



PROJEKT INDYWIDUALNY
DOKUMENTACJA

**APLIKACJA DO SYMULACJI METODY KANBAN
Z WYKORZYSTANIEM FLUTTER**

OPIEKUN:
mgr. inż. Krzysztof Marek

SPIS TREŚCI:

1. OPIS OGÓLNY	2
1.1. NAZWA PROGRAMU	2
1.2. PORUSZANY PROBLEM	2
1.3. UŻYTKOWNIK DOCELOWY	2
2. OPIS FUNKCJONALNOŚCI	2
2.1. URUCHOMIENIE PROGRAMU	2
2.2. KORZYSTANIE Z PROGRAMU	2
3. PRZYKŁADOWY SCENARIUSZ DZIAŁANIA PROGRAMU	3
3.1. SCENARIUSZ OGÓLNY	3
3.2. SCENARIUSZ SZCZEGÓŁOWY	3
4. FORMATY DANYCH I STRUKTURA PLIKÓW	4
4.1. POJĘCIA I OBIEKTY DZIEDZINY APLIKACJI (SŁOWNIK DZIEDZINY)	4
4.2. PLIKI ZAPISU STANU SYMULACJI	5
5. DRZEWO KATALOGU PROJEKTU I PRZEZNACZENIE POSZCZEGÓLNYCH PAKIETÓW	6
5.1. GŁÓWNY KATALOG PROJEKTU.	6
5.2. PODKATALOG „common”	7
5.3. PODKATALOG „features”	8
5.4. PODPODKATALOG „main_page”	9
6. ŹRÓDŁA I UŻYTE BIBLIOTEKI	10

Rozdział 1: OPIS OGÓLNY.

Podrozdział 1.1: Nazwa programu.

Pełna nazwa programu to **Kanban Method's Simulator**. Inne nazwy wykorzystywane w projekcie oraz dozwolone skróty to:

- KanbanSimApp
- kanbansim
- kanbansimapp
- KanbanSimulator

Podrozdział 1.2: Poruszany problem.

Problemem poruszonym i rozwiązywanym przez ten program jest zagadnienie pracy zespołu deweloperskiego zgodnie z postulatami Metody Kanban. Aplikacja ma za zadanie stanowić symulację takiej pracy pod nadzorem wirtualnego przewodnika. Program powinien być w stanie zasymulować pracę zgodnie z tą metodą tak, aby podkreślić i zwizualizować korzyści płynące z jej stosowania oraz pozwolić użytkownikowi jej się nauczyć.

Podrozdział 1.3: Użytkownik docelowy.

Odbiorca aplikacji może być zarówno osobą zaawansowaną w przedstawianej tematyce jak i początkującą. Użytkownikiem tego programu powinna być osoba chcąca poznać praktyczną stronę pracy zespołu deweloperskiego zgodnie z postulatami metody Kanban poprzez doświadczenie symulacji z przykładowym scenariuszem. Bardziej doświadczony użytkownik może również ignorować wirtualnego nadzorcę, przedłużać maksymalny czas symulacji i zagrać na swój sposób.

Rozdział 2: OPIS FUNKCJONALNOŚCI.

Podrozdział 2.1: Uruchomienie programu.

Dystrybucje desktopowe należy uruchamiać klikając LPM na plik wykonywalny aplikacji „*Kanban Method's Simulator*”. W przypadku dystrybucji webowej po otwarciu linku widoczny będzie biały ekran, podczas którego należy poczekać aż aplikacja zostanie pobrana przez przeglądarkę internetową i uruchomiona w bieżącym oknie.

Podrozdział 2.2: Korzystanie z programu.

Korzystanie z programu nie powinno sprawiać problemu nawet w małym stopniu zaawansowanym użytkownikom komputera. Aplikacja jest przystosowana do szerokiego grona odbiorców w różnym stopniu zaawansowania, wieku, narodowościach, platformach i preferencjach.

