

Zespołowe przedsięwzięcie inżynierskie

Rok akademicki 2019/20 Semestr letni

Termin zajęć: Czwartek, 15:15-18:45

Zespół nr 4

Skład zespołu: Karol Belina, Dominik Bordzio, Piotrek Kosturek, Patryk Mazurek

Calendarola

Release 2, sprint 5

Historyjki użytkownika pozostałe do wykonania po poprzednich sprintach:

1. Jako użytkownik chcę móc zaimportować wszystkie wydarzenia z mojego kalendarza Google, aby móc od razu planować na ich podstawie.
2. Jako użytkownik chcę móc przesuwać zaplanowane wydarzenia i zadania w kalendarzu, aby w szybki sposób zmienić jego termin.
3. Jako użytkownik chcę podać swoje preferencje (np. godziny snu), aby automatyczne harmonogramowanie je uwzględniało.
4. Jako użytkownik chcę usuwać zadania i wydarzenia wykonane lub anulowane, aby nie zajmowały niepotrzebnie miejsce.
5. Jako użytkownik chcę, aby harmonogramowanie zaplanowanych zadań i wydarzeń odbywało się automatycznie, ponieważ ułatwi mi to organizację czasu.
6. Jako użytkownik chcę mieć możliwość rejestracji i logowania, aby mieć dostęp do własnego, prywatnego kalendarza.
7. Jako użytkownik chcę definiować i modyfikować nazwę, opis, planowany czas trwania, miejsce odbycia, ograniczenia (czasowe, następstwa, podzielność, cykliczność) i priorytet zadania, aby w kolejność zadań mogła być lepiej dla mnie dopasowana.

Cel sprintu:

Poprawa funkcjonalności dodawania, edycji oraz ponownego ustawienia wydarzeń oraz zadań w kalendarzu. Dodanie nowego menu nawigacyjnego. Dodanie prerekwizytów do zadań. Usprawnienie algorytmu harmonogramowania.

Data rozpoczęcia / zakończenia:

07.05.2020 / 21.05.2020

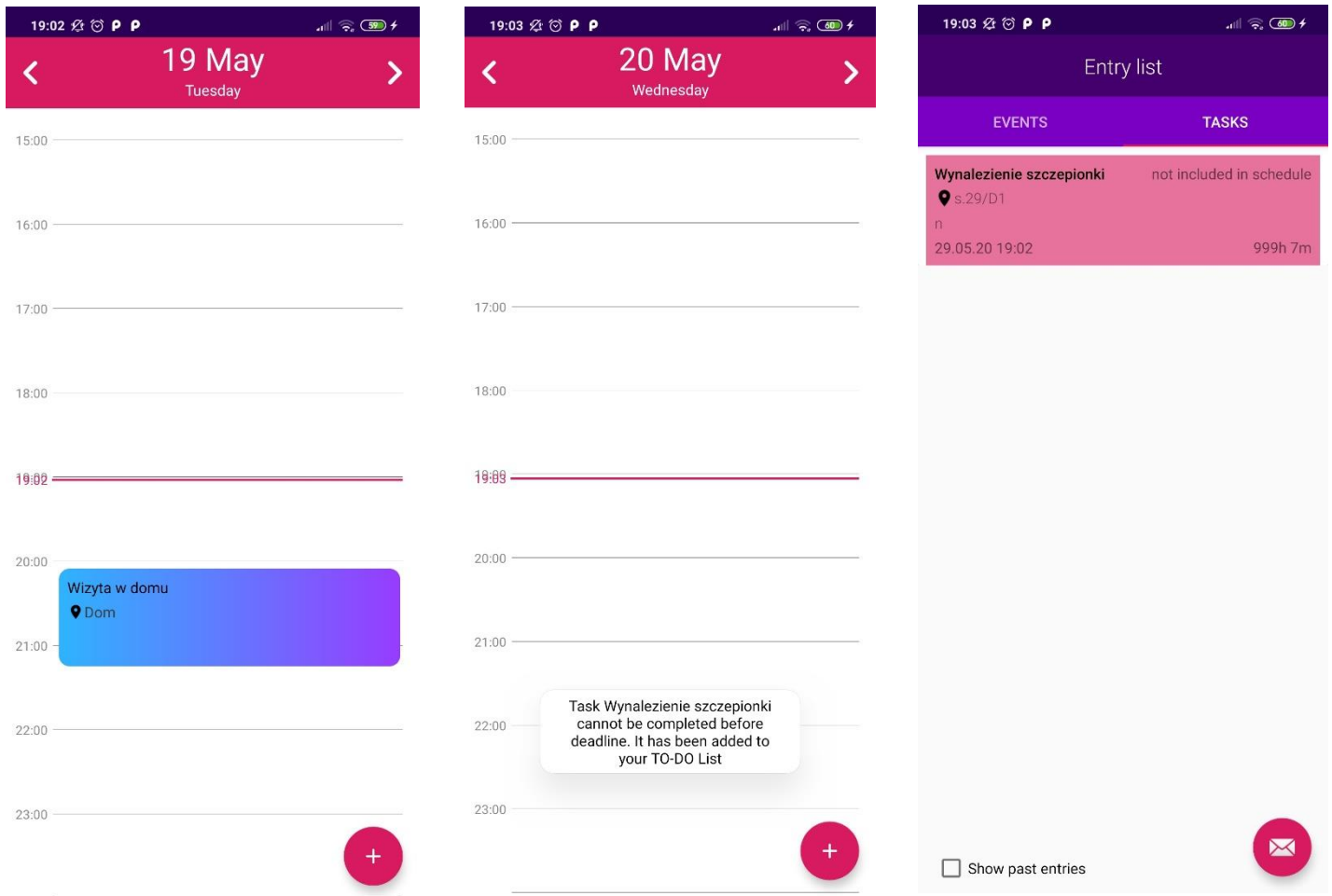
Historyjki użytkownika wybrane do wykonania w ramach sprintu:

- 2. Jako użytkownik chcę móc przesuwać zaplanowane wydarzenia i zadania w kalendarzu, aby w szybki sposób zmienić jego termin.
- 4. Jako użytkownik chcę usuwać zadania i wydarzenia wykonane lub anulowane, aby nie zajmowały niepotrzebnie miejsce.
- 5. Jako użytkownik chcę, aby harmonogramowanie zaplanowanych zadań i wydarzeń odbywało się automatycznie, ponieważ ułatwi mi to organizację czasu.
- 7. Jako użytkownik chcę definiować i modyfikować nazwę, opis, planowany czas trwania, miejsce odbycia, ograniczenia (czasowe, następstwa, podzielność, cykliczność) i priorytet zadania, aby w kolejność zadań mogła być lepiej dla mnie dopasowana.

Zadania sprintu:

	Historyjka	Wykonawca	Status	Estymacja	Wykonanie	Pozostało
Swipe'owanie dni i wyświetlanie dnia w widoku agendy	4.	PK	Wykonane	3h	2h	1h
Dodawanie niewykonanych zadań do listy bez wyświetlania ich w kalendarzu	4.	PK	Wykonane	1h	1h 30min	-30min
Wyświetlanie przeszłych zadań/wydarzeń jako opcja w widoku listy	4.	PK	Wykonane	1h	1h	0h
Zmiana priorytetu zadania na opisowy	7.	DB	Wykonane	1h	1h	0h
Uwzględnienie minimalnej długości segmentu w algorytmie	5.	PK	Wykonane	2h	45min	1h 15min
Przechowywanie listy lokalizacja zadania	7.	DB	Wykonane	2h 30min	4h 30 min	-2h
Refaktoryzacja aktywności aplikacji na fragmenty		KB	Wykonane	3h	4h	-1h
Dodanie menu nawigacyjnego	2.	KB	Wykonane	3h	2h	1h
Dodawanie i edycja prerekwizytów zadania	5., 7.	PM	Wykonane	5h	5,5h	-30min

- Swipe'owanie dni i wyświetlanie dnia w widoku agendy. Do przełączenia między poniższymi ekranami należy zastosować gest swipe lub użyć przycisków (strzałek). Dodanie Taska, który nie mieści się w kalendarzu spowoduje dodanie go do zadań niewykonalnych w aktualnym harmonogramie. Do widoku Entry list została dodana opcja Show past entries, do wyświetlania ukończonych zadań bądź wydarzeń.



