POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA (INF)

SPECJALNOŚĆ: INŻYNIERIA SYSTEMÓW INFORMATYCZNYCH (INS)

PRACA DYPLOMOWA

INŻYNIERSKA

Wirtualna kwiaciarnia – interaktywny system sprzedaży kwiatów

Virtual florist – an interactive system for selling flowers

AUTOR:

Karol Maśluch

PROWADZĄCY PRACĘ:

Dr inż. Tomasz Babczyński, K9

OCENA PRACY:

WROCŁAW, 2019

**Spis treści**

[Spis rysunków 4](#_Toc25584978)

[Spis tabel 5](#_Toc25584979)

[Spis listingów 6](#_Toc25584980)

[Skróty 7](#_Toc25584981)

[1. Wstęp 8](#_Toc25584982)

[1.1. Wprowadzenie 8](#_Toc25584983)

[1.2. Cel i zakres pracy 8](#_Toc25584984)

[2. Analiza wymagań systemu 9](#_Toc25584985)

[2.1. Opis aktorów 9](#_Toc25584986)

[2.2. Wymagania funkcjonalne 9](#_Toc25584987)

[2.3. Wymagania niefunkcjonalne 11](#_Toc25584988)

[2.4. Przyjęte założenia projektowe 11](#_Toc25584989)

[3. Wykorzystane technologie i narzędzia programistyczne 11](#_Toc25584990)

[3.1. JDK 11](#_Toc25584991)

[3.2. Eclipse 12](#_Toc25584992)

[3.3. Spring 12](#_Toc25584993)

[3.4. Spring Boot 12](#_Toc25584994)

[3.5. Maven 13](#_Toc25584995)

[3.6. Git 14](#_Toc25584996)

[3.7. Hibernate 14](#_Toc25584997)

[3.8. MySQL 14](#_Toc25584998)

[3.9. NPM 14](#_Toc25584999)

[3.10. React.js 14](#_Toc25585000)

[3.11. Visual Studio Code 14](#_Toc25585001)

[3.12. Redux 14](#_Toc25585002)

[3.13. Axios 14](#_Toc25585003)

[3.14. Material-UI 14](#_Toc25585004)

[3.15. MDBReact 14](#_Toc25585005)

[3.16. Pozostałe 14](#_Toc25585006)

[4. Projekt systemu 15](#_Toc25585007)

[4.1. Architektura aplikacji 15](#_Toc25585008)

[4.2. Projekt bazy danych 15](#_Toc25585009)

[4.2.1. Model konceptualny 15](#_Toc25585010)

[4.2.2. Model logiczny 15](#_Toc25585011)

[4.2.3. Model fizyczny z ograniczeniami integralności danych 15](#_Toc25585012)

[5. Implementacja systemu 15](#_Toc25585013)

[5.1. Back-End 15](#_Toc25585014)

[5.2. Front-End 15](#_Toc25585015)

[5.3. Projekt wybranych funkcji 15](#_Toc25585016)

[5.4. Struktura interfejsu graficznego 15](#_Toc25585017)

[6. Testowanie wybranych funkcji systemu 15](#_Toc25585018)

[7. Podsumowanie i wnioski 15](#_Toc25585019)

[Literatura 16](#_Toc25585020)

[Dodatek A 17](#_Toc25585021)

# Spis rysunków

[Rys. 1. Przykład podpisu rysunku 3](#_Toc465685478)

# Spis tabel

[Tab. 1. Przykład podpisu tabeli 3](#_Toc465685652)

# Spis listingów

[Listing. 1. Początkowe żadanie HTTP 3](#_Toc465685644)

# Skróty

**JDK**(ang. *Java Development Kit*)

**JVM** (ang. *Java Virtual Machine*)

JRE (ang. Java Runtime Environment)

**REST** (ang. *Representational state transfer*)

**ORM** (ang. *Object-Relational Mapping*)

**SQL** (ang. *Structured Query Language*)

**JS** (ang. *JavaScript*)

**SPA** (ang. *Single-page application*)

**JSON** (ang. *JavaScript Object Notation*)

**XML** (ang. *Extensible Markup Language*)

**API** (ang. *Application programming interface*)

JPA(ang. Java Persistence API)

**CSS** (ang. *Cascading Style Sheets*)

**POM** (ang. *Project Object Model*)

**UML** (ang. *Unified Modeling Language*)

**ISO** (ang. *International Standards Organization*)

**IT** (ang. *Information Technology*)

IDE (ang. Integrated Development Environment)

JDBC

IoC

AOP

EJB

JAR

URL

1. Wstęp

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus. Aliquam erat volutpat. Integer maximus est turpis, ut bibendum ligula accumsan et. Ut eget vestibulum libero. Aliquam erat volutpat. Nullam placerat mauris a lectus tincidunt, et aliquet turpis aliquam. Etiam in malesuada lacus. Proin dignissim augue sit amet auctor elementum. Suspendisse potenti. Vivamus suscipit vulputate massa ac molestie. Suspendisse a justo porttitor, commodo mi at, placerat risus. Integer lobortis augue ac neque suscipit, vel sodales lacus fringilla.