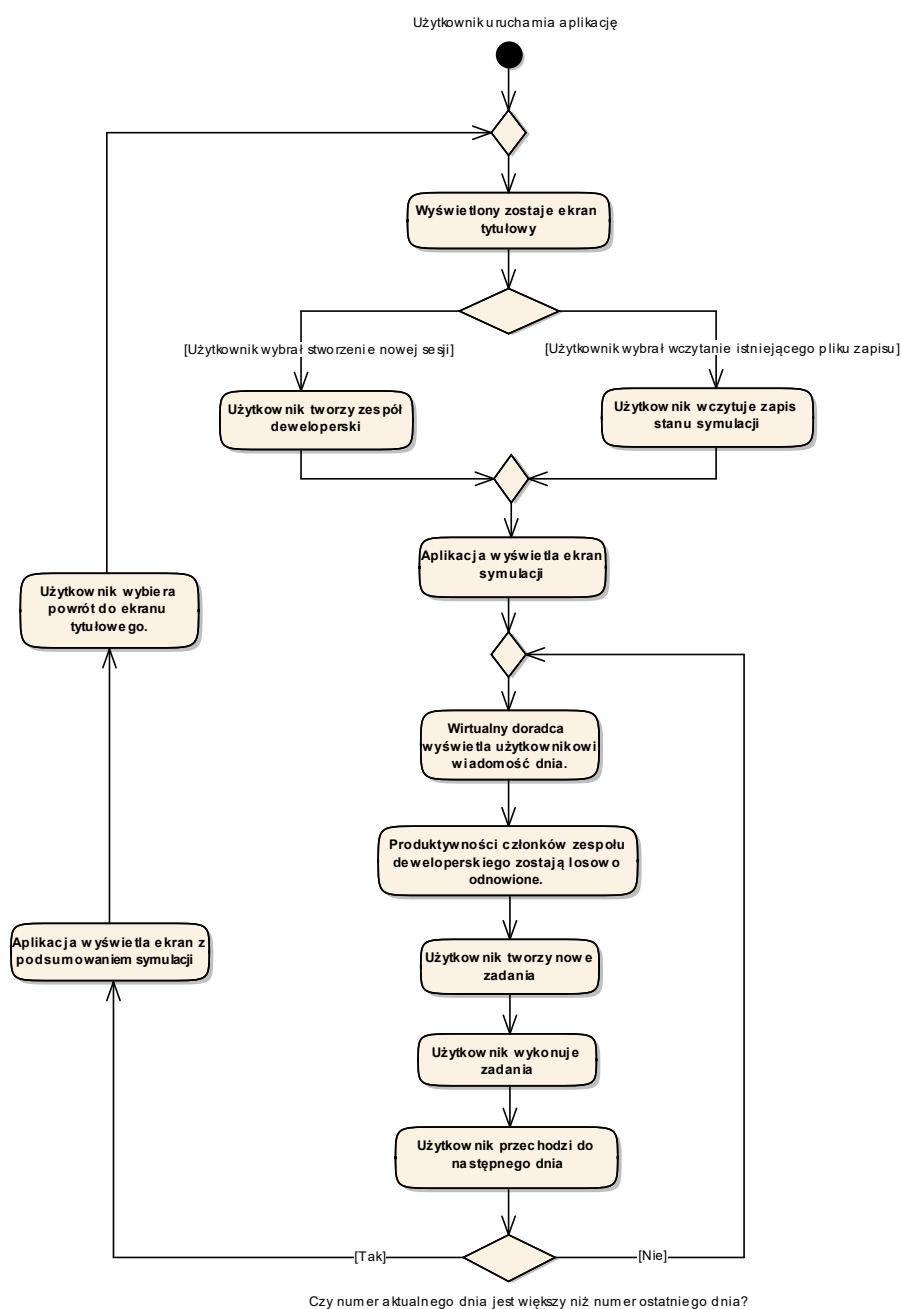
- **Aplikacja opatrzona jest w przyjemny dla oka i intuicyjny interfejs graficzny** wykorzystujący elementy Material Design oraz zaimplementowany został przełącznik pomiędzy ciemnym, a jasnym motywem interfejsu graficznego, w programie na pasku wyboru menu.
- **Aplikacja jest dostępna zarówno w języku polskim jak i angielskim.** Domyślny język w programie dostosowuje się do języka systemu operacyjnego użytkownika. Dla polskiego użytkownika wyświetlony zostanie język polski, a dla zagranicznego język angielski. Opcja zmiany języka jest także dostępna w programie na pasku wyboru menu.
- **Aplikacja dostępna jest dla większości obecnie najpopularniejszych systemów operacyjnych.** Wersja desktopowa dostępna jest dla systemów operacyjnych z rodziny Windows oraz Linux. System macOS oraz systemy mobilne (iOS, Android) NIE są wspierane. Użytkownicy niewspieranych systemów operacyjnych mogą skorzystać z wersji webowej aplikacji.
- **Aplikacja dostosowana jest do poziomu zaawansowania użytkownika w metodyce Kanban.** Domyślne ustawienia symulacji są wystarczające do prawidłowego jej przeprowadzenia bez potrzeby konfiguracji. Bardziej zaawansowani użytkownicy mogą na przykład ustawić limity zadań na poszczególne kolumny tablicy Kanban.

Rozdział 3: PRZYKŁADOWY SCENARIUSZ DZIAŁANIA PROGRAMU.

Podrozdział 3.1: Scenariusz ogólny.

1. Użytkownik uruchamia aplikację.
2. Użytkownik tworzy zespół deweloperski.
3. Użytkownik rozpoczyna sesję.
4. Użytkownik przechodzi przez kolejne dni symulacji.
5. Użytkownik kończy symulację.
6. Aplikacja wyświetla użytkownikowi podsumowanie jego sesji.

Podrozdział 3.2: Scenariusz szczegółowy.



Podrozdział 4.1: Pojęcia i obiekty dziedziny aplikacji (słownik dziedziny).

1. Task (in. zadanie, zadanie do wykonania):

- **TaskID** – numer identyfikacyjny zadania. Musi być unikatowy.
- **Title** – tytuł bądź nazwa danego zadania, wyświetlana na tablicy Kanban.
- **Type** – parametr wskazujący na typ danego zadania (patrz TaskType).
- **StartDay** – numer dnia, w którym zadanie zostało rozpoczęte (opuściło pierwszą kolumnę).
- **EndDay** – numer dnia, w którym zadanie zostało zakończone (dotarło do ostatniej kolumny).
- **Stage** – numer etapu, na którym przebywa zadanie (0 – pierwsza kolumna, 1 – etap pierwszy, 2 – etap drugi, 3 – zakończone).
- **DeadlineDay** – parametr zadań typu FixedDate, określający numer dnia, w którym upływa termin wykonania zadania.
- **Owner** – referencja do obiektu właściciela zadania (patrz Task Owner).
- **ProductivityRequiredToUnlock** – ilość produktywności jaką potrzeba zużyć, aby odblokować zadanie po losowym zablokowaniu.
- **Progress** – obiekt przechowujący stopień ukończenia zadania w danym etapie, zaopatrzony w szereg metod oraz listę zainwestowanych produktywności przechowujących referencje do użytkowników.
- **LatestTaskID** – statyczny parametr przechowujący numer ID ostatnio utworzonego zadania. Jest ważny przy określaniu numeru następnego zadania. Należy do zaktualizować po wczytaniu pliku z symulacją.

