Projekt Programistyczny



# Aplikacja do zarządzania projektami informatycznymi

## Autorzy:

Regina Giałbas

Patryk Gruszka

Łukasz Jamróz

Andrzej Mazurowski

Olena Zhuk

Spis treści

[Aplikacja do zarządzania projektami informatycznymi 1](#_Toc493622981)

[Autorzy: 1](#_Toc493622982)

[1. Założenia 3](#_Toc493622983)

[2. Wymagania funkcjonalne 3](#_Toc493622984)

[a. Autoryzacja użytkowników 3](#_Toc493622985)

[b. Tworzenie zadań, projektów i sprintów ich podział i przypisywanie 4](#_Toc493622986)

[c. Zarządzanie użytkownikami 7](#_Toc493622987)

[3. Wymagania techniczne 9](#_Toc493622988)

[a) Serwer WWW z PHP w wersji 5.6 i bazą danych MySQL 9](#_Toc493622989)

[b) Composer - Menadżer zależności dla PHP. 9](#_Toc493622990)

[c) Node.js - Środowisko programistyczne dla skalowalnych aplikacji internetowych. 9](#_Toc493622991)

[d) Grunt - JavaScript'owy system automatyzacji pracy. 9](#_Toc493622992)

[4. Instalacja 9](#_Toc493622993)

[Konfiguracja vhostów (na przykładzie serwera XAMPP) 9](#_Toc493622994)

[Utworzenie bazy danych 10](#_Toc493622995)

[Instalacja bazy danych oraz wymaganych bibliotek przed pierwszym uruchomieniem 10](#_Toc493622996)

[Instalacja zależności projektu Node.js 11](#_Toc493622997)

[Budowanie projektu za pomocą Grunt'a 11](#_Toc493622998)

[Praca nad projektem 11](#_Toc493622999)

[Dodatkowe pomocne informacje 12](#_Toc493623000)

[Wstępny podział projektu **Error! Bookmark not defined.**](#_Toc493623001)

[5. Instrukcja Obsługi 12](#_Toc493623002)

[6. Przykłady Testów 14](#_Toc493623003)

[7. Przykładowe Makiety 17](#_Toc493623004)

[8. Podział zadań w projekcie 18](#_Toc493623005)

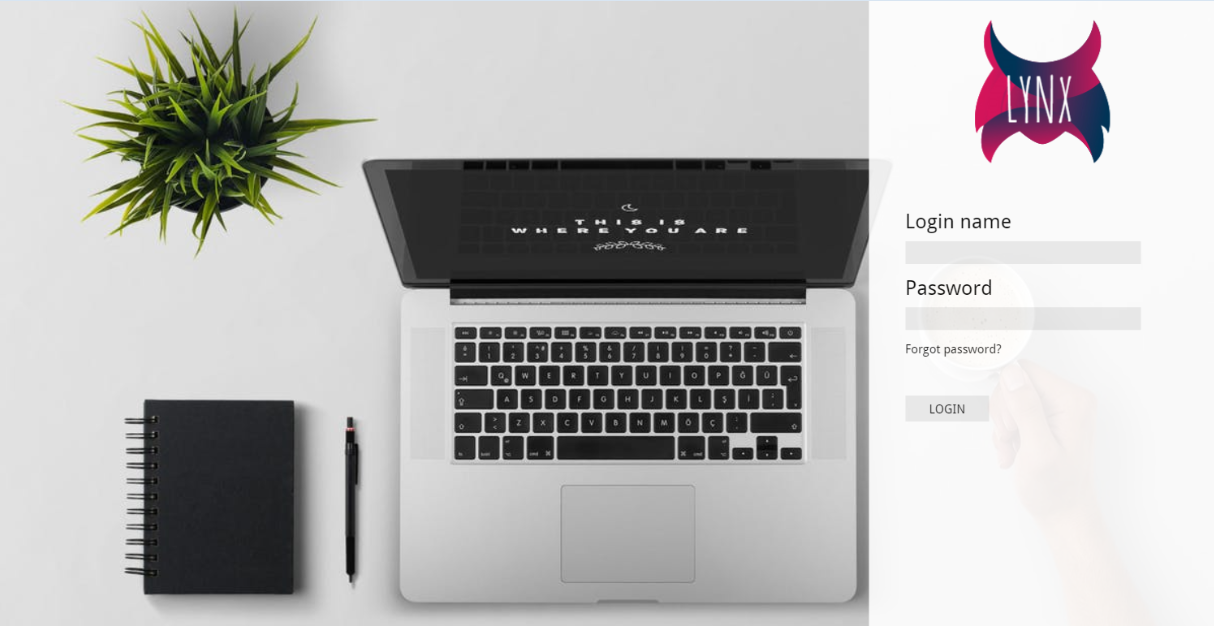
# Założenia

Serwis ma spełniać podstawowe założenia narzędzia umożliwiającego kontrolę pracy nad projektami. Główny użytkownik (właściciel) ma mieć możliwość definiowania projektów, kamieni milowych, poszczególnych zadań oraz ich statusów i priorytetów. Tworzy również pozostałych użytkowników i ustala dla nich typ. Wszystkie statusy, priorytety oraz typy użytkowników są z góry ustalone przez serwis. Serwis będzie wyświetlać dane w sposób przejrzysty i możliwy do dostosowania dla użytkownika, wyświetlane dane będą odświeżane w czasie rzeczywistym z opóźnieniem zależnym od docelowego serwera oraz jakości połączenia użytkownik-serwis.

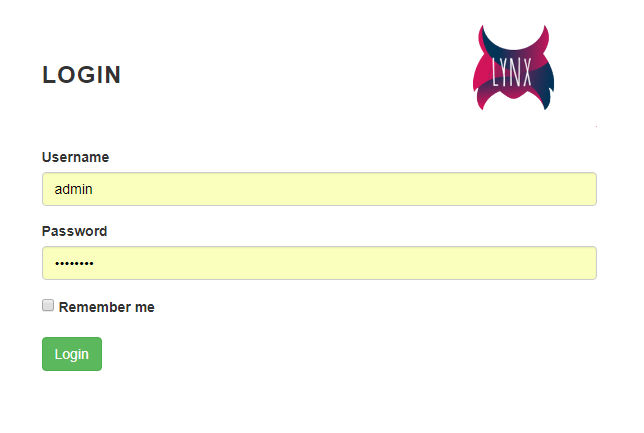
# Wymagania funkcjonalne

## Autoryzacja użytkowników

Celem autoryzacji jest kontrola dostępu (ang. access control), która potwierdza, czy dany podmiot jest uprawniony do korzystania z żądanego zasobu. Dla określenia uprawnień danego podmiotu konieczne jest najpierw stwierdzenie jego tożsamości, dlatego w typowym zastosowaniu autoryzacja następuje dopiero po potwierdzeniu tożsamości podmiotu za pomocą identyfikacji i uwierzytelnienia.



