

Projekt zaliczeniowy – Timesheet

Dokumentacja aplikacji do zarządzania zadaniami

Lider zespołu:

Jagoda Wykusz

Pozostali członkowie:

Ryszard Białas, Maksym Bielec, Marcin Słota, Oskar Ziomek

Uniwersytet Śląski

2017

Spis treści

1	Założenia projektowe	3
1.1	Przeznaczenie aplikacji	3
1.2	Elementy składowe	3
2	Specyfikacja techniczna	5
2.1	Język programowania i frameworki	5
2.1.1	PHP oraz Laravel	5
2.1.2	Bootstrap	5
2.2	Baza danych	6
2.3	Repozytorium kodu	6
2.4	Zarządzanie pracą w zespole	7
2.5	Uruchomienie aplikacji	7
3	Realizacja	9
3.1	Zasada tworzenia widoków	9
3.2	Wybrane widoki	10
3.2.1	Główny ekran aplikacji	10
3.2.2	Tworzenie nowego projektu	10
3.2.3	Coś jeszcze (?)	10
3.3	Mechanizm logowania użytkownika	10
3.4	Mechanizm rejestracji użytkownika	10
4	Testowanie	10

4.1	Metody testowania	10
4.2	Jakieś wnioski?	10
5	Dalszy rozwój aplikacji	10
5.1	Założenia, które nie zostały zrealizowane	10
5.2	Możliwość rozbudowania aplikacji w przyszłości	10

1 Założenia projektowe

W tym rozdziale opisane zostaną założenia, które przyświecały zespołowi podczas tworzenia aplikacji. Przedstawiony zostanie zarówno cel stworzenia programu, jak i funkcjonalności, które będą w nim docelowo dostępne.

1.1 Przeznaczenie aplikacji

Celem niniejszego projektu jest stworzenie aplikacji webowej, pozwalającej na zarządzanie zadaniami podczas pracy w zespole. Umożliwiać ona będzie tworzenie wielu projektów, a w ich ramach zadań, do realizacji których przydzielać będzie można konkretnych użytkowników. Każdy projekt posiadać będzie menadżera, który ma większe uprawnienia, niż pozostali członkowie zespołu. Administrator aplikacji posiada maksymalne uprawnienia i może zarządzać również menadżerami.

Jednym z przykładów innych aplikacji podobnego rodzaju (choć w tym kontekście z nieco mniejszymi możliwościami), które są dostępne już na rynku, może być serwis Trello: <https://trello.com>.

1.2 Elementy składowe

Zgodnie z przedstawionymi wymaganiami, tworzona aplikacja składać się będzie z następujących elementów (cech):

- Ogólne:
 - Witryna internetowa,

- Należy zastosować ORM lub Micro-ORM,
 - Dane powinny być walidowane po stronie front i back endu,
 - Mechanizm logowania błędów,
 - Projektu musi się znajdować na repozytorium,
 - Dane muszą być zapisywane i odczytywane z bazy danych,
 - Projekt będzie oceniany również ze względu na estetykę,
 - Projekt musi posiadać czytelną dokumentację,
- Aplikacja:
 - Rejestrację nowego użytkownika wraz z podaniem danych osobowych,
 - Mechanizm "Przypominania hasła",
 - W systemie muszą występować 3 role - Administrator, Manager i Pracownik,
 - Moduł przesyłania wiadomości pomiędzy użytkownikami systemu,
 - We wszystkich tabelach w systemie powinna być możliwość filtracji danych,
 - Należy zaimplementować mechanizm aprobowania wpisów.
 - Pracownik:
 - Możliwość dodawania i edycji wpisów za danych dzień pracy. Wpis musi zawierać: datę, czas pracy, projekt, zadanie, komentarz,
 - Wyświetlanie listy wszystkich wpisów (wraz z możliwością ich filtracji).
 - Manager:

- Wyświetlanie, dodawanie, edycja i usuwanie zadań (zadania powinny posiadać przewidywaną estymację),
 - Wyświetlanie wszystkich projektów, które są prowadzone przez danego managera,
 - Wyświetlanie listy pracowników w danym projekcie,
 - Generowanie raportu z przebiegu prac w projekcie,
 - Aprobowanie wpisów pracowników.
- Administrator:
 - Wyświetlanie listy użytkowników, zadań i projektów,
 - Możliwość zablokowania użytkownika lub wymuszenie zmiany hasła,
 - Dodawanie, edycja i usuwanie projektów,
 - Przypisywanie managerów i pracowników do projektów.

2 Specyfikacja techniczna

W tym rozdziale opisane zostaną techniczne aspekty wykonania projektu, w tym zastosowane języki programowania, biblioteki oraz technologie. Zostanie również wyjaśniona kwestia zarządzania pracą w zespole, opisane repozytorium kodu oraz struktura bazy danych.