2. User (in. członek zespołu deweloperskiego, „team member”):

- **ID** – Numer identyfikacyjny. Musi być unikalny.
- **Name** – Pseudonim użytkownika.
- **MaxProductivity** – Maksymalna ilość punktów produktywności, jaką użytkownik może jednocześnie posiadać.
- **Color** – obiekt koloru będącego znakiem rozpoznawczym użytkownika.
- **Productivity** – Aktualna ilość przechowywanej produktywności.

3. TaskType (in. typ zadania):

- **FixedDate** (in. Z Ustaloną datą „Fixed Date”).
- **Expedite** (in. Pilne, „Expedite”).
- **Standard** (in. Standardowe, „Standard”).

4. Productivity (in. produktywność):

- Ilość punktów, jaką użytkownik przechowuje i którą może zainwestować w wybrane zadanie, aby zwiększyć jego pasek postępu w danej fazie lub odblokować po losowym zablokowaniu zadania w trakcie pracy. Zapełnienie paska produktywności do pełna oznacza możliwość przesunięcia go w dalsze etapy. Produktywność jest odnawiana w losowym stopniu każdego nowego dnia symulacji.

5. AllTasksContainer (in. pojemnik ze wszystkimi listami zadań):

- **IdleTasksColumn** (in. kolumna dostępnych zadań, „available tasks column”) – lista zadań w kolumnie z dostępnymi zadaniami. Kolumna pierwsza.
- **StageOneInProgressTasksColumn** (in. kolumna zadań „w trakcie” etapu pierwszego) – lista zadań w kolumnie etapu pierwszego w „w trakcie”.
- **StageOneDoneTasksColumn** (in. kolumna zadań „wykonanych” etapu pierwszego) – lista zadań w kolumnie etapu pierwszego w „wykonane”.
- **StageTwoTasksColumn** (in. kolumna zadań etapu drugiego) – lista zadań w kolumnie etapu drugiego.
- **FinishedTasksColumn** (in. kolumna zadań ukończonych) – lista ukończonych zadań. Kolumna ostateczna.

6. SimState (in. stan symulacji, bieżący stan symulacji, „simulation state”):

- Obiekt symbolizujący bieżący stan symulacji. Wykorzystywany w plikach zapisu stanu symulacji.
- **Users** – lista użytkowników w trwającej symulacji.

- **AllTasks** – obiekt typu AllTasksContainer przechowujący stan wszystkich kolumn w trwającej symulacji.
 - **CurrentDay** – numer aktualnego dnia w trwającej symulacji.
 - **StageOneInProgressColumnLimit** – narzucony limit zadań w kolumnie „w trakcie” etapu pierwszego w trwającej symulacji.
 - **StageOneDoneColumnLimit** – narzucony limit zadań w kolumnie wykonanych etapu pierwszego w trwającej symulacji.
 - **StageTwoColumnLimit** – narzucony limit zadań w kolumnie etapu drugiego w trwającej symulacji.
7. **SimEngine (in. silnik symulacji, „simulation engine”):**
- Obiekt zawierający zbiór metod odpowiedzialnych za działanie symulacji w obszarze weryfikacji działań użytkownika, reagowania na nie oraz poprawnym przebiegu danych w programie.
8. **Stage One/Two (in. etap pierwszy/etap drugi):**
- Nazwy kolumn przedstawiających dany etap wykonywania zadania. Kolumna etapu pierwszego jest podwójna i składa się z dwóch podkolumn „w trakcie” oraz „wykonane”.
9. **Blockade:**
- Blokada zadania pojawiająca się w sposób losowy dla zadań znajdujących się w kolumnach roboczych. Szansa na zablokowanie danego zadania po przejściu do następnego dnia określana jest przez odpowiedni atrybut w klasie *StoryModule*. Zablokowanie zadania znacząco graniczy interakcje użytkownika z nim. Zadanie nie może być zasilone produktywnością bądź przeniesione do następnej kolumny dopóki nie zostanie odblokowane kosztem pewnej ilości produktywności, która również jest określana w klasie *StoryModule*.
10. **Savefile:**
- Plik z rozszerzeniem typu *ksim* przechowujący strukturę znaków określającą stan trwającej symulacji w czasie utworzenia pliku. Użytkownik może tworzyć takie pliki, a następnie wczytywać je, aby powracać do trwającej w przeszłości symulacji.

Podrozdział 4.2: Pliki zapisu stanu symulacji.

Za zapisywanie oraz odczytywanie plików zapisu stanu symulacji są odpowiedzialne klasy w pakiecie „*savefile_parsers*”.

