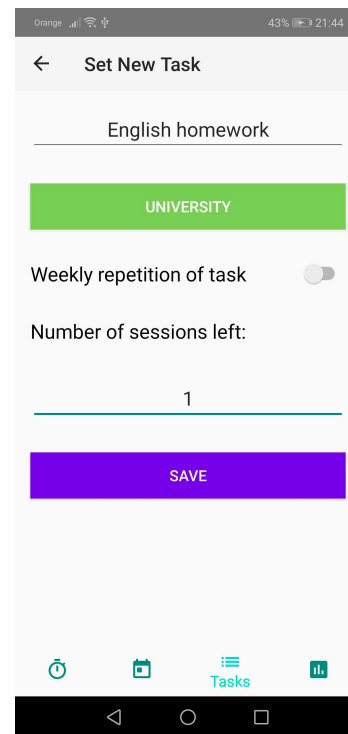
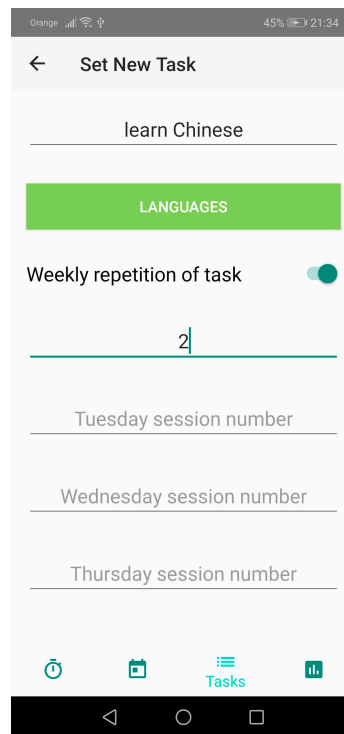
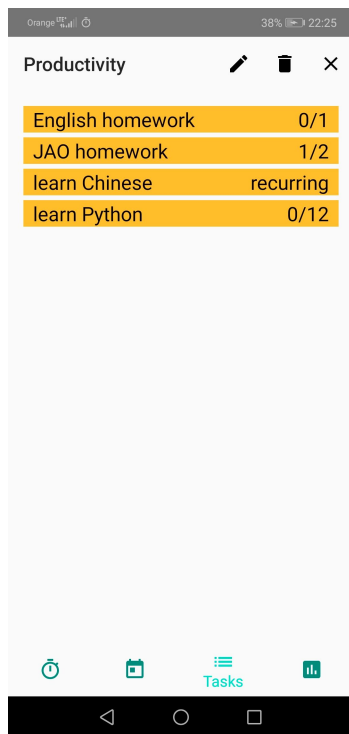
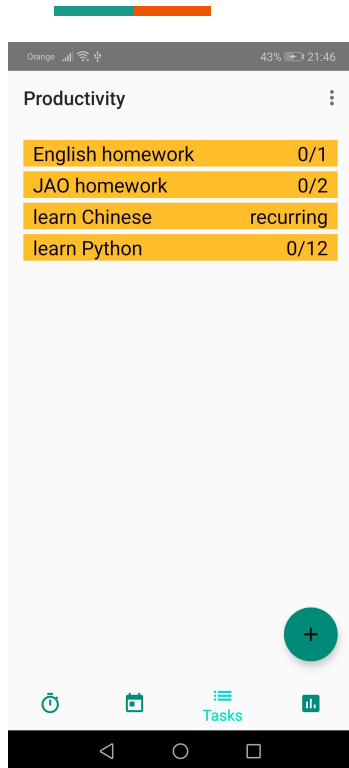
Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
<ul style="list-style-type: none">→ Podstawowa baza danych dla zadań→ Dokumentacja→ Edycja i tworzenie zadań	<ul style="list-style-type: none">→ Wzbogacenie zadania o kategorię i tygodniowy plan→ Ustawianie długości sesji pracy i przerw	<ul style="list-style-type: none">→ Ekran timera→ Wprowadzenie kolejki zadań→ Możliwość edycji i czyszczenia kolejki	<ul style="list-style-type: none">→ Odliczanie czasu sesji i przerwy przez timer→ Pobieranie zadań z kolejki→ Powiadomienia o upływie czasu	<ul style="list-style-type: none">→ Wyświetlanie statystyk na temat wykonanych zadań→ Przygotowanie prezentacji projektu