Obraz1. Przykładowy interfejs logowania



Obraz2. Istniejący interfejs logowania

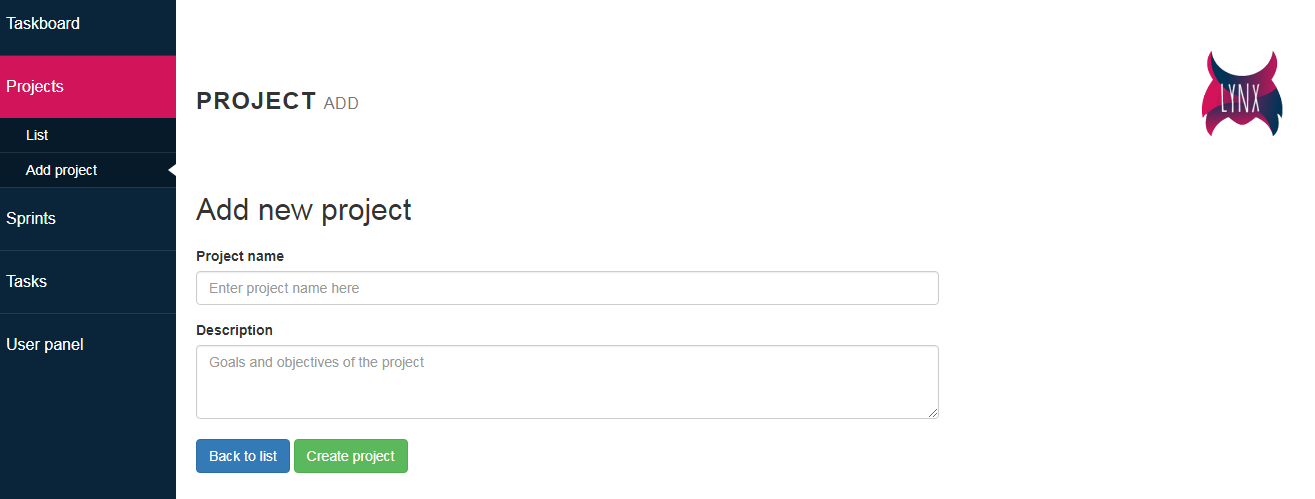
## Tworzenie zadań, projektów i sprintów ich podział i przypisywanie

**Projekt** - jest tymczasowym przedsięwzięciem, który ma na celu stworzenie unikalnej usługi lub produktu, gdzie charakterystyka tymczasowości określa, że to przedsięwzięcie ma bardzo ściśle określony początek i koniec, zaś unikalność oznacza, że usługa lub produkt w zauważalny sposób jest całkowicie inna niż wszystkie podobne usługi bądź produkty. Pojęcie projektu wymaga sprecyzowania głównych cech. Nalezą do nich: cel, niepowtarzalność, złożoność, określoność, zmienność, zaangażowanie zasobów ludzkich, ograniczoność czasowa.

Aplikacja powinna pozwolić użytkownikowi na stworzenie nowego projektu i jego modyfikacje.