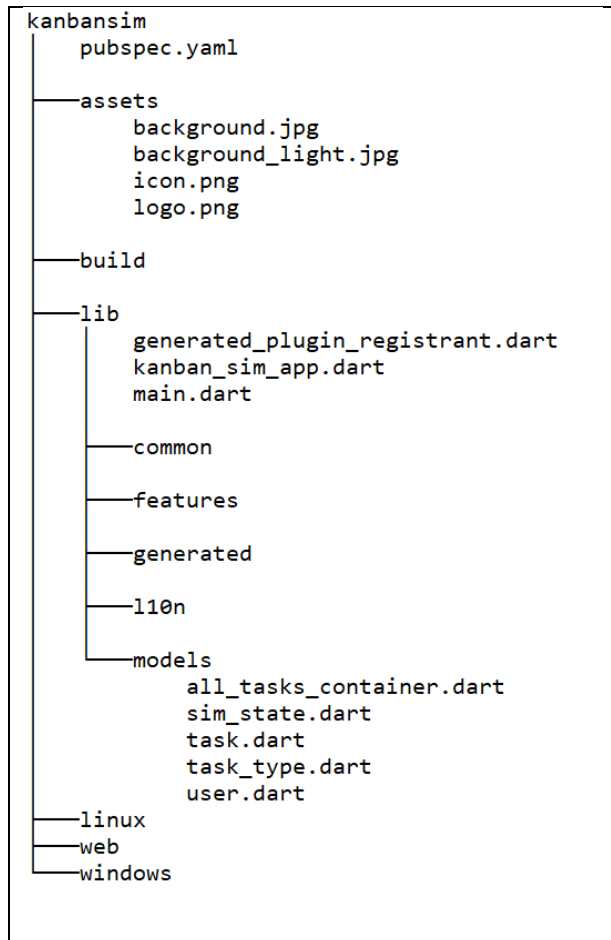
- TaskProgress przechowuje numery identyfikacyjne użytkownika, który dany punkt produktywności zainwestował. Wartość ‘-1’ oznacza, że punkt jest pusty i nikt jeszcze go nie wypełnił.
- Color jest przechowywany w wartościach RGB.
- Wartość END_DAY ‘-1’ oznacza, że zadanie nie jest ukończone.
- Wartość START_DAY ‘-1’ oznacza, że zadanie nie zostało jeszcze rozpoczęte (nie wyszło z kolumny dostępnych zadań).
- Wartość DEADLINE ‘-1’ oznacza, że zadanie nie jest typu FixedDate i nie posiada dnia deadline’u.
- Wartości ‘-1’ w parametrach limitów kolumn w sekcji SIM_STATE_PARAMS oznaczają brak narzuconych limitów (nieskończoność).

```
[BEGIN_SIM_STATE_PARAMS]
[CURRENT_DAY] 1
[LATEST_TASK_ID] 0
[STAGE_ONE_IN_PROGRESS_COLUMN_LIMIT] -1
[STAGE_ONE_DONE_COLUMN_LIMIT] -1
[STAGE_TWO_COLUMN_LIMIT] -1
[END_SIM_STATE_PARAMS]

[BEGIN_USERS_LIST]
[BEGIN_USER]
[ID] 0
[NAME] {Bhutan}
[COLOR] {0,188,212}
[Max_PRODUCTIVITY] 5
[PRODUCTIVITY] 0
[END_USER]
[BEGIN_USER]
[ID] 1
[NAME] {Lebanon}
[COLOR] {255,152,0}
[Max_PRODUCTIVITY] 5
[PRODUCTIVITY] 0
[END_USER]
[BEGIN_USER]
[ID] 2
[NAME] {Dallas}
[COLOR] {0,150,130}
[Max_PRODUCTIVITY] 5
[PRODUCTIVITY] 0
[END_USER]
[END_USERS_LIST]

[BEGIN_TASK_LISTS]
[BEGIN_IDLE_TASKS_COLUMN]
[END_IDLE_TASKS_COLUMN]
[BEGIN_STAGE_ONE_IN_PROGRESS_TASKS_COLUMN]
[BEGIN_TASK]
[TITLE] {Task #2}
[TASKTYPE] TaskType-Expedite
[TASK_ID] 2
[OWNER_ID] 1
[PROD_REQ_TO_UNLOCK] 0
[PROD_REQ_TO_COMPLETE] 5
[DEADLINE] -1
[START_DAY] 1
[END_DAY] -1
[CURRENT_STAGE] 1
[BEGIN_TASKPROGRESS]
[PART] 1
[PART] 1
[PART] -1
[PART] -1
[PART] -1
[END_TASKPROGRESS]
[END_TASK]
[BEGIN_TASK]
[TITLE] {Task #1}
[TASKTYPE] TaskType-Standard
[TASK_ID] 1
[OWNER_ID] 2
[PROD_REQ_TO_UNLOCK] 0
[PROD_REQ_TO_COMPLETE] 5
[DEADLINE] 10
[START_DAY] 1
[END_DAY] -1
[CURRENT_STAGE] 1
[BEGIN_TASKPROGRESS]
[PART] 2
[PART] -1
[PART] -1
[PART] -1
[PART] -1
[END_TASKPROGRESS]
[END_TASK]
[BEGIN_TASK]
[TITLE] {Task #3}
[TASKTYPE] TaskType-FixedDate
[TASK_ID] 3
[OWNER_ID] -1
[PROD_REQ_TO_UNLOCK] 0
[PROD_REQ_TO_COMPLETE] 5
[DEADLINE] 10
[START_DAY] 1
[END_DAY] -1
[CURRENT_STAGE] 1
[BEGIN_TASKPROGRESS]
[PART] -1
[PART] -1
[PART] -1
[PART] -1
[PART] -1
[END_TASKPROGRESS]
[END_TASK]
[END_STAGE_ONE_IN_PROGRESS_TASKS_COLUMN]
[END_STAGE_ONE_DONE_TASKS_COLUMN]
[END_STAGE_TWO_TASKS_COLUMN]
[END_STAGE_THO_TASKS_COLUMN]
[END_FINISHED_TASKS_COLUMN]
[END_TASK_LISTS]
```