Organizacja software development process

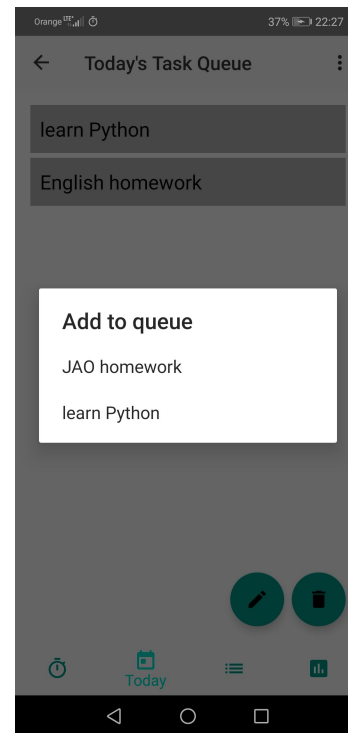
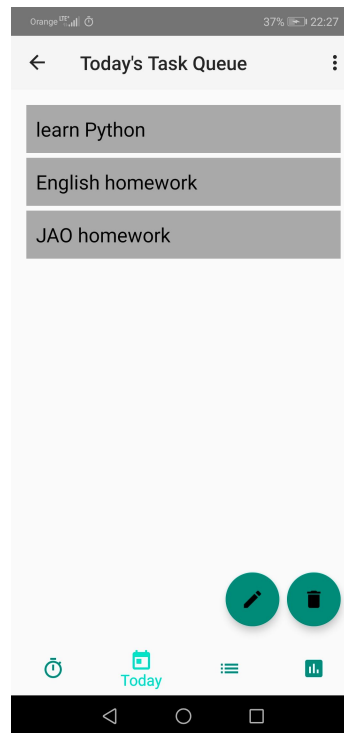
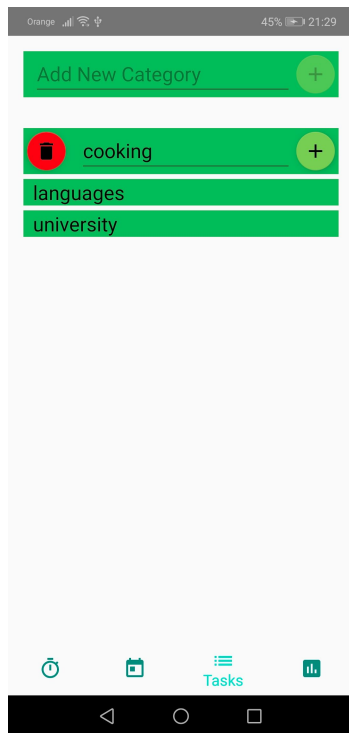
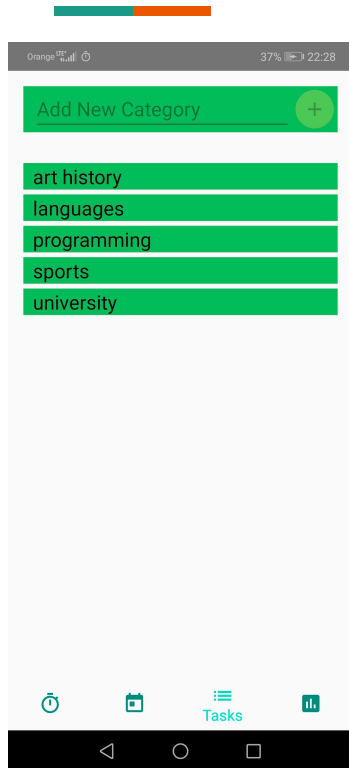
Wykorzystane narzędzia