* 1. Wprowadzenie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. Cel i zakres pracy

Celem projektu jest stworzenie responsywnej aplikacji webowej. System zostanie stworzony w oparciu o trójwarstwową architekturę. System będzie podzielony na dwa części:

* Część transakcyjna – umożliwia sprzedaż kwiatów
* Część informacyjna – udostępnia szczegółowych informacji na temat kwiatów

Projekt zakłada stworzenie trzech elementów aplikacji:

* Baza danych
* Back-End
* Front-End

1. Analiza wymagań systemu

Wymagania tworzonej aplikacji zostały sformułowane na podstawie analizy istniejących produktów, ich zalet oraz braków. Głównym celem tych wymagań powinno być osiągniecie jak największej wygody użytkowania. Można to osiągnąć poprzez wygląd aplikacji oraz zapewnianą funkcjonalność.

* 1. Opis aktorów

W systemie możemy wyróżnić trzech aktorów: Gościa, Klienta i Pracownika. Każdy z nich cechuje się funkcjami jakimi powinien być w stanie dokonywać w systemie.

* **Gość** – Jest niezalogowanym użytkownikiem systemu, posiada on możliwość oglądania produktów, dodawania ich do koszyka, kupowania produktów po wprowadzeniu poprawnych informacji. Dodatkowo każdy gość ma dostęp do informacji znajdujących się w wiki. Gość ma możliwość rejestracji, po udanej rejestracji i logowaniu gość staje się klientem.
* **Klient** – Klient jest zalogowaną osobą posiadającą konto. Posiada on wszystkie możliwości Gościa oraz rozszerza niektóre z nich. Klient ma dostęp do historii kupionych produktów, jego aktualny koszyk jest zapisywany na serwerze, przez co dostępny jest on z wielu urządzeń.
* **Pracownik** – Jest osobą która dodaje/modyfikuje/usuwa produkty oraz informacje dostępne w systemie. Jest osobą, która fizyczne realizuje zamówienia klientów i modyfikuje ich status w systemie.
  1. Wymagania funkcjonalne

Tabela 1 Wymagania funkcjonalne aplikacji

|  |  |  |
| --- | --- | --- |
| **Wymagania funkcjonalne** | | |
| Id | Nazwa | Opis |
| **Gość** | | |
| FU01 | Dostęp do informacji o kwiatach | Gość ma dostęp do szczegółowych informacji o kwiatach znajdujących się w wiki |
| FU02 | Dostęp do produktów | Gość ma dostęp do produktów sprzedawanych w aplikacji |
| FU03 | Dodaj produkt do koszyka | Gość ma możliwość dodania produktów do koszyka |
| FU04 | Usuń produkt z koszyka | Gość ma możliwość usunięcia produktu z koszyka |
| FU05 | Złóż zamówienie | Gość może złożyć zamówienie na produktu znajdujące się w koszyku po poprawnym wypełnieniu wymaganych informacji |
| FU06 | Zarejestruj się | Gość ma możliwość utworzenia konta |
| FU07 | Zaloguj się | Gość ma możliwość zalogowania się do systemu sprzedaży pod warunkiem, że posiada on zarejestrowane konto. Gość po udanym zalogowaniu się staje się Klientem |
| **Klient** | | |
| FU08 | Dostęp do historii | Klient ma dostęp po historii swoich wcześniejszych zakupów |
| FU09 | Modyfikuj dane personalne | Klient ma możliwość modyfikacji danych personalnych wprowadzonych podczas procesu rejestracji |
| FU10 | Wyloguj się | Klient ma możliwość wylogowania się z systemu sprzedaży po poprawnym wylogowaniu staje się Gościem |
| **Pracownik** | | |
| FU11 | Zaloguj się do systemu | Pracownik ma możliwość zalogowanie się do systemu |
| FU12 | Wyloguj się z systemu | Pracownik ma możliwość wylogowania się z systemu |
| FU13 | Dodaj wpis | Pracownik ma możliwość dodania wpisu do wiki |
| FU14 | Modyfikuj wpis | Pracownik może modyfikować istniejący wpis na wiki |
| FU15 | Usuń wpis | Pracownik ma możliwość usunięcia wpisu z wiki |
| FU16 | Dostęp do produktu | Pracownik ma |
| FU17 | Dodaj produkt | Pracownik może dodać nowy produkt |
| FU18 | Modyfikuj produkt | Pracownik może modyfikować istniejący produkt |
| FU19 | Usuń produkt | Pracownik może usunąć produkt |
| FU20 | Spełnij zamówienie | Pracownik może zrealizować zamówienie, jednocześnie zmieniając mu status |
| FU21 | Oblicz przychody | Pracownik ma podgląd na przychody w danym miesiącu |
| FU22 | Przeglądaj magazyn | Pracownik ma wgląd w liczność produktów |