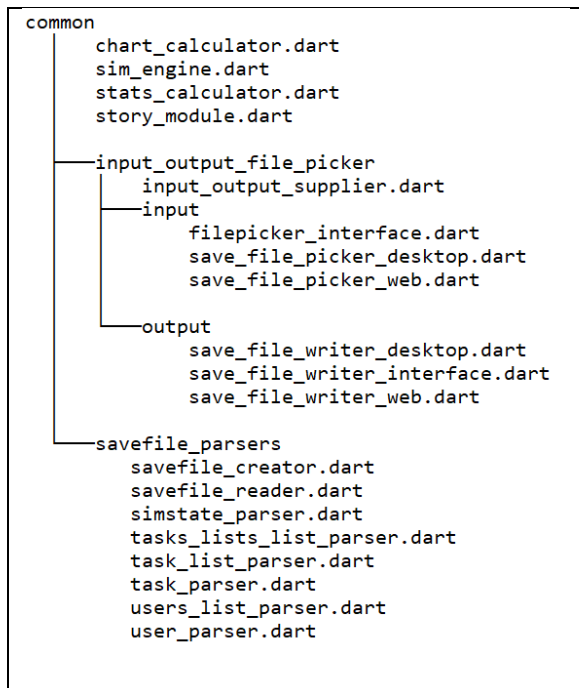
Podrozdział 4.1: Główny katalog projektu.



- **pubspec.yaml** – plik konfiguracyjny określający zasoby, z których korzysta projekt. To tutaj ustala się wersję silnika Fluttera oraz Dart, z którego korzysta projekt. Tutaj także umieszcza się odniesienia do publicznych pakietów (tutaj „dependencies”), które mają zostać użyte w projekcie.
- **assets** – katalog zawierający materiały graficzne wykorzystywane w projekcie. Znajdują się tu między innymi obrazy tła, ikony aplikacji oraz logo widoczne na ekranie tytułowym.
- **build** – Folder zawierający automatycznie zbudowane dystrybucje przez silnik Flutter. Po wygenerowaniu dystrybucji (np. *flutter build windows*) jego katalog pojawi się wewnątrz tego katalogu. Podobna sytuacja występuje w przypadku wersji typu debug.
- **linux** – katalog zawierający materiały oraz silnik, z którego budowana jest wersja projektu na platformę Linux. Można go konfigurować celem edycji bezpośrednio dystrybucji projektu na tę platformę.
- **web** – katalog zawierający materiały oraz silnik, z którego budowana jest wersja projektu na platformę webową. Można go konfigurować celem edycji dystrybucji projektu bezpośrednio na tę platformę.
- **windows** – katalog zawierający materiały oraz silnik, z którego budowana jest wersja projektu na platformę Windows. Można go konfigurować celem edycji dystrybucji projektu bezpośrednio na tę platformę.

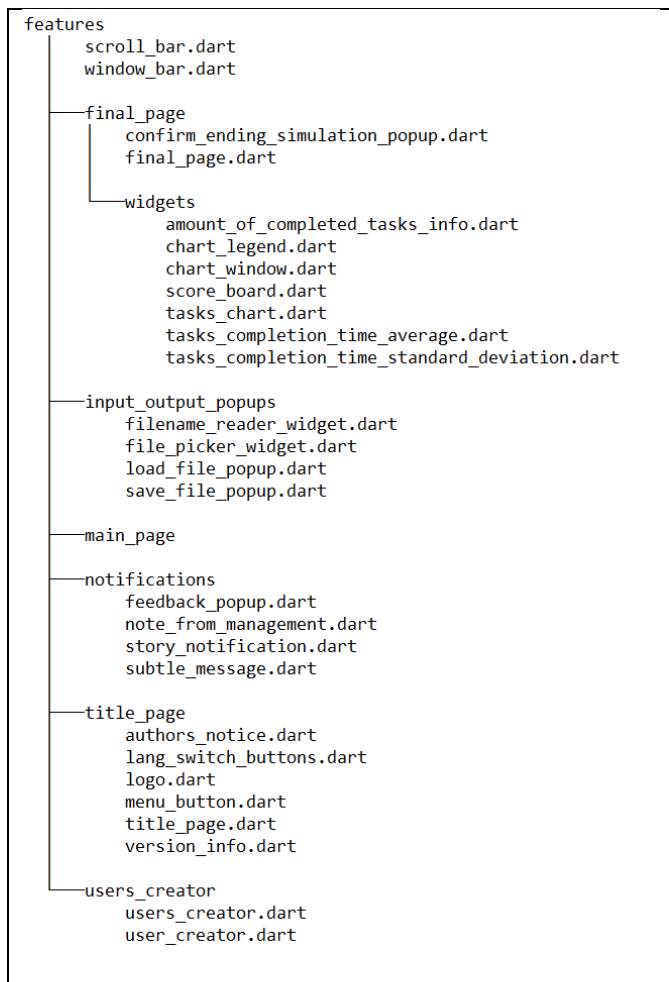
- **lib** – Katalog przechowujący główny kod projektu, czyli wszystkie klasy logiczne oraz klasy elementów interfejsu graficznego.
 - **main.dart** – początkowa instrukcja, uruchamiająca aplikację oraz dostosowująca właściwości jej okna.
 - **kanban_sim_app.dart** – główna klasa aplikacji zawierająca najważniejsze i najogólniejsze parametry konfiguracyjne, do których można odnosić się w pozostałej części kodu. Uruchamia domyślnie ekran tytułowy.
 - **common** – katalog zawierający klasy ściśle warstwy back-endowej. Opisany dalej.
 - **features** – katalog zawierający klasy warstwy front-endowej oraz mid-endowej.
 - **generated** – katalog automatycznie generowany, przechowujący pliki konfiguracyjne pakietu odpowiadającego za internacjonalizację aplikacji pod kątem językowym.
 - **l10n** – katalog plików zawierających tłumaczenia polskie oraz angielskie, część pakietu odpowiadającego za internacjonalizację aplikacji pod kątem językowym.
 - **models** – katalog zawierający klasy stanowiące logikę dziedziny aplikacji.
 - **all_tasks_container.dart** – Klasa-kontener zawierająca stan list zadań we wszystkich kolumnach tablicy Kanban w czasie bieżącej symulacji.
 - **sim_state.dart** – Klasa zawierająca informacje na temat stanu bieżącej symulacji. Oprócz bieżącego przechowywania wartości symulacji służy także do tworzenia plików zapisu.
 - **task.dart** – Klasa reprezentująca zadanie w ramach pracy zespołu deweloperskiego.
 - **task_type.dart** – Klasa typu wyliczeniowego reprezentująca typy zadań (task.dart).
 - **user.dart** – Klasa reprezentująca członka zespołu deweloperskiego, których użytkownik kreuje podczas tworzenia nowej sesji symulacji.

