



# INTERNETOVÝ OBCHOD NA BÁZI SERVLETŮ

## Bakalářská práce

*Studijní program:* B2341- Strojírenství  
*Studijní obor:* 2301R030 Výrobní systémy  
*Autor práce:* **Vo Tran Tai**  
*Vedoucí práce:* Ing. Michal Moučka, Ph.D.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: **Tai VO TRAN**  
Studijní program: **B2341 Strojírenství**  
Obor: **2301R030 Výrobní systémy**  
Zaměření: **Inženýrská informatika**

Ve smyslu zákona č. 111/1998 Sb. o vysokých školách se Vám určuje bakalářská práce na téma:

### Internetový obchod na bázi servletů

#### **Zásady pro vypracování:**

(uved'te hlavní cíle bakalářské práce, doporučené metody pro vypracování)

1. Nastudujte problematiku vývoje dynamických webových aplikací na bázi servletů a JSP s využitím doporučené literatury.
2. Navrhněte strukturu univerzální aplikace pro podporu internetového obchodu.
3. Aplikaci naprogramujte. Jako vývojového prostředku využijte programovací jazyk Java EE ve verzi 7.
4. Aplikaci nainstalujte na některém z volných serverů. Ověřte jeho funkčnost.



Forma zpracování bakalářské práce:

- průvodní zpráva: cca 35 stran + přílohy
- grafické práce: dle potřeby

Seznam literatury (uved'te doporučenou odbornou literaturu):


- [1] CASTRO, E. - HYSLOP, B.: *HTML5 a CSS3* (české vydání). Brno: Computer Press, 2012
- [2] HALL, M.: *Java servlety a stránky JSP* (české vydání). Praha: Neocortex, 2011
- [3] HEROUT, P.: *Učebnice jazyka Java* (5. vydání). České Budějovice, KOPP nakladatelství, 2010
- [4] RYAN, S.: *Naučte se SQL za 28 dní* (české vydání). Brno: Computer Press, 2010
- [5] SEDLÁK, M. - MIKULÁŠKOVÁ, P.: *Jak vytvořit úspěšný a výdělečný internetový obchod*. Brno: Computer Press, 2012
- [6] SCHILDT, H.: *Java 7 – Výukový kurz* (české vydání). Brno: Computer Press, 2012

Vedoucí bakalářské práce:

Ing. Michal Moučka, Ph.D.

Konzultant bakalářské práce:



  
prof. Ing. Miroslav Olehla, CSc.  
vedoucí katedry

  
prof. Dr. Ing. Petr Lenfeld  
děkan

V Liberci dne 28. 2. 2014

## **Prohlášení**

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářskou práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

.....

V Liberci, 21.května 2014

Vo Tran Tai

**Poděkování:**

Rád bych poděkoval panu Ing. Michal Moučka, Ph.D. za vedení mé bakalářské práce, jeho odborné konzultace a připomínky.

Dále bych rád poděkoval v neposlední řadě mé rodině za podporu během studia.

Vo Tran Tai

**TÉMA : INTERNETOVÝ OBCHOD NA BÁZI SERVLETŮ**

**ABSTRAKT :** Stavební webové aplikace s funkcemi pro splnění základních požadavků při provádění on-line transakcí. Tato aplikace je založena na platformě Java 2 Enterprise Edition a využívá technologie, které jsou její součástí, zejména servlety a stránky jsp. Webové rozhraní této aplikace respektuje standardy html a css. Zabezpečení používá autentifikaci na základě uživatelského jména a hesla. Datová vrstva aplikace je tvořena relační databází.

**KLÍČOVÁ SLOVA:** Internetový obchod, Java, JSP, servlet, J2EE.

**THEME : SERVLET BASED E-SHOP**

**ABSTRACT:** Creating a web applications with the roles to meet of the essential requirements in the on-line transactions. This application is based on Java 2 Enterprise Edition Platform and uses technology, mainly servlets and JSP pages, which are its components. The Interface of this application respects the HTML and CSS standards. Security is based on authentication of username and password. Data layer of the application consists of a relational database.

**KEYWORDS:** E-shop, Java, JSP, servlet, J2EE.

**Rozsah práce a příloh:**

Počet stran : 44

Počet tabulek : 8

Počet příloh : 2

Počet obrázků : 12

**Datum:** 21.5.2014

# Obsah

|  |           |
|--|-----------|
| <b>1 Použité technologie.....</b>                      | <b>10</b> |
| 1.1 HTML .....   | 10        |
| 1.2 Kaskádové styly (CSS).....                         | 14        |
| 1.3 JSP a servlet .....                                | 14        |
| 1.3.2 JSP (Java Server Pages).....                     | 15        |
| 1.3.3 Servlet .....                                    | 17        |
| 1.4 Relační database .....                             | 21        |
| 1.4.1 Jazyk SQL (Structured Query Language).....       | 21        |
| 1.4.2 MySQL Community Server.....                      | 22        |
| 1.5 JDBC (Java Database Connectivity) .....            | 22        |
| <b>2 Webová aplikace Internetový obchod.....</b>       | <b>25</b> |
| 2.1 Požadavky.....                                     | 25        |
| 2.2 Návrh.....   | 25        |
| 2.2.1 Model.....                                       | 27        |
| 2.2.2 Pohled (View).....                               | 31        |
| 2.2.3 Řadič (Controller).....                          | 31        |
| <b>3 Uživatelská příručka.....</b>                     | <b>32</b> |
| 3.1 Požadavky na systém.....                           | 32        |
| 3.2 Nástroje k použití : .....                         | 32        |
| 3.3 Správa aplikace .....                              | 33        |
| 3.4 Používání aplikace zákazníkem .....                | 37        |
| <b>Závěr.....</b>                                      | <b>40</b> |
| <b>Seznam Literatury .....</b>                         | <b>41</b> |
| <b>Přílohy .....</b>                                   | <b>43</b> |
| Příloha A - Obsah CD.....                              | 43        |
| Příloha B - Instalace a nasazení webové aplikace ..... | 44        |

## Seznam použitých zkratk

|      |  |
|------|--|
| API  | <i>Application Programming Interface</i> - Označuje v informatice rozhraní pro programování aplikací.                                      |
| CSS  | <i>Cascading Style Sheets</i> - Kaskádové styly, jazyk pro popis stylu zobrazení dokumentů napsaných v jazycích HTML, XHTML nebo XML.      |
| DAO  | <i>Data Access Objects</i> - Objekt pro přístup k datům(povětšinou v databázi).  |
| DTO  | <i>Data transfer object</i> .  |
| DDL  | <i>Data Definition Language</i> - Jazyk pro definici dat.  |
| DML  | <i>Data Manipulation Language</i> - Jazyk pro manipulaci s daty.   |
| HTTP | <i>HyperText Transfer Protocol</i> - Protokol použitý webovými servery a klienty pro odesílání požadavků a tvorbu odezvy.                  |
| HTML | <i>Hypertext Markup Language</i> - Hypertextový jazyk značek.  |
| J2EE | <i>Java 2 Platform, Enterprise Edition</i> - Součást platformy Java určená pro vývoj a provoz podnikových aplikací a informačních systémů. |
| JDBC | <i>Java DataBase Connectivity</i> - Rozhraní umožňující jednotný přístup k různým relačním databázím.                                      |
| JDK  | <i>Java Development Kit</i> - Obsahuje soubor základních nástrojů pro vývoj aplikací pro platformu Java.                                   |
| JSP  | <i>JavaServer Pages</i> - HTML stránky používající jazyk Java pro generování dynamického obsahu.   |
| MVC  | <i>Model-View-Controller</i> - Softwarová architektura rozdělující vrstvy aplikace.  |
| SQL  | <i>Structured Query Language</i> - Strukturovaný dotazovací jazyk.   |
| URL  | <i>Uniform resource locator</i> - Adresa určující umístění dokumentu na internetu.   |
| WAR  | <i>Web application archive file</i> - Jsou formátu komprimované soubory.   |
| XML  | <i>Extensible Markup Language</i> - Rozšiřitelný značkovací jazyk.   |



# Úvod

Prodej zboží prostřednictvím webových stránek je velmi častým využitím Internetu a technologie WWW. Pro zajištění funkcí, které s takovým prodejem souvisí, se používají nejrůznější druhy webových aplikací, od jednoduchých stránek založených pouze na HTML, které obsahují pouze kontaktní informace na prodejní místa, po komplexní podnikové aplikace, které kromě prezentace výrobků a jejich prodeje zajišťují i nejrůznější další funkce a mohou být propojeny s podnikovými informačními systémy.

Jednou z nejrozšířenějších technologií pro vývoj serverových aplikací je Java 2 Enterprise Edition (J2EE), která využívá výhod platformy Java (multiplatformnost, rozsáhlá standardní knihovna, správa paměti) a přidává rozsáhlou podporu pro tvorbu a provozování aplikací spouštěných na serverech (z technologií použitých v této bakalářské práci jsou to servlety, stránky JSP, JDBC, JavaMail).

Tato bakalářská práce si klade za cíl vytvořit webovou aplikaci v J2EE, která bude zajišťovat prezentaci a prodej zboží pro malou firmu. Jako příklad prodávaného zboží slouží elektronika a telefon.

V první kapitole uvádím přehled technologií a nástrojů, které jsem při tvorbě internetového obchodu použil.

Ve druhé kapitole předkládám popis požadavků na aplikaci a návrhu aplikace.

Třetí kapitolu potom tvoří uživatelská příručka pro práci s touto aplikací.

# 1 Použité technologie

## 1.1 HTML

HTML je HyperText Markup Language, neboli hypertextový jazyk značek. Je hlavním z jazyků pro vytváření stránek v systému World Wide Web(www.) který umožňuje publikaci dokumentů na Internetu. Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka. [3]

Nejkratší HTML dokument vypadá takhle:

```
<html>
<head> <title> Titulek stránky </title> </head>
<body bgcolor="Barva pozadí" text="Barva textu" \
      link="Barva odkazů">
...text dokumentu...
</body>
</html>
```

Jako značkovací jazyk je nazýván z důvodu své charakteristiky, která spočívá v používání značek a jejich atributů. Níže následuje podrobný rozpis HTML (značka spíše pro pokročilé) : [4][5][6]

Struktura dokumentu : Každý HTML soubor by měl obsahovat několik základních značek pro hlavičky a těla. Tím se zaručí, že všichni klienti (zejm. prohlížeče) pochopí, o co v dokumentu vlastně jde.