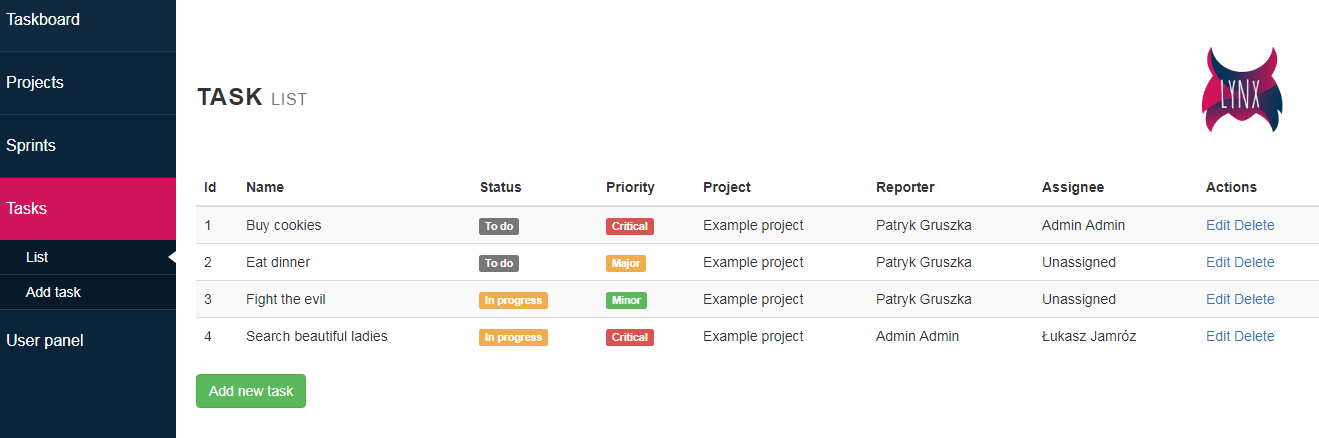


Obraz3. Lista projektów z przyciskiem umożliwiającym tworzenie projektów

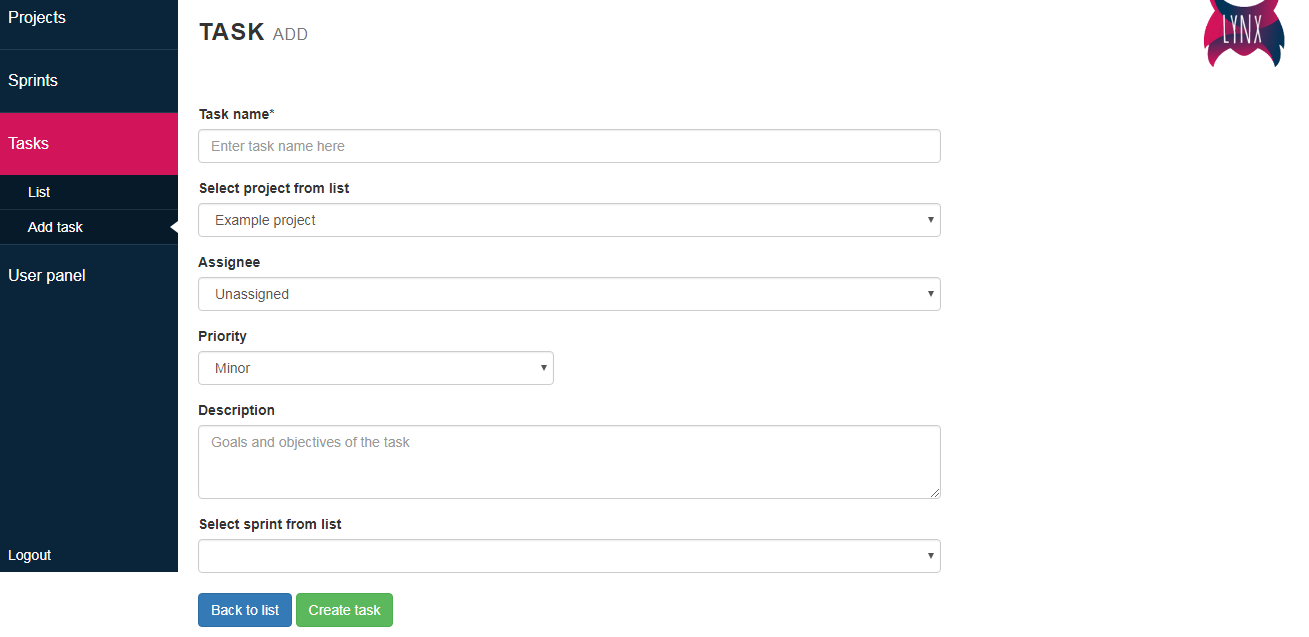


Obraz4. Formularz projektu

. **Task** jest to zadanie do zrealizowania. Charakteryzuje się Nazwą, Statusem, Priorytetem, Opisem Projektem (do którego jest przypisany), osobą tworzącą podane zadanie oraz osobą przypisaną do jego rozwiązania.