Podrozdział 4.2: Podkatalog „common”.



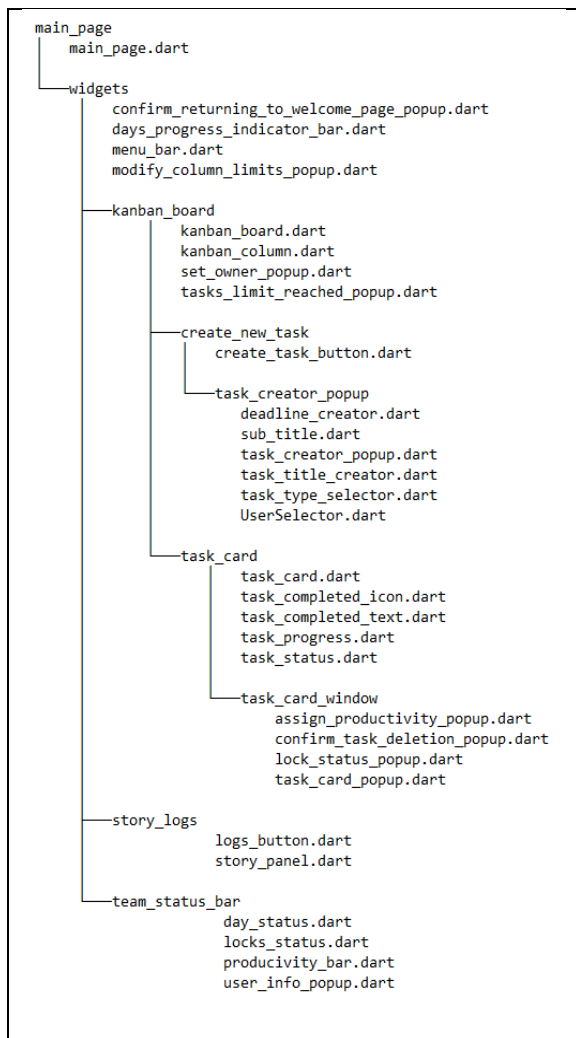
- **chart_calculator.dart** – moduł zawierający zestaw metod przydatnych przy konstrukcji finalnego wykresu, podsumowującego zależność ilości wykonanych zadań od ich czasu wykonania.
- **sim_engine.dart** – klasa zawierająca najważniejszych dla czystej mechaniki symulacji metod. Jej metody są wykorzystywane przez kluczowe action listenery, które podają im uzyskane parametry.
- **story_module.dart** – klasa odpowiedzialna za tryb historii w symulacji. To tutaj konfiguruje się ilość maksymalną dni, wiadomości od wirtualnego przewodnika oraz zasady rozgrywki takie jak szanse na blokowania się zadań czy ilość przywracanych produktywności każdego dnia. Klasa wyświetla także komunikaty, powiadomienia oraz wypisuje logi.
- **input_output_file_picker** – pakiet zawierający zestaw klas dla obsługi wejścia/wyjścia dla plików zapisu dla dystrybucji webowych oraz desktopowych, które muszą mieć oddzielne systemy ze względu na swoją odmienność.
 - **input_output_supplier.dart** – klasa zaopatrzeniowa, zawierająca zestaw wspólnych metod wykorzystywanych zarówno przez moduły wejścia i wyjścia dla obu dystrybucji platformowych.
- **input** – pakiet zawierający wszelkie klasy obsługi wejścia plików zapisu.
 - **filepicker_interface.dart** – interfejs implementowany przez klasy obsługi wejścia plików zapisu. Definiuje metody, które powinny być wspólne niezależnie od obsługiwanej platformy.
 - **save_file_picker_desktop.dart** – klasa odpowiedzialna za dostarczenie użytkownikowi interfejsu do wybrania pliku zapisu dla wersji desktopowej aplikacji.
 - **save_file_picker_web.dart** – klasa odpowiedzialna za dostarczenie użytkownikowi interfejsu do wybrania pliku zapisu dla wersji webowej aplikacji.
- **output** – pakiet zawierający wszelkie klasy obsługi wyjścia plików zapisu.
 - **save_file_writer_interface.dart** – interfejs implementowane przez klasy obsługi wyjścia plików zapisu. Definiuje metody, które powinny być wspólne niezależnie od obsługiwanej platformy.
 - **save_file_writer_desktop.dart** – klasa odpowiedzialna za stworzenie, zapisanie pliku zapisu na dysku użytkownika w katalogu saves aplikacji dla wersji desktopowych aplikacji.
 - **save_file_writer_web.dart** – klasa odpowiedzialna za stworzenie pliku zapisu i pobranie go na komputer użytkownika dla wersji webowych aplikacji.
- **savefile_parsers** – pakiet zawierający zestaw klas odpowiedzialnych tworzenie i odczytywanie plików zapisu, przetłumaczenie aktualnego stanu symulacji na typ danych *String* i z powrotem.
 - **savefile_creator.dart** – główna klasa zrzeszająca pozostałe klasy w funkcji tworzenia całego pliku zapisu.
 - **savefile_reader.dart** – główna klasa zrzeszająca pozostałe klasy w funkcji odczytywanie całego pliku zapisu i odtwarzania z niego obiektów ważnych dla symulacji.
 - **user_parser.dart** - klasa odpowiedzialna za przetłumaczenie na *String* obiektu typu User oraz z powrotem.
 - **task_list_parser.dart** – klasa odpowiedzialna za przetłumaczenie listy zadań (z kolumny) na *String* oraz z powrotem.
 - **user_list_parser.dart** – klasa odpowiedzialna za przetłumaczenie listy członków zespołu deweloperskiego na *String* oraz z powrotem.
 - **task_lists_list_parser.dart** – klasa odpowiedzialna za przetłumaczenia listy list zadań (kolumn), czyli całej tablicy kanbanowej (np. AllTaskContainer) na typ *String* oraz z powrotem.