Tabulka 1.1: HTML značky v Struktura dokumentu

| Značka                      | Význam                         | Párový | Výskyt                       |
|-----------------------------|--------------------------------|--------|------------------------------|
| <code>&lt;html&gt;</code>   | Začátek HTML dokumentu         | Ano    | Na začátku souboru           |
| <code>&lt;head&gt;</code>   | Hlavička stránky               | Ano    | Na začátku souboru           |
| <code>&lt;body&gt;</code>   | Tělo stránky + definice pozadí | Ano    | Za <code>&lt;head&gt;</code> |
| <code>&lt;!-- --&gt;</code> | Poznámka                       | Ano    | Kdekoliv                     |

Úprava textu: Určují vzhled textu při zobrazení. Zde je seznam nejpoužívanějších značky:

Tabulka 1.2: HTML značky pro Úprava textu

| Značka | Význam          | Párový |
|--------|-----------------|--------|
| <b>    | Tučné písmo     | Ano    |
| <i>    | Kurzíva         | Ano    |
| <u>    | Podtržení textu | Ano    |
| <font> | Písmo           | Ano    |

Bloky : Do přehledu bloků jsem kromě blokových prvků zahrnul všechny HTML značky, které způsobí zalomení řádky (kromě tabulek a seznamů).

Tabulka 1.3: HTML značky pro Bloky

| Značka   | Význam         | Párový    | značka | Význam           | Párový |
|----------|----------------|-----------|--------|------------------|--------|
| <p>      | Odstavec       | nepovinně | <h1>   | Nadpis 1. úrovně | Ano    |
| <br>     | Řádkový zlom   | Ne        | <h2>   | Nadpis 2. úrovně | Ano    |
| <div>    | Oddíl          | Ano       | <h3>   | Nadpis 3. úrovně | Ano    |
| <center> | Vycentrování   | Ano       | <h4>   | Nadpis 4. úrovně | Ano    |
| <hr>     | Vodorovná čára | Ne        | <h5>   | Nadpis 5. úrovně | Ano    |
|          |                |           | <h6>   | Nadpis 6. úrovně | Ano    |

Seznamy : Říká se tomu také odrážky a číslování. HTML zná také definiční výčty, které se používají velmi málo.

Tabulka 1.4: HTML značky pro Seznamy

| Značka | Význam           | Párový    |
|--------|------------------|-----------|
| <li>   | Položka seznamu  | Nepovinně |
| <ol>   | Číslovaný seznam | Ano       |
| <ul>   | Odrážkový seznam | Ano       |

Odkazky: Odkaz se v HTML dělá značkou <a>.

Příklad: <a href="http://www.seznam.cz">Seznam</a>

Obrázky: (<img>)

Obrázek (angl. image). Nepárová značka. Do stránky se vloží obrázek načtený z jiného souboru. Značka <img> má kromě obecných atributů spoustu dalších:

Tabulka 1.5: HTML atributů pro značky <img> (Obrázek)

| Atribut       | Význam                      | Hodnoty                          |
|---------------|-----------------------------|----------------------------------|
| <i>src</i>    | Umístění souboru s obrázkem | URL (Unique Resource Locator)    |
| <i>alt</i>    | Alternativní popis          | Libovolný text                   |
| <i>width</i>  | Šířka                       | Délka nebo procento              |
| <i>height</i> | Výška                       | Délka nebo procento              |
| <i>border</i> | Tloušťka rámečku            | Délka                            |
| <i>align</i>  | Zarovnání obrázku           | left, right, top, middle, bottom |

Tabulky : Tabulky jsou v HTML na strukturované zobrazování dat, používají se ale i na celkové rozvržení stránky (což je zastaralé). Podstatné jsou pouze značky <table>, <tr> a <td>.

Tabulka 1.6: HTML značky pro Tabulky

| Značka  | Význam                   | Párový    | Výskyt                          |
|---------|--------------------------|-----------|---------------------------------|
| <table> | Tabulka                  | Ano       |                                 |
| <tr>    | Řádek tabulky            | Nepovinně | <table>,<tbody>,<thead>,<tfoot> |
| <td>    | Buňka tabulky            | Nepovinně | <tr>                            |
| <th>    | Hlavičková buňka tabulky | Nepovinně | <tr>                            |
| <tbody> | Tělo tabulky             | Ano       | <table>                         |
| <thead> | Hlavička tabulky         | Ano       | <table>                         |
| <tfoot> | Patička tabulky          | Ano       | <table>                         |

Formuláře : Formuláře najdete téměř na každém webu. Slouží například pro vyhledávání, pro uživatelské komentáře, pro komunikaci mezi tvůrci webu a uživateli nebo také pro přihlášení uživatele atd.

Tabulka 1.7: HTML značky pro Formuláře

| Značka                        | Význam             | Párový | Výskyt  |
|-------------------------------|--------------------|--------|---|
| <code>&lt;form&gt;</code>     | Formulář           | Ano    |   |
| <code>&lt;input&gt;</code>    | Vstupní pole       | Ne     | <code>&lt;form&gt;</code>                                   |
| <code>&lt;select&gt;</code>   | Výběrové pole      | Ano    | <code>&lt;form&gt;</code>                                   |
| <code>&lt;option&gt;</code>   | Volba              | Ne     | <code>&lt;select&gt;</code> , <code>&lt;datalist&gt;</code> |
| <code>&lt;textarea&gt;</code> | Velké vstupní pole | Ano    | <code>&lt;form&gt;</code>                                   |
| <code>&lt;label&gt;</code>    | Popis pole         | Ano    | <code>&lt;form&gt;</code>                                   |
| <code>&lt;button&gt;</code>   | Tlačítko           | Ano    | <code>&lt;form&gt;</code>                                   |

Hlavička : HTML hlavička (značka `<head>`) obsahuje značky, jejichž obsah se na stránce nezobrazí, ale jsou důležité.

Tabulka 1.8: HTML značky pro Hlavičku

| Značka                     | Význam                | Párový | Výskyt                    |
|----------------------------|-----------------------|--------|---------------------------|
| <code>&lt;title&gt;</code> | Titulek stránky       | Ano    | <code>&lt;head&gt;</code> |
| <code>&lt;link&gt;</code>  | Nezobrazovaný odkaz   | Ano    | <code>&lt;head&gt;</code> |
| <code>&lt;meta&gt;</code>  | Informace o dokumentu | Ne     | <code>&lt;head&gt;</code> |

Obecné atributy : Atributy použitelné u skoro každého HTML značky.

+ *Class* : Třída elementu používaná pro CSS. Umožňuje jednoduché označení elementu, který má mít jiný vzhled. V zápisu stylu se ke jménu třídy předřazuje tečka. Příklad:

```
<style>
.cervene {color: red}
</style>
...
<p class="cervene">Odstavec červeným písmem</p>
```

+ *Id* : Identifikátor, který by měl být jednoznačný. Používá se pro CSS a skripty. V zápisu CSS se jménu identifikátoru předřazuje křížek # . Příklad:

```
<style>
#mrnavej {font-size: 5px}
</style>
...
<p id="mrnavej">Odstavec děsně mrňavým písmem.</p>
```

## 1.2 Kaskádové styly (CSS)

**Kaskádové styly** (v anglickém originále Cascading Style Sheets se zkratkou CSS) je jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML nebo XML. [7]

**Syntax:** Definice kaskádových stylů sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarací. Každý blok deklarací pak obsahuje seznam deklarací oddělených středníky ; a každá deklarace sestává z identifikátoru vlastnosti, následuje dvojtečka : a hodnota vlastnosti. [7]

Příklad pravidla:

```
body{background-color: white; color: black; padding: 10px;}
```

Celý blok nazveme pravidlo, „ body “ je selektor, část v závorkách je blok deklarací, řádek „ background-color: white;“ je deklarace samotná, „ background-color“ je identifikátor vlastnosti a „white“ její hodnota. Celý kód pak nastavuje barvu pozadí stránky na bílou, barvu textu na černou a okraj na 10 pixelů.[7]

- Selektory : CSS definuje mnoho různých selektorů, které obvykle můžeme kombinovat. Mezi základní patří:

- *body* – Tyto deklarace budou platit pro všechny výskyty elementu body.
- *.trida* – Tyto deklarace budou platit pro všechny elementy, které mají v HTML nastavenou třídu *trida*. To se provádí pomocí HTML atributu *class*.
- *#id* – Tyto deklarace budou platit pro všechny elementy, které mají v HTML nastavený identifikátor id. To se provádí pomocí HTML atributu *id*. [7]
- Připojení kaskádových stylů do HTML stránky : Existuje několik možných způsobů, jak aplikovat kaskádové styly v HTML dokumentu; v praxi se nejčastěji používá odkaz na externí soubor.

• Zápis stylů do elementu style. Takové styly se aplikují na celou stránku podle předepsaných selektorů. Příklad:

```
<style type="text/css">
#hlavicka{width: 200px; height: 450px;}
</style>
```

• Připojení externího souboru pomocí elementu link. Příklad:

```
<head>
  <link rel='stylesheet' href='soubor.css' type='text/css'>
</head> [7]
```

## 1.3 JSP a servlet

### 1.3.2 JSP (Java Server Pages)

JSP (Java Server Pages) je technologie, která umožňuje vytvářet dynamické webové stránky pomocí značek podobně jako statické HTML. Stránka se potom převádí na třídu servletu.

JSP umožňují vývojářům a návrhářům webu rychlý vývoj a jednoduchou údržbu bohatých a dynamických webových stránek. Stejně jako ostatní Java produkty, jsou všechny kódy platformně nezávislé. JSP technologie odděluje uživatelské rozhraní od generování obsahu, což umožňuje návrhářům měnit celé stránky bez toho, aniž by museli měnit již vytvořený dynamický obsah.

Skriptovací značky JSP:

Komentář: `<%-- komentář --%>` či `<!-- komentář -->`

Komentuje JSP stranu, ale není vkládán do odpovědi.

Deklarace: `<%! int i = 0; Ctverec a = new Ctverec (5,3); ...%>`

Deklaruje jednu nebo více proměnných nebo metod, které můžete používat v pozdějším Java kódu v JSP. Proměnné je nutné deklarovat před použitím v JSP, tyto jsou pak přístupné v celém dokumentu včetně vložených vnějších zdrojů (*include*). Proměnné a metody importované z balíčků jsou již deklarované.

Výraz: `<h1>Hodnota a+5 je <%= a+5 %> </h1>`

Výrazový element musí obsahovat validní výraz skriptovacího jazyka, který je vyhodnocen zleva doprava a převeden na String, takže lze výraz použít jako část textu, který může být označován HTML značky. V elementu se nepoužívá středník k uzavření výrazu.

Includování:

Dvě možnosti

- `<%@ include file="header.jsp" %>`
  - V podstatě vložení JSP kódu z daného souboru.
  - Provádí staticky se při kompilaci!
- `<jsp:include page="header.jsp" />`

- Provedení kódu z daného souboru.
- Provádí se dynamicky při zpracování každého požadavku!

