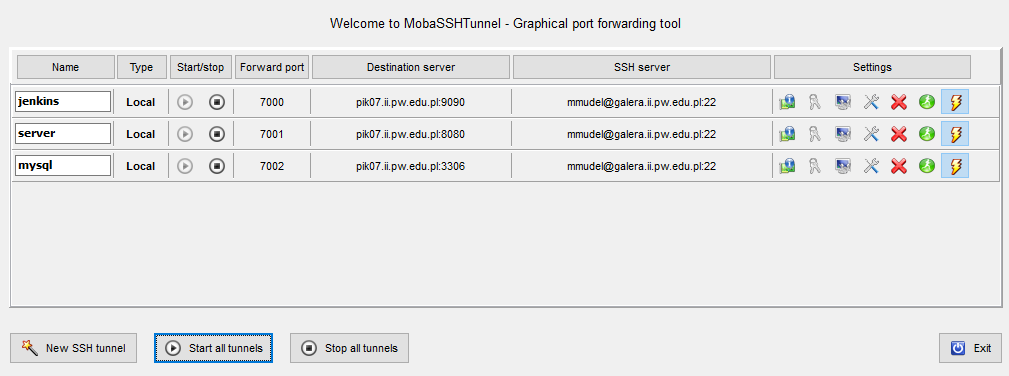
Dokumentacja środowiska

Autorzy:  
- Joanna Raczyńska,  
- Michał Mudel,  
- Wojtek Zieliński.

1. Opis projektu

Tematem projektu było stworzenie systemu służącego do zarządzania projektami informatycznymi. System ten został nazwany Project Management Tool. W założeniu, system miał pozwalać na tworzenie grup developerskich, skupionych wokół prywatnych bądź publicznych projektów. W obrębie tych projektów członkowie grup mogli definiować swoje tablice (np. związane z konkretnym sprintem, częścią projektu), w obrębie każdej z tych tablic możliwe było definiowanie list zadań (np. „w trakcie”, „do zrobienia”, „zakończone”), a w obrębie każdej z tych list miało być możliwe definiowanie zadań. Każde z zadań mogłoby zostać przydzielone do konkretnej osoby, a każda z pracujących nad nim osób mogłaby zalogować swój czas pracy nad tym zadaniem.

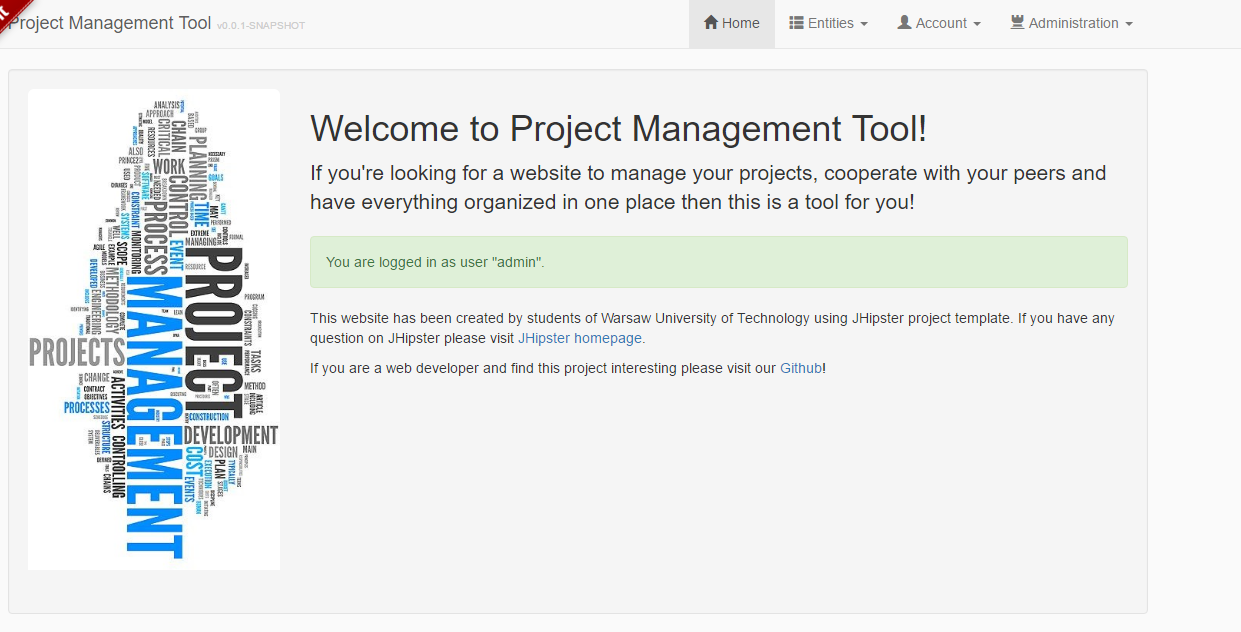
1. Wykonanie projektu
   1. Projekt podzielony jest na dwie części:
      1. Back-end – cała logika aplikacji, REST Api, serwisy
      2. Front-end – strona projektu, wyświetlana za pomocą przeglądarki
   2. Back-end
      1. Back-end został w całości napisany przy pomocy języka Java. Frameworkiem, zapewniającym biblioteki do wiekszości potrzebnych funkcjonalności jest Spring
      2. Skorzystaliśmy z narzędzia springboot
      3. Do testów wykorzystane zostały frameworki JUnit oraz Hamcrest
   3. Front-end
      1. Front-end został napisany przy użyciu HTML5, CSS oraz języka javascript,
      2. Javascriptowym frameworkiem wykorzystanym przy tworzeniu front-endu jest AngularJS
      3. Front-end aplikacji jest aplikacją typu SPA (Single Page Application)
2. Opis środowiska
   1. Projekt został stworzony z użyciem generatora projektów Jhipster. Wytyczne podanie przy tworzeniu projektu:
      1. Type: monolithic application
      2. Default java package name: pw.elka.pik.mkdev1
      3. Authentication type: classical HTTP session authentication
      4. Dependency manager: Maven
      5. Production database type: MySQL
      6. Developmnet database type: MySQL
      7. Hibernate 2nd level cache
      8. No search engine
      9. No HTTP clustered sessions
      10. No Websockets
   2. Używanie środowiska programistyczne w czasie developowania aplikacji
      1. Eclipse
      2. Intellij
   3. Repozytorium projektu, a zarazem narzędziem przez które prowadzone było code review jest github.
      1. Adres repozytorium: <https://github.com/mmudel/Pw.Elka.Pik.Mkdev1>
      2. Dostęp do repozytorium jest publiczny
   4. Zarządzanie projektem
      1. Narzędziem do zarządzanie projektem był system Trello
      2. Adres projektu: <https://trello.com/pwelkapik16lmkdev1>
      3. Dostęp do projektu jest prywatny
   5. Serwerem, na którym stawiane były narzędzia służące do wspomagania pracy nad projektem, a także wystawiony był sam projekt był serwer na wydziale Elektroniki i Technik Informacyjnych
      1. Adres serwera: pik07.ii.pw.edu.pl
      2. Jedyny dostęp do serwera był możliwy poprzez inny serwer na tym samym wydziale: galera.ii.pw.edu.pl
   6. W celu dostępu do narzędzi skonfigurowaliśmy tunelowanie. Na systemach Windows użyliśmy w tym celu programu MobaXterm