* 1. Wymagania niefunkcjonalne

|  |  |  |
| --- | --- | --- |
| **Wymagania niefunkcjonalne** | | |
| Id | Nazwa | Opis |
| NFU01 | Obsługa wielu użytkowników | Aplikacja powinna być w stanie obsłużyć do 30 jednoczesnych użytkowników |
| NFU02 | Przejrzysty interfejs użytkownika | Aplikacja powinna posiadać jednolity, intuicyjny interfejs użytkownika |
| NFU03 | Aplikacja powinna być dostępna z poziomu przeglądarki | Aplikacja powinna wspierać najpopularniejsze przeglądarki, mobilne i desktopowe |
| NFU04 | Aplikacja powinna być skalowalna | Dodawanie nowych sunk |
| NFU05 | Serwer wymaga stałego połączenia z Internetem |  |
| NFU06 | Użytkownik aplikacji musi posiadać połączenie z Internetem |  |
| NFU07 | Aplikacja powinna być odporna na ataki |  |
|  |  |  |

* 1. Przyjęte założenia projektowe

1. Wykorzystane technologie i narzędzia programistyczne

Ten rozdział pokazuje oraz opisuje technologie i narzędzia, które były użyte podczas tworzenia aplikacji.

* 1. JDK

JDK jest paczką darmowego oprogramowania firmy Sun Microsytems (obecnie firma należy do Oracle Corporation), w której skład wchodzą trzy elementy: JVM, JRE, narzędzia programistyczne Javy (kompilator, debugger, generator dokumentacji itd.). Produkt ten jest skierowany głównie do programistów tworzących oprogramowanie w języku Java. Java korzysta z maszyny wirtualnej, oznacza to, że oprogramowanie napisane w tym języku może zostać uruchomione na każdym urządzeniu wspierającym maszynę wirtualną, bez potrzeby modyfikowania, ani ponownej kompilacji kodu. W tworzeniu aplikacji została użyta wersja JDK 1.8.

* 1. Eclipse

Jest to darmowe IDE firmy Eclipse Foundation dla profesjonalnych deweloperów tworzących oprogramowanie. Zintegrowane środowiska deweloperskie są świetnym narzędziem w kontekście wytwarzania oprogramowania. Zawierają one w sobie środowiska uruchomieniowe, edytory kodu oraz podpowiadanie składni. W znaczny sposób przyspieszają oraz ułatwiają pracę programisty. Eclipse wydawany jest w dwóch wersjach: standardowej oraz EE. Wersja EE zawiera w sobie kilka dodatków takich jak: integracja z gitem, obsługa mavena, edytory XML, JPA i wiele innych.

* 1. Spring

Spring Framework wydany po raz pierwszy w roku 2002 miał na celu eliminować wiele problemów związanych z EJB, takich jak: narzucony model programowania, duży nakład kodu do osiągnięcia niewielkiego efektu, wiele projektów korzystało tylko z niewielkiej części EJB. Spring nie narzuca żadnego modelu programowania. Głównymi cechami tego frameworka są inwersja kontroli (IoC) poprzez wstrzykiwanie zależności i programowanie aspektowe (AOP). Cechy te w znaczący sposób wpływają na przejrzystość tworzonego kodu oraz pozwalają go znacznie zredukować. …dopisać

* 1. Spring Boot

Spring Boot jest paczką konfiguracji/oprogramowania, rodzajem udogodnienia bazującym na springu, wydawany jest on w oddzielnych wersjach niż spring. Łączy on w sobie kilka elementów:

* Wbudowany serwer – Spring Boot posiada wbudowany serwer(domyślnie Tomcat), na którym aplikacja jest automatycznie uruchamiana, nie wymaga to od programisty instalacji żadnego dodatkowego oprogramowania do uruchomienia aplikacji.
* Automatyczną konfigurację – Do uruchomienia aplikacji nie jest potrzebna żadna dodatkowa konfiguracją, nadal istnieje możliwość konfigurowania wielu elementów Springa.
* Szybkość – Spring Boot zawsze dostarcza kompatybilne ze sobą wersje Springa, pozwala to uniknąć problemów związanych z kompatybilnością pakietów. Dodatkowo w Spring Bootowych aplikacjach możemy znaleźć plik konfiguracyjny application.properties w którym możemy modyfikować różne aspekty konfiguracji bez potrzeby używania XMLa.
  1. Maven

Apache Maven jest narzędziem ułatwiającym tworzenie projektów w Javie. Udostępnia on jednolity system budowy projektów, jak i narzuca dobre praktyki programowania. Maven Posiada możliwości automatyzacji tworzenia oprogramowania. Budowa jest oparta na informacjach zawartych w pliku POM.xml. Plik POM.xml zawiera:

* Informacje o projekcie – nazwa, id grupy, id projektu, wersję projektu, URL projektu
* Zależności – wymagane pliki jar, które możemy pobrać z repozytorium
* Pluginy – wtyczki np. failsave (uruchamia testy JUnit), install(instaluje zależności w lokalnym repozytorium), compiler (kompiluje pliki Javy), których czynności zostają wykonane podczas budowy projektu
  1. Git

Git jest systemem kontroli wersji, służy on głównie po przechowywania oraz synchronizowania projektów. Git przechowuje projekty w centralnych repozytoriach, mogą one być lokalne bądź zdalne. Praca w gitcie jest oparta na komitach. Komit to stan plików(ich różnica w bajtach w stosunku do poprzedniego komita) wraz z komentarzem. Git pozwala nam na odtworzenie stanu naszego projektu do danego komita. Git składa się z gałęzi, początkową i główną jest gałąź master. Od istniejących gałęzi możemy dodawać dowolne nowe gałęzie i do nich przypisywać nasze komity. Gałęzie możemy ze sobą scalać łącząc ich zawartość.

* 1. Hibernate

Hibernate jest frameworkiem pozwalającym na automatyczne mapować obiekty javy na struktury danych bazy danych i struktury danych na obiekty. Hibernate spełnia specyfikację JPA. Programista dzięki Hibernate programista nie musi pisać zapytań SQL do bazy danych.

* 1. MySQL

Oprogramowanie MySQL udostępnia wielowątkowy, niezawodny, wieloużytknikowy serwer SQL. Serwer jest zaprojektowany do obsługi dużej ilości zapytań. Nadaje się do użycia w małych jak i dużych aplikacjach.

* 1. NPM

NPM(Node Package Manager) to zbiór narzędzi przydatnych w tworzeniu aplikacji w środowisku JavaScript. NPM w swojej zasadzie działania jest bardzo podobny do Mavena. Plikiem z informacjami o projekcie jest package.json którego zawartość musi być napisana w konwencji JSON. Repozytorium NPM zawiera ponad 800,000 paczek z kodem.

* 1. React.js
  2. Visual Studio Code

Visual Studio Code to IDE od firmy Microsoft.

* 1. Redux

Redux jest kontenerem stanu dla aplikacji napisanych w JavaScripcie. Pozwala na wyeliminowanie jednej z wad React.js, jaką jest niemożliwość przekazywania stanu do komponentu rodzica. Redux przechowuje stan w pojedynczym kontenerze nazwanym „store”. Stan możemy zmienić jedynie poprzez wywołanie akcji, w których zawarta jest informacja o zmianie. Akcje są przechwytywane przez reduktory, ich zawartość jest analizowana, a stan aplikacji jest zmieniany w zależności od danych zawartych w akcji.

* 1. Axios
  2. Material-UI
  3. MDBReact
  4. Pozostałe
* Xamp
* Visio
* Visual paradigm
* Postman
* Opera
* Word

1. Projekt systemu
   1. Architektura aplikacji

Aplikacja została stworzona w trójwarstwowym modelu.

* 1. Projekt bazy danych
     1. Model konceptualny
     2. Model logiczny
     3. Model fizyczny z ograniczeniami integralności danych

1. Implementacja systemu
   1. Back-End
   2. Front-End
   3. Projekt wybranych funkcji
   4. Struktura interfejsu graficznego
2. Testowanie wybranych funkcji systemu
3. Podsumowanie i wnioski

# Literatura

[1] M. Bickley, C. Slominski. A MySQL-based data archiver: preliminary results. Proceedings of ICALEPCS07, Paz. 2007. http://www.osti.gov/scitech/servlets/purl/922267 [dostęp dnia 20 czerwca 2015].

[2] J. Jędrzejczyk, B. Sródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

# Dodatek A