

**VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA**  
**Hornicko-geologická fakulta**  
**institut geoinformatiky**

**TVORBA APLIKACE PRO EVIDENCI RYBÁŘSKÝCH  
ÚLOVKŮ, VYUŽÍVAJÍCÍ SLUŽBU WWW**

**diplomová práce**

Autor:  
Vedoucí diplomové práce:

**Karel Vlček**  
**Ing. Kateřina Růžičková, Ph.D.**

**Ostrava 2004**

**zde je vložen zadávací protokol**

*Prohlašuji, že*

- *celou diplomovou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.*
- *jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo*
- *beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevydělečně, ke své vnitřní potřebě, práci užít (§ 35 odst. 3)*
- *souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové práce, budou zveřejněny v informačním systému VŠB-TUO.*
- *bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst.4 autorského zákona.*
- *bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).*

*V Ostravě dne 20.4.2004*

*Karel Vlček*

*Karel Vlček  
Partyzánská 1107  
756 61 Rožnov Pod Radhoštěm*

## **ANOTACE DIPLOMOVÉ PRÁCE**

Tato diplomová práce je zaměřena na rozšiřování geografických informačních systémů mezi laickou veřejností. V úvodu je vysvětlena problematika vizualizace prostorových dat v prostředí WWW. Následně je vybráno vhodné programové vybavení pro tvorbu aplikace, jenž má sloužit pro vizualizaci prostorových dat. V další části je zpracována analýza informačního systému, ze které následně vzešla aplikace pro evidenci rybářských úlovků, využívající službu WWW. Aplikace je určena především pro rybářskou veřejnost, která si může interaktivně zjišťovat informace o revírech a především úlovcích na těchto revírech. Uživatelé mají také možnost tyto úlovky zadávat. Také slouží jako mapová prohlížečka rybářských revírů na Severní Moravě a ve Slezsku.

## **ANOTATION OF THESIS**

This disseration is about dissemination of geographical information systems to the common people. Introduction explains visualisation of space datas in WWW. Recommended software equipment is selected as next. Following chapetr includes analysis of information system that was used as the base of WWW application for fishing trophies. This application enables interactive overview of fishing areas and tropies. All users can add own catches. The application is assigned to fishers and is useful as the fishing map of Northern Moravia and Silesia too.

## OBSAH

|   |           |
|---|-----------|
| <b>SEZNAM ZKRATEK.....</b>  | <b>7</b>  |
| <b>1 Úvod .....</b>   | <b>8</b>  |
| <b>2 Postup zpracování .....</b>  | <b>9</b>  |
| <b>3 Český rybářský svaz.....</b>   | <b>10</b> |
| <b>4 Popis použitých dat .....</b>  | <b>11</b> |
| 4.1 Geodata.....  | 11        |
| 4.2 Navazující popisná data .....   | 12        |
| 4.2.1 Datový model.....   | 12        |
| 4.2.2 Databáze „rybydb“ .....   | 13        |
| <b>5 Výběr programového vybavení.....</b>   | <b>15</b> |
| 5.1 Internetové prostředí .....   | 15        |
| 5.1.1 Klient-server .....   | 15        |
| 5.1.2 Třívrstvá architektura webové databázové aplikace .....                         | 17        |
| 5.2 Webové servery.....   | 19        |
| 5.2.1 Apache .....  | 19        |
| 5.2.2 Internet Information Server.....  | 19        |
| 5.3 Skriptovací jazyk PHP .....   | 20        |
| 5.4 Databázový systém.....  | 20        |
| 5.4.1 MySQL .....   | 20        |
| 5.5 Vizualizace prostorových dat v prostředí WWW .....                                | 21        |
| 5.5.1 Mapový server MapServer.....  | 21        |
| 5.6 Skriptovací jazyk JavaScript .....  | 21        |
| 5.7 ArcView GIS 3.2.....  | 22        |
| 5.8 Přehled zvoleného programového vybavení .....                                     | 22        |
| <b>6 Analýza a návrh informačního systému .....</b>                                   | <b>23</b> |
| 6.1 Slovní specifikace požadavků .....  | 23        |
| 6.2 Model byznys procesů a doménový model .....                                       | 25        |
| 6.3 Diagram aktivit.....  | 28        |
| 6.4 Model případů užití .....   | 31        |
| 6.5 Sekvenční diagramy .....  | 33        |
| 6.5.1 Sekvenční diagram – práce s mapou full extent .....                             | 33        |
| 6.5.2 Sekvenční diagram – práce s mapou pan .....                                     | 34        |
| 6.5.3 Sekvenční diagram – práce s mapou zoom-in .....                                 | 35        |
| 6.5.4 Sekvenční diagram – práce s mapou zoom-out .....                                | 36        |
| 6.5.5 Sekvenční diagram – zadání nového úlovku .....                                  | 37        |
| 6.5.6 Sekvenční diagram – vyhledání revíru .....                                      | 38        |
| 6.5.7 Sekvenční diagram – vyhledávání všech úlovků a zobrazení požadovaného úlovku .. | 40        |
| 6.6 Diagram tříd .....  | 41        |
| 6.7 Diagram spolupráce.....   | 43        |
| 6.8 Stavový diagram.....  | 44        |
| <b>7 Vlastní návrh informačního systému .....</b>                                     | <b>45</b> |

|          |   |           |
|----------|---|-----------|
| 7.1      | Načtení mapových vrstev .....                       | 46        |
| 7.1.1    | Mapový soubor MAPFILE .....                         | 46        |
| 7.1.2    | Mapové okno .....                                   | 46        |
| 7.2      | Funkce aplikace .....                               | 47        |
| 7.2.1    | Vyhledávání informací o jednotlivých revírech ..... | 47        |
| 7.2.2    | Vyhledávání informací o všech revírech .....        | 49        |
| 7.2.3    | Vyhledávání informací o všech úlovcích .....        | 50        |
| 7.2.4    | Zobrazovací operace .....                           | 52        |
| 7.2.5    | Vyhledávat informace o úlovcích na revírech .....   | 54        |
| 7.2.6    | Vkládání nových údajů o lovu a úlovku .....         | 55        |
| 7.2.7    | Doplňkové funkce .....                              | 56        |
| 7.3      | Adresářová struktura aplikace .....                 | 57        |
| <b>8</b> | <b>Závěr .....</b>                                  | <b>59</b> |
|          | <b>SEZNAM LITERATURY .....</b>                      | <b>60</b> |
|          | <b>SEZNAM OBRÁZKŮ .....</b>                         | <b>60</b> |
|          | <b>SEZNAM PŘÍLOH .....</b>                          | <b>61</b> |
|          | <b>Příloha 1 .....</b>                              | <b>1</b>  |
|          | <b>Příloha 2 .....</b>                              | <b>17</b> |
|          | <b>Příloha 3 .....</b>                              | <b>20</b> |
|          | <b>Příloha 4 .....</b>                              | <b>23</b> |
|          | <b>Příloha 5 .....</b>                              | <b>24</b> |
|          | <b>Příloha 6 .....</b>                              | <b>25</b> |
|          | <b>Příloha 7 .....</b>                              | <b>28</b> |
|          | <b>Příloha 8 .....</b>                              | <b>33</b> |
|          | <b>Příloha 9 .....</b>                              | <b>34</b> |

# SEZNAM ZKRATEK

## České zkratky

|         |  |
|---------|--|
| ATD.    | a tak dále   |
| V K. Ú. | v katastrálním území                               |
| VŠB-TUO | Vysoká škola báňská – Technická univerzita Ostrava |
| T. J.   | to jest  |

## Cizojazyčné zkratky

|           |   |
|-----------|---|
| ESRI      | Environmental Systems Research Institute        |
| HTML      | Hypertext Markup Language                       |
| HTTP      | Hypertext Transfer Protocol                     |
| IP        | Internet Protocol                               |
| JPG, JPEG | Joint Photographic Experts Group                |
| PHP       | Hypertext Preprocessor                          |
| SQL       | Structured Query Language                       |
| TCP/IP    | Transmission Control Protocol/Internet Protocol |
| UML       | Unified Modeling Language                       |
| WWW       | World Wide Web                                  |

# 1 Úvod

Informační systémy hrají v dnešní době významnou roli. Uživatelé těchto systémů si přejí co nejefektivněji využívat zdroje informací a co nejrychleji informace získávat. Využívání informačních systémů není jen doménou expertů na danou oblast, tedy např. manažerů, kteří se dle získaných informací rozhodují. Manažer požaduje maximalizaci využití zdrojů a naopak minimalizaci času k jejich získání. Zároveň musí někdo zdroje aktualizovat, aby jejich využívání bylo efektivní. Dnes jsou informační systémy vyhledávány i širokou veřejností, neboť informace získané z těchto systémů mohou uživatelům ušetřit čas, finance, atd.

Tato diplomová práce se zabývá tvorbou informačního systému v prostředí internetu. Prostor internetu je voleno kvůli přístupnosti pro co největší spektrum uživatelů. Implementace v prostředí internetu většinou neobtěžuje uživatele instalací nových produktů a je vesměs poskytována zdarma. Nejinak je to s aplikací, která vznikla v rámci této diplomové práce. Aplikace pro evidenci rybářských úlovků nepožaduje instalaci žádného podpůrného programu a je poskytována zdarma. Aplikace je praktickou ukázkou toho, že geografické informační technologie mohou být využívány téměř ve všech oborech.

Tento systém by měl sloužit všem sportovním rybářům, ale mohou jej využívat i ostatní uživatelé. Měl by poskytovat informace o rybářských revírech, úlovcích na těchto revírech i s jejich lokalizací. O vyhledaném revíru jsou uživatelům poskytnuty veškeré informace, které obsahuje Soupis revírů pro Severní Moravu a Slezsko, jehož vydavatelem je Český rybářský svaz (viz kapitola 3). Uživatel si bude moci vyhledat revír v zájmové oblasti a zjistit si úlovky na tomto revíru, a dle této informace se rozhodnout o své návštěvě zvoleného revíru. Taktéž se předpokládá, že uživatel zaeviduje svůj lov do tohoto systému, pokud navštíví nějaký revír a uloví rybu. Zájmovou oblastí je Severní Morava a Slezsko.



## 2 Postup zpracování

**Stanovil jsem následující postup při zpracování diplomové práce:**

- studium problematiky publikace a vizualizace prostorových dat v prostředí WWW
- volba programových prostředků
- analýza a návrh informačního systému
- navržení datového modelu
- vytvoření databáze
- naplnění databáze
- tvorba informačního systému
- testování aplikace

### 3 Český rybářský svaz

V České republice jsou aktivity související se sportovním rybolovem organizovány Českým rybářským svazem. Mezi hlavní aktivity Českého rybářského svazu patří:

- vykonávat rybářské právo ve smyslu zákona o rybářství
- poskytovat služby pro organizační jednotky svazu a jeho členy včetně nákupu a distribuce rybích násad pro další produkci a pro zarybňování rybářských revírů
- rozvíjet a popularizovat rybářský sport

Sportovním rybářstvím se zabývají 3% populace v České republice. V současné době se Český rybářský svaz skládá ze 7 územních svazů, pod které spadá 479 místních organizací. V místních organizacích je evidováno 216 221 dospělých členů a 49 889 členů ve věku od 8 do 18 let. Rybářské revíry v České republice tvoří 34 100 hektarů vodních ploch. Ročně se uloví více jak 3 751 tun různých druhů ryb. (Číselné údaje jsou vztažené k roku 2002.)

Územní výbor pro Severní Moravu a Slezsko sdružuje přes 50 000 členů a to včetně mládeže. O hospodaření na rybářských revírech se podílí 60 místních organizací. Organizace pečují o 5 000 hektarů vodních ploch, a to jak pstruhových, tak i mimopstruhových.

## 4 Popis použitých dat

V diplomové práci jsem pracoval s vektorovými geotady, které mi vymezovaly prostorové rozmístění rybářských revírů. Dále jsem pracoval s popisnými údaji o revírech.

### 4.1 Geodata

Při rozhodování o vektorových zdrojích pro mapový výstup jsem chtěl využít vektorovou databázi ArcČR500, ale při jejím prohledání jsem zjistil, že tato databáze je neúplná. Neobsahovala některé vodní plochy v zájmové oblasti – Severní Morava a Slezsko. Další alternativou bylo použití vektorové databáze DMÚ200, nicméně ani tato databáze neobsahovala veškeré vodní plochy a vodní toky v zájmové oblasti. Z těchto dvou variant byla pro daný účel vhodnější vektorová databáze DMÚ200. Aby ji bylo možno použít, musel jsem vrstvy „*Vodní toky*“ a „*Vodní plochy*“ doplnit o chybějící elementy a atributy.

Jako podklad pro doplnění chybějících dat jsem použil mapy Klubu českých turistů v měřítku 1:50 000, jež byly již geometricky transformovány. Tyto mapy pokrývají celou zájmovou oblast. Pro digitalizaci chybějících vodních ploch jsem použil produkt ArcView GIS 3.2. Nově zdigitalizované plochy byly ukládány do vrstev „*Vodní toky*“ a „*Vodní plochy*“. Dále jsem pomocí ArcView rozdělil nebo naopak sloučil vodní toky tak, aby tvořily revír daný popisem v Rybářském řádu a soupisu revírů platném od 1.1.2003. Dle tohoto soupisu revírů byly oběma vrstvám dopsány nové atributy. Do nového sloupce „*c\_rev*“ bylo připsáno číslo revíru a do sloupce „*id\_pod*“ identifikátor lokality. Přes tyto identifikátory byla později připojena navazující popisná data

Z vektorové databáze DMÚ200 jsem použil tyto vrstvy:

- kraje
- okresy
- lesy
- silnice
- sídla
- vodní plochy

- vodní toky

Dalším zdrojem byla publikace „*Rybářské revíry Severní Moravy a Slezska*“ [1]. Z tohoto zdroje jsem čerpal informace o lokalizaci jednotlivých revírů na mapě.

