



# Inżynieria Oprogramowania & Bazy Danych - Projekt

## 1. Wybór technologii

Opracował: Maciej Penar

## Spis treści

1. Wybór technologii.....	3
2. Proof of Concept + Prezentacja .....	3
3. Grupa L5 .....	4
4. Grupa L6 .....	5

## 1. Wybór technologii

Przygotować rozdział do sprawozdania końcowego które:

1. **Krótko** opisuje rozważane dostępne technologie np.:
  - a. Front-Endu
  - b. Back-Endu
  - c. Aplikacji typu Desktop
  - d. Aplikacji mobilnych
  - e. Konteneryzacja (np. Docker) / Wirtualizacja (np. Vagrant)
2. Zebrać zbiór języków w których piszemy w ramach wybranych technologii i znaleźć przydatne biblioteki. Opisać je zwięźle (1-3 zdania). Interesują nas przede wszystkim:
  - a. Metoda zapisu/odczytu z Bazy Danych (np. ORM'y)
  - b. Metoda prezentacji danych użytkownikowi
  - c. **Biblioteki wspierające testy jednostkowe**
  - d. Biblioteki wspierające serializację/deserializację JSON'ów/XML/YAML
  - e. Biblioteki wspierające autentykację (np. z użyciem SSO)
  - f. Pozostałe
3. Przedstawiona jest analiza SWOT dla każdej z technologii. Pomyśleć o:
  - a. Warunkach w jakich będziecie stosować
  - b. Powszechność języków, paradygmaty, dojrzałość języków / frameworków, zwięzłość kodu
  - c. Bibliotekach zewnętrznych
4. Podana jest ostateczna rekomendacja dot. wybranej technologii – **dopóki nie ma tematu to można opuścić.**

Każda technologia powinna być opisana na **minimum** 1 pełną stroną – dobrze jest pokazać przykład w kodzie, zobrazować diagramami. Dobór technologii odbędzie się na pierwszych zajęciach. Proszę o skoordynowanie się i sklejenie jednego zbiorczego dokumentu z opisem dostępnych narzędzi.

Szablon dokumentu powinien być w katalogu.

## 2. Proof of Concept + Prezentacja

Żeby dokument z zadania 1 był wiarygodny należy przygotować 5-10 minutowe prezentacje dot. dostępnych technologii wraz z prezentacją działającego kodu który realizuje:

- Umieszczenie bytu w bazie danych
- Prezentację tego bytu na liście
- Umożliwia usunięcie bytu z bazy danych
- Wystawienie aplikacji jako strony internetowej na **jakimś porcie** na **localhost**

Rozważyć liczbę dostępnych bibliotek, integrację z chmurami/VCS'ami, wspieranie debugowania.

Szablon prezentacji powinien być w katalogu.

### 3. Grupa L5

Wybraliście tematy:

1. Java (3 osoby)
  - Jak wystawić jakiś port na świat / jak stworzyć stronę webową (np. Spring / Sockety ☺ / Servlety)
  - Jakiś ORM (np. OrmLite / Java Persistence API / Hibernate) lub inny sposób komunikacji z BD (np. JDBC)
2. JS (+ HTML + CSS) (3 osoby)
  - Server-side (np. NodeJS)
  - Client-side
  - Frameworki (np. ExtJS / Angular / React)
3. Python (3 osoby)
  - Jak wystawić jakiś port na świat / jak stworzyć stronę
  - Jakiś ORM lub inny sposób komunikacji z BD
4. Wirtualizacja (3 osoby) = **Vagrant** (hostowany np. na VirtualBox)
  - Jaki jest concept?
  - Jak go wykorzystać w projekcie?
  - Postawić jakikolwiek tutorial lub usługę np. Jenkins / Redis / Memcached / Tomcat
5. Konteneryzacja (3 osoby) = **Docker** (hostowany na VirtualBox lub Hyper-V)
  - Jaki jest concept?
  - Jak go wykorzystać w projekcie?
  - Postawić jakikolwiek tutorial lub usługę np. Jenkins / Redis / Memcached / Tomcat

## 4. Grupa L6

Wybraliście tematy:

1. **Java (2 osoby)**
  - Jak wystawić jakiś port na świat / jak stworzyć stronę webową (np. Spring / Sockety ☺ / Servlety)
  - Jakiś ORM (np. OrmLite / Java Persistence API / Hibernate) lub inny sposób komunikacji z BD (np. JDBC)
2. **C# (2 osoby)**
  - Jak wystawić jakiś port na świat / jak stworzyć stronę webową (**NIE**: WebForms, Silverlight) (np. ASP.NET MVC)
  - Jakiś ORM (np. Entity Framework / Dapper) lub inny sposób komunikacji z BD (np. surowe połączenie)
3. **JS (+ HTML + CSS) (3 osoby)**
  - Server-side (np. NodeJS)
  - Client-side
  - Frameworki (np. ExtJS / Angular / React)
4. **Python (2 osoby)**
  - Jak wystawić jakiś port na świat / jak stworzyć stronę
  - Jakiś ORM lub inny sposób komunikacji z BD
5. **Wirtualizacja (3 osoby) = Vagrant** (hostowany np. na VirtualBox)
  - Jaki jest concept?
  - Jak go wykorzystać w projekcie?
  - Postawić jakikolwiek tutorial lub usługę np. Jenkins / Redis / Memcached / Tomcat
6. **Konteneryzacja (3 osoby) = Docker** (hostowany na VirtualBox lub Hyper-V)
  - Jaki jest concept?
  - Jak go wykorzystać w projekcie?
  - Postawić jakikolwiek tutorial lub usługę np. Jenkins / Redis / Memcached / Tomcat