- Refaktoryzacja aktywności do fragmentów

```

22 - class AddUserActivity : AppCompatActivity() {
27 + class AddUserFragment : Fragment() {

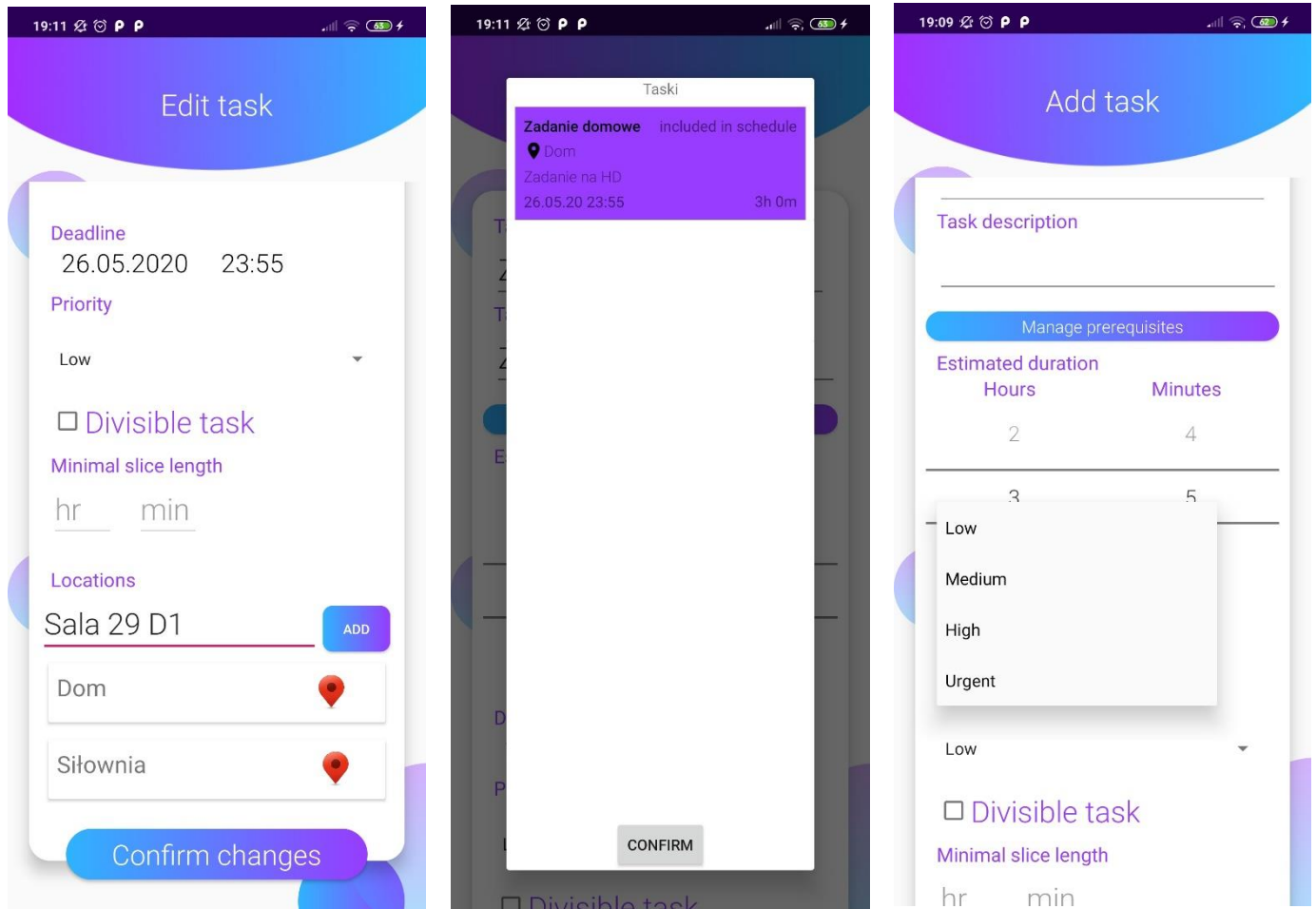
```

```

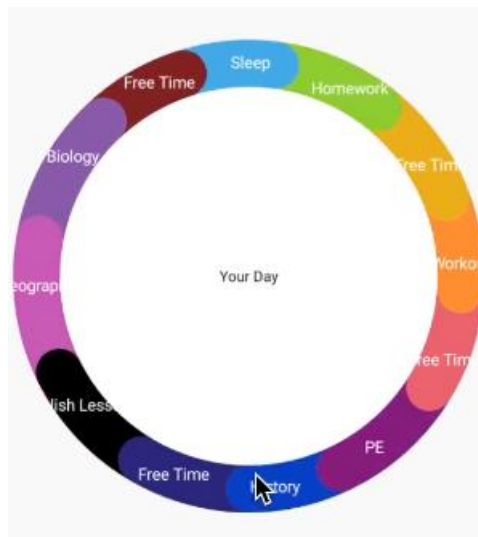
26 + class StartupFragment : Fragment() {
27 +
28 +     @ExperimentalTime
29 +     override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
30 +         loadTasksFromMemory(requireContext())
31 +         loadEventsFromMemory(requireContext())