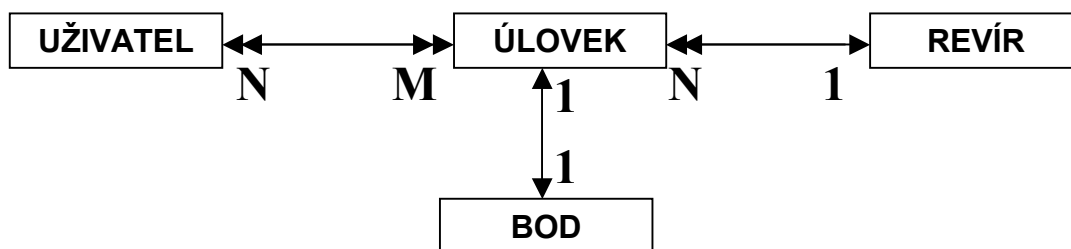
## 4.2 Navazující popisná data

Z předchozího ročníkového projektu, v němž předmětem řešení byla aplikace pro vedení „Místní organizace v Rožnově pod Radhoštěm“, jsem měl již hotovou databázi, která mimo jiné obsahovala tabulku s rybářskými revíry Severní Moravy a Slezska. Tato tabulka nese záznamy, jenž jsou uvedeny v „*Rybářském řádu a soupisu revírů platném od 1.1.2003*“. Měl-li revír více lokalit, byla tato skutečnost zaznamenána v nové tabulce, jenž byla propojena s tabulkou „*Revír*“ vazbou 1..N. Databáze byla ve formátu Microsoft Access. Pro účely této diplomové práce jsem použil tabulku se všemi lokalitami určenými k lovu ryb na Severní Moravě a ve Slezsku. Tabulku jsem vyexportoval do textového formátu, jednotlivé záznamy byly odděleny oddělovačem.

Při řešení této diplomové práce nebylo nutné rozkládat revíry a lokality na nich vazbou 1..N, neboť práce pouze s jednou tabulkou byla v tomto případě efektivnější a pro potřeby aplikace dostačující. Tabulku s lokalitami jsem následně naimportoval do databázového systému MySQL pomocí SQL dotazu.

### 4.2.1 Datový model

Datový model relační databáze je popsán pomocí diagramu entit a vztahů.



Obr. 1 ER diagram databáze

Vztah mezi tabulkou „*Úlovek*“ a „*Bod*“ je 1:1, a to z důvodu práce s výslednou aplikací. Tato vazba akceptuje požadavky dále navrženého řešení.

#### 4.2.2 Databáze „rybydb“

Pro ukládání dat do databáze jsem zvolil databázový systém MySQL a v tomto prostředí jsem vytvořil tabulky pro ukládání dat. Tabulka s lokalitami určenými pro lov ryb byla tedy naimportována pod jménem *Revir*.

##### Výpis atributů z jednotlivých tabulek:

Tabulka **REVIR** (výčet atributů viz *Tab. 1*) již byla podrobně popsána výše, není nutno ji tedy již více specifikovat. Je zde evidováno 444 lokalit určených k lovu ryb na Severní Moravě a ve Slezsku.

| Název pole | Datový typ | Primární klíč | Poznámka  |
|------------|------------|---------------|---|
| id_r       | Integer    | ano           | jedinečný identifikátor pro revír                 |
| c_rev      | integer    | ne            | číslo revíru                                      |
| naz_r      | text       | ne            | název revíru                                      |
| lokal      | text       | ne            | lokalita na revíru                                |
| v_k_u      | text       | ne            | katastrální území, v němž se vodní plocha nachází |
| delka      | float      | ne            | délka toku  |
| rozl       | float      | ne            | rozloha vodní plochy                              |
| info       | text       | ne            | informace o revíru                                |
| omez       | text       | ne            | omezení na revíru                                 |

*Tab. 1* Tabulka atributů pro revíry

Tabulka **ULOVEK** nese záznamy o jednotlivých úlovcích. Do této tabulky jsou ukládány údaje, které vyplní samotní rybáři v zadávacím formuláři ve výsledném informačním systému. (Výčet atributů viz *Tab. 2*)

| Název pole | Datový typ | Primární klíč | Poznámka                           |
|------------|------------|---------------|------------------------------------|
| id_ul      | integer    | ano           | jedinečný identifikátor pro úlovek |
| id_uz      | integer    | ne            | identifikátor rybáře               |
| datum      | date       | ne            | datum uskutečnění lovu             |
| popis      | float      | ne            | popis lovu                         |
| id_rev     | integer    | ne            | revír, na němž došlo k úlovku      |
| foto       | text       | ne            | název souboru fotografie           |

Tab. 2 Tabulka atributů pro jednotlivé úlovky

Tabulka **BOD**. Zde jsou ukládány údaje, jenž udávají místo lovu. Jsou zde zaznamenávány X a Y souřadnice bodu lovu, dále číslo revíru, na němž došlo k úlovku. (Výčet atributů viz Tab. 3)

| Název pole | Datový typ | Primární klíč | Poznámka                               |
|------------|------------|---------------|--|
| id_bod     | integer    | ano           | jedinečný identifikátor pro bod        |
| x_sour     | float      | ne            | x souřadnice bodu na mapě              |
| y_sour     | float      | ne            | y souřadnice bodu na mapě              |
| id_rev     | integer    | ne            | číslo revíru, kde došlo k ulovení ryby |
| id_ul      | integer    | ne            | úlovek, jemuž náleží daný bod          |

Tab. 3 Tabulka atributů pro ukládání X a Y souřadnice bodu lovu

Tabulka **UZIVATEL** nese záznamy o rybářích, tedy jméno uživatele či jeho přezdívku. Taktéž je zde ukládán e-mail na uživatele. (Výčet atributů viz Tab. 4)

| Název pole | Datový typ | Primární klíč | Poznámka  |
|------------|------------|---------------|---|
| id_uz      | integer    | ano           | jedinečný identifikátor pro uživatele                     |
| jmeno      | char       | ne            | jméno uživatele - může se jednat o jméno, přezdívkou, ... |
| email      | char       | ne            | e-mail na uživatele                                       |

Tab. 4 Tabulka atributů s údaji o uživateli

## 5 Výběr programového vybavení

Z práce vyplývá, že informační systém má fungovat v prostředí internetu, tedy bez lokálně uložených programových aplikací a dat. K řešení tohoto problému bylo nutno nejprve nalézt odpovídající technologie. Zde uvádím přehled softwarových produktů, které umožňují vytvoření a běh informačních systémů v prostředí WWW.

### 5.1 Internetové prostředí

S nárůstem internetu za poslední desetiletí souvisí i velký nárůst poskytovaných služeb, které jsou dostupné jeho prostřednictvím. V dřívějších dobách byly síťové aplikace dostupné pouze pro velké organizace, neboť náklady na zřízení a provozování vlastní sítě byly vysoké.

Naproti tomu internet dnes představuje levnou a téměř všudepřítomnou síť. Uživatelem internetu může být kdokoli, kdo splňuje určité požadavky na jeho funkčnost. Uživatel musí mít počítač se standardizovaným webovým prohlížečem (Internet Explorer, Netscape, Mozilla, ...). Tento počítač musí mít připojení k Internetu. Aby mohl webový prohlížeč reagovat na požadavky uživatele, musí být tento požadavek zasílán na vzdálený server. Ukažme si tedy princip možné síťové architektury pro komunikaci v prostředí internetu.

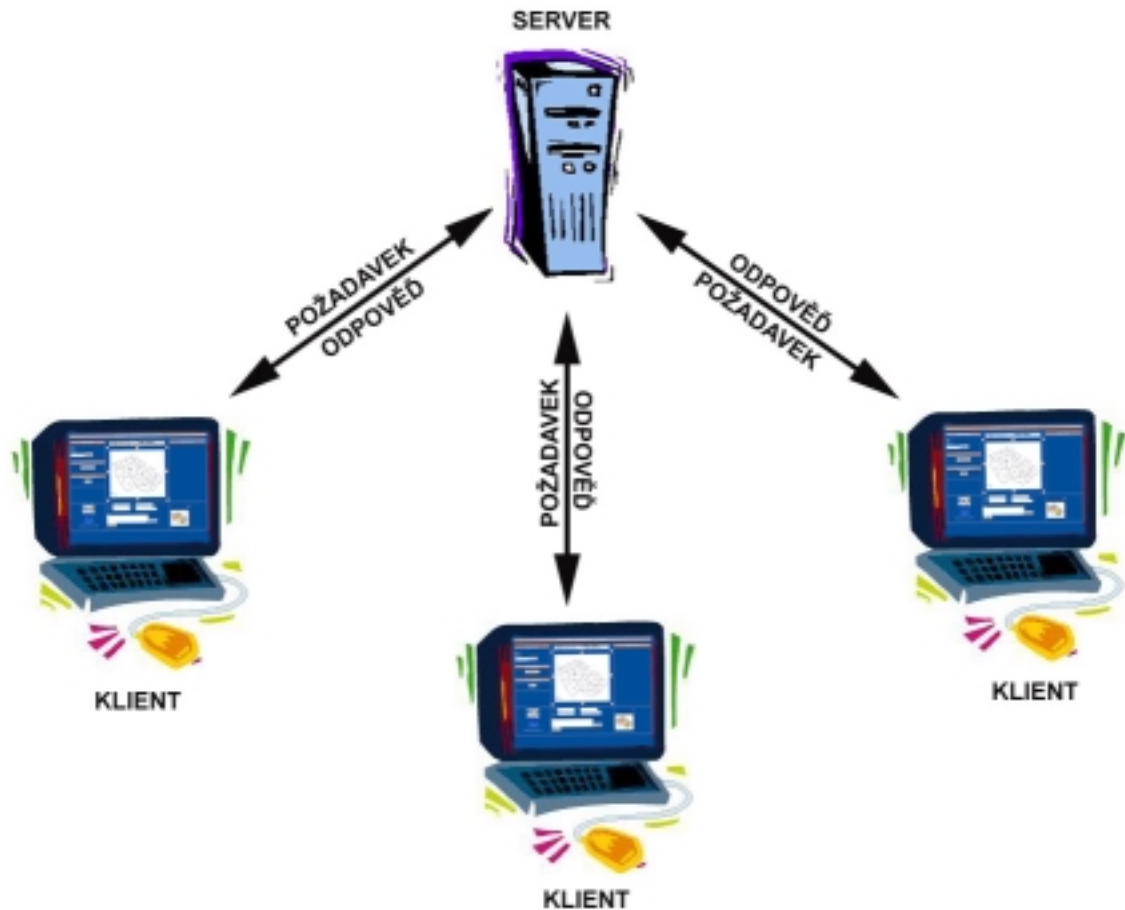
#### 5.1.1 Klient-server

Technologie využívaná k řešení problémů v prostředí WWW se nazývá klient-server. Klientem je uzel (pracovní stanice) nebo program, jenž vyžaduje službu po serveru

(viz *Obr. 2*). Takto je pracovně zatížen především server. Výstupní stránka webového prohlížeče je v co největší míře zpracována na straně serveru. Použitá technologie neklade tak vysoké nároky na komponentové a programové vybavení u klienta, jako spíše na straně severu. Obsluha klientského prostředí má být v internetových aplikacích nenáročná. Při vzájemné komunikaci klienta se serverem se využívají přenosové protokoly HTTP (služba WWW). Síť by neměla přenášet velké datové balíky, aby nedocházelo k velkým prodlevám při čekání klienta na zpracování požadavku. Spolehlivost komunikace není až tak závislá na klientských stanicích, jako spíše na správně „vyladěném“ serveru. Kvalita zobrazované informace nesmí záviset na schopnostech koncového uživatele. Použitá technologie musí vytvářet dokonale bezpečný systém. Musí být zabezpečen proti zhroucení, chybným zápisům, hackerům, a také proti úniku informací.

Jednoduchý princip architektury klient-server je prezentován na názorném příkladu (viz *Obr. 2*). Jedná se o službu WWW, klientem je libovolný internetový prohlížeč (Internet Explorer, Netscape, Mozilla, ...). S tímto klientem komunikuje uživatel, který mu zadává požadavky. Klient převede požadavky uživatele do protokolu HTTP a odešle je serveru. Server tyto požadavky přijme, zpracuje a odpoví klientovi. Přesný formát požadavku a odpovědi je definován ve specifikaci protokolu HTTP. Odpověď zobrazí klient.



