

Projekt Programistyczny			
Rok akademicki	Rodzaj studiów	Kierunek	Grupa
2011/2012	Dzienne	INF	GKiO2

Dokument Projektowy P1

Przygotowanie środowiska

Skład sekcji:

Piotr Tomala
 Mateusz Sulima
 Paweł Grzywocz
 Michał Krupiński
 Marek Machnik

Opis środowiska pracy

Narzędzie CASE

Jako narzędzie CASE wybraliśmy Enterprise Architect 8, ponieważ jest to narzędzie cieszące się dużą renomą i mające wszystkie potrzebne nam do zbudowania diagramów UML funkcje. Ponadto wszyscy członkowie grupy korzystali już z niego przy projekcie z Inżynierii Oprogramowania.

Narzędzia wersjonujące

Wybraliśmy git (<http://git-scm.com/>) jako system kontroli wersji. Większość członków sekcji nie miała z nim styczności, jednak dla osób, które miały do czynienia np. z Subversion nauczanie się nowego narzędzia nie powinno stanowić problemu. Głównym powodem, dla którego zdecydowaliśmy się na git jest jego integracja z github.

Narzędzie pracy grupowej

Postanowiliśmy utworzyć projekt w serwisie github (<http://github.com/>). Do udostępnianych przez ten serwis narzędzi należą:

- Repozytorium systemu kontroli wersji git
- Zarządzanie zadaniami (issue tracker), w tym przydzielanie ich do członków zespołu i ich etykietowanie
- Tworzenie kamieni milowych
- Przeglądanie kodu
- Komentarze dotyczące commitów do repozytorium
- Wiki

W porównaniu do innych serwisów udostępniających podobne narzędzia (np. Assembla), główną zaletą githuba jest to, że w projektach oznaczonych jako opensourcowe jest bezpłatny. Nasz projekt dostępny jest pod adresem: <https://github.com/ProjektPPiBD/Baza-organizacji>

Narzędzia deweloperskie

Kod programu mamy zamiar napisać przy użyciu środowiska SpringSource Tool Suite. Jest to narzędzie oparte o Eclipse, mające wbudowane kilka narzędzi ułatwiających pracę w Spring Framework, jak graficzne edytory plików konfiguracyjnych. Ponadto działają z nim wszystkie wtyczki do Eclipse, takie jak wspierające git, Maven itp.

Biblioteki

Całość aplikacji oparta będzie o Spring Framework. Wykorzystamy z niego takie cechy jak wstrzykiwanie zależności (Dependency Injection), wsparcie integracji z Hibernate, zarządzanie bezpieczeństwem (Spring Security). Wykorzystamy również ORM w postaci biblioteki Hibernate, a do stworzenia GUI bibliotekę JavaFX 2.

Środowisko testowania i zarządzania zmianami

Do testowania użyjemy biblioteki JUnit 4, gdyż jest ona bardzo wygodna w użyciu i rozwinięta, ponadto Spring udostępnia dla niej wsparcie dla testów wykorzystujących transakcyjne bazy danych. Do zarządzania konfiguracją projektu użyjemy Maven 2. Pozwala on w prosty sposób zarządzać

zależnościami projektu od bibliotek oraz wygodnie konfigurować etapy kompilacji projektu. W dodatku można z niego w prosty sposób uruchomić wszystkie testy jednostkowe w projekcie.