### Skriptlet :

Jsou části kódu v jazyku Java, které jsou při překládání stránky převedeny na odpovídající kód ve výsledném servletu. Kód skriptletu se musí nacházet mezi značkami `<%` a `%>`. Pro zjednodušení práce mají vývojáři při psaní skriptletů k dispozici osm implicitních objektů, které poskytují přístup k některým důležitým informacím:

|               |  |
|---------------|--|
| out :         | Výstupní proud, do kterého lze zapisovat.                              |
| request :     | Informace o požadavku.   |
| response :    | Informace o odezvě.  |
| session :     | Informace o sezení (relaci).   |
| application : | Informace dostupné v rámci celé aplikace.                              |
| config :      | Informace o konfiguraci servletu.                                      |
| pageContext : | Kontext stránky umožňující manipulaci se stránkou a objekty stránky.   |
| Page :        | Reprezentuje instanci servletu, který bude vygenerován z této stránky. |

### Příklad:

```
<%
String name = null;
if (request.getParameter("name") == null)
{
%>
<%@ include file="chyba.html" %>
<%
}else{
name = request.getParameter("name");
out.println ("Jméno zní: „+name);
}
%>
```

Skriptlet může obsahovat množství příkazů, proměnných apod. Uvnitř skriptletu lze:

- + Deklarovat proměnné či metody použitelné dále v JSP – viz značka deklarace
- + Psát výrazy v platném skriptovacím jazyce
- + Použít příkaz, který je platný kdekoli na JSP stránce (napr. příkazový blok po if).



### 1.3.3 Servlet

Servlety jsou programy v jazyce Java, které je spouštějí na straně serveru, používají se pro dynamické vytváření dokumentů a interakci s nimi prostřednictvím určitého protokolu. Vytvářenými dokumenty mohou být webové stránky v jazyku HTML nebo dokumenty XML, prosté textové dokumenty a jiné. V naprosté většině se však servlety používají pro vytváření dynamických webových stránek a pro řízení toku webových aplikací. Protokolem, který zprostředkovává interakci se servlety, je obvykle HTTP<sup>[19]</sup> (Hypertext Transfer Protocol). Lze použít i jiný protokol, avšak na Internetu se obvykle používá právě jen HTTP a pro jiné protokoly většinou ani neexistují implementace kontejneru servletů. V dalším textu budeme hovořit již jen o servletech pracujících nad protokolem HTTP.

Struktura servletu : Standardní servlet dělí od třídy *HttpServlet* a jeho nejčastější akcí je předefinování metody *doGet* nebo *doPost*. Tyto metody zpracovávají požadavky *GET* a *POST*. Požadavky *GET* jsou implicitním typem požadavků prohlížeče webových stránek. Požadavky typu *POST* se obvykle vytvářejí v případě, když uživatel odešle HTML formulář s nastavením *METHOD* = "*POST*". Jak *doGet* tak *doPost* mají dva parametry, jimiž jsou instance rozhraní *HttpServletRequest* a *HttpServletResponse*. První z nich specifikuje vlastní požadavek, který dostaneme z webového prohlížeče. Jsou v něm uložena všechna data, která se požadavku týkají (např. Formulářová data). Druhý je zástupcem odezvy, kterou vytváříme (sem budeme např. zapisovat výslednou stránku a nastavovat stavové kódy). Obě metody mohou vyhodit dvě výjimky a proto je požadováno, abychom je zahrnuli do deklarace.

Příklad 1 – HelloWorld.java

```
package votrantai;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title></title>");
        }
```

```

        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet HelloWorld </h1>");
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
    }
}

public void doPost(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}
}

```

Příklad 1 uvádí zdrojový text servletu, který vytvoří HTML stránku, jejímž obsahem bude zformátovaný nápis HelloWorld.

Hned první řádek obsahuje definici balíčku. Balíčku slouží k tomu, abychom logicky strukturovali svoje zdrojové kódy a oddělili od ostatních. Servlety nemusí být v balíčcích, ale je dobré tento postup používat, protože se tím vyhneme problémům s volbou názvů.

Další tři řádky určují importované knihovny (balíčky tříd) *java.io* se zde používá mj. pro *PrintWriter*, *javax.servlet* např. pro *HttpServlet* a *javax.servlet.http* pro *HttpServletRequest* a *HttpServletResponse*.

Následuje deklarace třídy, která je potomkem třídy *HttpServlet*. Tím vlastně určujeme, že jde o servlet. Pokud chceme, aby náš servlet reagoval na požadavky zaslané prohlížečem, musíme předefinovat metodu *doGet*, která odpovídá na akci *GET*. Pokud budeme chtít, aby servlet odpovídal stejným způsobem i na požadavky *POST*, můžeme jednoduše předefinovat metodu *doPost* a z ní vyvolat *doGet*. Samozřejmě to lze zapsat i opačně.

Na dalším řádku získáme instanci objektu *PrintWriter* z proměnné *response*. Tato instance reprezentuje výstupní datový tok. Tedy slouží k ukládání námi vytvořené výstupní stránky. Jak je vidět dále, to se provádí metodou *println*.

### Životní cyklus servletu probíhá takto:

1. Inicializační část (volá se jen jednou):

a) vyvoláním servletu (např. zadáním URL v prohlížeči) se vytvoří jeho instance v paměti.

b) vyvolání metody *init* - zde dochází k inicializaci.

2. Při každém uživatelském požadavku:

a) vytvoření nového vlákna;

b) vyvolání metody *service* - zjišťuje typ požadavku, podle něhož se následně zavolá některá z metod *doGet*, *doPost*, apod.

3. Odstranění instance servletu z paměti ( ještě než k odstranění dojde, je vyvolána metoda *destroy*):

a) v případě dlouhé nečinnosti servletu.

b) při odstranění administrátorem

### Sledování sezení (session):

HTTP je bezstavový protokol, což znamená, že si neuchovává žádné informace o stavu klienta, tudíž každý klientský požadavek je zcela samostatný a nijak nesouvisí s tím předchozím. Což je v mnoha případech velmi nepříjemné (např. autentifikace). Servlet poskytují způsob řešení tohoto problému. Nazývá se sledování sezení a je k tomu použito rozhraní *HttpSession*. Objekt třídy *HttpSession* vytvoříme metodou *HttpServletRequest.getSession*, jako parametr jí předáme konstantu *true*, která zajistí, že pokud session spojená s tímto klientem neexistuje, bude vytvořena. Takže váš první krok vypadá obvykle následovně :

```
HttpSession session = request.getSession(true) ;
```

Když už máme objekt rozhraní *HttpSession*, můžeme z něj číst hodnoty zadaných atributů metodami: [2]

- **Public Object getValue(String name);**

**Public Object getAttribute(String name)**

Tyto metody extrahují z objektu sezení již uloženou hodnotu. Vrací *null* v případě, že s daným názvem není sdružena žádná hodnota.

- **Public void putValue(String name, Object value);**

**Public void setAttribute(String name, Object value)**

Tyto metody sdružují hodnotu s názvem. Implementuje

- **Public void removeValue(String name);**  
**Public void removeAttribute(String name)**  
Tyto metody odstraňují všechny hodnoty asociované s označeným názvem.
- **Public String[] getValueNames();**  
**Public Enumeration getAttributeNames()**  
Tyto metody vracejí názvy všech atributů v sezení.
- **Public String getId()**  
Tato metoda vrací jedinečný identifikátor vytvořený pro každé sezení.
- **Public boolean isNew()**  
Tato metoda vrací *true* v případě, když klient (prohlížeč) nikdy sezení nepoužil; jinak se vrátí *false*
- **Public long getCreationTime()**  
Tato metoda vrací čas, kdy bylo sezení poprvé vytvořeno (v milisekundách od roku 1970). Abychom dostali hodnotu užitečnou pro vytištění, předejte tuto hodnotu konstruktoru třídy *Date*
- **Public long getLastAccessedTime()**  
Tato metoda vrací čas, kdy bylo sezení naposledy odesláno uživatelem.
- **Public int getMaxInactiveInterval()**  
**Public void setMaxInactiveInterval(int seconds)**  
Tyto metody čtou nebo nastavují časový úsek bez přístupu (v sekundách), než sezení automaticky pozbude platnosti. Záporná hodnota udává, že sezení má neomezenou platnost.
- **Public void invalidate()**  
Tato metoda ruší platnost sezení a uvolňuje všechny objekty s ní sdružené.

Sezení typické využití :

- internetový obchod
- každý klient má svůj nákupní košík

## 1.4 Relační database

### 1.4.1 Jazyk SQL (Structured Query Language)

SQL zkratka anglických slov **Structured Query Language** (strukturovaný dotazovací jazyk), je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. Jazyk SQL se často dělí na následující součásti : [10]

- Data Definition Language slouží k vytváření, změnám a rušení tabulek. Do DDL patří příkazy CREATE TABLE, DROP TABLE a ALTER TABLE.

- Data Manipulation Language slouží k přidávání, změnám, rušení a čtení záznamů v tabulkách. Do DML jsou zařazovány příkazy INSERT, UPDATE, DELETE a SELECT.

+ INSERT: vkládá do databáze nová data, základní forma příkazu INSERT vypadá takto:

```
INSERT INTO <tabulka> [( <sloupec>[, ...n] ) ] VALUES  
( <hodnota>[, ...n] );
```

+ UPDATE : mění data v databázi (editace), UPDATE má následující syntaxi :

```
UPDATE <tabulka>  
  
SET <nazev_sloupecku> = <hodnota> [, ...n]  
  
[FROM <zdrojova tabulka (tabulky)>]  
  
[WHERE <podmínka>]
```

+ DELETE – odstraňuje data (záznamy) z databáze. Obecný formát příkazu je:

```
DELETE FROM <tab_name> [WHERE <condition>]
```

Kde :

- <tab\_name> je jméno datové tabulky, ze které budou záznamy odstraněny
- <condition> je logická podmínka, kterou mají splňovat odstraňované záznamy

+ SELECT – vybírá data z databáze, umožňuje výběr podmnožiny a řazení dat.

### 1.4.2 MySQL Community Server

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Velkou výhodou tohoto databázového systému je cena: je k dispozici zdarma. Nevýhodou pak mohou být částečné odchylky od standardu SQL a některé chybějící funkce. [11][12]

## 1.5 JDBC (Java Database Connectivity)

Je API(programové rozhraní) pro programátory v programovacím jazyku Java, které definuje jednotné rozhraní pro přístup k relačním databázím.

Pro přístup ke konkrétnímu databázovému serveru je potřeba JDBC driver (ovladač), který poskytuje tvůrce databázového serveru.