## 5. Linki

### 1. Java:

- a. Java SDK
  - i. SE: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
  - ii. EE: <https://projects.eclipse.org/proposals/eclipse-jakarta-ee-platform>
- b. IDE:
  - i. IntelliJ: <https://www.jetbrains.com/idea/download/>
  - ii. Eclipse: <https://www.eclipse.org/downloads/packages/release/mars/r/eclipse-ide-java-developers>
  - iii. Netbeans: <https://netbeans.org/>
- c. Framework/kombajn Spring: <https://spring.io/>
- d. Dostęp do danych:
  - i. ORMLite: <http://ormlite.com/>
  - ii. JPI: <https://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html>
  - iii. Hibernate: <http://hibernate.org/>
- e. Serwery aplikacji:
  - i. Tomcat: <http://tomcat.apache.org/>
  - ii. Glassfish: <https://javaee.github.io/glassfish/>
- f. Inne:
  - i. Maven (budowanie): <https://maven.apache.org/>
  - ii. Gradle (budowanie): <https://gradle.org/>
  - iii. Apache commons (productivity): <https://commons.apache.org/>
  - iv. Guava (productivity): <https://github.com/google/guava>
  - v. Lombok (productivity): <https://projectlombok.org/>
- g. Testy:
  - i. Junit: <https://junit.org/junit5/>
  - ii. Mockito: <https://site.mockito.org/>

### 2. C#

- a. SDK:
  - i. .NET 4.7.2: <https://dotnet.microsoft.com/download/dotnet-framework-runtime/net472>
- b. IDE:
  - i. Visual Studio: <https://visualstudio.microsoft.com/pl/?rr=https%3A%2F%2Fwww.google.com%2F>
  - ii. Visual Studio Code: <https://code.visualstudio.com/>
- c. Dostęp do danych:
  - i. ORMLite: <https://github.com/ServiceStack/ServiceStack.OrmLite>
  - ii. Dapper: <https://github.com/StackExchange/Dapper>
  - iii. Entity Framework: <https://docs.microsoft.com/pl-pl/ef/>
- d. Serwery aplikacji IIS: <https://www.iis.net/> (poszukać instalacji z poziomu Panelu Sterowania w Windows Home)
- e. Inne:
  - i. MSBuild (budowanie): <https://docs.microsoft.com/pl-pl/visualstudio/msbuild/msbuild?view=vs-2017>
  - ii. NANT: <http://nant.sourceforge.net/>
  - iii. LINQ: <https://docs.microsoft.com/pl-pl/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>
- f. Testy:

- i. Nunit: <https://nunit.org/>
- 3. JS
  - a. NodeJS: <https://nodejs.org/en/>
  - b. ExtJS: <https://www.sencha.com/products/extjs/>
  - c. Angular: <https://angular.io/>
  - d. React: <https://reactjs.org/>
  - e. ExpressJS: <https://expressjs.com/>
- 4. Python
  - a. SDK:
    - i. SDK: <https://www.python.org/downloads/>
  - b. IDE:
    - i. Eclipse (patrz: Java)
    - ii. Visual Studio (patrz: C#)
    - iii. Visual Studio Code (patrz: C#)
  - c. Framework: <https://www.djangoproject.com/start/>
  - d. Dostęp do danych:
    - i. ORM: <https://www.fullstackpython.com/object-relational-mappers-orms.html>
  - e. Testowanie:
    - i. Unittest: <https://docs.python.org/3/library/unittest.html>
- 5. Wirtualizacja
  - a. Wiki: <https://pl.wikipedia.org/wiki/Wirtualizacja>
  - b. Wiki: <https://pl.wikipedia.org/wiki/Hyper-V>
  - c. VirtualBox: <https://www.virtualbox.org/>
  - d. Vagrant: <https://www.vagrantup.com/>
- 6. Konteneryzacja:
  - a. Wiki: <https://pl.wikipedia.org/wiki/Wirtualizacja>
  - b. Wiki: <https://pl.wikipedia.org/wiki/Hyper-V>
  - c. Wiki: [https://pl.wikipedia.org/wiki/Docker\\_\(oprogramowanie\)](https://pl.wikipedia.org/wiki/Docker_(oprogramowanie))
  - d. VirtualBox: <https://www.virtualbox.org/>
  - e. Docker: <https://www.docker.com/>

## 6. W pozostałych rolach

- 1. Upublicznianie adresów lokalnych na potrzeby developementu: <https://ngrok.com/>
- 2. Repozytoria kodu źródłowego **VCS**:
  - a. Oprogramowanie:
    - i. GIT: <https://git-scm.com/>
    - ii. Tortoise: <https://tortoisesvn.net/>
  - b. Repozytoria:
    - i. Github: <https://github.com/>
    - ii. Bitbucket: <https://bitbucket.org/>
- 3. Organizacja pracy:
  - a. Trello: <https://trello.com/>
  - b. JIRA: <https://pl.atlassian.com/software/jira>
- 4. **Single-Sign On**, czyli lenistwo na wyższym poziomie:
  - a. Wiki: [https://en.wikipedia.org/wiki/Single\\_sign-on](https://en.wikipedia.org/wiki/Single_sign-on)
  - b. Protokoły: WS-FED / SAML / SAML2 / OAuth / OAuth2 / CAS
  - c. Via facebook: <https://developers.facebook.com/docs/facebook-login/web>