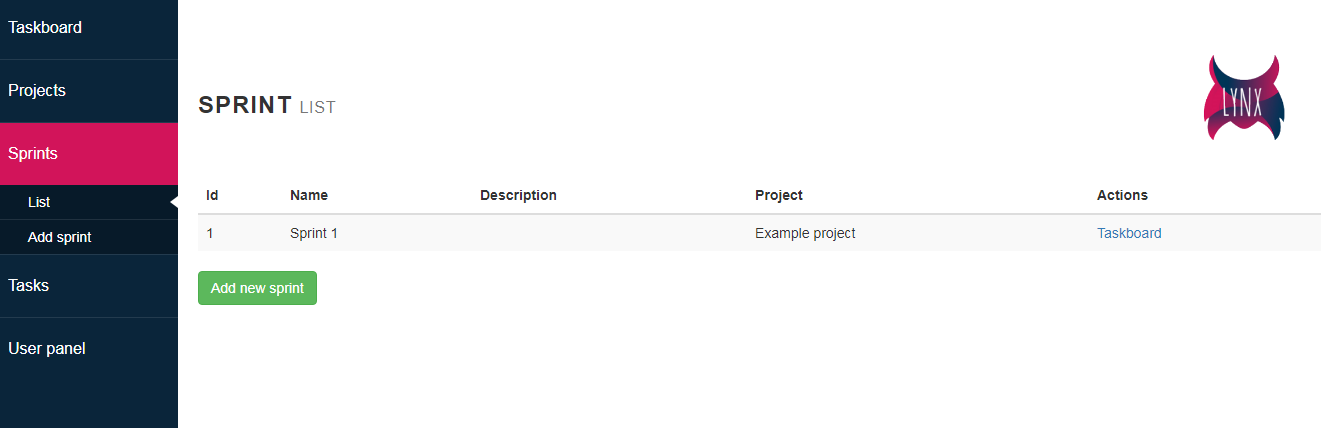


Obraz4. Lista zadań

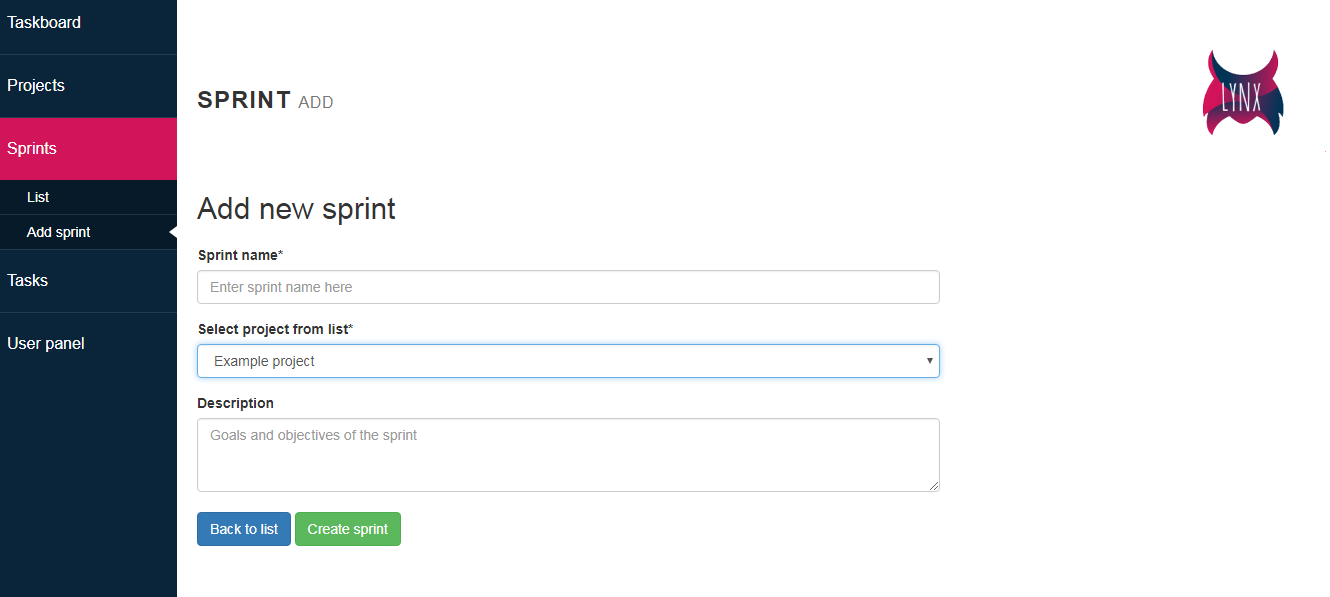


Obraz5. Formularz zadania

**Sprint** to główny element Scruma, cykl pracy, który mieści się w okresie maksymalnie jednego miesiąca. Czas trwania Sprinta pozostaje niezmienny w całym okresie tworzenia i pracy nad produktem. Sprinty mają charakter cykliczny, następują po sobie. Przy okazji każdego używa się tej samej struktury scrumowej.

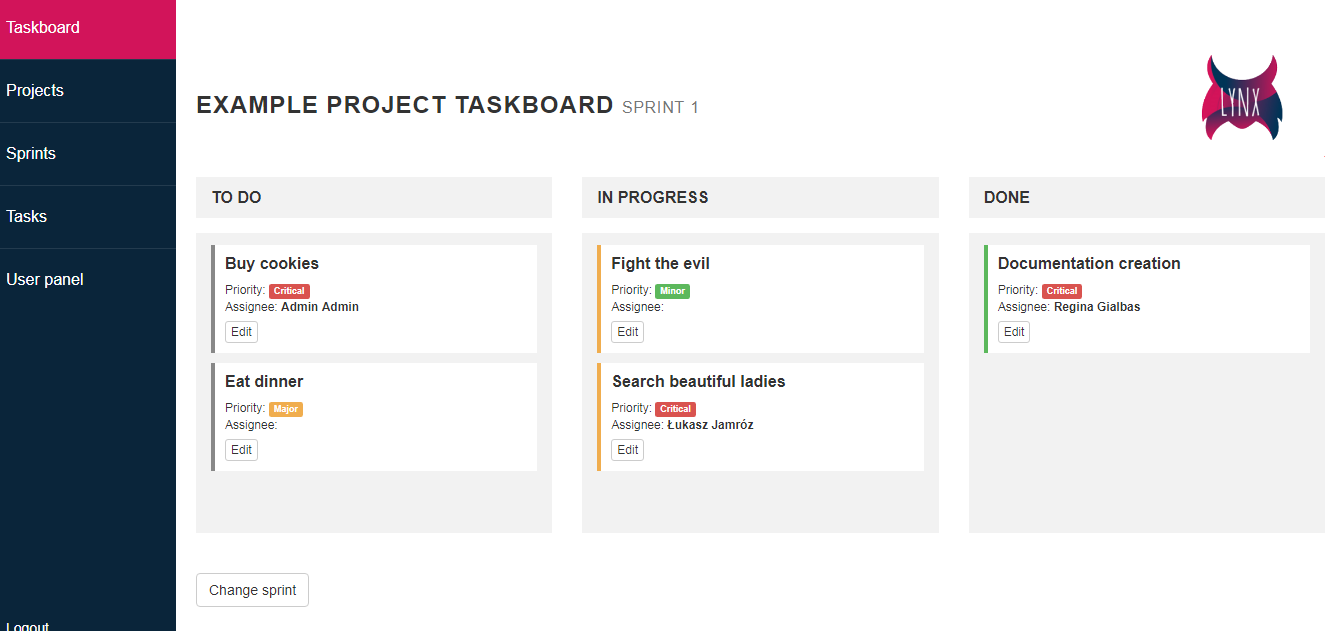


Obraz6. Lista sprintów



Obraz7. Formularz sprintów

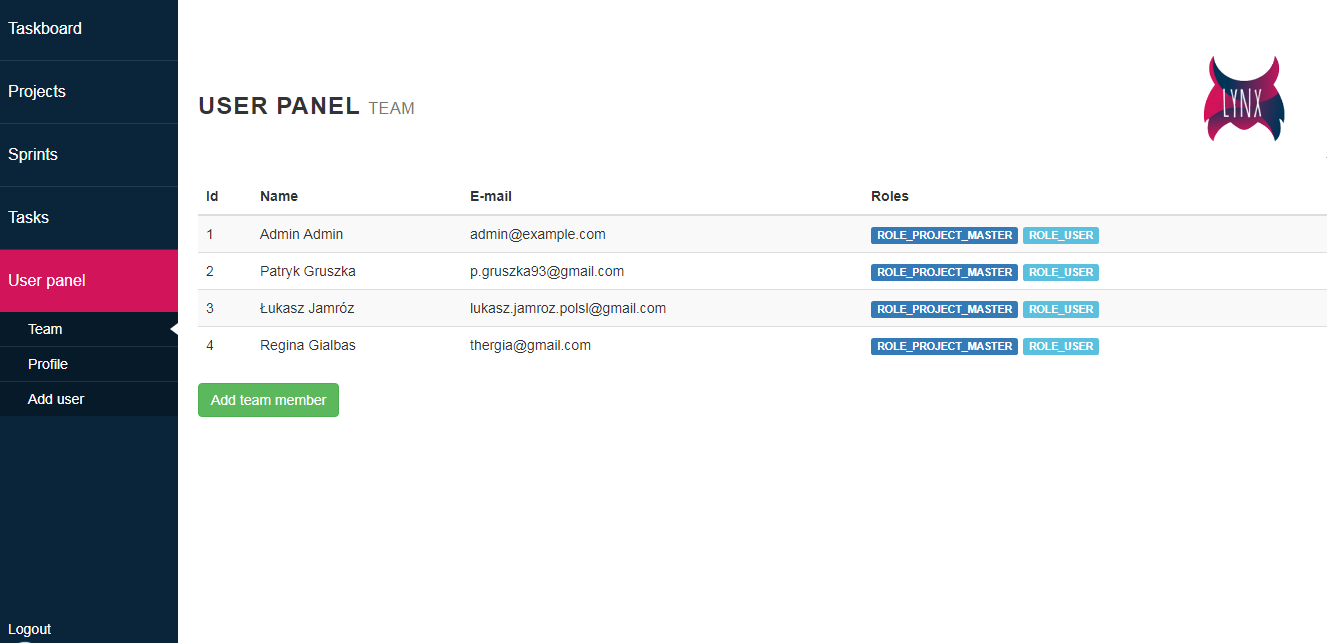
**Taskboard** (kanban) - Tablice kanban są odmianą tradycyjnego systemu kanban, polegającego na takim organizowaniu procesu wytwórczego, aby każda komórka organizacyjna produkowała dokładnie tyle, ile w danej chwili jest potrzebne, opartego na cyrkulacji kart wyrobów oraz analizie przepływu kart