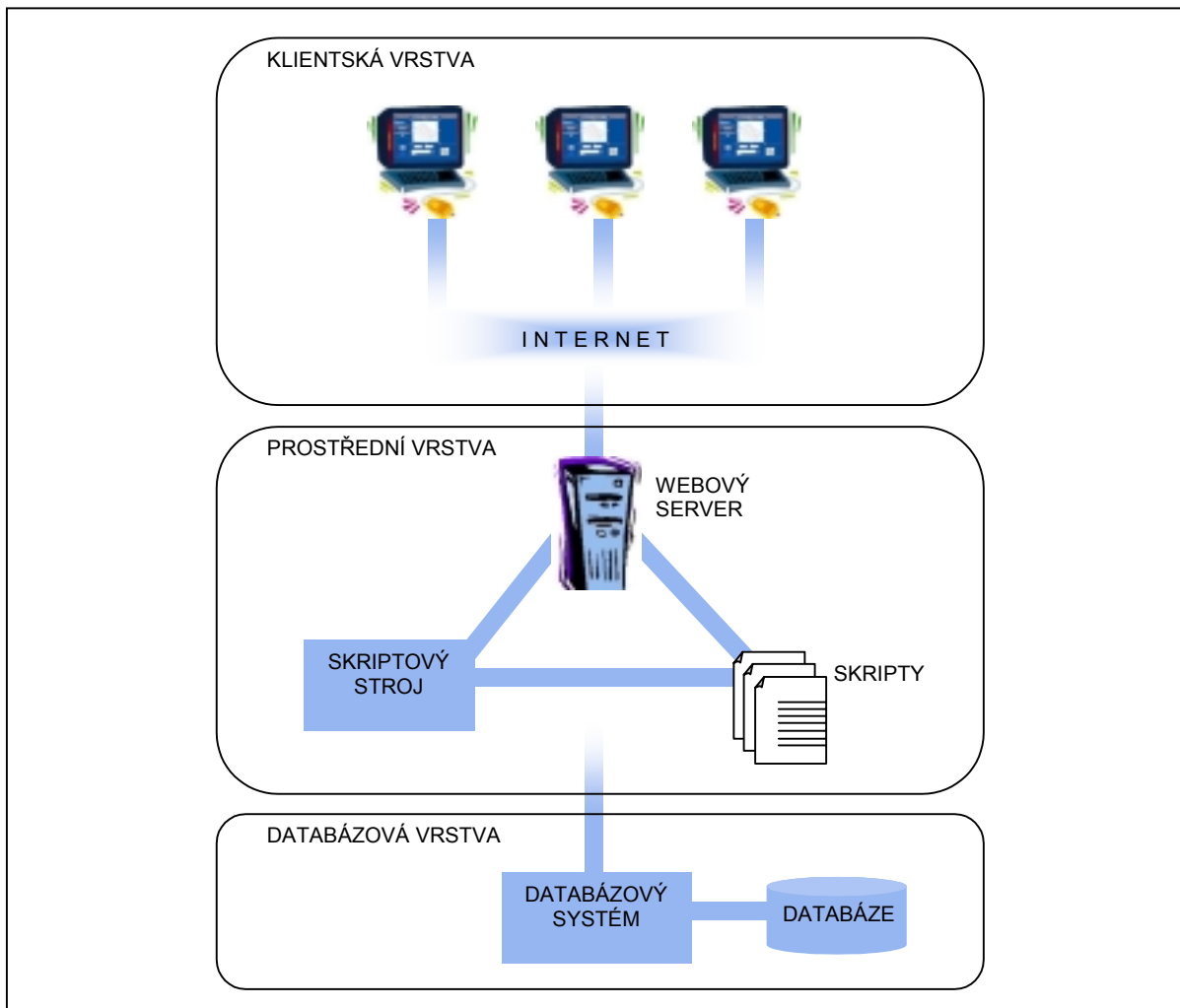


Obr. 2 Architektura komunikace klient – server

Pro technologii klient-server platí rozdělené zpracování mezi klienta a server, případně mezi tři vrstvy (kapitola 5.1.2 Třívrstvá architektura webové databázové aplikace).

### 5.1.2 Třívrstvá architektura webové databázové aplikace

Základem aplikace je *databázová vrstva* (viz Obr. 3), kterou tvoří databázový systém, jenž obstarává správu databáze obsahující data, jejich vytváření, odstraňování, modifikaci. Nad databázovou vrstvou je *prostřední vrstva*, jenž obsahuje aplikační logiku a zajišťuje komunikaci mezi ostatními vrstvami. Třetí vrstvou je *klientská vrstva*, jež je obvykle reprezentována webovým prohlížečem, který umožňuje komunikaci uživatele s aplikací.



Obr. 3 Třívrstvá architektura webové databázové aplikace [2]

Třívrstvá architektura je využívána především u *tenkých klientů*, lze se s ní setkat i u silnějšího klienta. Tenkého klienta představuje webový prohlížeč, v klientské vrstvě pracuje jen velmi málo aplikační logiky. Webový prohlížeč pouze zasílá požadavky pomocí HTTP protokolu a poté zobrazuje odpovědi, nejčastěji jako dokumenty HTML. HTML je formátovací jazyk, jenž je nástrojem pro strukturování a prezentaci informací v aplikacích webového prohlížeče.

Příkladem silnějšího klienta lze prezentovat javový applet vlastní výroby. Uživatel si tento applet stáhne na svůj počítač, kde na vlastní platformě spouští část aplikační logiky. Applet nadále komunikuje s prostřední vrstvou, která tvoří rozhraní pro komunikaci k databázové vrstvě.

*Silný klient* je součástí dvouvrstvého řešení (architektura klient-server). Klientská vrstva obsahuje většinu aplikační logiky, zatímco serverovou vrstvu tvoří samotný databázový systém.

Pro řešení diplomové práce jsem se na základě funkčnosti systému rozhodl pro architekturu **klient-server**. Tento model zajišťuje všechna kritéria kladená na výsledný informační systém.

## **5.2 Webové servery**

Webové servery bývají také označovány jako servery HTTP. Funkcí webových serverů je naslouchání požadavkům HTTP, příjem příchozích požadavků HTTP, obsluha požadavků a vrácení odpovědí HTTP s požadovanými výstupy. Existuje více webových serverů, a to jak komerčních, tak i nekomerčních. Ukažme si dva nejpoužívanější webové servery.

### **5.2.1 Apache**

Jedná se o webový server s otevřeným zdrojovým kódem, jde tedy o nekomerční produkt a je možno si jej stáhnout z adresy <http://www.apache.org>. Instalace a konfigurace pro účely provozování webových aplikací je docela jednoduchá. Je určen pro operační systém Linux i Windows 32bit. Server Apache je dnes ve verzích 1.3 a 2.0. Verze 2.0 se od verze 1.3 liší především lepší podporou, výkonností a stabilitou na jiných pracovních stanicích než na Unixových. Podporuje nový protokol IPv6, což je nová verze protokolu IP z balíku síťových protokolů TCP/IP. Jsou doplněny filtrovacími moduly.

### **5.2.2 Internet Information Server**

Jedná se o produkt firmy Microsoft Corporation. Existuje ve více verzích a je součástí Windows .NET, tedy serverových variant Windows 2000, XP. Nejnovější verzí IIS je verze 6.0, ta se od IIS 5 liší větší bezpečností, jednodušší správou a odolností. IIS 6 je standardně instalován se všemi službami vypnutými a administrátor serveru si požadované služby zapíná sám.

Nyní se objevuje zcela nový produkt Windows Server 2003 – Web Edition, ovšem ten nebude nabízen v maloobchodní síti. Systém Windows Server 2003 je vytvořen na základě systému Windows 2000 Server. Windows Server 2003 je určen pro tvorbu

a poskytování hostitelských služeb webovým aplikacím, webovým stránkám a webovým službám XML. Veškeré produkty jsou komerční.

Pro práci jsem vybral HTTP server **Apache 1.3**. Hlavní prioritou oproti IIS je skutečnost, že se nejedná o komerční produkt. Verze Apache 2.0 nebylo třeba, neboť pro účely diplomové práce postačovala verze 1.3.

### **5.3 Skriptovací jazyk PHP**

Jedná se o skriptovací jazyk s otevřeným zdrojovým kódem. Skripty PHP se dají vkládat do statických souborů HTML. Jako Apache, tak i PHP je flexibilní vůči operačnímu systému a platformě. Verzí jazyka PHP existuje velká řada a je neustále obměňována novými verzemi. Nyní se jazyk PHP nachází ve verzi 4. Oproti verzi PHP3 je interní skriptový stroj kompletně přepsaný. Ve verzi 3 byly skripty vykonávány po řádcích, tedy každý řádek skriptu bylo nutné načíst, analyzovat a provést. Pokud se příkaz v těle opakoval víckrát, znamenalo to jeho opakovanou interpretaci. Z toho vyplývá velké zpomalení. PHP 4 pracuje jiným způsobem. Skript se přečte, analyzuje a zkompiluje do mezilehlého formátu, který je potom vykonán tzv. exekutorem skriptového zdroje PHP4 Zend. Skriptovací jazyk PHP4 je rozdělen do dvou komponent. Jedná se o knihovny funkcí a hlavní stroj Zend pro zpracovávání skriptů.

Z předešlého výkladu vidíme výhody verze PHP4, tudíž jsem použil verzi PHP 4.0.6.

### **5.4 Databázový systém**

Hlavní úlohou databázového systému je ukládání, vyhledávání a správa dat. Nejvíce využívaným databázovým systémem v rámci internetu je MySQL. Na základě studia jeho výhod a nevýhod, jsem se jej rozhodl použít.

#### **5.4.1 MySQL**

MySQL je relační databázový systém střední velikosti. Svoji konstrukcí se hodí ke správě webových databází. Pro dotazování nad databází je využíván strukturovaný dotazovací jazyk SQL. Je to standardní jazyk pro komunikaci s relačními databázemi. Nejedná se však pouze o dotazovací jazyk, ale taky slouží coby nástroj pro vytváření databází, správu a zabezpečení. MySQL podporuje skriptovací jazyk PHP.

## 5.5 Vizualizace prostorových dat v prostředí WWW

V dnešní době již máme mnoho možností, jak prezentovat prostorová data v prostředí WWW. Ať již se jedná o komerční, tak i nekomerční využití. Ač by to mohlo vypadat, že nekomerční řešení nám neposkytne tak silné nástroje jako komerční produkt, tato domněnka by byla mylná. Existuje již řada “open source” projektů, tedy volně dostupných produktů, jenž mohou směle sekundovat komerčnímu řešení. Mezi nejznámější produkty z řad komerčních můžeme zařadit *ArcIMS*, jedná se o produkt firmy ESRI. Dalším produktem je *GeoMedia Web Map* od společnosti Intergraph. O komerčních produktech se nebudu více rozepisovat, neboť jejich pořizovací náklady jsou vysoké, a proto nemohou být využity v této diplomové práci. K nejpoužívanějším “open source” projektům patří produkt *MapServer* z Minesotské univerzity.

### 5.5.1 Mapový server MapServer

Jedná se o mapový server z Minesotské univerzity, který je volně ke stažení. Tento mapový server generuje na základě uživatelem zadaných parametrů mapové výstupy. Ty jsou ukládány ve formě rastrových obrázků, které následně načítáme do internetových aplikací. Nemusí se ovšem jednat pouze o internetové aplikace, ale i o intranetové či extranetové aplikace. Uživatel tedy pracuje s rastrovými obrázky. Jako vstupy mohou sloužit vektorové soubory, rastrové soubory, ale mohou být také ve formě číselných souřadnic, z nichž se následně vykreslí vrstva. Vše se odehrává na straně serveru, z toho vyplývá, že uživatel nemusí mít nainstalován žádný speciální software pro zobrazování potřebných dat. PHP/Mapscript je modulem MapServeru. Jedná se o knihovnu \*.dll, kterou kopírujeme do extenzí jazyka PHP. Tím jsem již nastínil, že tato knihovna obsahuje kolekci tříd pro práci s digitálními mapami. Dovoluje využívat výhody jazyka PHP. S ohledem na použití modulu PHP/Mapscript verze 3.6 bylo nutné použít skriptovací jazyk PHP 4.0.6, neboť přímo pro tuto verzi PHP byla knihovna Mapserveru zkompileovaná.

## 5.6 Skriptovací jazyk JavaScript

JavaScripty mohou být vkládány přímo do HTML stránek. JavaScript je klientský skript, to znamená, že se odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván. Ačkoli JavaScript připomíná Javu, je zde velký rozdíl. Java je programovací jazyk, který je komplexnější než JavaScript. Javascript byl vytvořen jako lehce srozumitelný jazyk použitelný v internetových stránkách.

## 5.7 ArcView GIS 3.2

Produkt firmy Esri, Inc. Jedná se o aplikaci, která umožňuje práci se všemi informacemi, které mají vztah k mapě. Přistupuje ke geografickým i k tabulkovým datům, umožňuje jejich zobrazování, dotazy, výběry, analýzy a samozřejmě i prezentaci výsledků práce.

## 5.8 Přehled zvoleného programového vybavení

|   |                                |
|---|--------------------------------|
| <b>Webový server:</b>                             | Apache, verze 1.3              |
| <b>Skriptovací jazyk:</b>                         | PHP, verze 4.0.6<br>JavaScript |
| <b>Mapový server:</b>                             | MapServer 3.6, php-mapscript   |
| <b>Databázový systém:</b>                         | MySQL, verze 3.23              |
| <b>Editace a digitalizace vektorových vrstev:</b> | ArcView GIS 3.2                |

## 6 Analýza a návrh informačního systému

Pro analýzu systému jsem použil objektově orientovanou analýzu. Celý proces realizace produktu je při této analýze popsán jistými modely, jenž nám umožní lépe pochopit vyvíjený systém. K tomuto popisu jsem použil jazyka UML (Unified Modeling Language). „UML je jazyk umožňující specifikaci, vizualizaci, konstrukci a dokumentaci artefaktů softwarového systému“ [2].

### 6.1 Slovní specifikace požadavků

Cílem práce je vytvořit informační systém, který bude sloužit k evidenci rybářských úlovků v prostředí internetu. Po systému je požadováno, aby byl schopen zobrazovat mapu revírů

a měl možnost zobrazovat podrobné informace k vybranému revíru. Dále systém musí umožňovat vyhledávat revíry dle jeho čísla nebo názvu revíru. Informační systém musí být schopen předkládat uživateli podrobný přehled o úlovcích a to jak na jednotlivých revírech, tak i v rámci celého Severomoravského kraje. Uživatel může zadávat nový úlovek nebo vyhledávat informace o revíru či úlovcích. Pro práci s mapovým oknem musí být použity translokační nástroje, jako jsou zoom-in, zoom-out, pan, centrování, pohyb s mapou pomocí šipek. Pro práci s mapou je používán produkt MapServer z Minesoty. O údržbu by se měl starat administrátor informačního systému.