- GitHub - współdzielone repozytorium z kodem, zakładka “Projects” z podziałem zadań
- Slack - narzędzie do komunikacji w wątkach
- Discord - narzędzie do komunikacji głosowej, ze współdzieleniem ekranu
- Google Jamboard - wirtualna tablica umożliwiająca współdzielenie w czasie rzeczywistym

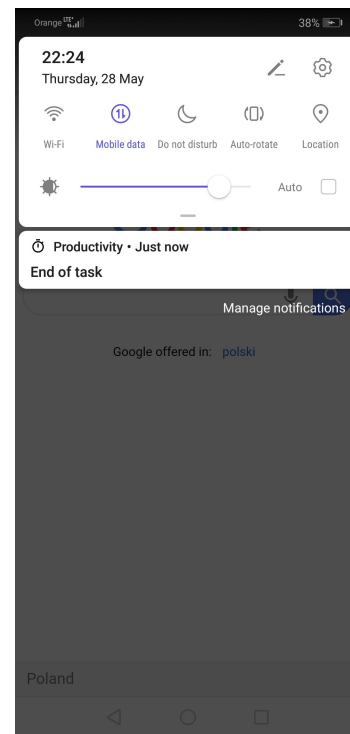
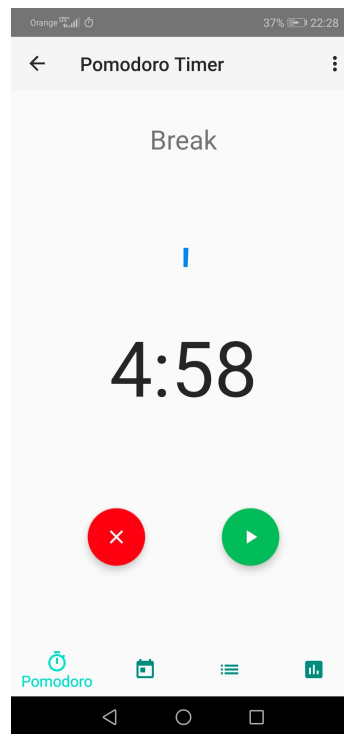
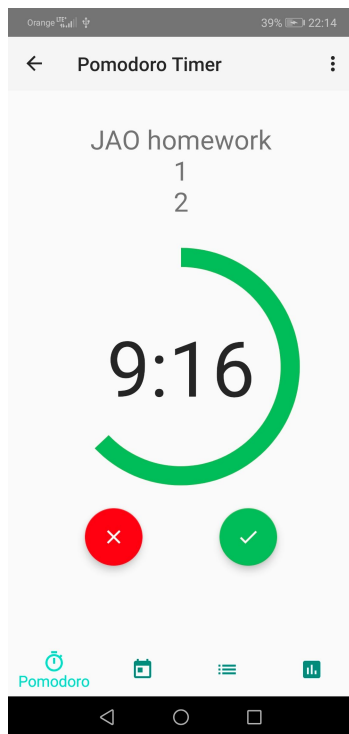
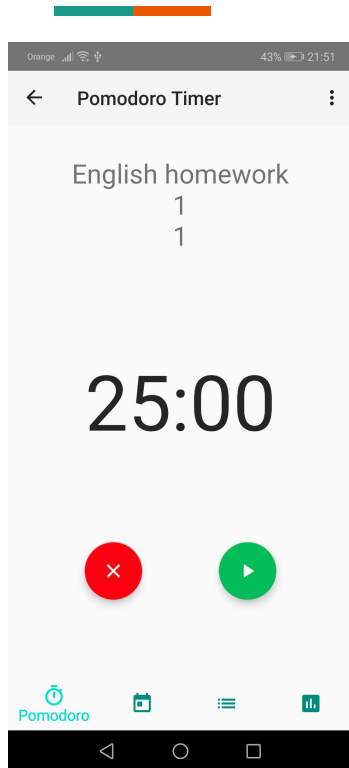
Lista zadań, dodawanie i edycja