2.1 Język programowania i frameworki

2.1.1 PHP oraz Laravel

Aplikacja stworzona została w oparciu o skryptowy, interpretowany o język PHP. Od początku stosowano także framework Laravel w wersji 5.4 rozszerzający możliwości języka między innymi o modułowość. Framework ten rozwijany jest w ramach licencji MIT. Jego kod źródłowy pozostaje dostępny w serwisie GitHub, pod adresem: <https://github.com/laravel>. Laravel uznawany jest za jeden z najpopularniejszych frameworków języka PHP. Szacuje się, iż wykorzystuje go obecnie kilkadziesiąt milionów aplikacji webowych.

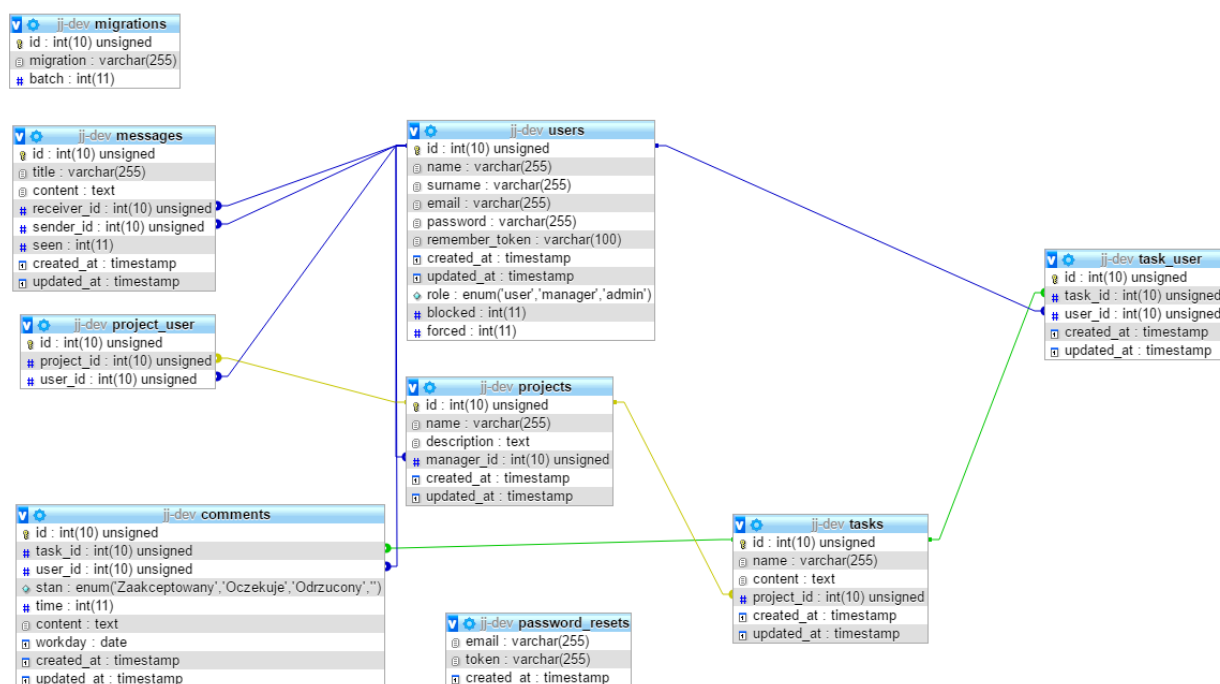
2.1.2 Bootstrap

Do stworzenia widoków w aplikacji wykorzystano framework Bootstrap. Gwarantuje on elementy zachowujące się zgodnie z ideologią RWD (Responsive Web Design) – stron internetowych zachowujących responsywność w zależności od rozdzielczości ekranu, na którym wyświetlana jest witryna. Framework ten składa się z odpowiednio skonfigurowanych wstępnie plików CSS oraz JS, których dołączenie do projektu, a później tworzenie elementów interfejsu w oparciu o wspólne klasy, gwarantuje odpowiednie zachowanie strony.

W naszym projekcie elementy Bootstrapa zaimportowane zostały do projektu bezpośrednio z wzorca *AdminLTE Control Panel Template* dostępnego w Internecie pod adresem: <https://adminlte.io>.

2.2 Baza danych

Na poniższym schemacie przedstawiono strukturę bazy danych wykorzystanej w projekcie:



2.3 Repozytorium kodu

Aby zapewnić najwyższą wygodę pracy w zespole oraz bezpieczeństwo tworzonego kodu aplikacji, pliki projektu umieszczone zostały w repozytorium w serwisie GitHub, pod adresem: <https://github.com/Rys922/Timesheet>. Znajduje się tam najnowsza wersja kodu, która na bieżąco aktualizowana jest przez członków zespołu, wraz z postępem w pracach.