Informační systém by měl být navržen takovým způsobem, aby nebyla potřeba při jeho provozu používat komerční software.

#### **Byly stanoveny základní požadavky na informační systém:**

- evidovat jednotlivé úlovky na určitém revíru včetně jejich lokalizace a údajů o:
  - jménu lovce
  - e-mailu
  - datumu lovu
  - revíru, na němž se uskutečnil lov
  - popisu lovu
  - fotografii úlovku
- zobrazovat mapu revírů a k ní příslušné informace

- vytvořit translokační nástroje pro práci s mapou
- zobrazit informace o revíru + možnost jeho lokalizace v mapě
- zobrazit seznam úlovků k danému revíru + lokalizace úlovku
- zobrazit fotografii k úlovku
- zobrazit seznam všech úlovků na všech revírech

Již zde se objevují první pojmy, které je potřeba vysvětlit pro lepší chápání dané problematiky:

|                             |   |
|-----------------------------|---|
| <b><i>Administrátor</i></b> | osoba, která je pověřena či oprávněna ke správě informačního systému  |
| <b><i>MapServer</i></b>     | mapový server, jenž generuje na straně serveru dle zadaných požadavků rastrové obrázky, které se vracejí klientovi          |
| <b><i>Uživatel</i></b>      | osoba, která využívá informační systém, prohlíží data a vkládá nové záznamy o úlovcích                                      |
| <b><i>Revír</i></b>         | jedná se o vodní plochu určenou k lovu ryb, která je uvedena v Soupisu revírů pro Severní Moravu a Slezsko                  |
| <b><i>Úlovek</i></b>        | reprezentuje jeden úlovek jednoho rybáře na revíru  |
| <b><i>Jméno</i></b>         | identifikace osoby, jenž zaznamenala ulovení ryby   |
| <b><i>E-mail</i></b>        | spojení na uživatele pomocí elektronické pošty  |
| <b><i>Datum</i></b>         | uvedení data, kdy došlo k ulovení ryby  |
| <b><i>Popis lovu</i></b>    | přesnější popis lovu rybáře; měly by zde být uloženy informace o způsobu lovu, nástraze, času, místu, okolnostech lovu, ... |
| <b><i>Zoom-in</i></b>       | translokační funkce, jenž nám umožňuje změnu měřítka daného území, tedy zmenšení měřítka zobrazení dat                      |



|                               |  |
|-------------------------------|--|
| <b><i>Zoom-out</i></b>        | translokační funkce, jenž nám umožňuje změnu měřítka daného území, tedy zvětšení měřítka zobrazení dat |
| <b><i>Pan</i></b>             | translokační funkce, jenž nám umožňuje posun daného území  |
| <b><i>Centrování mapy</i></b> | translokační funkce, jenž nám umožňuje centrování daného území na střed mapového okna                  |
| <b><i>Pohyb s mapou</i></b>   | translokační funkce pro pohyb se uskutečňuje pomocí šipek  |
| <b><i>Mapové okno</i></b>     | okno, do kterého je načten vygenerovaný obrázek mapovým serverem                                       |

## 6.2 Model byznys procesů a doménový model

Doménová analýza se nám snaží poskytnout pohled na objekty v informačním systému, jejich vztahy a chování.

„Třída je popis množiny objektů majících společnou strukturu, chování, vztahy a sémantiku“[3]. Pro přehlednost a pochopení tříd uvádím jejich podrobnější textový popis.

### TŘÍDA

### POPIS

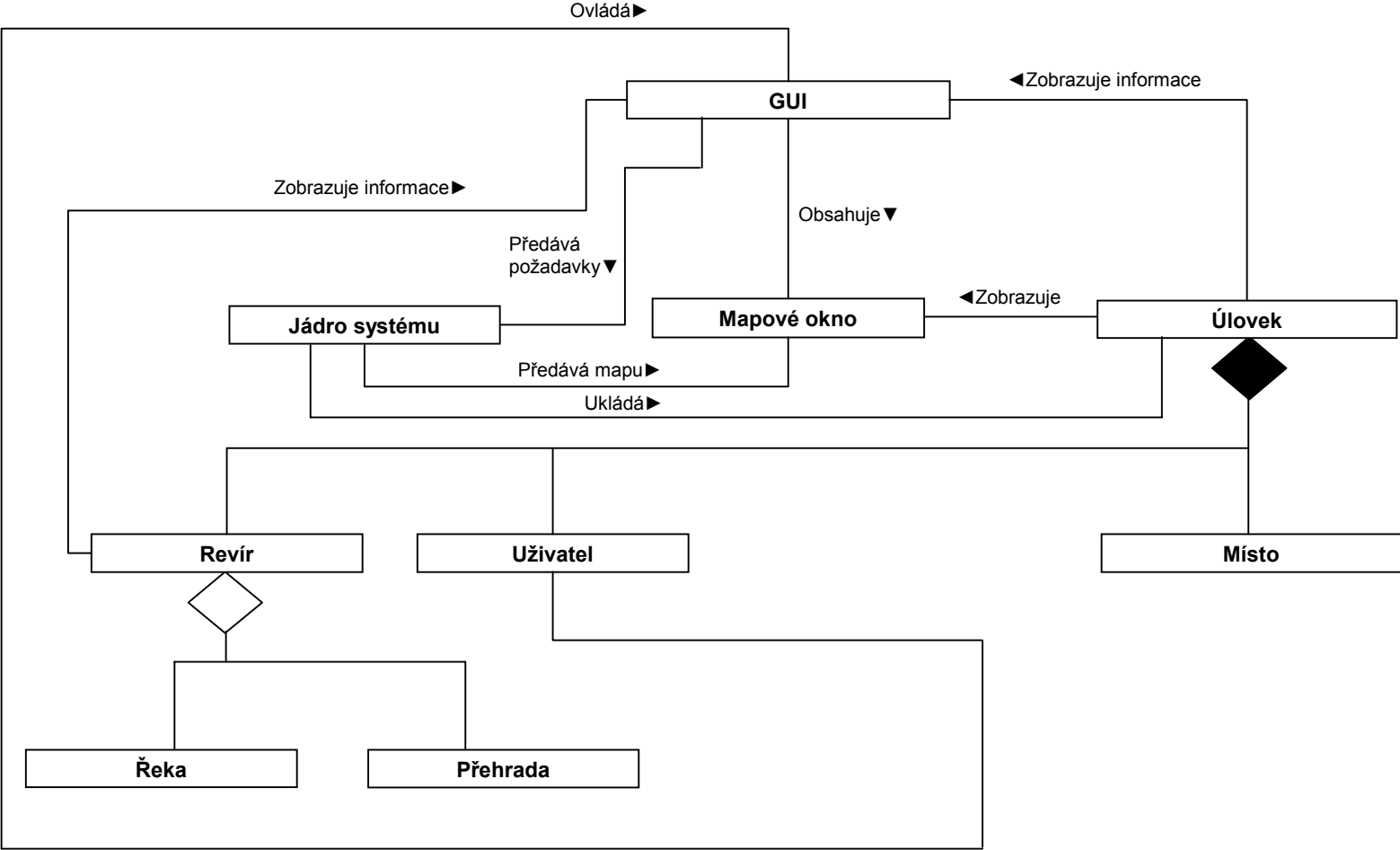
***Mapové okno*** Zde je načítán rastrový obrázek, jenž je vygenerován mapovým serverem. Obrázek je generován na podkladě předaných parametrů, jež určuje uživatel. Tento obrázek je vkládán do HTML stránky. Jakmile provedeme jakoukoli změnu (zoom, pan, ...), je opět generován nový rastr. V našem případě jsou na Mapové okno používány translokační funkce popsané výše (kapitola 6.1 Slovní specifikace požadavků). V této třídě je používána metoda pro zjištění souřadnice při kliknutí do Mapového okna.

### ***Uživatel***

Představuje uživatele jakožto osobu, jež má možnost prohlížet rybářské revíry a informace k nim vztahované. Dále může procházet úlovky na daných revírech a prohlížet jejich lokalizaci v Mapovém okně. U uživatele je evidováno jméno a e-mailová adresa. Uživatel zadává nové úlovky, které jsou ukládány do databáze, a posléze jsou přístupné pro všechny uživatele.

- Jádro systému** Jedná se o imaginární třídu, kterou v informačním systému reprezentuje databázový server, zde se jedná o MySQL. Do této databáze jsou ukládány informace o úlovcích a bodech k těmto úlovkům. Dále jsou zde uloženy informace o revírech a uživatelích. Jádro systému hlídá přihlašování uživatelů a správu účtů. Dále reprezentuje mapový server, jenž slouží pro obsluhu Mapového okna. Tato třída nemá tudíž žádné vlastnosti, ale pouze metody.
- Revír** Třída, která reprezentuje rybářské revíry (v našem případě Severní Moravy a Slezska). U této třídy je evidováno číslo revíru, které je celorepublikově jedinečné. Název revíru u této třídy představuje název daného toku, přehrady, .... Dále je zde uveden atribut délka (pouze u toků), rozloha vodní plochy či toku. Jsou zde uvedeny informace o daném revíru a omezení na revíru. Každý revír ještě nese informaci, v kterém katastrálním území se nachází.
- GUI** Zkratka pro grafické uživatelské rozhraní, což znamená prostředí pro komunikaci s uživatelem. Uživatel zde určuje, co se v daném okamžiku bude provádět. Ať již jde o translokační funkce, nebo o prohlížení revírů a úlovků.
- Úlovek** Třída, která představuje jednotlivé úlovky na revírech. Z předchozích popisů je zřejmé, k čemu tato třída slouží, tudíž ji nebudu dále rozepisovat.
- Bod** Jakmile dojde k zadání úlovku do systému, má uživatel možnost lokalizovat tuto událost v mapě. Právě k tomuto účelu je zde třída Bod. Atributy jsou revír, na kterém došlo k úlovku, souřadnice ve směru osy X a souřadnice ve směru osy Y a číslo úlovku, k němuž se bod vztahuje.
- Řeka** Třída, jež reprezentuje revír danou řeku.
- Přehrada** Třída, která reprezentuje revír danou přehradu, rybník, jezero, ....

Doménový model



Obr. 4 Doménový model

## 6.3 Diagram aktivit

Diagram aktivit slouží k popisu jednotlivých procesů a reprezentuje jejich stavy a přechody mezi nimi. Při vlastní tvorbě informačního systému jsem vypustil část přihlašování uživatele (login), jež je zde zaznamenána. Přihlašování uživatele by bylo vzhledem k účelu této aplikace poněkud zbytečné a mohlo by nejednoho uživatele odradit od jejího použití. Přihlašování jsem v tomto diagramu ponechal kvůli eventuálnímu rozšíření informačního systému.

Vysvětlení akčních stavů a přechodů (řízení), jenž by mohly být nesrozumitelné, jsou popsány zde.

| <b>AKČNÍ STAV</b>                     | <b>VYSVĚTLENÍ</b>   |
|---------------------------------------|---|
| <i>Login</i>                          | umožňuje přihlášení uživatele do informačního systému; po přihlášení již může pracovat s aplikací |
| <i>Odhlášení</i>                      | slouží k zabezpečení, které umožní bezpečné odhlášení z informačního systému                      |
| <i>Zobrazení podrobností o revíru</i> | vypíše informace o revíru (název, číslo, délku, rozlohu, v k. u., informace, omezení)             |
| <i>Zobrazení lokality v mapě</i>      | tento stav zobrazí v mapovém okně požadovanou lokalitu  |
| <i>Umístění bodu v mapě</i>           | kliknutím do mapového okna se převezmou souřadnice kurzoru a uloží se do databáze                 |
| <i>Zadání podrobností o úlovku</i>    | zadáme podrobnosti o úlovku (jméno, email, datum, revír, popis lovu)                              |
| <i>Seznam rybářských úlovků</i>       | vypíše seznam všech rybářských úlovků na všech revírech   |
| <i>Výpis informací o úlovku</i>       | vypíše informace o jednom konkrétním úlovku   |