```

- Nowy ekran nawigacyjny

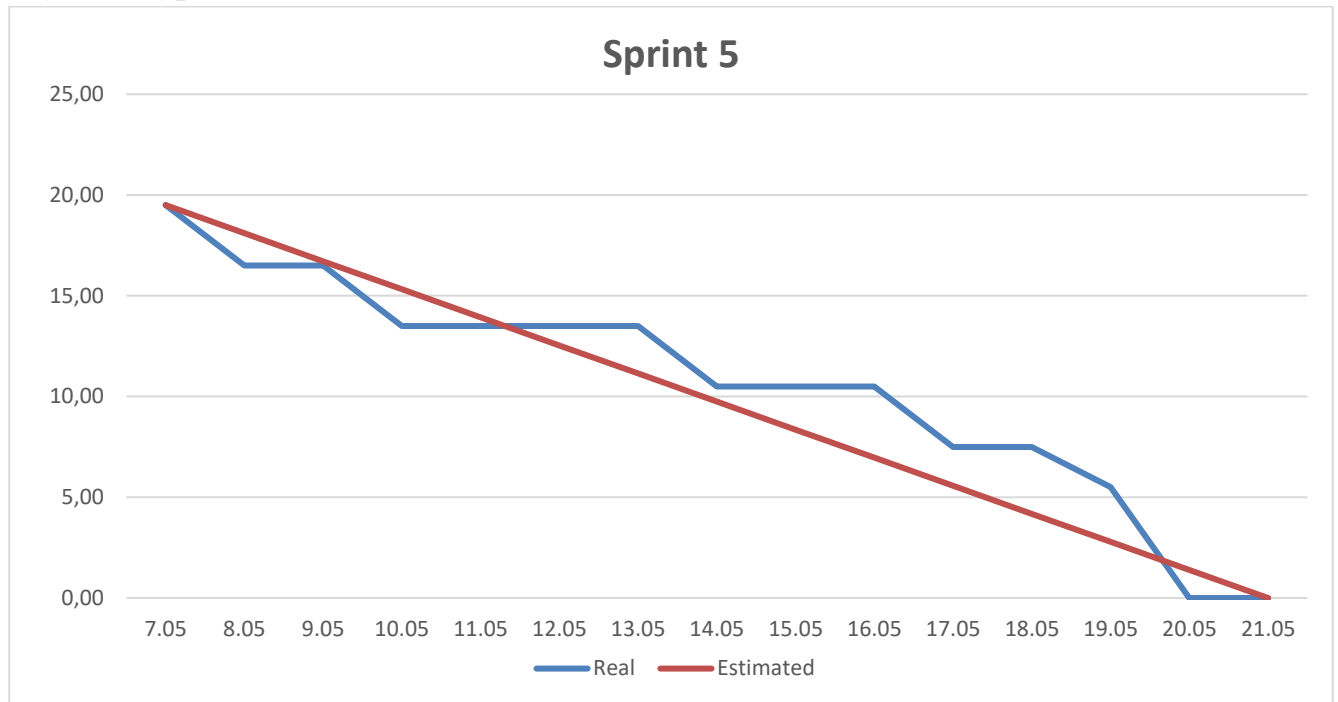


- Nowy ekran nawigacyjny



- W programie dodano również uwzględnianie minimalnej długości segmentu wydarzenia przez algorytm harmonogramowania

Wykres wypalania (*burndown chart*)



Historyjki użytkownika do wykonania w ramach kolejnych sprintów (pozostałe; stan końcowy *Product Backlog*):

1. Jako użytkownik chcę móc zaimportować wszystkie wydarzenia z mojego kalendarza Google, aby móc od razu planować na ich podstawie.
2. Jako użytkownik chcę podać swoje preferencje (np. godziny snu), aby automatyczne harmonogramowanie je uwzględniało.
3. Jako użytkownik chcę usuwać zadania i wydarzenia wykonane lub anulowane, aby nie zajmowały niepotrzebnie miejsca.
4. Jako użytkownik chcę, aby harmonogramowanie zaplanowanych zadań i wydarzeń odbywało się automatycznie, ponieważ ułatwi mi to organizację czasu. – historyjka niedokończona, harmonogramowanie nie uwzględnia niektórych zależności
5. Jako użytkownik chcę mieć możliwość rejestracji i logowania, aby mieć dostęp do własnego, prywatnego kalendarza.

Plan kolejnego sprintu (zadania następnego sprintu):

Zadanie	Historyjka	Wykonawca	Estymacja
Usprawnienie algorytmu harmonogramowania	4.	PK	5h
Dodanie ekranu preferencji	2.	PM	5h
Usunięcie taska/eventu	3.	DB	2h
Zastosowanie fragmentów zamiast aktywności, refaktoryzacja		KB	5h
Zmiana wyglądu wyświetlanych tasków		DB	3h