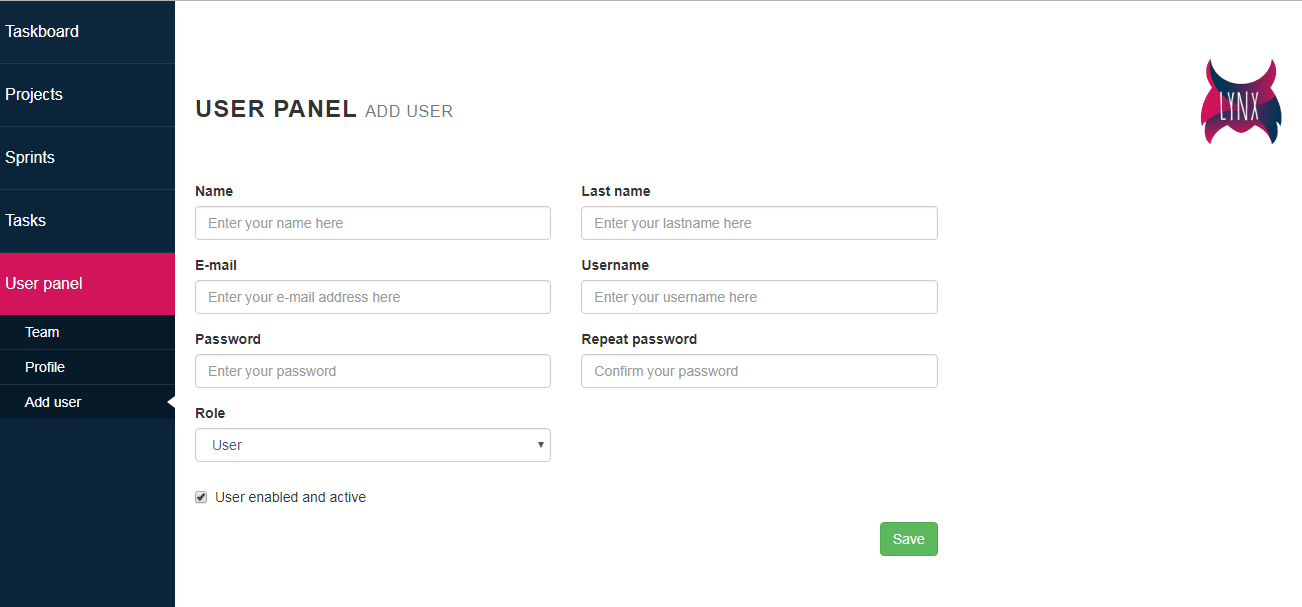
Obraz8. Taskboard

## Zarządzanie użytkownikami

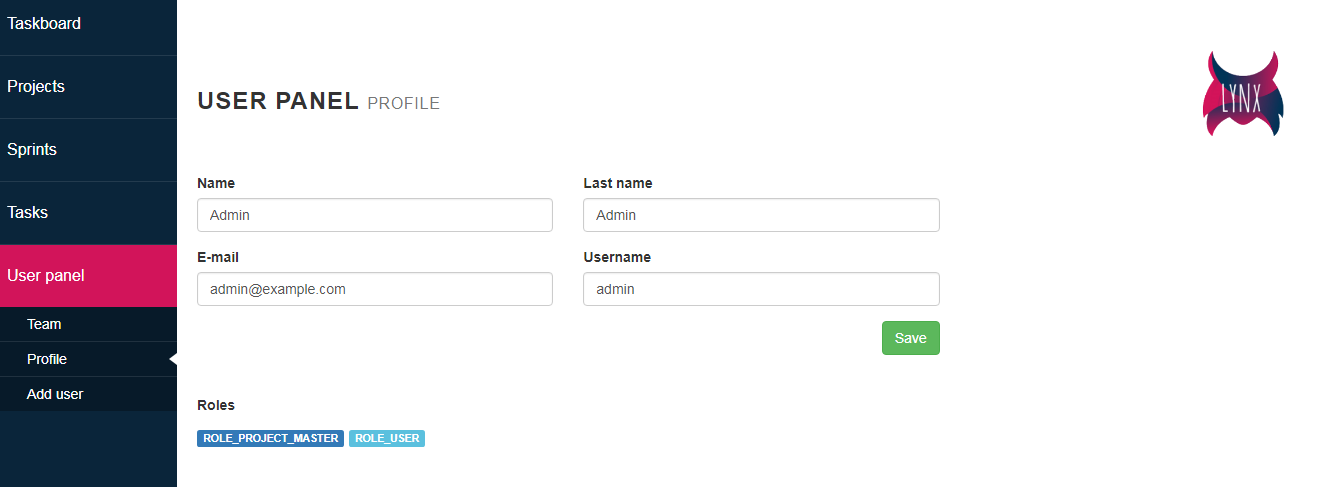
Zarządzanie użytkownikami (ang. user management) – całość działań w zakresie zakładania i usuwania kont użytkowników w systemie operacyjnym, tworzenia środowiska pracy użytkowników, ewidencji wykorzystywanych przez nich zasobów, rozliczania i godzenia sprzecznych interesów użytkowników.



Obraz9. Panel zarządzania użytkownikami



Obraz10. Formularz tworzenia użytkownika



Obraz10. Formularz profilu użytkownika

# Wymagania techniczne

## Serwer WWW z PHP w wersji 7.1 i bazą danych MySQL

Można korzystać z dowolnego serwera WWW, który posiada PHP w wersji 7.1 i serwer baz danych MySQL. W przykładach będziemy korzystali z serwera XAMPP - należy pobrać wersję z PHP 7.1 i zainstalować w systemie.