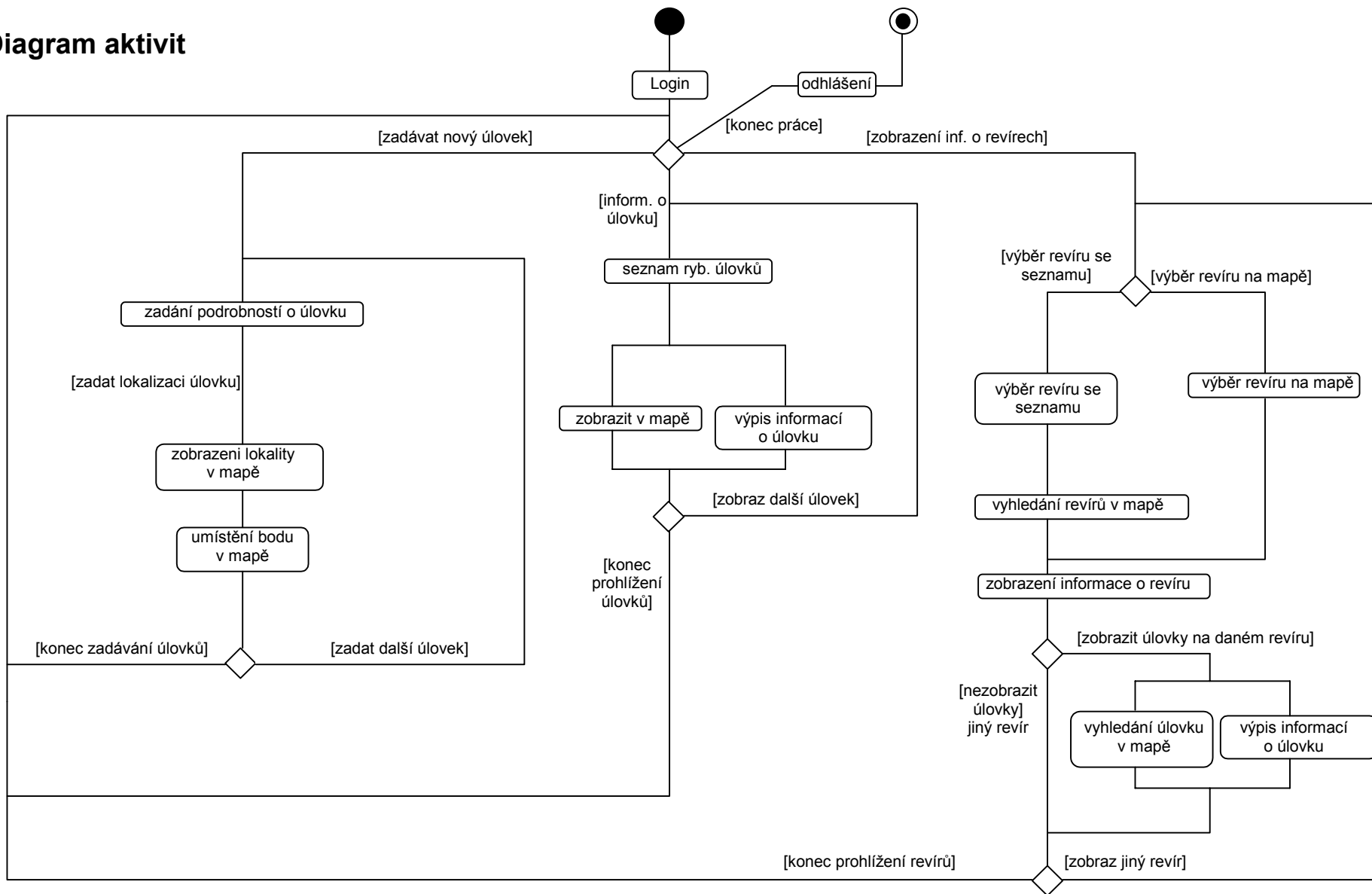
|  |   |
|--|---|
| <b><i>Výběr revíru ze seznamu</i></b>      | možnost vybrat revír z výběrového seznamu                       |
| <b><i>Výběr revíru na mapě</i></b>         | pomocí translokačních funkcí vyhledám revír, jenž mne zajímá    |
| <b><i>Vyhledání revíru v mapě</i></b>      | při výběru revíru ze seznamu mám možnost zobrazit revír na mapě |
| <b><i>zobrazení informace o revíru</i></b> | vypíše se podrobnosti o jednom specifickém revíru               |

*Přechody (řízení)* nebudu blíže vysvětlovat, neboť jsou zřejmé z podrobností o akčních stavech.

Na další straně je uveden již popsany diagram aktivit (viz Obr. 5).

# Diagram aktivit

30



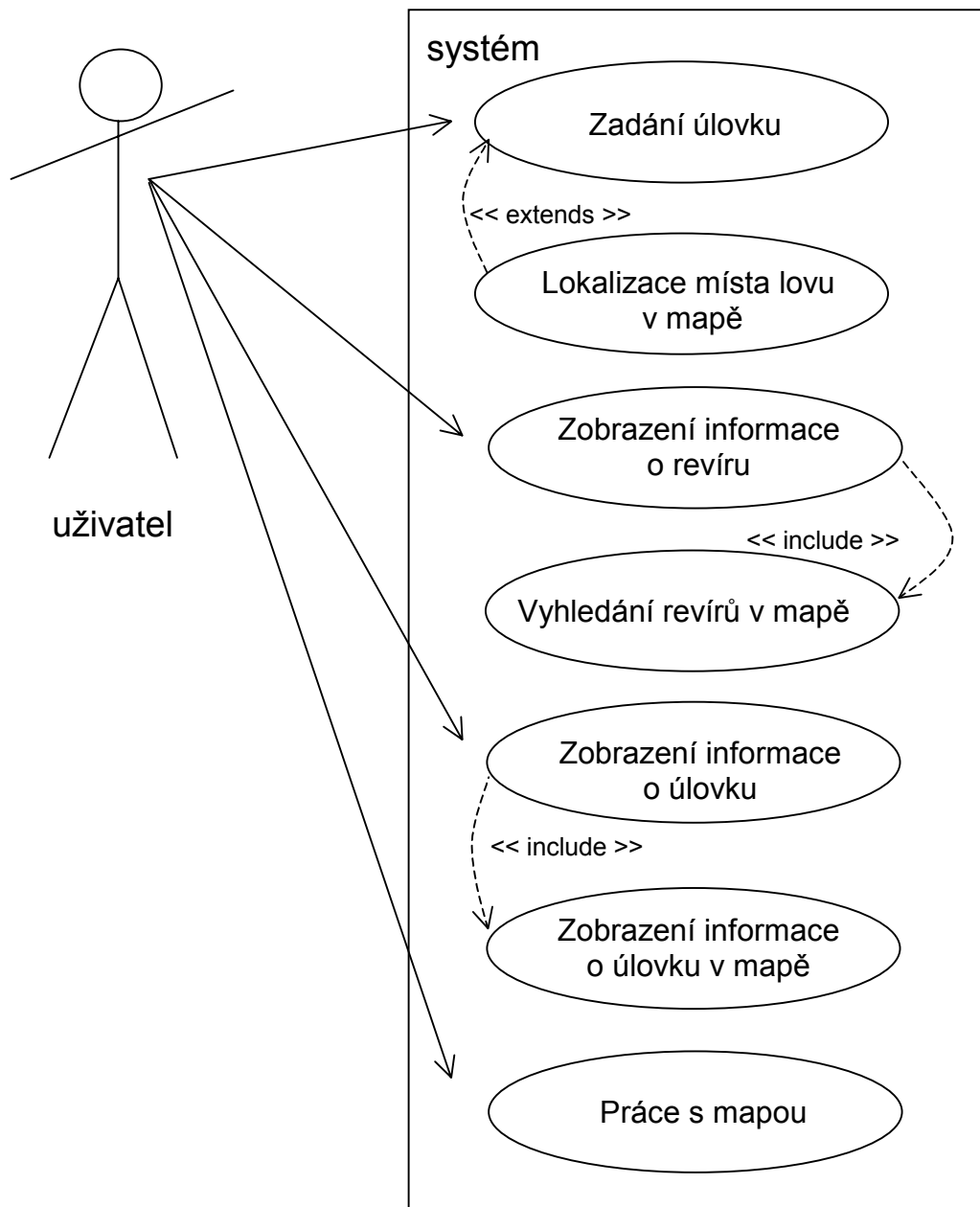
Obr. 5 Diagram aktivit

## 6.4 Model případů užití

Případ užití neboli USE CASE slouží k vymezení toho, co existuje vně systému (v našem případě uživatel) a toho, co působí uvnitř systému – tj. to, co je systémem prováděno.

### Vysvětlení pojmů:

|  |   |
|--|---|
| <i>Zadání úlovku</i>                       | případ užití, který vyplývá z předchozích popisů, tedy zadávání požadovaných údajů  |
| <i>Lokalizace místa lovu v mapě</i>        | kliknutím do mapového okna se převezmou souřadnice kurzoru a uloží se do databáze   |
| <i>Zobrazení informace o revíru</i>        | zde se vypíše podrobnosti o revírech  |
| <i>Vyhledání revíru v mapě</i>             | při výběru revíru mám možnost zobrazit revír na mapě  |
| <i>Zobrazení informace o úlovku</i>        | vypíše informace o úlovcích   |
| <i>Zobrazení informace o úlovku v mapě</i> | zobrazí vybraný úlovek v mapě   |
| <i>Práce s mapou</i>                       | zde má uživatel možnost využít translokačních funkcí informačního systému, tedy zoom – in, zoom – out, pan, center, full extent |



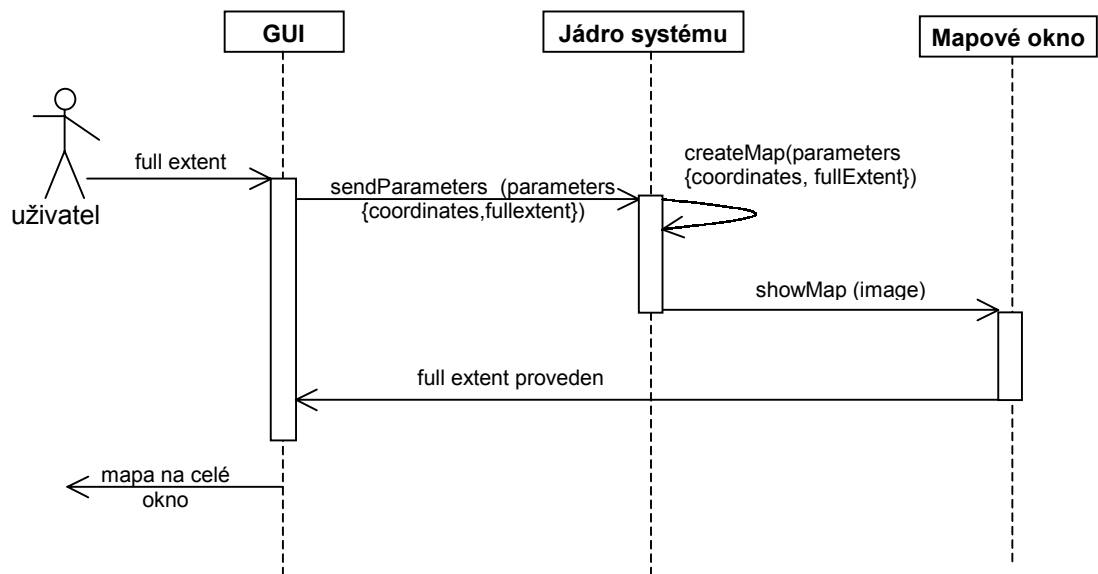
Obr. 6 Diagram případů užití



## 6.5 Sekvenční diagramy

Jedná se o diagramy, jenž sledují, jaké zprávy jsou mezi objekty zaslány z pohledu času.

### 6.5.1 Sekvenční diagram – práce s mapou full extent

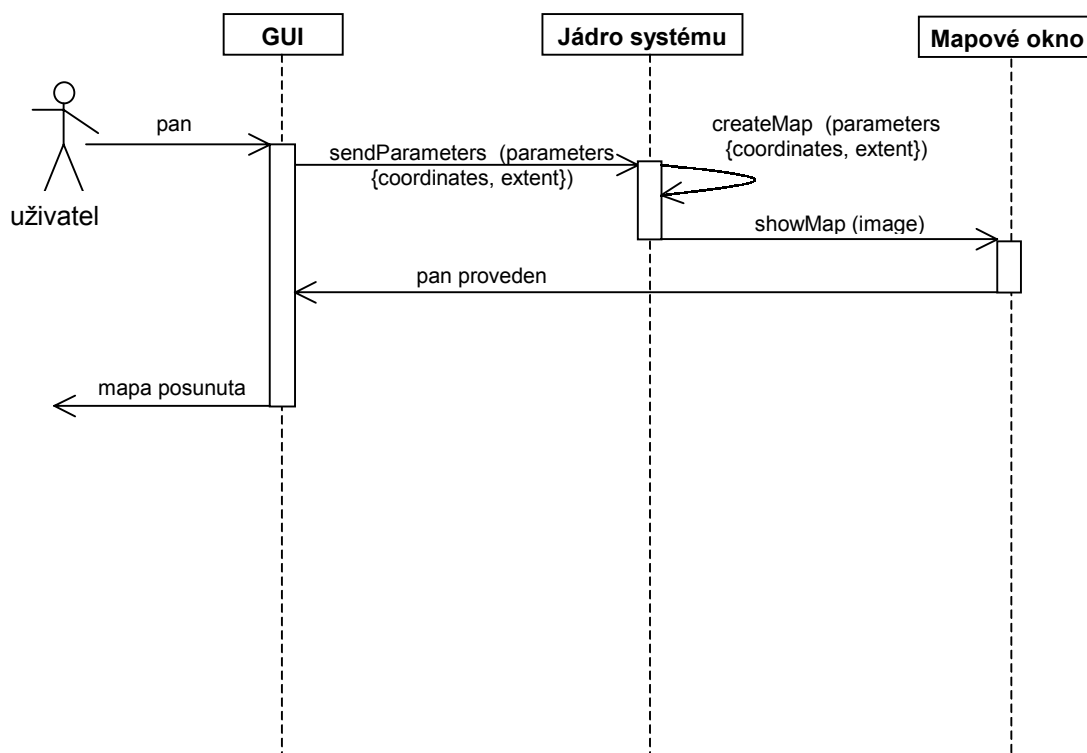


Obr. 7 Sekvenční diagram pro práci s mapou (full extent)

Scénář:

|   |
|---|
| <b>Práce s mapou full extent</b>  |
| <b>Účastníci:</b> uživatel  |
| <b>Vstupní podmínky:</b> vybrání požadované trans. funkce   |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"><li>1. uživatel vybere požadovanou trans. funkci</li><li>2. GUI pošle dotaz na Jádro systému</li><li>3. Jádro systému vygeneruje nový obrázek</li><li>4. do Mapového okna se načte vygenerovaný obrázek</li><li>5. uživatel vidí celou mapu</li></ol> |
| <b>Výstupní podmínky:</b> výstup odpovídá požadované funkci   |
| <b>Alternativní tok:</b> možnost vybrání jiné funkce  |