Podrozdział 4.3: Podkatalog „features”.



- **scroll_bar.dart** – klasa widgetu suwaka do tablicy Kanban.
- **window_bar.dart** – klasa paska sterowania oknem (tylko dystrybucje desktopowe).
- **input_output_popups** – pakiet zawierający popupy wczytywania i zapisywania stanu symulacji (pliki zapisu).
 - **filename_reader_widget.dart** – klasa widgetu odpowiadającego za możliwość wpisania nazwy pliku zapisu, który użytkownik chce utworzyć.
 - **file_picker_widget.dart** – klasa widgetu odpowiadającego za możliwość otworzenia przeglądarki plików, a następnie wyświetlenia nazwy/ścieżki do pliku zależnie od dystrybucji desktop/web.
 - **load_file_popup.dart** – klasa okna typu popup od wczytywania pliku zapisu.
 - **save_file_popup.dart** – klasa okna typu popup od zapisywania pliku zapisu.
- **main_page** – pakiet zawierający klasy i podpakiety odpowiedzialne za stan strony głównej, wyświetlanej w czasie symulacji.
- **notifications** – pakiet zawierający klasy powiadomień oraz komunikatów wyświetlanych użytkownikowi w czasie symulacji.
 - **feedback_popup.dart** – klasa prostego, pustego popup'u przyjmująca jego zawartość.
 - **note_from_management.dart** – klasa okna, w którym pojawiają się codzienne wiadomości od wirtualnego przewodnika.
 - **Story_notification.dart** – klasa powiadomienia typu Card pojawiającego się na górze ekranu.
- **subtle_message.dart** – klasa powiadomień typu Snackbar.
- **users_creator** – pakiet zawierający okna tworzenia zespołu deweloperskiego.
 - **users_creator.dart** – klasa okna tworzenia zespołu deweloperskiego.
 - **user_creator.dart** – klasa okna tworzenia pojedynczego członka zespołu deweloperskiego.
- **title_page** – pakiet zawierający klasy składające się na okno tytułowe aplikacji.
 - **authors_notice.dart** – klasa widgetu zawierającego informacje o autorze aplikacji.
 - **lang_switch_buttons.dart** – klasa widgetu przycisków zmiany języka na stronie tytułowej.
 - **logo.dart** – klasa widgetu wyświetlająca logo aplikacji.
 - **menu_button.dart** – klasa przycisku w menu głównym, przyjmująca tekst oraz akcję po wciśnięciu.
 - **title_page.dart** – główna klasa strony tytułowej, zrzeszająca wszystkie widgety.
 - **version_info.dart** – klasa widgetu wyświetlającego informacje o wersji aplikacji oraz dystrybucji web/desktop.
- **final_page** – pakiet zawierający widgety składające się na finałowe okno z podsumowaniem statystyk symulacji.
 - **confirm_ending_simulation_popup.dart** – klasa okienka typu popup wyskakującego przed zakończeniem symulacji i przejściem do ekranu końcowego.
 - **final_page.dart** – klasa główna okna końcowego, na którym wyświetlone są statystyki i podsumowanie symulacji.
 - **widgets** – pakiet widgetów składających się na końcowe okno ze statystykami.
 - **amount_of_completed_tasks_info.dart** – klasa widgetu wyświetlającego ilość ukończonych zadań.
 - **chart_legend.dart** – klasa legendy wykresu.
 - **chart_window.dart** – klasa widgetu-okna zawierającego wykres.
 - **score_board.dart** – klasa widget pokazującego ogólne statystyki z symulacji.
 - **tasks_chart.dart** – klasa widgetu wykresu zależności ilości zadań od czasów ich ukończenia.
 - **tasks_completion_time_average.dart** – klasa widgetu wyświetlającego średni czas ukończenia zadania.
 - **tasks_completion_time_standard_deviation.dart** – klasa widgetu wyświetlającego odchylenie standardowe od czasu ukończenia zadania.

Podrozdział 4.3: Podkatalog „main_page”.