## Composer - Menadżer zależności dla PHP.

Instalacja globalna w systemie

Jedną z opcji jest zainstalowanie composera globalnie w systemie operacyjnym, w tym celu należy pobrać i zainstalować odpowiedni plik ze strony internetowej Composera.

## Node.js - Środowisko programistyczne dla skalowalnych aplikacji internetowych.

Środowisko programistyczne node.js należy pobrać ze strony internetowej i zainstalować w systemie operacyjnym.

## Grunt - JavaScript'owy system automatyzacji pracy.

Grunt jest instalowany globalnie za pomocą wiersza poleceń. Więcej informacji na stronie internetowej Grunta.

npm install -g grunt-cli

# Instalacja

### Konfiguracja vhostów (na przykładzie serwera XAMPP)

Konfiguracja vhostów serwera XAMPP znajduje się w katalogu serwera, np. C:\xampp\apache\conf\extra\httpd-vhosts.conf

Do pliku należy dodać wpis vhost dla przykładowej domeny lynx.dev oraz lynx.prod dla katalogu w którym znajduje się projekt

<VirtualHost \*:80>

DocumentRoot "SCIEZKA/DO/PROJEKTU/web"

ServerName lynx.dev

ErrorLog "logs/lynx.dev-error.log"

CustomLog "logs/lynx.dev-access.log" common

DirectoryIndex "app\_dev.php"

<Directory "SCIEZKA/DO/PROJEKTU">

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

</VirtualHost>

<VirtualHost \*:80>

DocumentRoot "SCIEZKA/DO/PROJEKTU/web"

ServerName lynx.prod

ErrorLog "logs/lynx.prod-error.log"

CustomLog "logs/lynx.prod-access.log" common

DirectoryIndex "app.php"

<Directory "SCIEZKA/DO/PROJEKTU>

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

</VirtualHost>

Przy wykorzystaniu vhosta należy pamietać o przekierowaniu domen na adres lokalny w pliku hosts. W systemach Windows znajduje się on w folderze:

#C:\Windows\System32\drivers\etc\hosts

127.0.0.1 lynx.dev

127.0.0.1 lynx.prod

### Utworzenie bazy danych

Należy utworzyć bazę danych dla projektu lynx. Dla serwera XAMPP jest dostępny panel phpMyAdmin.

* W panelu XAMPP control panel uruchamiamy usługi Apache oraz MySql
* Uruchamiamy akcję admin dla usługi MySql
* Tworzymy nową bazę np. lynx z metodą porównywania wpisów ustawioną na utf8\_general\_ci

### Instalacja bazy danych oraz wymaganych bibliotek przed pierwszym uruchomieniem

* composer install - Jeśli composer został zainstalowany globalnie
* W czasie instalacji zostanie wygenerowany także plik konfiguracyjny na podstawie podanych informacji.
* Podczas instalacji znaczące są informace dotyczące połączenia z bazą danych. Poniżej konfiguracja domyślna dla bazy danych lynx na porcie 3306 i użytkownika root bez hasła

database\_host: 127.0.0.1

database\_port: 3306

database\_name: lynx

database\_user: root

database\_password: null

* Po udanej instalacji należy wykonać następującą komendę aby utworzyć tablice na podstawie wstępnych encji: php app/console doctrine:database:create

### Instalacja zależności projektu Node.js

Aby zainstalować zależności projektu Node.js w głównym katalogu projektu należy wykonać polecenie: npm install

W katalogu projektu zostanie utworzony folder 'node\_modules' zawierający określone paczki npm wykorzystywane w projekcie.

### Budowanie projektu za pomocą Grunt'a

Aby zbudować odpowiednie pliki końcowe (np. CSS i JavaScript) należy w głównym katalogu projektu uruchomić polecenie:grunt

grunt watch - Tryb obserwacji zmian w plikach źródłowych (js, scss) - automatycznie wygeneruje odpowiednie pliki: skompilowane pliki CSS oraz pliki JS. Umożliwia także korzystanie z livereload: [https://github.com/gruntjs/grunt-contrib-watch#optionslivereload](https://github.com/gruntjs/grunt-contrib-watch" \l "optionslivereload).

### Praca nad projektem

Po każdej aktualizacji gita należy wykonać poniższe polecenia w podanej kolejności:

composer update

php app/console doctrine:schema:update -f

php app/console cache:clear --env=prod

npm install

grunt

#### Ładowanie inicjalizujących danych do bazy

php app/console doctrine:fixtures:load

### Dodatkowe pomocne informacje

Wszystkie polecenia można wykonać z poziomu Netbeans IDE poprzez PPM na projekcie w przypadku: \*php app/console po wybraniu z menu Symfony polecenia Run Command... \*composer po wybraniu z menu Composer podanego polecenia \*gruntpo wybraniu z menu Grunt Tasks podanego polecenia \*npm install po wybraniu polecenia npm Install

Aplikacja została także udostępniona pod adresem <http://lynx.aranea.usermd.net>

### Wstępny podział projektu

App/ApiBundle - Bundle odpowiedzialny za logowanie do serwisu, stronę startową oraz menu

App/ManagerBundle - Bundle odpowiedzialny za obsługę zdarzeń w serwisie App/UserBundle - Bundle odpowiedzialny za użytkowników, ich rejestrację i podstawowe operacje związane z kontem

Lynx/ProjectBundle - Bundle odpowiedzialny za Projekt i wszystkie rzeczy z nim związane

Lynx/TaskBundle - Bundle odpowiedzialny za pojedyńczy Task i wszystkie rzeczy z nim związane (wyświetlanie i edycja)

Lynx/TaskboardBundle - Bundle odpowiedzialny za wyświetlanie listy tasków

Lynx/UserpanelBundle - Bundle odpowiedzialny za ustawienia dotyczące Taskboardu związane z użytkownikiem

# Instrukcja Obsługi

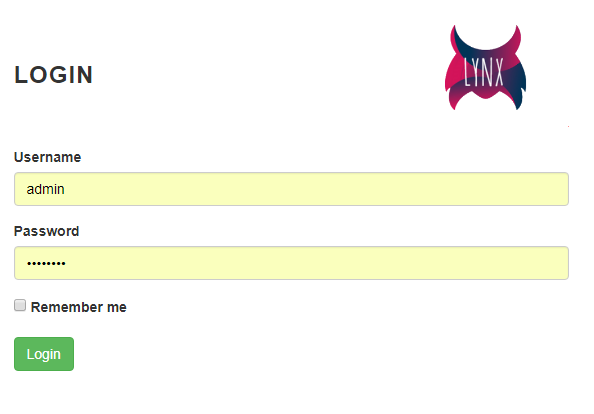
* 1. Logowanie i tworzenie użytkowników

Aby zalogować się do aplikacji wymagana jest unikalna nazwa użytkownika oraz jego hasło. Podstawowy użytkownik systemu:

Login: Admin

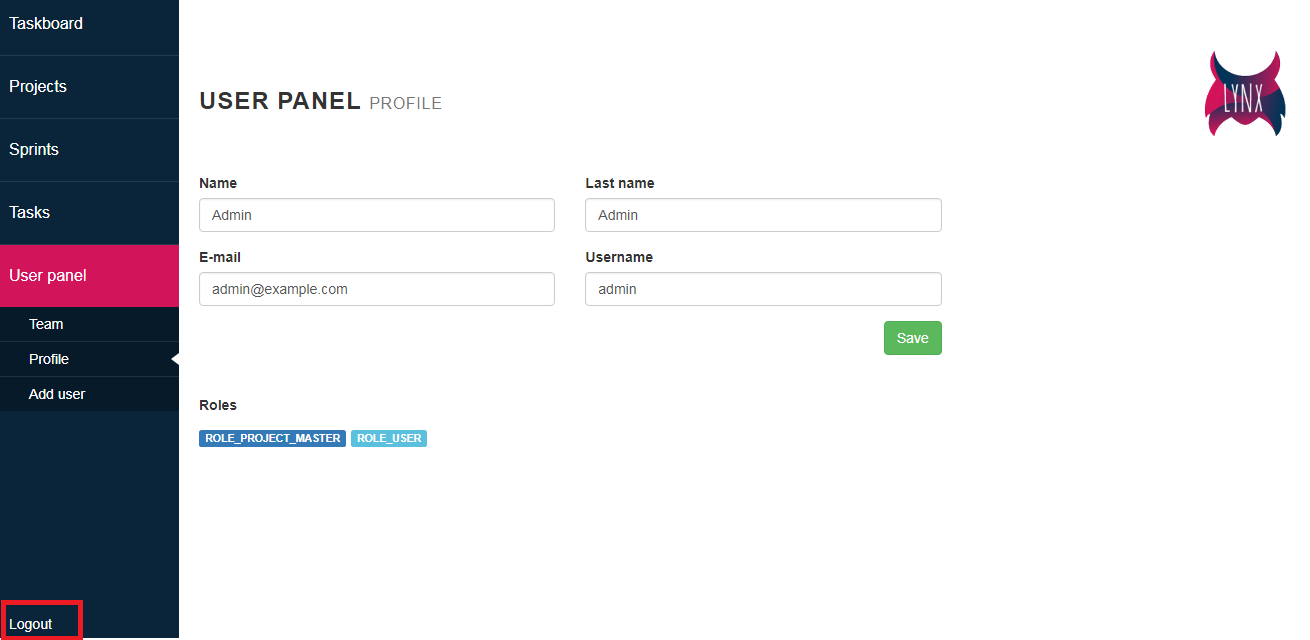
Hasło:testPass

Zatwierdzając chęć zalogowania przyciśnięciem przycisku Login



Obraz12. Przycisk do zalogowania

Aby wylogować się z systemu użyj przycisku Logut znajdującego się w prawym dolnym rogu.



Obraz12. Przycisk do wylogowania

Aby zmienić swoje dane przejdź do zakładki User Panel i po wprowadzeniu zmian zatwierdź swoją decyzję używając przycisku Save.

Aby stworzyć nowego użytkownika przejdź do zakładki Team i naciśnij przycisk Add team member. Po naciśnięciu przycisku pojawi się formularz tworzenia użytkownika. Aby zatwierdzić stworzenie użytkownika naciśnij przycisk Save .

* 1. Tworzenie i zarządzanie projektami

Aby utworzyć nowy projekt przejdź do zakładki Project i naciśnij przycisk Add new project. Po naciśnięciu przycisku pojawi się formularz tworzenia projektu. Aby zatwierdzić projekt naciśnij przycisk Create Project.

* 1. Tworzenie i zarządzanie sprintami

Aby utworzyć Sprint przejdź do zakładki Sprints i naciśnij przycisk Add new sprint. Następnie pojawi się formularz tworzenia sprintu. W celu zatwierdzenia utworzenia nowego sprintu potwierdź swoją decyzję przyciskiem Create Sprint.

* 1. Tworzenie i zarządzanie taskami

Aby uworzyć task przejdź do zakładki Task i kliknij przycisk Add new task. Następnie pojawi się formularz tworzenia nowego zadania. W celu utworzenia zadania zatwierdź przyciskiem Create Task.

Aby zedytować zadanie użyj przycisku Edit. Aby usunąć zadanie użyj przycisku Delete.

Wszystkie zadania są widoczne na Task Liscie oraz na Taskboardzie. Przeciągając zadania pomiędzy poszczególnymi kolumnami możesz łatwo i sprawnie zmieniać ich status.

# Przykłady Testów

Name: Logowanie do Aplikacji - pomyślne

Preconditions: Użytkownik istniejący w systemie

Steps:

Wpisz nazwę użytkownika, która istnieje w systemie.

Wpisz poprawne hasło użytkownika

Wciśnij przycisk Login

Expected result:

Logowanie pomyślne, przekierowanie do aplikacji.