## 6.5.2 Sekvenční diagram – práce s mapou pan

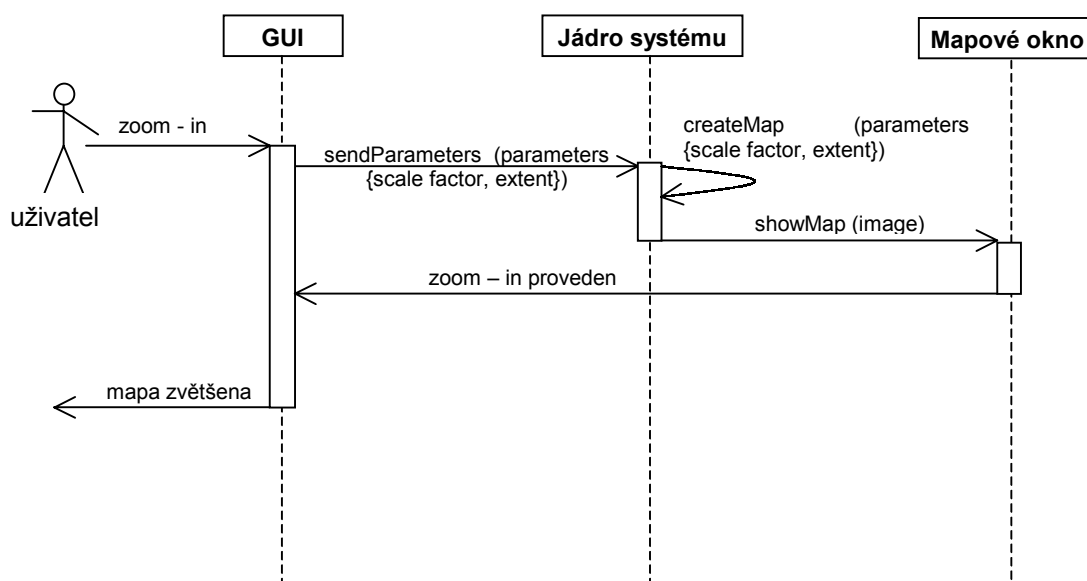


Obr. 8 Sekvenční diagram pro práci s mapou (pan)

### Scénář:

|  |
|--|
| <b>Práce s mapou pan</b>   |
| <b>Účastníci:</b> uživatel   |
| <b>Vstupní podmínky:</b> vybrání požadované trans. funkce + určení posunu  |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"> <li>1. uživatel vybere požadovanou trans. funkci</li> <li>2. klikne do Mapového okna, tím určí posun mapy</li> <li>3. GUI pošle dotaz na Jádro systému</li> <li>4. Jádro systému vygeneruje nový obrázek</li> <li>5. do Mapového okna se načte vygenerovaný obrázek</li> <li>6. uživatel vidí posunutou mapu</li> </ol> |
| <b>Výstupní podmínky:</b> posun v mapě   |
| <b>Alternativní tok:</b> možnost vybrání jiné funkce   |

### 6.5.3 Sekvenční diagram – práce s mapou zoom-in

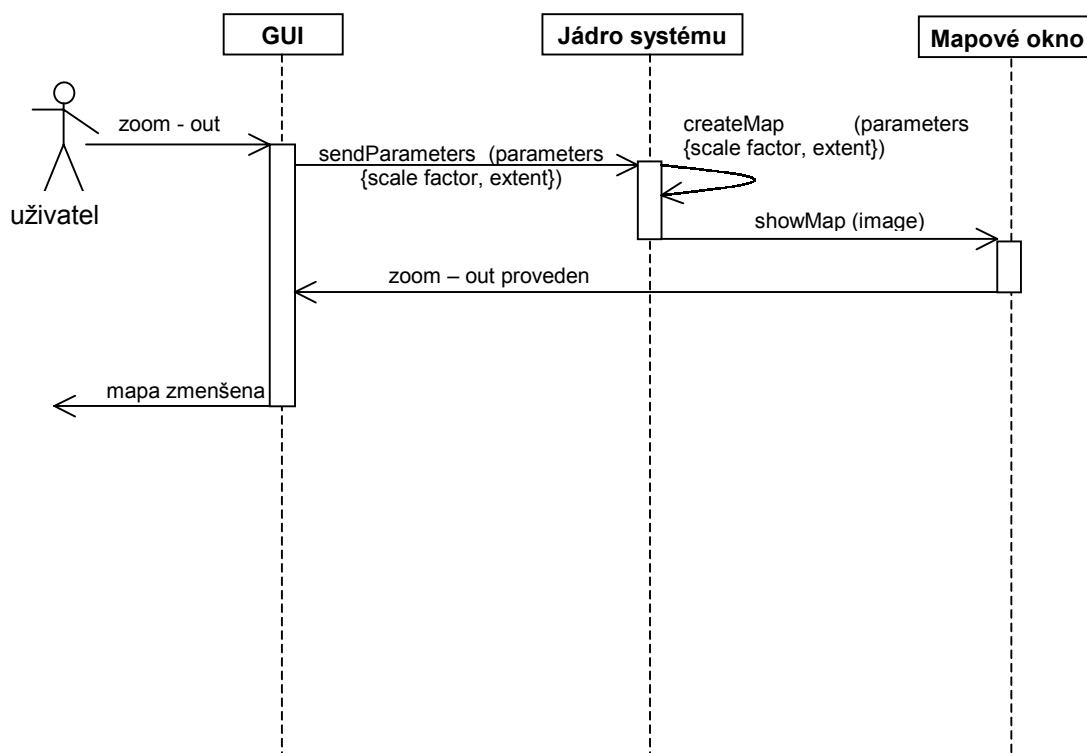


Obr. 9 Sekvenční diagram pro práci s mapou (zmenšení měřítka zobrazení dat)

#### Scénář:

|  |
|--|
| <b>Práce s mapou zoom - in</b>   |
| <b>Účastníci:</b> uživatel   |
| <b>Vstupní podmínky:</b> vybrání požadované trans. funkce + určení míry změny měřítka  |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"> <li>1. uživatel vybere požadovanou trans. funkci</li> <li>2. použije funkci na mapě</li> <li>3. GUI pošle dotaz na Jádro systému</li> <li>4. Jádro systému vygeneruje nový obrázek</li> <li>5. do Mapového okna se načte vygenerovaný obrázek</li> <li>6. uživatel vidí zvětšenou mapu</li> </ol> |
| <b>Výstupní podmínky:</b> zmenšení měřítka   |
| <b>Alternativní tok:</b> možnost vybrání jiné funkce   |

### 6.5.4 Sekvenční diagram – práce s mapou zoom-out

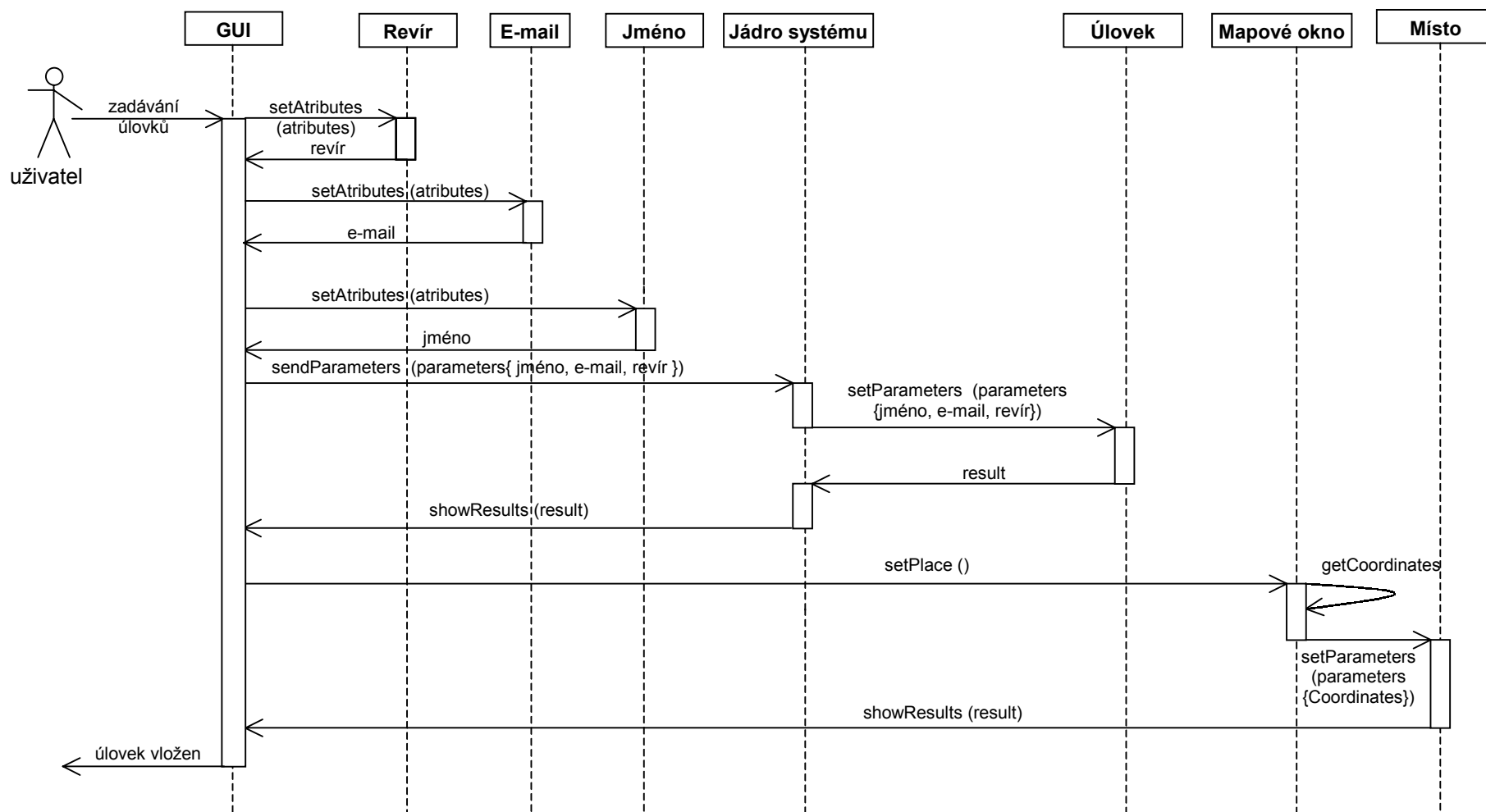


Obr. 10 Sekvenční diagram pro práci s mapou (zvětšení měřítka zobrazení dat)

#### Scénář:

|  |
|--|
| <b>Práce s mapou zoom - out</b>  |
| <b>Účastníci:</b> uživatel   |
| <b>Vstupní podmínky:</b> vybrání požadované trans. funkce + určení bodu  |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"> <li>1. uživatel vybere požadovanou trans. funkci</li> <li>2. použije funkci na mapě</li> <li>3. GUI pošle dotaz na Jádro systému</li> <li>4. Jádro systému vygeneruje nový obrázek</li> <li>5. do Mapového okna se načte vygenerovaný obrázek</li> <li>6. uživatel vidí zmenšenou mapu</li> </ol> |
| <b>Výstupní podmínky:</b> zvětšení měřítka   |
| <b>Alternativní tok:</b> možnost vybrání jiné funkce   |

### 6.5.5 Sekvenční diagram – zadání nového úlovku

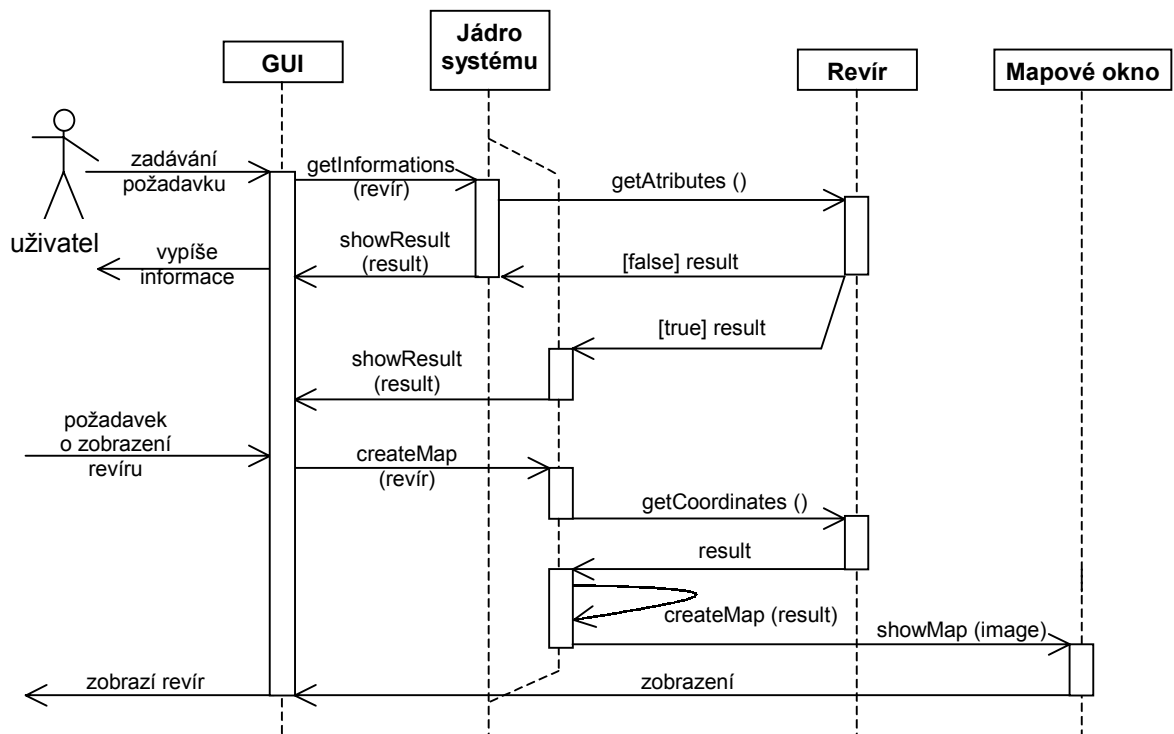


Obr. 11 Sekvenční diagram pro zadání nového úlovku

**Scénář:**

|   |
|---|
| <b>Zadat nový úlovek</b>  |
| <b>Účastníci:</b> uživatel  |
| <b>Vstupní podmínky:</b> uživatel musí ulovit úlovek a zvolit zadávání nového úlovku  |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"> <li>1. uživatel se přepne do okna se zadáváním nových úlovků</li> <li>2. uživatel zadá své Jméno, E-mail</li> <li>3. uživatel vybere Revír, kde ulovil rybu</li> <li>4. uživatel zadá informace o lovu</li> <li>5. uloží se nové údaje o úlovku</li> <li>6. uživatel lokalizuje Místo úlovku na revíru</li> <li>7. uloží se Místo lovu k úlovku</li> <li>8. záznam se zobrazí mezi úlovky</li> </ol> |
| <b>Výstupní podmínky:</b> záznamy odpovídají realitě  |
| <b>Alternativní tok:</b> uživatel si může zjistit informace o revíru  |