- **main_page.dart** – klasa główna okna symulacji. Przechowuje obiekty logiki oraz dziedziny aplikacji typu SimState, SimEngine oraz StoryModule, do których odnoszą się jej podwidgety.
- **widgets** – pakiet zawierający widgety składające się na stronę główną aplikacji, wyświetlanej w czasie symulacji.
 - **confirm_returning_to_welcome_page_popup.dart** – klasa popupu wymagającego potwierdzenia w razie, gdyby użytkownik wyraził chęć powrotu do strony tytułowej.
 - **days_progress_indicator_bar.dart** – klasa paska postępu wizualizującego stosunek przebytych dni symulacji do całości.
 - **menu_bar.dart** – widget paska narzędzi, zawierający także pasek kontroli okna.
 - **modify_column_limits_popup.dart** – klasa popupu pozwalającego ustalić limity zadań w kolumnach roboczych tablicy Kanban.
- **story_logs** – pakiet zawierający element odpowiedzialny za wyświetlanie logów symulacji.
 - **logs_button.dart** – klasa widgetu typu FloatingActionButton przycisku, który wyzwala okno z listą logów.
 - **story_panel.dart** – klasa popupu, która wyświetla listę logów z bieżącej sesji symulacji.
- **team_status_bar** – pakiet zawierający klasy widжетów składających się na pasek statusów drużyny deweloperskiej oraz symulacji.
 - **day_status.dart** – klasa widgetu wyświetlającego numer aktualnego dnia symulacji i pozwalającego na przełączanie się pomiędzy dniami.
 - **locks_status.dart** – klasa widgetu wyświetlającego informację, czy na tablicy są jakieś zablokowane zadania.
 - **productivity_bar.dart** – klasa widgetu wyświetlającego listę członków zespołu deweloperskiego i aktualny stan ich punktów produktywności.
 - **user_info_popup.dart** – Prosty popup wyświetlający informacje o członku zespołu deweloperskiego i jego aktualny stan punktów produktywności.

- **kanban_board** – pakiet zawierający klasy widжетów składających się na wygląd i funkcjonowanie podelementów tablicy Kanban.
 - **kanban_board.dart** – klasa główna widgetu tablicy Kanban, określająca zachowania (np. wymagania na przyjęcie zadania) swoich kolumn oraz zaopatrująca swoje podwidgety w obiekty stanu symulacji.
 - **kanban_column.dart** – klasa widgetu kolumny Kanban, przyjmująca swoją zawartość oraz akcje na poszczególne zachowania użytkownika.
 - **set_owner_popup.dart** – klasa popupu wyświetlanego, gdy użytkownik przesunie nierozpoczęte w danym etapie zadanie do kolumny roboczej, w którym użytkownik wskazuje głównego właściciela zadania.
 - **tasks_limit_reached_popup.dart** – klasa popupu wyświetlanego, gdy użytkownik próbuje przerzucić do kolumny nowe zadanie, a aktualna ilość zadań w danej kolumnie jest równa narzuconemu limitowi.
 - **create_new_task** – pakiet zawierający składowe elementy biorące udział w procesie tworzenia nowego zadania.
 - **create_task_button.dart** – widget przycisku wywołującego okno tworzenia zadania.
 - **task_creator_popup** – pakiet zawierający widgety składające się na popup stworzenia nowego zadania.
 - **task_card** – pakiet zawierający widgety składające się na wyświetlaną na tablicy Kanban kartkę z opisem przypisanego do niej zadania.
 - **task_card.dart** – główna klasa widgetu reprezentującego zadania na tablicy Kanban.
 - **task_card_window** – pakiet zawierający elementy okna wyskakującego po kliknięciu na kartkę zadania na tablicy Kanban. Wyświetla informacje o zadaniu oraz umożliwia interakcję z nim.
 - ❖ **assign_productivity_popup.dart** – klasa popupu umożliwiającego przypisanie produktywności od wybranego członka zespołu deweloperskiego.
 - ❖ **confirm_task_deletion_popup.dart** – klasa popupu wymagającego potwierdzenia chęci usunięcia danego zadania.
 - ❖ **lock_status_popup.dart** – klasa popupu, w którym można odblokować zablokowane zadanie.
 - ❖ **task_card_popup.dart** – klasa główna wyświetlanego popupu z informacjami oraz dostępnymi akcjami zadania po kliknięciu na nie na tablicy Kanban.

Rozdział 6: ŹRÓDŁA I UŻYTE PAKIETY.

- **overlay_support** (yangbinyhbn) https://pub.dev/packages/overlay_support
- **vs_scrollbar** (vickysalunkhe) https://pub.dev/packages/vs_scrollbar
- **pluto_menu_bar** (weblaze) https://pub.dev/packages/pluto_menu_bar
- **filesystem_picker** (chentsov) https://pub.dev/packages/filesystem_picker
- **google_fonts** (material.io) https://pub.dev/packages/google_fonts
- **file_picker_cross** (testapp.schule) https://pub.dev/packages/file_picker_cross
- **intl** (official dart.dev) <https://pub.dev/packages/intl>
- **universal_html** (dint.dev) https://pub.dev/packages/universal_html
- **bitsdojo_window** (bitsdojo.com) https://pub.dev/packages/bitsdojo_window
- **syncfusion_flutter_sliders** (syncfusion.com) https://pub.dev/packages/syncfusion_flutter_sliders
- **flutter_material_color_picker** (jeancharles) https://pub.dev/packages/flutter_material_color_picker
- **cupertino_icons** (official flutter.dev) https://pub.dev/packages/cupertino_icons
- **animated_background** (andreabaltazar.com) https://pub.dev/packages/animated_background
- **fl_chart** (ikhoshabi.com) https://pub.dev/packages/fl_chart