JDBC umožňuje paralelní použití několika implementací v jediné aplikaci. API poskytuje možnost dynamického načtení příslušných Java knihoven a jejich registraci pomocí JDBC Driver Manager. Driver Manager obsahuje tovární metodu pro vytvoření připojení s databází přes JDBC. Připojení k databázi přes JDBC umožňuje tvorbu a vykonání jednotlivých příkazů. Patří mezi ně SQL příkazy pro modifikaci dat: CREATE, INSERT, UPDATE a DELETE a také dotazovací příkaz SELECT. [13]

### ***Základní kroky při použití JDBC:*** [2]

Pro dotazování databází existuje sedm standardních kroků.

1. Zaved' ovladač JDBC
2. Definuji připojovací URL
3. Navaž spojení
4. Vytvoř si příkazový objekt
5. Proveď dotaz nebo aktualizaci
6. Zpracuj výsledky
7. Uzavři spojení

Zde je uvedeno několik podrobností o tomto procesu.

#### 1. Zavedení ovladače :

Ovladač je část software, která umí komunikovat s databázovým serverem. Vše, co potřebujete pro zavedení ovladače, je zavedení příslušné třídy; statický inicializátor v samotné třídě vytvoří automaticky instanci ovladače a registruje ji pomocí správce ovladačů JDBC.

Registrace ovladače u aplikace použít metodu *Class.forName*. Tato metoda požívá řetězec obsahující plně kvalifikovaný název třídy (tj. název, který zahrnuje názvy balíků) a zavádí odpovídající třídu. Zde je uveden příklad :

```
Class.forName("com.mysql.jdbc.Driver");
```

## 2. Definování připojovací URL

Pokud jste již zavedli ovladač JDBC, tak potřebujete specifikovat umístění databázového serveru. URL vztahující se na databáze obsahují jako protokol `jdbc` : a je v nich uložen serverový hostitel, port a název databáze (nebo reference). Přesný formát bude definován v dokumentaci, která se dodává s příslušným ovladačem, zde je ukázkové příklad :

```
String url = "jdbc:mysql://localhost:3306/database";
```

## 3. Navázání spojení

Pro skutečné navázání síťového spojení URL, uživatelské jméno pro danou databázi a heslo metodě `getConnection` třídy `DriverManager` tak, jak je uvedeno v následujícím příkladu. Poznamenejme, že `getConnection` může vyvolat výjimku `SQLException`, takže musíte použít blok `try/catch`. Zde je MySQL příklad připojení:

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    String url = "jdbc:mysql://localhost:3306/database";
    String uzivatel = "username";
    String heslo = "password";
    Connection connection =
        DriverManager.getConnection(url,uzivatel,heslo);
} catch (Exception ex)

{

}
```

## 4. Vytvoření příkazu

Objekt `Statement` se používá pro odesílání dotazů a příkazů databázi a je vytvářen na základě třídy `Connection` následovně :

```
Statement statement = connection.createStatement();
```

## 5. Provedení dotazu

Pokud již máte objekt `Statement`, můžete ho použít pro odeslání dotazů SQL pomocí metody `executeQuery`, které vrátí objekt typu `ResultSet`. Zde je uveden příklad :

```
String query = "SELECT col1, col2, col3 FROM sometable";  
  
ResultSet resultSet = statement.executeQuery(query);
```

Pro modifikaci databáze použijte `executeUpdate` místo `executeQuery` a dodejte řetězec, který používá `UPDATE`, `INSERT` nebo `DELETE`.

## 6. Zpracování výsledků

Použijte metodu `next` (třída `ResultSet`) pro získání nové řádky. Tato metoda prochází tabulku po jednotlivých řádcích. Uvnitř řádky pak `ResultSet` poskytuje různé metody `getXxx`, které dostanou index nebo název sloupce jako argument a vrátí výsledek jako různé typy jazyka Java. Například použijte `getInt` v případě, že hodnota je typu `Int`, `getString` pro typ `String` a podobně pro většinu dalších typ sloupce.

Důležité upozornění je první sloupec na řádce `ResultSet` má index 1, nikoliv index 0.

Zde je uveden příklad, který vytiskne hodnoty prvních tří sloupců na všech řádkách `ResultSet`.

```
while(resultSet.next()) {  
  
    System.out.println( resultSet.getString(1)+ " " +  
                        resultSet.getString(2)+ " " +  
                        resultSet.getString(3));  
  
}
```

## 7. Uzavření spojení

Pro uzavření spojení explicitním způsobem byste zavolali :

```
Connection.close() ;
```



## 2 Webová aplikace Internetový obchod

Tato kapitola popisuje požadavky kladené na webovou aplikaci Internetový obchod, návrh této aplikace na základě požadavku a implementaci návrhu.

### 2.1 Požadavky

Analýza požadavku je velmi důležitou součástí vývoje jakékoliv aplikace. Je to proces, při kterém se snažíme stanovit funkční a nefunkční požadavky. Funkční požadavky popisují služby, které by aplikace měla poskytovat svým uživatelem. Nefunkční požadavky určují omezující podmínky, za jakých bude aplikace pracovat.

Pro aplikace Internetový obchod byly stanoveny následující funkční požadavky:

- + Aplikace poskytne rozhraní, jehož prostřednictvím bude uživatel moci pohodlně prohlížet zboží a přidávat kupované zboží do nákupního košíku.
- + V uživatelském rozhraní bude možné vyhledávat produkty na základě názvu, klíčových slov v popisu produktu nebo případně dalších kritérií.
- + Aplikace poskytne možnost objednat zboží, které se nachází v nákupním košíku, na zadanou adresu.
- + Aplikace poskytne zákazníkům možnost zaregistrovat se pod uživatelským jménem a heslem.
- + Po uskutečnění objednávky bude zákazníkovi zasláno e-mailové potvrzení.
- + Součástí aplikace bude administrátorské rozhraní, které poskytne správci aplikace následující funkce:
  - Možnost přidat, změnit, hledat nebo odebrat druh prodávaného zboží
  - Možnost přidat, změnit nebo odebrat kategorii zboží
  - Možnost procházet případně rušit zadané objednávky

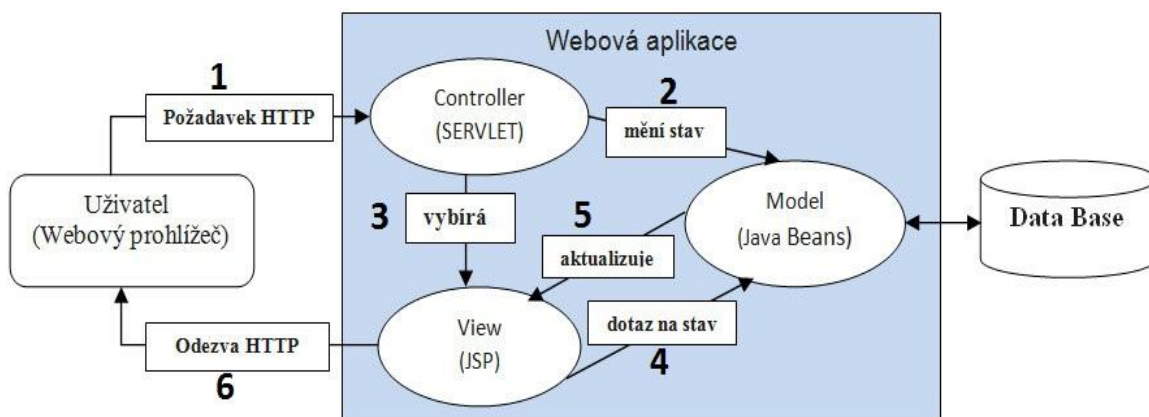
### 2.2 Návrh

Návrh aplikace používá návrhový vzor Model-View-Controller ( MVC nebo Model 2 ) . Vzor MVC dělí aplikaci na tři základní části : Model, pohled (View) a řadič (Controller). Schéma tohoto vzoru se nachází na obrázku 2.1. [18]

- + Model (model) : Je funkčním a datovým základem celé aplikace. Poskytuje prostředky jak pro přístup k datové základně a stavům aplikace, tak pro jejich ukládání a aktualizaci.

+ View (pohled) : Má na starosti prezentaci. Proto používá data z modelu prostřednictvím *Controlleru*. Také *view* získává vstup od uživatele a ten jde potom do *Controlleru*.

+ Controller (řadič) : Analyzuje vstup z requestu a rozhodne o tom, jaký má význam pro *model*. „Updatuje” model a posílá data do *view* vrstvy. [17][18]



Obrázek 2.1 : Schéma webové aplikace používající vzor Model-View-Controller.

### Činnost MVC aplikace

1. Uživatel provede nějakou akci v uživatelském rozhraní (např. stiskne tlačítko), tak prohlížeč pošle požadovaná data do Controlleru.
2. Controller (servlet) obdrží oznámení o této akci z objektu uživatelského rozhraní. Controller přistoupí k modelu a v případě potřeby ho mění stav na základě provedené uživatelské akce (např. mění stav nákupní košík uživatele).
3. Controller (servlet) předá dále požadavek klienta stránce JSP (View).
4. JSP (View) dotaz na stav modelu.
5. JSP (View) aktualizuje stav modelu
6. JSP (View) vrací stránku uživateli.

Uživatelské rozhraní čeká na další akci uživatele, která celý cyklus zahájí znovu. [18][17]

### 2.2.1 Model

Model ve vzoru MVC reprezentuje data v aplikaci a soubor pravidel, podle kterých se tato data chovají. Zahrnuje i vnitřní operace aplikace, které nemají přímý vliv na zobrazení dat.

V aplikaci Internetový obchod je model tvořen relační databází, balíkem tříd `Model.DAO` (třídy pro přístup k databázi) a částí balíku `Model.DTO`.

- Přístup k databázi se řídí návrhovým vzorem Data Access Object (DAO), v balíku `Model.DAO`, tento má java třídy :

+ `catalogDAO.java` : má metody

- *`selectCatalog()`* : vybrat kategorii zboží.
- *`addCatalog()`* : přidat kategorii zboží.
- *`deleteCatalog()`* : odstranit kategorii zboží.
- *`updateCatalog()`* : změnit kategorii zboží.

+ `productDAO.java` : má metody

- *`selectProduct()`* : vybrat zboží.
- *`selectProductByCatalog()`* : vybrat zboží podle kategorie.
- *`addProduct()`* : přidat zboží.
- *`deleteProduct()`* : odstranit zboží.
- *`updateProduct()`* : změnit zboží.
- *`findByLikeName()`* : hledat zboží podle názvu.

+ `customerDAO.java` : má metody

- *`addCustomer()`* : přidat zákazníka.
- *`selectCustomer()`* : vybrat zákazníka.
- *`selectCustomerById()`* : vybrat zákazníka podle kódu (Id).
- *`deleteCustomer()`* : odstranit zákazníka.
- *`maxId()`* : vybrat maximálního zákazníka Id.

+ `orderDAO.java` : má metody

- `addOrder()` : přidat objednávku.
- `selectOrderById()` : vybrat objednávku podle Id (zákazníka Id).

+ `userDAO.java` : má metody

- `addUser()` : přidat uživatele.
- `getUser()` : vybrat uživatele.
- `encryptPass()` : zašifrovat heslo.

- Částí balíku `Model.DTO` : Včetně Java třídy, tato třída používá metody (*get* a *set*) získat a nastavit atributy v databázi.

Příklad:

```
public class catalogDTO {  
    Integer catalogId; String catalogName;  
    public Integer getCatalogId() {  
        return catalogId; }  
    public String getCatalogName () {  
        return catalogName; }  
    public void setCatalogId(Integer catalogId) {  
        this.catalogId = catalogId; }  
    public void setCatalogName(String catalogName) {  
        this.catalogName = catalogName; } } }
```

- Vytvářet databáze :

a. Tabulky v databázi :

+ **catalog** : Obsahuje Informace o kategorii zboží.

- `catalogId` : kategorie kódů.
- `catalogName` : kategorie název.

+ **product** : Obsahuje Informace o zboží.

- *productId* : zboží kódů.
- *productName* : zboží název.
- *productPrice* : zboží cena.
- *productImage* : zboží obraz.
- *productDetail* : zboží detail.
- *productQuantity* : zboží množství.
- *catalogId* : kategorie kódů podle zboží.

+ **customer** : Obsahuje Informace o zákazníkovi

- *customerId* : zákazníkův kód.
- *email* : zákazníkův email.
- *telefon* : zákazníkovo číslo telefonní.
- *name* : zákazníkovo jméno
- *firstName* : zákazníkovo příjmení
- *street* : zákazníkova ulice
- *city* : zákazníkovo město.
- *zipCode* : zákazníkovo Poštovní Směrovací Číslo

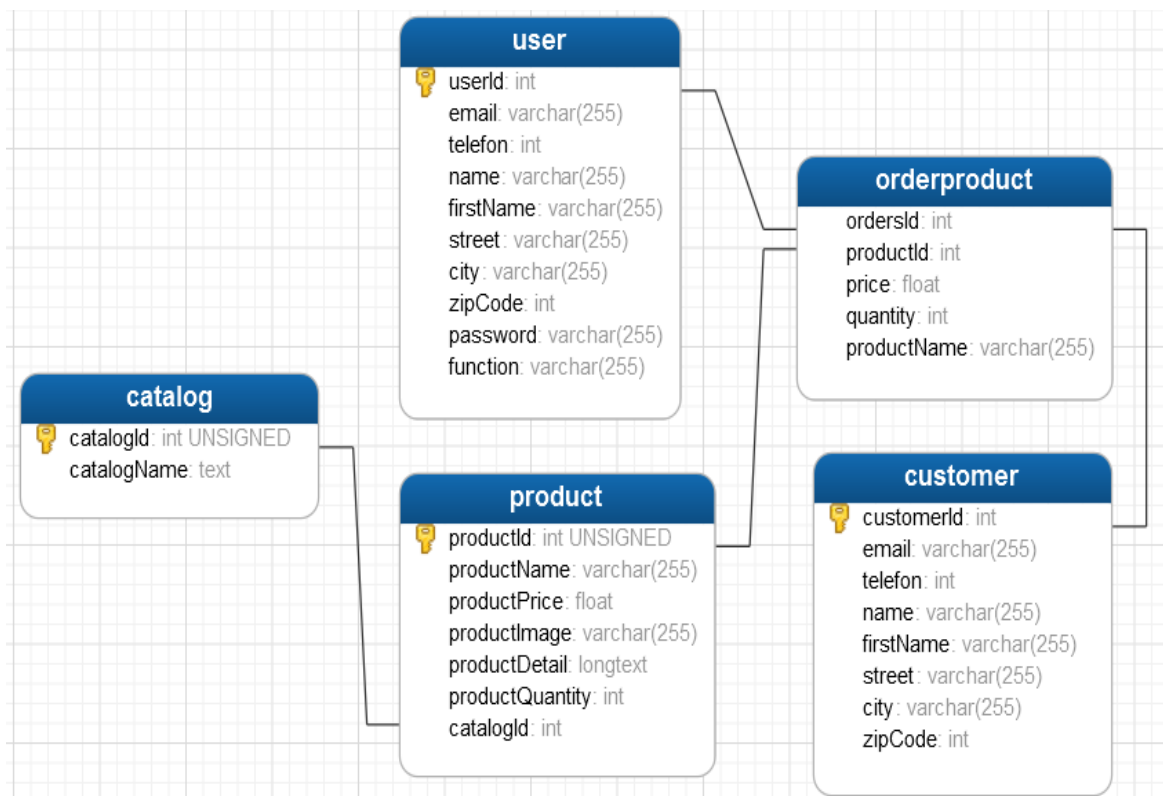
+ **orderproduct** : Obsahuje Informace o objednávce.

- *ordersId* : kód objednávky.
- *productId* : zboží kódů podle objednávky.
- *price* : cena zboží .
- *quantity* : množství zboží v objednávce.
- *productName* : název zboží

+ **user** : Obsahuje Informace o uživateli a administrátorovi.

- *userId* : uživatel kódů.
- *email* : uživatel emailu.
- *telefon* : číslo telefonní uživatele.
- *name* : jméno uživatele.
- *firstName* : příjmení uživatele.
- *street* : ulice uživatele.
- *city* : město uživatele.
- *zipCode* : Poštovní Směrovací Číslo uživatele
- *password* : heslo uživatele.
- *function* : funkční uživatel (Správce nebo uživatel )

b. Schéma relačních databází.



Obrázek 2.2: Navržené schéma databáze

### 2.2.2 Pohled (View)

Je uživatelské rozhraní, včetně aktuálních dat na obrazovce, poskytuje menu, tlačítka, editační pole, volby ..., takže uživatelé mohou přidávat, mazat, editovat, vyhledávat a provádět další operace na data v systému. Typicky zobrazené informace jsou převzaty z modelů. Pro tuto aplikaci slouží pohled stránky JSP.

### 2.2.3 Řadič (Controller)

Reaguje na události (typicky pocházející od uživatele) a zajišťuje změny v modelu nebo v pohledu.

Pro aplikaci Internetový obchod jsem vytvořil následující akce:

- addCatalogControl : přidat kategorii zboží  
(volání funkce z třídy `Model.DAO.catalogDAO`).
- addProductControl : přidat produkt  
(volání funkce z třídy `Model.DAO.productDAO`).
- deleteKatalogControl : Smaže kategorii  
(volání funkce z třídy `Model.DAO.catalogDAO`).
- deleteProductControl : Smaže produkt  
(volání funkce z třídy `Model.DAO.productDAO`).
- deleteCustomerControl : Smaže zákazníka  
(volání funkce z třídy `Model.DAO.customerDAO`).
- editCatalogControl : Načte kategorii a zobrazí stránku pro její úpravu.  
(volání funkce z třídy `Model.DAO.catalogDAO`).
- editProductControl : Načte produkt a zobrazí stránku pro jeho úpravu.  
(volání funkce z třídy `Model.DAO.productDAO`).
- registraceControl : registraci uživatele  
(volání funkce z třídy `Model.DAO.userDAO`).

- `LoginControl` : Pokusí se ověřit uživatele. Pokud uspěje, zapíše jej do objektu `HttpSession`  
(volání funkce z třídy `Model.DAO.userDAO`).
- `shoppingcartControl` : kontrola session, pokud neexistuje session "cart" (nákupní košík), tak se vytvoří nový nákupní košík používáním session. Provádí vložení zboží do košíku. Řízení předá stránce `shoppingcartPro.jsp`, která zobrazuje obsah košíku.
- `UpdateCartControl` : Změnit zboží, smazat zboží v nákupním košíku a koupit zboží a vložit do nákupního košíku
- `buyControl` : vybrat informaci o nakupujících, potom vyplnit do objednávky
- `SendMailControl` : poslat email pro zákazníka.

## 3 Uživatelská příručka

### 3.1 Požadavky na systém

Systém, na kterém je aplikace nainstalována, by měl obsahovat následující nástroje :

- + Java Development Kit (JDK) 1.3 nebo novější
- + Web Server jako Jrun, Apache Tomcat, GlassFish ...
- + Přístup k databázi MySQL verze 5.0 nebo novější. Aplikace může fungovat i se staršími verzemi MySQL, nicméně nebyla pro ně testována a mohou se proto vyskytovat chyby.
- + Webový prohlížeč

### 3.2 Nástroje k použití :

- NetBeans IDE
- Správa databáze
- + My SQL
- + Navicat



### 3.3 Správa aplikace

Administrátorské rozhraní lze vyvolat zobrazením stránky `http://<doména>/<kontext>/LoginControl`. (<doména> je doménové jméno Vašeho počítače, <kontext> je kontextová cesta zadaná při nasazení). Zde musíte přihlásit správně email a heslo (administrátorský email a administrátorské heslo) (viz obrázek 3.1), tak se zobrazí úvodní stránka administrátorského rozhraní (viz obrázek 3.2).

| Internetový obchod  |                                      |         |            |            |
|---|--------------------------------------|---------|------------|------------|
| Home  | Nákupní Košík                        | Kontakt | Registrace | Přihlášení |
| <b>Katalog</b>  | Váš e-mail: <input type="text"/>     |         |            |            |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | Vaše heslo: <input type="password"/> |         |            |            |
|   | <input type="button" value="Login"/> |         |            |            |
|   |                                      |         |            |            |

Obrázek 3.1: Přihlášení rozhraní

| Internetový obchod  |   |         |            |            |
|---|---|---------|------------|------------|
| Home  | Nákupní Košík                                       | Kontakt | Registrace | Přihlášení |
| <b>Katalog</b>  | Ahoj! Administrator: TaiVotran, <a href="#">out</a> |         |            |            |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | <b>Administrátorské rozhraní</b>                    |         |            |            |
|   | <a href="#">+&gt; Manage Katalog</a>                |         |            |            |
|   | <a href="#">+&gt; Manage Zboží</a>                  |         |            |            |
|   | <a href="#">+&gt; Manage Zákazník</a>               |         |            |            |
|   |   |         |            |            |

Obrázek 3.2: Administrátorské rozhraní

Zde si můžete zvolit jednu z akcí, k nimž je oprávněn administrátor systému:

+ **Manage Katalog** : Správa kategorie (viz obrázek 3.2.1) , tady můžete provést :










- Přidat kategorii zboží
- Změnit kategorii zboží
- Odstranit kategorii zboží

| Home  | Nákupní Košík  | Kontakt              | Registrace                               | Přihlášení |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
|---|--|----------------------|--|------------|-----------|-------------|---------|--------|---|-------|----------------------|--|---|-------|----------------------|--|---|---------|----------------------|--|---|-----|----------------------|--|---|----------|----------------------|--|---|---------------|----------------------|--|---|----|----------------------|--|
| Katalog   | Manage Katalog   |                      |  |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | <div><div><b>Přidat katalog:</b><br/>Nazev : <input type="text"/> <input type="button" value="Add catalog"/></div><div><b>Katalog Opravit</b><br/>Novy Nazev : <input type="text"/> <input type="button" value="Opravit"/></div></div> <table border="1"><thead><tr><th>catalogID</th><th>catalogName</th><th>Opravit</th><th>delete</th></tr></thead><tbody><tr><td>1</td><td>Apple</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>2</td><td>Nokia</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>3</td><td>Samsung</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>4</td><td>HTC</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>5</td><td>Motorola</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>6</td><td>Sony Ericsson</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr><tr><td>7</td><td>LG</td><td><a href="#">Edit</a></td><td><input type="button" value="odstranit"/></td></tr></tbody></table> |                      |  |            | catalogID | catalogName | Opravit | delete | 1 | Apple | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 2 | Nokia | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 3 | Samsung | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 4 | HTC | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 5 | Motorola | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 6 | Sony Ericsson | <a href="#">Edit</a> | <input type="button" value="odstranit"/> | 7 | LG | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |
| catalogID   | catalogName  | Opravit              | delete                                   |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 1   | Apple  | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 2   | Nokia  | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 3   | Samsung  | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 4   | HTC  | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 5   | Motorola   | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 6   | Sony Ericsson  | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |
| 7   | LG   | <a href="#">Edit</a> | <input type="button" value="odstranit"/> |            |           |             |         |        |   |       |                      |  |   |       |                      |  |   |         |                      |  |   |     |                      |  |   |          |                      |  |   |               |                      |  |   |    |                      |  |

Obrázek 3.2.1: Správa kategorie rozhraní

+ **Manage Zboží** : Správa zboží (viz obrázek 3.2.2) , tady můžete provést :

- Přidat druh prodáváného zboží
- Změnit druh prodáváného zboží
- Hledat druh prodáváného zboží
- Odstranit druh prodáváného zboží
- Zobrazit seznam zboží

| Home  | Nákupní Košík | Kontakt  | Registrace   | Přihlášení  |                 |             |               |                      |                        |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
|---|---------------|--|--|---|-----------------|-------------|---------------|----------------------|------------------------|-----------------|-----------|---------|---------|--------|---|---------------|------|---|--------------|---|---|-------|----------------------|------------------------|---|-------|-------|---|---------|---|---|---------------|----------------------|------------------------|----|-------|-------|---|---------|---|---|-------|----------------------|------------------------|
| <b>Katalog</b><br><a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a>   |               | <b>Manage Product</b>  |  |   |                 |             |               |                      |                        |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
|   |               | <b>Přidat product</b><br>Name : <input type="text"/><br>Catalog : <input type="text"/><br>Price : <input type="text"/><br>Quantity : <input type="text"/><br><br>Description: <input type="text"/><br>Image : <input type="text"/><br><input type="button" value="Vybrat soubor"/> Soubor nevybrán<br><input type="button" value="Add"/> | <b>Opravit product</b><br>Name : <input type="text"/><br>Catalog : <input type="text"/><br>Price : <input type="text"/><br>Quantity : <input type="text"/><br><br>Description: <input type="text"/><br>Image : <input type="text"/><br><input type="button" value="Vybrat soubor"/> Soubor nevybrán<br><input type="button" value="Edit"/> | <b>Hledat product</b><br>Hledat zboží : <input type="text"/><br><input type="button" value="Hledat"/> |                 |             |               |                      |                        |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
|   |               | <b>Seznam zboží</b>  |  |   |                 |             |               |                      |                        |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
| <table border="1"> <thead> <tr> <th>productId</th> <th>productName</th> <th>productPrice</th> <th>productImage</th> <th>productDetail</th> <th>productQuantity</th> <th>catalogId</th> <th>katalog</th> <th>Opravit</th> <th>delete</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>808 Pure View</td> <td>6000</td> <td></td> <td>eruytgijyhjk</td> <td>5</td> <td>2</td> <td>Nokia</td> <td><a href="#">Edit</a></td> <td><a href="#">Delete</a></td> </tr> <tr> <td>2</td> <td>Arc S</td> <td>11499</td> <td></td> <td>dfdgdfg</td> <td>6</td> <td>6</td> <td>Sony Ericsson</td> <td><a href="#">Edit</a></td> <td><a href="#">Delete</a></td> </tr> <tr> <td>19</td> <td>Ipad2</td> <td>12000</td> <td></td> <td>xdsfsdg</td> <td>5</td> <td>1</td> <td>Apple</td> <td><a href="#">Edit</a></td> <td><a href="#">Delete</a></td> </tr> </tbody> </table> |               |  |  |   | productId       | productName | productPrice  | productImage         | productDetail          | productQuantity | catalogId | katalog | Opravit | delete | 1 | 808 Pure View | 6000 |  | eruytgijyhjk | 5 | 2 | Nokia | <a href="#">Edit</a> | <a href="#">Delete</a> | 2 | Arc S | 11499 |  | dfdgdfg | 6 | 6 | Sony Ericsson | <a href="#">Edit</a> | <a href="#">Delete</a> | 19 | Ipad2 | 12000 |  | xdsfsdg | 5 | 1 | Apple | <a href="#">Edit</a> | <a href="#">Delete</a> |
| productId   | productName   | productPrice   | productImage   | productDetail   | productQuantity | catalogId   | katalog       | Opravit              | delete                 |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
| 1   | 808 Pure View | 6000   |   | eruytgijyhjk  | 5               | 2           | Nokia         | <a href="#">Edit</a> | <a href="#">Delete</a> |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
| 2   | Arc S         | 11499  |   | dfdgdfg   | 6               | 6           | Sony Ericsson | <a href="#">Edit</a> | <a href="#">Delete</a> |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |
| 19  | Ipad2         | 12000  |   | xdsfsdg   | 5               | 1           | Apple         | <a href="#">Edit</a> | <a href="#">Delete</a> |                 |           |         |         |        |   |               |      |   |              |   |   |       |                      |                        |   |       |       |   |         |   |   |               |                      |                        |    |       |       |   |         |   |   |       |                      |                        |

Obrázek 3.2.2: Správa zboží rozhraní

+ **Manage Zákazník** : Správa zákazníka (viz obrázek 3.2.3) , tady můžete provést :

- Zobrazit informace o zákazníkovi.
- Odstranit zákazníka (tlačit na “Delete”) .
- Poslat e-mail pro zákazníka (tlačit na “send”).
- Informace o objednávce pro zákazníka, tlačit na “popis” (viz obrázek 3.2.3.1)

Home

Nákupní Košík

Kontakt

Registrace

Přihlášení

Katalog

Apple

Nokia

Samsung

HTC

Motorola

Sony Ericsson

LG

Manage Zákazníků

Seznam customer

| Kód | Jméno     | Příjmení   | Ulice     | Město      | PSČ  | Telefon   | E-mail                | Popis | Delete | sendEmail |
|-----|-----------|------------|-----------|------------|------|-----------|-----------------------|-------|--------|-----------|
| 16  | Tai       | VoTran     | listopadu | Liberec    | 3306 | 774030138 | taivotran@gmail.com   | Popis | Delete | send      |
| 18  | Joe       | Cole       | listopadu | Liberec    | 3306 | 774030138 | taivotran@gmail.com   | Popis | Delete | send      |
| 19  | andrey    | shevchenko | ukrej     | ukraina    | 9876 | 123456    | bbbbbb@gmail.com      | Popis | Delete | send      |
| 20  | Tai       | VoTran     | listopadu | Liberec    | 3306 | 774030138 | taivotran90@gmail.com | Popis | Delete | send      |
| 21  | Leonel    | Messi      | listopadu | Liberec    | 3306 | 774030138 | taivotran90@gmail.com | Popis | Delete | send      |
| 33  | Fernando  | Torres     | listopadu | Liberec 15 | 3306 | 987654321 | taivotran90@gmail.com | Popis | Delete | send      |
| 35  | Cristiano | Ronaldo    | real      | madrid     | 1212 | 7654321   | taivotran90@gmail.com | Popis | Delete | send      |
| 40  | Wayne     | Rooney     | listopadu | Liberec    | 3306 | 7654321   | taivotran90@gmail.com | Popis | Delete | send      |
| 47  | fernando  | Torres     | listopadu | Liberec    | 3306 | 7654321   | taivotran90@gmail.com | Popis | Delete | send      |

Informace zákazníků :

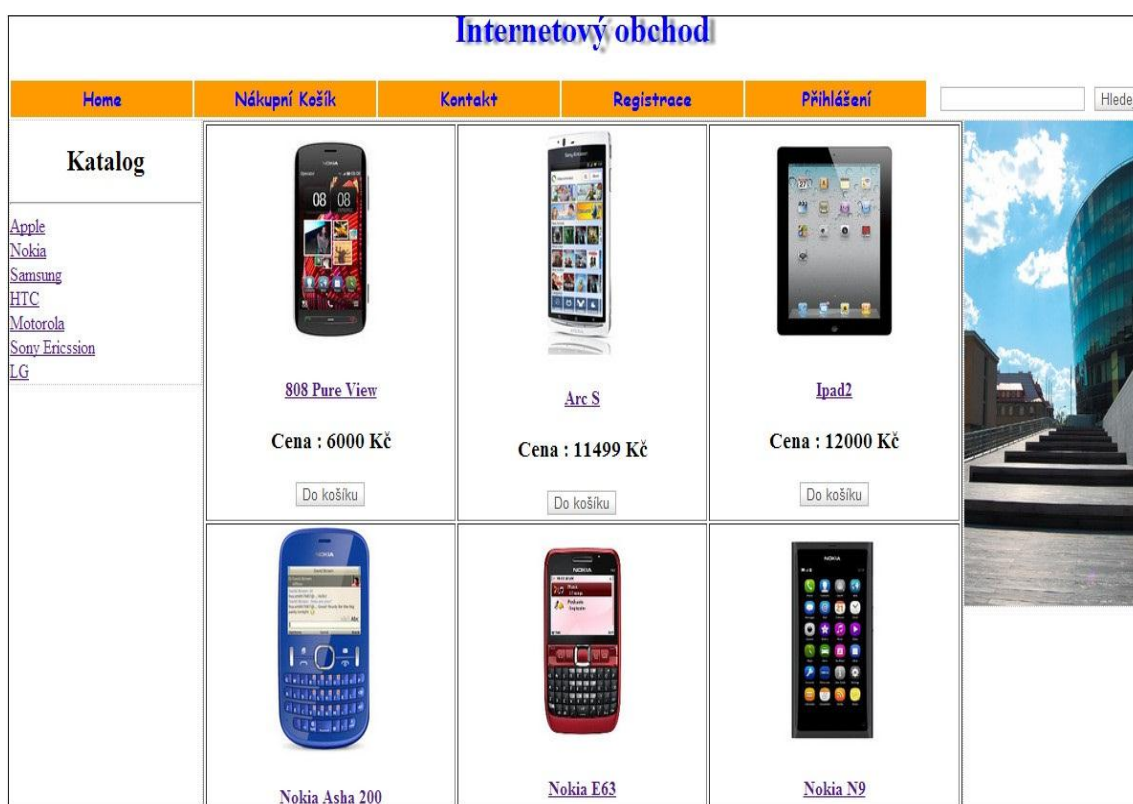
Obrázek 3.2.3: Správa o zákazníkovi rozhraní

|  |
|--|
| <b>Informace zákazníků :</b> <ul style="list-style-type: none"> <li>• <b>Jméno :</b> Tai</li> <li>• <b>Příjmení :</b> VoTran</li> <li>• <b>Ulice :</b> listopadu</li> <li>• <b>Město :</b> Liberec</li> <li>• <b>PSČ :</b> 3306</li> <li>• <b>Telefon :</b> 774030138</li> <li>• <b>E-mail :</b> taivotran90@gmail.com</li> </ul> <b>Objednané zboží:</b> <ul style="list-style-type: none"> <li>• <b>Název produktu :</b> Arc S</li> <li>• <b>Cena/ks :</b> 11499 ,- Kč</li> <li>• <b>Počet ks :</b> 2</li> <li>• <b>Cena celkem :</b> 22998 ,- Kč</li> <li>• <b>Název produktu :</b> Ipad2</li> <li>• <b>Cena/ks :</b> 12000 ,- Kč</li> <li>• <b>Počet ks :</b> 3</li> <li>• <b>Cena celkem :</b> 36000 ,- Kč</li> <li>• <b>Celková cena za zboží :</b> 58998 ,- Kč</li> </ul> |
|--|