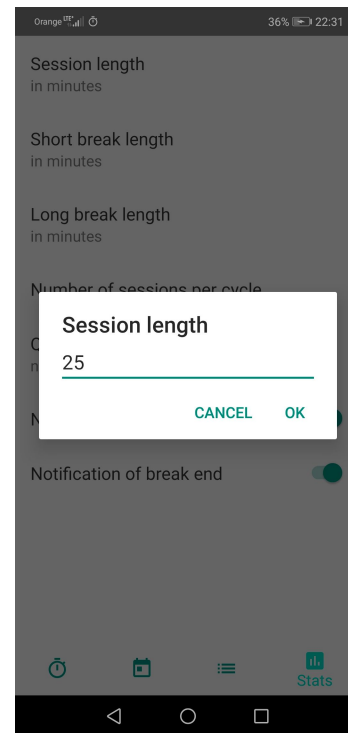
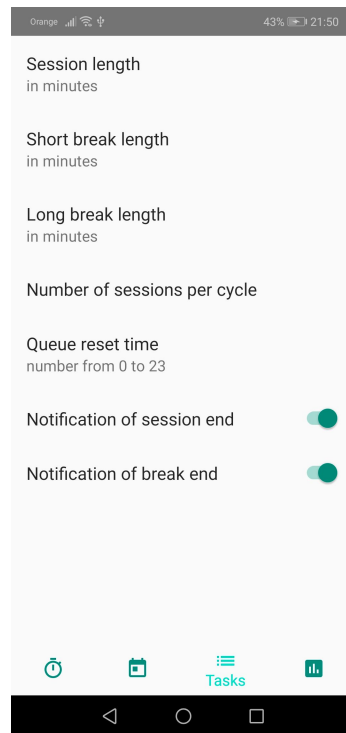
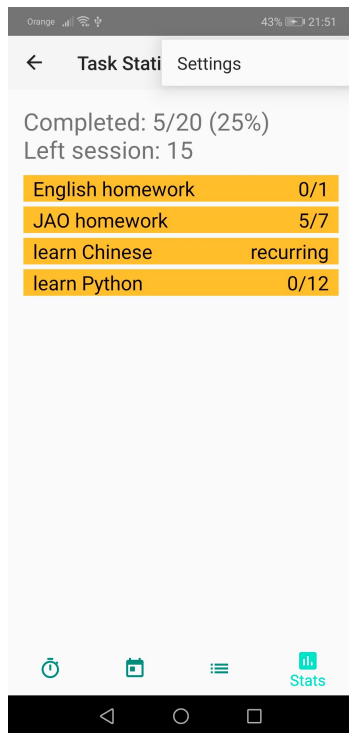
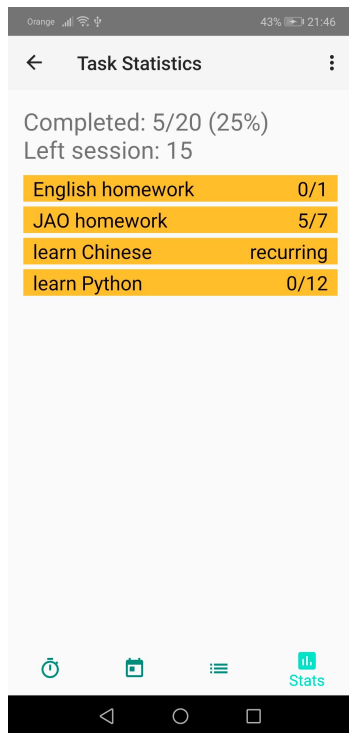
Lista kategorii i kolejka zadań



Timer



Statystyki i ustawienia



Dziękujemy za uwagę!