**6.5.6 Sekvenční diagram – vyhledání revíru**

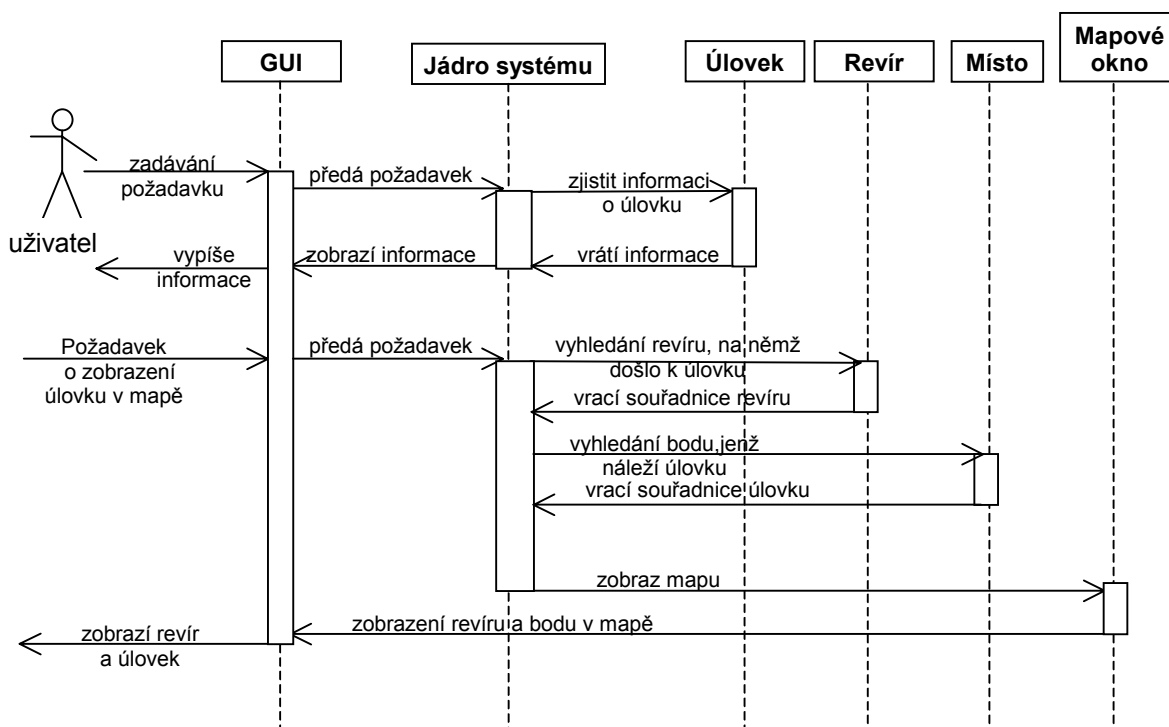


Obr. 12 Sekvenční diagram pro vyhledání revíru

**Scénář:**

|  |
|--|
| <b>Vyhledat revír</b>  |
| <b>Účastníci:</b> uživatel   |
| <b>Vstupní podmínky:</b> uživatel musí zadat číslo revíru nebo text,<br>dle kterého se revír vyhledá + vybrání funkce  |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"><li>1. uživatel zadá parametry pro vyhledání revíru a odešle</li><li>2. parametry jsou porovnány s revíry</li><li>3. KDYŽ nenalezen<ol style="list-style-type: none"><li>3.1. dle parametrů revír není nalezen</li><li>3.2. vypíše uživateli informaci o nenalezení revíru,<br/>dle zadaného klíče</li></ol></li><li>4. KDYŽ nalezen<ol style="list-style-type: none"><li>4.1. dle parametrů revír nalezen</li><li>4.2. vypíše uživateli informace o daném revíru</li><li>4.3. nabídne možnost zobrazení revíru v mapě</li><li>4.4. po potvrzení této možnosti se zobrazí revír v mapě</li></ol></li></ol> |
| <b>Výstupní podmínky:</b> záznamy odpovídají realitě   |
| <b>Alternativní tok:</b> uživatel může využít jinou funkci systému   |

### 6.5.7 Sekvenční diagram – vyhledávání všech úlovků a zobrazení požadovaného úlovku



Obr. 13 Sekvenční diagram pro vyhledávání úlovků a zobrazení požadovaného úlovku

#### Scénář:

|   |
|---|
| <b>Vyhledat úlovky a zobrazit požadovaný úlovek</b>   |
| <b>Účastníci:</b> uživatel  |
| <b>Vstupní podmínky:</b> vyhledání úlovků + zobrazení požadovaného úlovku   |
| <b>Kroky scénáře:</b> <ol style="list-style-type: none"> <li>1. uživatel odešle dotaz ke zjištění úlovků</li> <li>2. vypíší se informace o úlovcích na revírech</li> <li>3. uživatel má možnost si místo požadovaného úlovku prohlédnout v mapě pomocí tlačítka</li> <li>4. v mapě je lokalizován revír, na němž došlo k ulovení ryby + bod, jenž udává místo lovu</li> </ol> |
| <b>Výstupní podmínky:</b> záznamy odpovídají realitě  |
| <b>Alternativní tok:</b> uživatel může využít jinou funkci systému  |



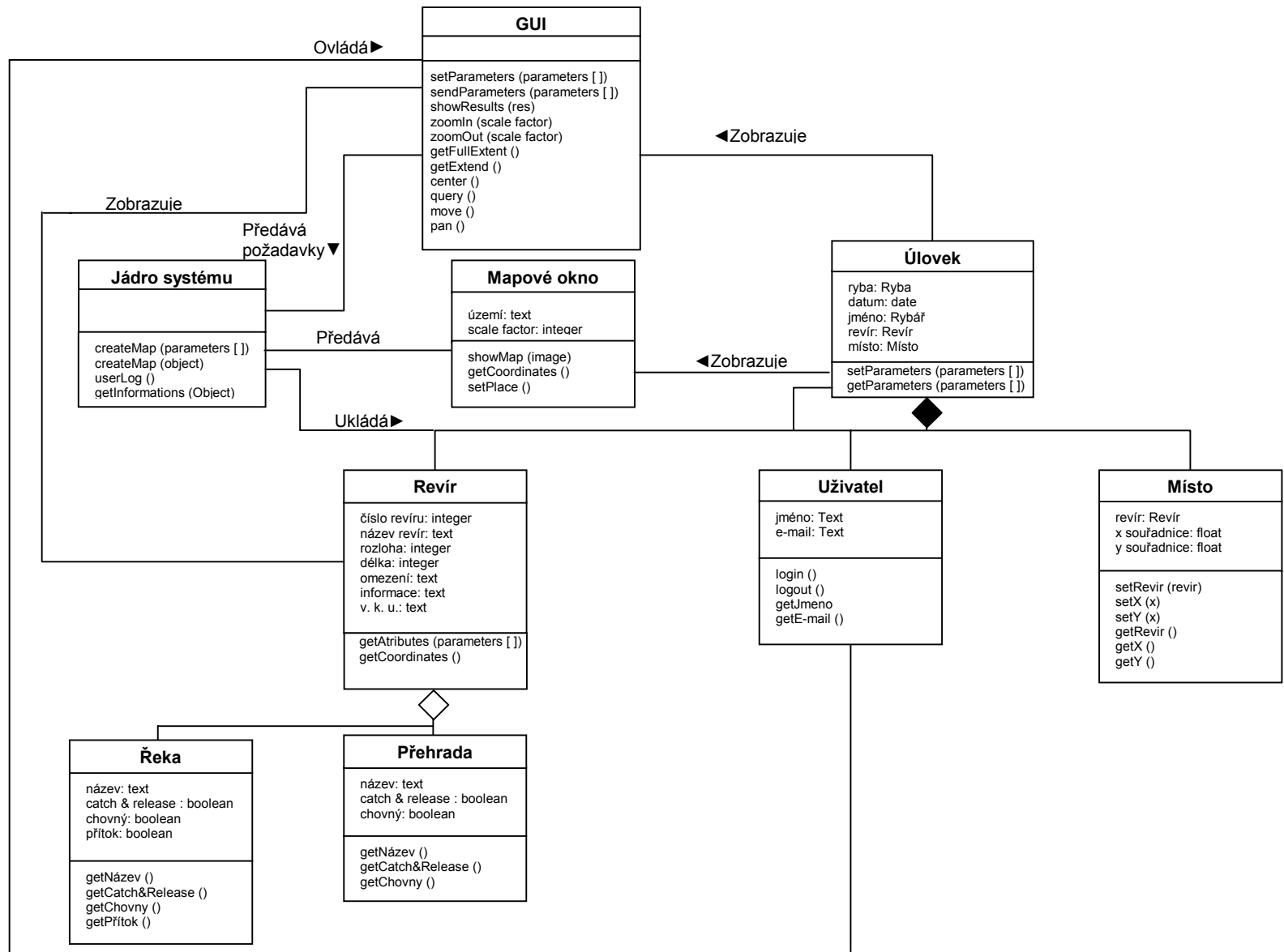
## 6.6 Diagram tříd

Je to pohled na množinu tříd, rozhraní a jejich vzájemné vztahy. Tento diagram nám vyjadřuje statický pohled na systém.

Podrobný slovní popis jednotlivých tříd viz kapitola 5.2.

Diagram tříd je zobrazen na další straně (*viz Obr. 14*).

# Diagram tříd

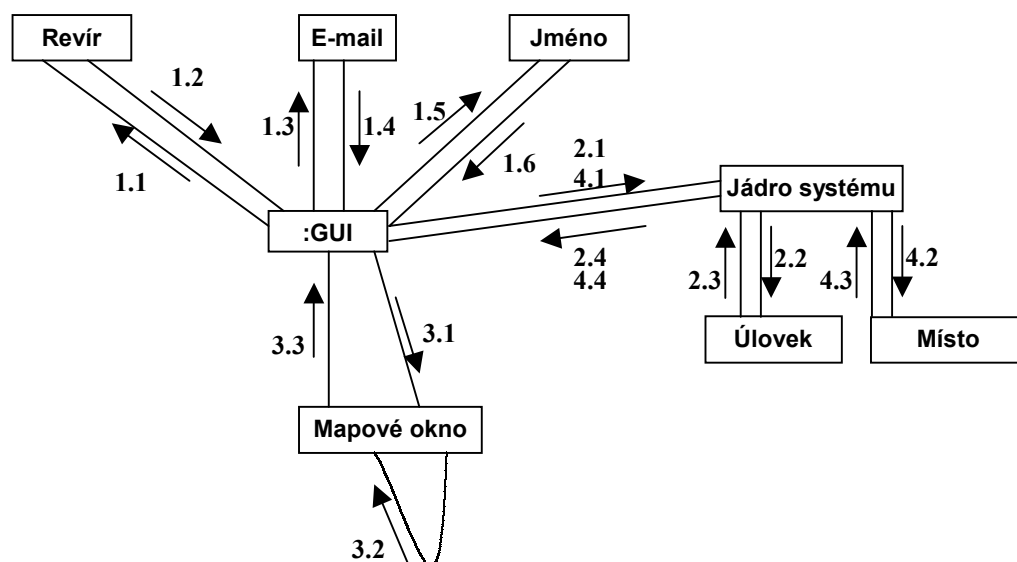


Obr. 14 Diagram tříd

## 6.7 Diagram spolupráce

Diagram spolupráce upřednostňuje při popisu komunikace mezi objekty topologií, tedy jejich rozložení a vzájemné spojení. Vychází ze sekvenčních diagramů. Diagram spolupráce jsem vytvořil jen pro zadávání nového úlovku.