Obrázek 3.2.3.1: Objednávka rozhraní

### 3.4 Používání aplikace zákazníkem

Pro zákazníky je aplikace intuitivní a podobá se většině současných internetových obchodů. V levé části okna prohlížeče se nachází menu, ve kterém jsou zobrazeny kategorie zboží. V horní části okna prohlížeče se nachází lišta menu, která může mít následující funkce: Zpět na úvodní stránku, zobrazení nákupního košíku, registrace, přihlášení a hledání produktů podle názvu nebo podle slova v jeho popisu. V centrální části okna prohlížeče je zobrazen seznam produktů, které se nacházejí v určité kategorii. Každý zobrazený produkt obsahuje odkaz (název produktu a jeho obrázek), jehož prostřednictvím lze zobrazit podrobnosti o produktu. Produkt lze vložit do košíku stisknutím tlačítka “Do košíku”. Na obrázku 3.3 je zobrazena úvodní stránka.



Obrázek 3.3: Úvodní stránka rozhraní

V nákupním košíku zákazníka lze : zadávat množství do příslušného editačního pole, odstranit zboží, odstranit košík, koupit zboží stisknutím tlačítka “Koupit”. (viz obrázek 3.3.1).

| <a href="#">Home</a>  | <a href="#">Nákupní Košík</a>   | <a href="#">Kontakt</a>  | <a href="#">Registrace</a> | <a href="#">Přihlášení</a> | <input type="checkbox"/> |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
|---|---|--|----------------------------|----------------------------|--------------------------|--------|-----------|---|-------|-----|-------|---------|--------------------------|---|-------|-----|-------|---------|--------------------------|--------------------------------|--|--|--|-----------------|--|-------------------------------|--|--|--|--|--|
| <b>Katalog</b>  | <b>Nákupní košík</b>  |  |                            |                            |                          |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | <table border="1"> <thead> <tr> <th>Obraz</th> <th>Název</th> <th>Množství</th> <th>Cena</th> <th>Součet</th> <th>Odstranit</th> </tr> </thead> <tbody> <tr> <td></td> <td>Arc S</td> <td>2 x</td> <td>11499</td> <td>= 22998</td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td>Ipad2</td> <td>1 x</td> <td>12000</td> <td>= 12000</td> <td><input type="checkbox"/></td> </tr> <tr> <td colspan="4"><b>Celková cena za zboží :</b></td> <td><b>34998 Kč</b></td> <td></td> </tr> <tr> <td colspan="2"><a href="#">Zpět k nákupu</a></td> <td colspan="4"> <input type="button" value="Update Cart"/> <input type="button" value="Empty Cart"/> <input type="button" value="Koupit"/> </td> </tr> </tbody> </table> | Obraz  | Název                      | Množství                   | Cena                     | Součet | Odstranit |  | Arc S | 2 x | 11499 | = 22998 | <input type="checkbox"/> |  | Ipad2 | 1 x | 12000 | = 12000 | <input type="checkbox"/> | <b>Celková cena za zboží :</b> |  |  |  | <b>34998 Kč</b> |  | <a href="#">Zpět k nákupu</a> |  | <input type="button" value="Update Cart"/> <input type="button" value="Empty Cart"/> <input type="button" value="Koupit"/> |  |  |  |
| Obraz   | Název   | Množství   | Cena                       | Součet                     | Odstranit                |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
|    | Arc S   | 2 x  | 11499                      | = 22998                    | <input type="checkbox"/> |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
|    | Ipad2   | 1 x  | 12000                      | = 12000                    | <input type="checkbox"/> |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
| <b>Celková cena za zboží :</b>  |   |  |                            | <b>34998 Kč</b>            |                          |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |
| <a href="#">Zpět k nákupu</a>   |   | <input type="button" value="Update Cart"/> <input type="button" value="Empty Cart"/> <input type="button" value="Koupit"/> |                            |                            |                          |        |           |   |       |     |       |         |                          |   |       |     |       |         |                          |                                |  |  |  |                 |  |                               |  |  |  |  |  |

Obrázek 3.3.1: Nákupní košík rozhraní

Když stiskneme tlačítko "Koupit", v centrální části okna prohlížeče budou zobrazeny dodací údaje (viz obrázek 3.3.2), zákazník musí vyplnit informace o zákazníkovi do editačního pole, potom klikne na tlačítko "Potvrdit", potvrdí a dokončí nákup, v centrální části okna prohlížeče bude zobrazen jeden dodací list, zde je informace o zákazníkovi a jeho objednávce (viz obrázek 3.3.3).

|   |  |                         |                            |                             |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
|---|--|-------------------------|----------------------------|-----------------------------|-----------|----------------------|------------|----------------------|--|----------|----------------------|-------------|----------------------|--|----------------|----------------------|----------|----------------------|-----------------------------|---|--|--|--|--|
| <a href="#">Home</a>  | <a href="#">Nákupní Košík</a>  | <a href="#">Kontakt</a> | <a href="#">Registrace</a> | <a href="#">Přihlášení</a>  |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
| <b>Katalog</b>  | <b>Dodací údaje</b>  |                         |                            |                             |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | <table> <tr> <td>E-mail: *</td> <td><input type="text"/></td> <td>Telefon: *</td> <td colspan="2"><input type="text"/></td> </tr> <tr> <td>Jméno: *</td> <td><input type="text"/></td> <td>Příjmení: *</td> <td colspan="2"><input type="text"/></td> </tr> <tr> <td>Ulice, č.p.: *</td> <td><input type="text"/></td> <td>Město: *</td> <td><input type="text"/></td> <td>PSC: * <input type="text"/></td> </tr> <tr> <td colspan="5"><input type="button" value="Potvrdit"/></td> </tr> </table> |                         |                            |                             | E-mail: * | <input type="text"/> | Telefon: * | <input type="text"/> |  | Jméno: * | <input type="text"/> | Příjmení: * | <input type="text"/> |  | Ulice, č.p.: * | <input type="text"/> | Město: * | <input type="text"/> | PSC: * <input type="text"/> | <input type="button" value="Potvrdit"/> |  |  |  |  |
| E-mail: *   | <input type="text"/>   | Telefon: *              | <input type="text"/>       |                             |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
| Jméno: *  | <input type="text"/>   | Příjmení: *             | <input type="text"/>       |                             |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
| Ulice, č.p.: *  | <input type="text"/>   | Město: *                | <input type="text"/>       | PSC: * <input type="text"/> |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |
| <input type="button" value="Potvrdit"/>   |  |                         |                            |                             |           |                      |            |                      |  |          |                      |             |                      |  |                |                      |          |                      |                             |   |  |  |  |  |

Obrázek 3.3.2: Dodací údaje rozhraní

| Home  | Nákupní Košík  | Kontakt | Registrace | Přihlášení |
|---|--|---------|------------|------------|
| Katalog   | Dodací list  |         |            |            |
| <a href="#">Apple</a><br><a href="#">Nokia</a><br><a href="#">Samsung</a><br><a href="#">HTC</a><br><a href="#">Motorola</a><br><a href="#">Sony Ericsson</a><br><a href="#">LG</a> | <p><u>Vámi zaslané údaje:</u></p> <ul style="list-style-type: none"> <li>• Jméno : Tai</li> <li>• Příjmení : VoTran</li> <li>• Ulice : Listopadu</li> <li>• Město : Liberec</li> <li>• PSČ : 46015</li> <li>• Telefon : 7654321</li> <li>• E-mail : taivotran90@gmail.com</li> </ul> <p><u>Objednané zboží:</u></p> <ul style="list-style-type: none"> <li>• Název produktu : Arc S</li> <li>• Cena/ks : 11499 ,- Kč</li> <li>• Počet ks : 2</li> <li>• Cena celkem : 22998 ,- Kč</li> <li>• Název produktu : Ipad2</li> <li>• Cena/ks : 12000 ,- Kč</li> <li>• Počet ks : 1</li> <li>• Cena celkem : 12000 ,- Kč</li> <li>• Celková cena za zboží : 34998 ,- Kč</li> </ul> <p>Děkujeme za Váš nákup !!!</p> |         |            |            |

Obrázek 3.3.3: Dodací list rozhraní

## Závěr

V této bakalářské práci jsem popsal základní postupy pro tvorbu webových aplikací na platformě J2EE a tyto postupy jsem využil při tvorbě ukázkové aplikace Internetový obchod. Uživatelským rozhraním aplikace je Internetový prohlížeč, ve kterém je vzhled zajištěn pomocí HTML (viz 1.1) a CSS (viz 1.2). Data pro celou aplikaci jsou uložena v MySQL databázi (viz 1.4.2), přičemž komunikace mezi uživatelem a databází je realizována pomocí JDBC (viz 1.5).