Wykorzystanie rozproszonego systemu kontroli wersji, jakim jest Git, gwarantuje dostępność kodu dla wszystkich członków zespołu i umożliwia jednoczesną pracę na plikach (aktualna wersja kodu zapisana jest zarówno lokalnie na komputerze programisty, jak i na głównym serwerze usługi).

2.4 Zarządzanie pracą w zespole

Do rozdzielania prac pomiędzy członków zespołu tworzącego niniejszy projekt, wykorzystano serwis *Trello* (<https://trello.com>). Jest to aplikacja pozwalająca dzielić zadania na grupy, przypisywać im osoby oraz wyznaczać terminy ich realizacji. Dzięki możliwości przenoszenia zrealizowanych zadań do innych grup, łatwo śledzić bieżący postęp w pracach. Pozwala to również w razie błędów cofnąć wybrany etap z fazy gotowości na przykład do fazy dalszych testów lub ponownej implementacji.

2.5 Uruchomienie aplikacji

Aby uruchomić aplikację na komputerze, konieczne jest jego wstępne skonfigurowanie. W przeciwieństwie do bazy danych, aplikacja nie została wdrożona na publiczny serwer, wobec czego niezbędne jest sklonowanie plików projektu z repozytorium, a następnie uruchamianie aplikacji lokalnie.

W tym celu konieczne jest pobranie aplikacji XAMPP (<https://www.apachefriends.org/pl>), a w niej uruchomienie modułu *Apache*. Niezbędne jest również zmodyfikowanie dwóch plików konfiguracyjnych, by odpowiednio odwoływały się do tworzonej aplikacji:

- plik `Windows\System32\drivers\etc\hosts` należy uzupełnić o wpisy:

```
127.0.0.1 timesheet.dev
127.0.0.1 www.timesheet.dev
```

- plik `xampp\apache\conf\extra\httpd-vhosts.conf` należy uzupełnić o:

```
<VirtualHost *:80>
    ServerName timesheet.dev
    ServerAlias www.timesheet.dev
    DocumentRoot "F:/xampp/htdocs/timesheet"
    <Directory "F:/xampp/htdocs/timesheet">
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

W drugim z powyższych przypadków należy zwrócić szczególną uwagę na ścieżkę, która musi odpowiadać faktycznej lokalizacji folderów XAMPP na dysku.

Kolejnym krokiem przed uruchomieniem aplikacji jest pobranie silnika frameworka i wypakowanie go w lokalizacji `xampp\htdocs\timesheet\vendor`. Silnik ten udostępniliśmy do pobrania pod adresem: <https://www.dropbox.com/s/ceo3a3tkqdo8hsl/vendor.rar>.

Ostatnim etapem jest utworzenie pliku `.env` w głównym katalogu z plikami projektu, którego zawartość ma być następująca:

```
APP_ENV=local

APP_KEY=base64:RFXLPxKbmADO9x167hZ5IkInLC0pAadBxWIBtQbt7Vg=

APP_DEBUG=true

APP_LOG_LEVEL=debug

APP_URL=http://www.timesheet.dev


DB_CONNECTION=mysql

DB_HOST=sql.jj-dev.nazwa.pl

DB_PORT=3306

DB_DATABASE=jj-dev

DB_USERNAME=jj-dev

DB_PASSWORD=Ancymony666


BROADCAST_DRIVER=log

CACHE_DRIVER=file

SESSION_DRIVER=file

QUEUE_DRIVER=sync


REDIS_HOST=127.0.0.1

REDIS_PASSWORD=null

REDIS_PORT=6379


MAIL_DRIVER=smtp

MAIL_HOST=mailtrap.io

MAIL_PORT=2525

MAIL_USERNAME=null
```

```
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
```

3 Realizacja projektu

W tym rozdziale przedstawione zostaną widoki ekranów, które dostępne są dla poszczególnych funkcji w tworzonej aplikacji.

3.1 Zasada tworzenia widoków

Był sobie główny widok, który później się rozszerza

3.2 Wybrane widoki

3.2.1 Główny ekran aplikacji

3.2.2 Tworzenie nowego projektu

3.2.3 Coś jeszcze (?)

3.3 Mechanizm logowania użytkownika

3.4 Mechanizm rejestracji użytkownika

4 Testowanie

4.1 Metody testowania

4.2 Jakież wnioski?

5 Dalszy rozwój aplikacji

5.1 Założenia, które nie zostały zrealizowane

5.2 Możliwość rozbudowania aplikacji w przyszłości