Name: Logowanie do aplikacji – błędne wartości

Preconditions:

Steps:

Pole name puste

Pole password puste

Nacisnij Save

Expected result:

Nie udało się zalogować

Name: Logowanie do aplikacji – nieistniejący użytkownik

Preconditions:

Steps:

Wpisz w pole name nazwe użytkownika którego nie ma w systemie

Wpisz w pole password haslo które istnieje w systemie

Expected result:

Nie udało się zalogować

Name: Logowanie do aplikacji – błędne hasło

Preconditions: Użytkownik istnieje w systemie

Steps:

W pole login wpisz nazwe użytkownika który istnieje w systemie

W pole Password wpisz nieprawidłowe hasło

Expected result:

Nie udało się zalogowac

Name: Wylogowanie

Preconditions:

Użytkownik zalogowany do systemu

Steps:

Przycisnij przycisk Logout

Expected result:

Użytkownik został wylogowany

Name: Zapamiętanie użytkownika

Preconditions: Użytkownik istnieje w systemie

Steps:

Zaznacz checkbox „Remember me”

Zaloguj się do aplikacji

Wyloguj się z aplikacji

Spróbuj ponownie się zalogować

Expected result:

System podpowiada nam nazwę ostatnio zalogowanego użytkownika

Name: Tworzenie nowego projektu

Preconditions: Użytkowik zalogowany do aplikacji

Steps:

Przejdz do zakładki Project

Naciśnij przycisk Add new project

Uzupełnij formularz

Nacisnij przycisk create

Expected result:

W zakładce project list widnieje nowo utworzony projekt

Name: Tworzenie sprintu

Preconditions: Użytkownik zalogowany do systemu

Steps:

Przejdz do zakładki sprints i naciśnij przycisk create

Wypelnij formularz i naciśnij przycisk create

Expected result:

W zakładce sprints list pojawił się nowo utworzony sprint

Name: Tworzenie taska

Preconditions: Użytkownik zalogowany do systemu

Steps:

Przejdz do zakładki task i naciśnij przycisk create

Wypełnij formularz i zatwierdz przyciskiem Create

Expected result:

W zakładce task pojawił się nowo dodany task

Name: Zmiana statusu za pomocą taskboarda

Preconditions: Użytkownik zalogowany do systemu, task istnieje w systemie

Steps:

Przejdz do zakładki taskboard

Wybierz sprint do którego przypisany jest task lub taski

Przeciągnij task z jednej kolumny do drugiej

Expected result:

Status został zmieniony zgodnie z nazwą tabeli

Name: Modyfikacja Taska

Preconditions: Użytkownik zalogowany do systemu, task istnieje w systemie

Steps:

Przejdz do zakładki Task

Nacisnij przycisk Edit

Expected result:

Otwarty zostanie formularz taska

Name: Usuwanie Taska

Preconditions: Użytkownik zalogowany do systemu, task istnieje w systemie

Steps:

Przejdz do zakładki Task

Nacisnij przycisk Dele

Expected result:

Wybrany task został usunięty z Task Listy i Taskboardu

Name: Tworzenie użytkownika

Preconditions: Użytkownik Admin zalogowany do systemu

Steps:

Przejdz do zakładki Users

Naciśnij przycisk Add new team memmber

Wypelnij formularz

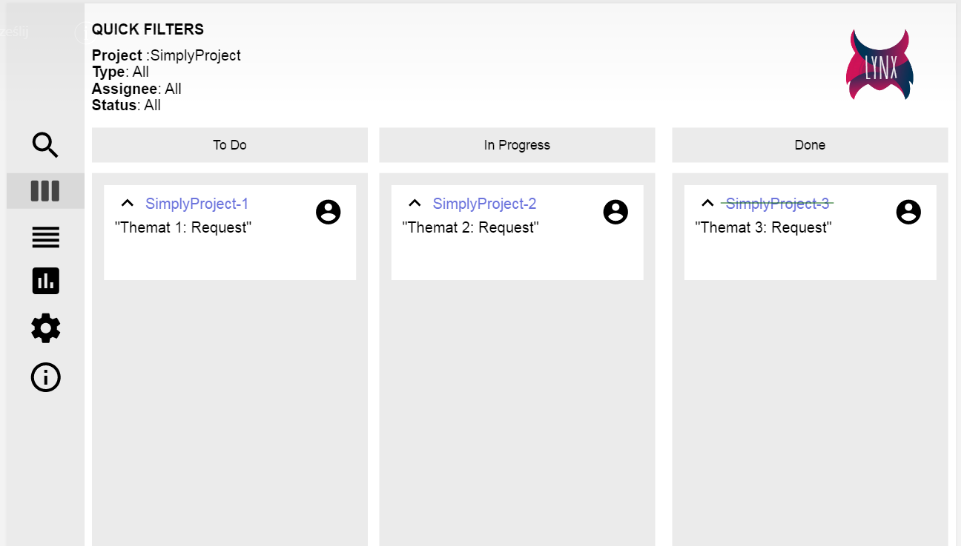
Naciśnij Create

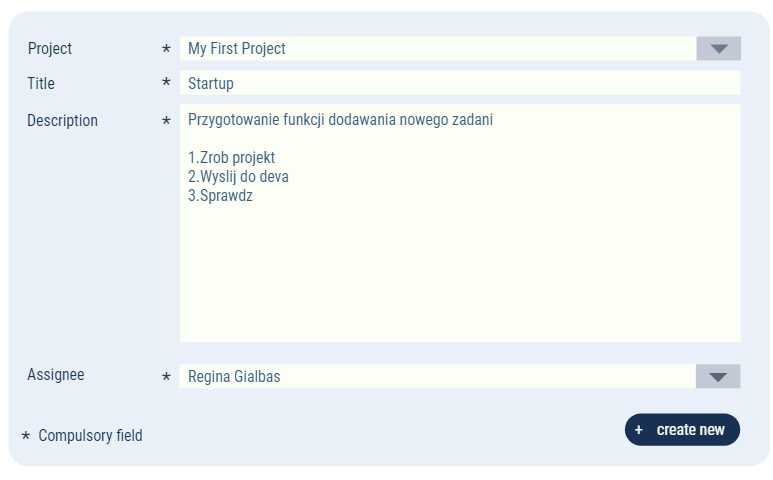
Expected result:

Urzytkownik został utworzony i jest wstanie zalogować się do systemu

# Przykładowe Makiety







# Podział zadań w projekcie

Regina Giałbas (thergia@gmail.com) - Dokumentacja, makiety, projekt, raport, testy, UX, QA, Project manager

Patryk Gruszka (p.gruszka93@gmail.com) - Aplikacja po stronie użytkownika (AngularJS), wdrożenie strony na zdalnym serwerze, developer, Web Designer, Personal Services

Łukasz Jamróz - Kod HTML, ostylowanie (CSS3), Web Developer

Olena Zhuk - Skruktura bazy danych i obiektów, ORM, API, Architect

Andrzej Mazurowski - Aplikacja po stronie serwera, logowanie, użytkownicy i role, Developer