Návrh této aplikace využívá návrhový vzor Model-View-Controller (viz 2.2), protože vzor MVC rozděluje aplikaci na tři samostatné části ( Model, View a Controller), které tak pomáhají vyvíjet aplikace rychle, jednoduše, snadno upgradovat, údržbu aplikace.

Níže jsou uvedeny výhody a nedostatky aplikace a možná vylepšení:

- Výhody:

+ Tato aplikace poskytuje základní funkce nezbytné pro provozování internetového obchodu (třeba objednávky zboží, pohled na zboží, hledání zboží).

+ Aplikace poskytuje Administrátorské rozhraní - pomáhají snadno spravovat zboží, objednávky, zákazníka : Administrátor může přidat, změnit, hledat nebo odebrat druh prodávaného zboží, procházet, případně rušit zadané objednávky, procházet informace o zákazníkovi a poslat email pro zákazníka, všechno to jsou akce pro Administrátora.

+ Aplikace je přátelská, rozhraní jednoduché a snadno použitelná

+ Aplikace se může provozovat a z hlediska skutečnosti (potřebuje mít hosting a domain na internetu)

- Nedostatky :

+ Nemá možnost platby bankovní kartou.

- Možná vylepšení :

+ Přidat funkci platby kreditní kartou

Psaní této práce mi přineslo mnoho zkušeností a vědomostí o programovacích jazycích, zejména o webových programovacích jazycích. Tato práce mi pomohla v tvorbě praktické aplikace, která slouží potřebám lidí.



# Seznam Literatury

[1] HEROUT, Pavel. *Učebnice jazyka Java*. Dotisk prvního vydání. České Budějovice: Kopp, 2000. ISBN 80-7232-1150-3.

[2] HALL, Marty. *JAVA servlety a stránky JSP*. První vydání. Praha: Neocortex, 2001. ISBN 80-86330-06-0.

Dokument dostupný v angličtině na URL:

<http://pdf.coreservlets.com> . [cit. 2014-03-03].

[3] HyperText Markup Language. [cit. 2014-03-03]. Dostupné na:

[http://cs.wikipedia.org/wiki/HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/HyperText_Markup_Language)

[4] Hypertextový jazyk značek. [cit. 2014-03-03]. Dostupné na:

<http://www.jakpsatweb.cz/html/>

[5] HTML tutorial. [cit. 2014-03-03]. Dostupné na:

<http://www.w3schools.com/html/default.asp>

[6] HTML Tag Reference. [cit. 2014-03-03]. Dostupné na:

<http://www.w3schools.com/tags/default.asp>

[7] Kaskádové styly. [cit. 2014-03-03]. Dostupné na:

[http://cs.wikipedia.org/wiki/Kaskádové\\_styl](http://cs.wikipedia.org/wiki/Kaskádové_styl)

[8] CSS tutorial. [cit. 2014-03-03]. Dostupné na:

<http://www.w3schools.com/css/default.asp>

[9] CSS Reference. [cit. 2014-03-03]. Dostupné na:

<http://www.w3schools.com/cssref/default.asp>

[10] Příkazy jazyka SQL. [cit. 2014-03-03]. Dostupné na:

[http://cs.wikipedia.org/wiki/Příkazy\\_jazyka\\_SQL](http://cs.wikipedia.org/wiki/Příkazy_jazyka_SQL)

[11] MySQL Community Server. [cit. 2014-03-03]. Dostupné na:

<http://cs.wikipedia.org/wiki/MySQL>

[12] MySQL 5.0 reference manual. [cit. 2014-03-03]. Dostupné na:

<http://dev.mysql.com/doc/refman/5.0/en/>

[13] Java Database Connectivity. [cit. 2014-03-03]. Dostupné na:

<http://cs.wikipedia.org/wiki/JDBC>

[14] Úvod do JDBC. [cit. 2014-03-03]. Dostupné na :

<http://interval.cz/clanky/uvod-do-jdbc/>

[15] Lukáš Fryč, Martin Večeřa, Ondřej Žížka: Tvorba pokročilých webových aplikací v Javě. [cit. 2014-03-03]. Dostupné na : <http://perchta.fit.vutbr.cz/~redhat/slajdy/VUT-course-01-Intro-JSP-JDBC.pdf>

[16] Accessing databases with JDBC. [cit. 2014-03-03]. Dostupné na : <http://pdf.coreservlets.com/Accessing-Databases-JDBC.pdf>

[17] Vzor Model-View-Controller. [cit. 2014-03-03]. Dostupné na : [http://fei.pepiczech.cz/wp-content/uploads/PIA06\\_Servlety2MVC.pdf](http://fei.pepiczech.cz/wp-content/uploads/PIA06_Servlety2MVC.pdf)

[18] Java Server Pages Model 2 architecture. [cit. 2014-03-03]. Dostupné na : <http://www.javaworld.com/article/2076557/java-web-development/understanding-javaserver-pages-model-2-architecture.html>

[19] Hypertext Transfer Protocol. [cit. 2014-03-03]. Dostupné na : [http://cs.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

#### **Jiné zdroje literatury:**

[20] Diskuzní fóra týkající se technologií okolo Javy. [cit. 2014-03-03]. Dostupné na: <http://www.java-forums.org/forum.php> .

# Přílohy

## Příloha A - Obsah CD

### 1. Adresář install

V tomto adresáři se nacházejí instalační soubory aplikací potřebných pro spuštění internetového obchodu. Jedná se o JDK 7, Tomcat 7.0.37, MySQL 5.0.41, Navicat 8, a netbeans 7.3 .

### 2. Adresář BP

Zde se nacházejí kompletní zdrojové kódy aplikace. Tento adresář byl vytvořen vývojovým prostředím NetBeans. Obsahuje následující důležité podadresáře:

- **build** obsahuje v podadresáři web zkompilované třídy aplikace, stránky JSP a konfigurační soubory tak, jak budou vloženy do komprimovaného souboru .WAR.
- **dist** obsahuje komprimovaný soubor BP.war, který je vhodný pro distribuci aplikace.
- **nbproject** obsahuje projekt vytvořený NetBeans.
- **src** obsahuje zdrojové kódy tříd aplikace.
- **web** obsahuje stránky JSP a konfigurační soubory.

Ostatní podadresáře nemají důležitý význam.

### 3. Adresář java\_dokument

Tento adresář obsahuje programátorskou dokumentaci aplikace ve formátu PDF.

### 4. Soubor *Bakalarska\_prace\_2014\_Vo\_Tran\_Tai.pdf*

Tento soubor obsahuje tuto bakalářskou práci ve formátu PDF.

### 5. Soubor bakalar.sql

Toto je formát databázového souboru.

## Příloha B - Instalace a nasazení webové aplikace

### 1) Instalace

Dříve než se pustíme do samotného programování webových aplikací, budeme si muset nainstalovat vývojové prostředí Javy, to bude složeno z Java Development Kitu, aplikačního serveru Tomcat a databázového system.

- Instalace JDK: Samotné JDK je možné bezplatně stáhnout z oficiálních stránek <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (nebo z disket CD) instalační soubor *jdk-7u45-windows-x64.exe* a nainstalujeme.

- Instalace MySQL : Stáhněte si z oficiálních stránek <http://dev.mysql.com/downloads/mysql/5.1.html> (nebo z disket CD). Následně rozbalte *setup.exe* a proveďte **custom** (uživatelskou) instalaci do *C:\Program Files\MySQL*. Po instalaci se spustí konfigurátor MySQL instance. Zvolte si zde konfiguraci pro vývojářskou stanici, default charset UTF8, a nastavte si nějaké heslo pro uživatele **root** (standardně je prázdné, pro tuto aplikaci nastavím heslo **root**).

- Instalace Tomcat: Na systémech Windows si stáhneme ze stránek <http://tomcat.apache.org/> (nebo z disket CD) instalační soubor *apache-tomcat-7.0.37.exe* a nainstalujeme. V průběhu instalace budeme dotázáni na uživatelské jméno a heslo pro přístup k webovému rozhraní Tomcatu.

+ Nastavení proměnné prostředí :

1. Zjistěte si cestu, kde je nainstalován Java Development Kit (JDK). Obvykle to bývá něco jako *C:\Program Files\Java\jdk1.7.0\_15*.
2. Klikněte pravým tlačítkem na *Tento počítač*, zvolte *Vlastnosti*.
3. Klikněte na *Upřesnit nastavení systému*.
4. Otevře se okno *Vlastnosti systému* s aktivní kartou *Upřesnit*. Klikněte na *Proměnné prostředí*.
5. V sekci *Systémové proměnné* zvolte *Nová*.
6. Jako název proměnné zadejte *JAVA\_HOME* a jako hodnotu cestu k JDK. Potvrďte *OK*.
7. V systémových proměnných dále najděte *Path*, vyberte, klikněte na *Upravit*.
8. Na konec hodnoty napište přesně: *%JAVA\_HOME%\bin*;
9. Vše potvrďte a restartujte počítač.

## 2) *Nasazení webové aplikace :*

Způsob nasazení aplikace se liší v závislosti na používaném webovém kontejneru. Většinou nasazení spočívá ve zkopírování souboru BP.war do příslušného adresáře webového kontejneru a případně v úpravě konfiguračního souboru kontejneru. Podrobné informace o nasazení naleznete v dokumentaci používaného webového kontejneru. Informace o nasazení webové aplikace ve webovém kontejneru Tomcat se nacházejí na <http://tomcat.apache.org/tomcat-7.0-doc/deployer-howto.html>

Import dat v MySQL : Naimportování zálohy pak můžeme provést pomocí řádkového klienta mysql:

```
mysql -u uživatel -p heslo jmeno_databaze < jmeno_souboru
```

*Uživatel* je název uživatelského účtu MySQL s administrátorskými právy, *heslo* je heslo k tomuto účtu, *jmeno\_databaze* je existující databáze v MySQL a *jmeno\_souboru* je ve formátu databázového souboru. Pro tuto aplikaci nastavím:

- *Uživatel* je : **root**
- *heslo* je : **root**
- *jmeno\_databaze* je : **bakalar**
- a *jmeno\_souboru* je : **bakalar.sql**

Spustit Tomcat, po spuštění by aplikace měla být dostupná na portu 8080, takže zadáme do webového prohlížeče adresu <localhost:8080/BP> a zobrazí se nám stránka webového aplikace rozhraní.

Vyvolejte *administrátorské rozhraní* : zadáme odkazu <localhost:8080/BP/LoginControl> do webového prohlížeče a zadejte uživatelské jméno je *tai.vo.tran@tul.cz* a heslo je *password*.