* 1. Do Ciągłej Integracji skorzystaliśmy z narzędzia Jenkins.
     1. Dostęp do narzędzia na wspomnianym wyżej serwerze na porcie 9090
     2. Jenkins został skonfigurowany w ten sposób, aby utworzył oddzielny job na każdy z branch’y projektu.
     3. Nie został skonfigurowany automatyczny build projektu przy każdym wypchnięciu zmian do repozytorium ze względu na utrudniony dostęp do serwera, na którym wystawiony jest jenkins
     4. Zamiast automatycznego budowania gałęzi po wypchnięciu zmian, skonfigurowany został mechanizm, który powoduje, że co 15 min jenkins sam odpytuje repozytorium czy były jakieś zmiany, i w razie jeśli były, buduje odpowiednią gałąź
     5. Pomyślne wykonanie zbudowania gałęzi master powoduje uruchomienie innego joba, którego zadaniem jest automatyczny deploy aplikacji na serwer aplikacji Apache Tomcat



* 1. Jako bazy danych użyliśmy serwera bazy danych MySQL
     1. Dostęp do bazy danych na wspomnianym wyżej serwerze na porcie 3306
     2. Lokalnie w celu połączenia z bazą korzystaliśmy z narzędzia MySQL workbench
  2. Aplikacja jest uruchamiana za pomocą narzędzia Apache Tomcat.
     1. Dostęp do aplikacji na wspomnianym wyżej serwerze na porcie 8080



1. Workflow projektu

W projekcie został wspólnie zdefiniowany następujący workflow, dotyczący developowania projektu:

1. Wybieramy jedno z user story w trello i przeciągamy je do kolumny In progress.
2. Zmiany wprowadzamy w branchu związanym z daną funkcjonalnością, jeśli taki branch nie istnieje to tworzymy nowy – feature/nazwa\_funkcjonalności.
3. Po wprowadzeniu całej funkcjonalności sprawdzamy czy branch buduje się
   1. W używanym IDE
   2. Po wpisaniu w konsoli znajdując się w głównym katalogu projektu mvn clean install
   3. W systemie Jenkins (odpowiedni job znajduje się w folderze mkdev1).

Jeśli branch buduje się bez błędów, możemy zmergować go z masterem, a następnie uruchomić job budujący gałąź master na jenkinsie.

1. W trello przerzucamy odpowiednie zadanie do listy Done. W komentarzu przy zadaniu podajemy hashe commitów oraz nazwę brancha który dotyczy tego zadania.