**zadání nového úlovku:**



Obr. 15 Diagram spolupráce pro zadání nového úlovku

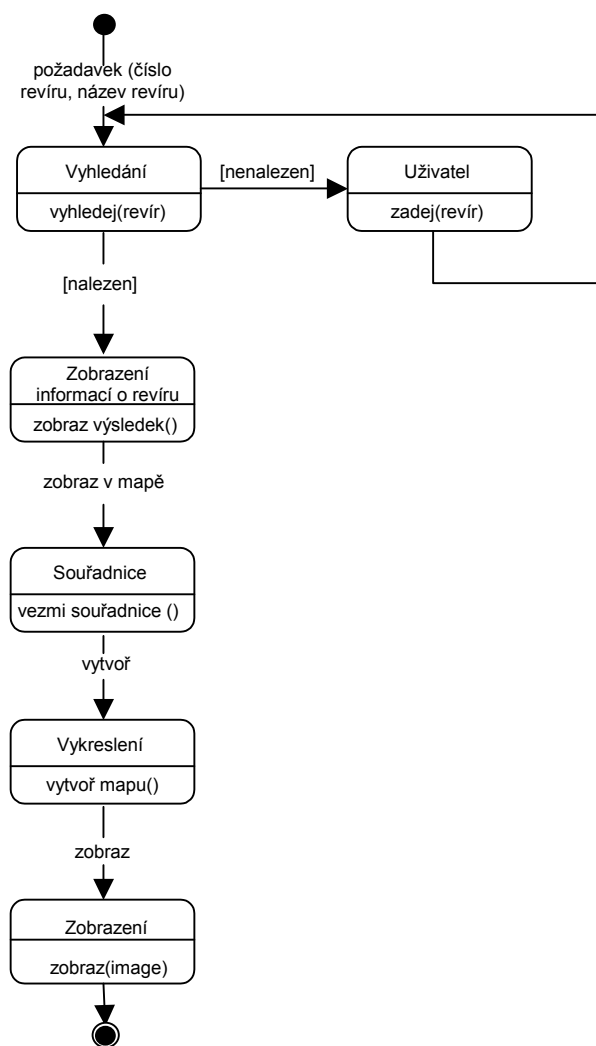
**POPIS:**

- |     |                   |     |                            |
|-----|-------------------|-----|----------------------------|
| 1.1 | Zadat revír       | 2.4 | Zobrazení zprávy           |
| 1.2 | Revír             | 3.1 | Zadat místo úlovku         |
| 1.3 | Zadat e-mail      | 3.2 | Zjistit souřadnice kurzoru |
| 1.4 | E-mail            | 3.3 | Zobrazení výsledku         |
| 1.5 | Zadat jméno       | 4.1 | Odeslat parametry          |
| 1.6 | Jméno             | 4.2 | Zadat parametry            |
| 2.1 | Odeslat parametry | 4.3 | Zpráva                     |
| 2.2 | Zadat parametry   | 4.4 | Zobrazení zprávy           |
| 2.3 | Zpráva            |     |                            |

## 6.8 Stavový diagram

Stavový diagram prezentuje životní cyklus objektu dané třídy z hlediska jeho stavů, přechodů mezi těmito stavy a událostmi, které je vyvolávají.

Stavový diagram pro revír je podrobně popsán (viz Obr. 16), takže zde nebudou uváděny bližší podrobnosti.



Obr. 16 Stavový diagram pro revír

## 7 Vlastní návrh informačního systému

Informační systém je určen především pro rybářskou veřejnost, kterou zajímají informace o revírech Severní Moravy a Slezska a úlovcích na nich, ale také pro evidenci ulovených ryb na těchto revírech.

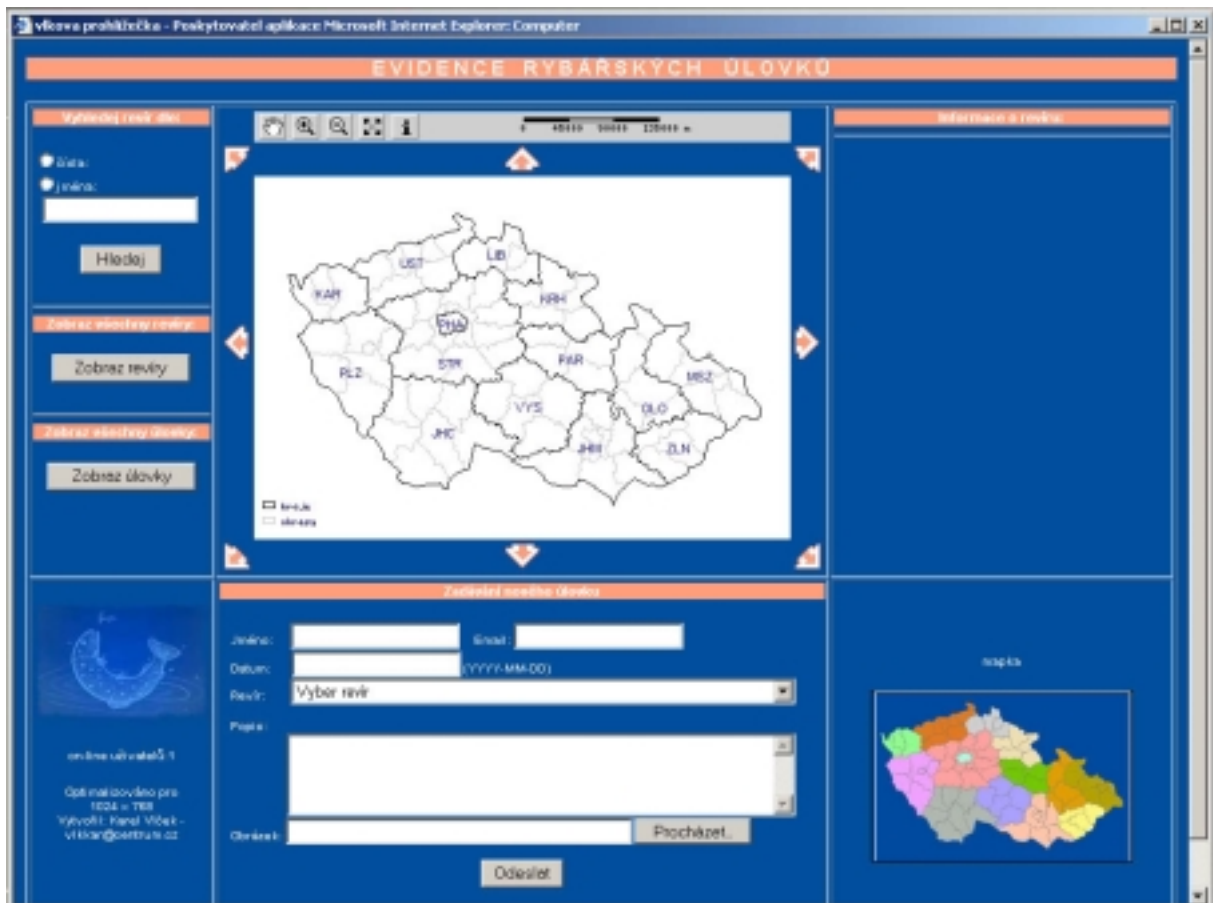
Informační systém je spouštěn přes odkaz na úvodní stránce (*Příloha 5*) internetového prohlížeče, t.j. kvůli otevření informačního systému v novém, specificky definovaném okně. Je zde užit javascript, který spustí aplikaci bez tlačítkové lišty, lišty pro adresu a textového menu. Naopak spouští informační systém s posuvnými lištami, zjistí si rozlišení monitoru a podle toho otevře okno přes celou obrazovku, ale to není adekvátní s funkcí *fullscreen*. Vše je vidět ve zdrojovém kódu javaskriptu, jenž definuje nové okno.

**Ukázka zdrojového kódu** – definice nového okna v internetovém prohlížeči:

```
window.open(url,jmeno,'resizable=1,scrollbars=1,top=0, left=0, menubar=0, width='+ (screen.availWidth - 10)+' , height=' + screen.availHeight)
```

Grafické uživatelské rozhraní informačního systému je vytvořeno pomocí jazyku HTML, skriptů PHP a javaskriptu. V aplikaci jsou načítány soubory ESRI Shapefile, jenž zobrazují zájmové území (Severní Morava a Slezsko). Vrstva toků a vodních ploch je propojena s databází MySQL „rybydb“. Aplikace je tvořena tabulkou, která rozděluje obrazovku na 6 základních částí a ty jsou všechny načteny do úvodní stránky.

Zdrojový kód viz *Příloha 1*



Obr. 17 Vzhled aplikace - rybářský průvodce

## 7.1 Načtení mapových vrstev

### 7.1.1 Mapový soubor MAPFILE

Jedná se o textový soubor (*Příloha 9*), v němž jsou deklarovány vstupy pro generování mapových výstupů. Je zde nastavena prvotní poloha zobrazovaného území v mapovém okně, cesty k souborům (vektorovým datům, textovým fontům, vykreslovacím symbolům, ...), definuje se zde legenda, měřítko a jednotlivé vrstvy s popisky, barvami, atd. viz dokumentace k *Mapfile*[4].

### 7.1.2 Mapové okno

Slouží k zobrazení uživatelem požadovaného mapového výstupu. Uživatel zadá požadavek na zobrazení mapy v mapovém okně. PHP/Mapscript na daný objekt *Map* (Mapové okno) nastaví EXTENT pro zobrazení. Metoda SETEXTENT nastaví nový rozsah zobrazovaného území v mapovém okně.

Jakmile se nastaví nová poloha zobrazovacího území pro objekt *Map*, musí se vytvořit mapový výstup metodou *DRAW*. Takto vytvořený mapový výstup se ukládá pomocí metody *SAVEWEBIMAGE*, jenž vrátí adresu uloženého obrázku.

Obrázek je následně vykreslen v mapovém okně. O zobrazení mapového okna se stará java applet, vytvořený vývojáři mapového serveru *Mapserver*. Ten zapříčiňuje pomalejší načítání mapového okna. Na druhou stranu umožňuje efektivnější práci se zobrazovacími funkcemi.

## 7.2 Funkce aplikace

Funkce systému byly popsány v kapitole *6.1 Slovní specifikace požadavků*. Uživatel bude moci využívat tyto funkce:

- **vyhledávat informace o jednotlivých revírech** – zobrazení informace na podkladě zadaného vyhledávacího znaku
- **vyhledávat informace o všech revírech** – zobrazení informací, jenž jsou dány Soupisem revírů pro Severní Moravu a Slezsko pro rok 2003
- **vyhledávat informace o všech úlovcích** – vypíše údaje o všech ulovených rybách na všech revírech
- **translokační operace** – Zoom-in, Zoom-out, Zoom-all, pan, informace o revíru vybraného v mapě, pohyb s mapovým výstupem pomocí šipek
- **vyhledávat informace o úlovcích na revírech** - údaj o rybách ulovených na jednotlivých revírech
- **vkldat nové údaje o lovu a úlovku**
- **doplňkové funkce**

### 7.2.1 Vyhledávání informací o jednotlivých revírech

Uživatel má možnost vyhledávat revír dle zadaných specifík. Buď zná číslo revíru (*číslo*), podle něhož si chce vyhledat informace, a nebo chce vyhledat revír za pomoci slova nacházejícího se v názvu revíru (*jména*). Vyhledávací hodnotu zadá do příslušné kolonky. Přepne vyhledávání dle čísla nebo dle jména a odešle pomocí tlačítka *Hledej*. Vstupní hodnota je porovnána s údaji v tabulce *revir* nacházející se v databázi „rybydb“.

Vyhledej revír dle:

čísla:  
 jména:

Obr. 18 Formulář pro vyhledávání revírů

K výpisu informací je využíván PHP skript – soupis.php (Příloha 6), jenž nám zobrazuje výsledek dotazu, který jsme zaslali. Je-li odeslaná hodnota nalezena, zobrazí se nová stránka s výpisem názvů revírů, jenž obsahují klíčová slova zadaná ve vstupním parametru. Není-li hodnota v databázi nalezena, je tato skutečnost taktéž sdělena uživateli.


V databázi bylo vyhledáno 2 záznamů

| číslo revíru | název revíru  | lokality                | v. k. ú.    | délka | rozloha | informace  | omezení   | mapa |
|--------------|---|-------------------------|-------------|-------|---------|--|---|------|
| 473002       | Bečvka<br>ROŽNOVSKÁ 2 - MO<br>Rožnov pod<br>Radhoštěm (přítok<br>Bečvy, Hněvín) |                         |             | 30    | 18,0    | Od jezů pod přírodním mostem do<br>otřepě nad Bečvou až k pramenům<br>včetně všech přítoků | Tok Bečvy nad mostem u Hněvínů až k pramenům a včetně<br>přítoky, kromě přítoky Šolman, Hněvín, Hněvín a Šolman<br>po zjevké a potokem Žákovci a Úpek toku pod s mostem<br>k 1. Hněvíněmu přívodu v Rožnově pod Radhoštěm -<br>vystačeno tabulkami - jasu O1910 - lov ryb zakázan. Měra<br>kopaná 27 cm šířkou. Kapka omezen na 4 kg ročně a 40 kg<br>ročně na jednu pozvědi. |      |
| 473002       | Bečvka<br>ROŽNOVSKÁ 2 - MO<br>Rožnov pod<br>Radhoštěm (přítok<br>Bečvy, Hněvín) | 01 - nádrž u<br>Macečků | Horní Bečva |       | 18,0    |  | Tok Bečvy nad mostem u Hněvínů až k pramenům a včetně<br>přítoky, kromě přítoky Šolman, Hněvín, Hněvín a Šolman<br>po zjevké a potokem Žákovci a Úpek toku pod s mostem<br>k 1. Hněvíněmu přívodu v Rožnově pod Radhoštěm -<br>vystačeno tabulkami - jasu O1910 - lov ryb zakázan. Měra<br>kopaná 27 cm šířkou. Kapka omezen na 4 kg ročně a 40 kg<br>ročně na jednu pozvědi. |      |

Obr. 19 Formulář s vyhledanými revírů

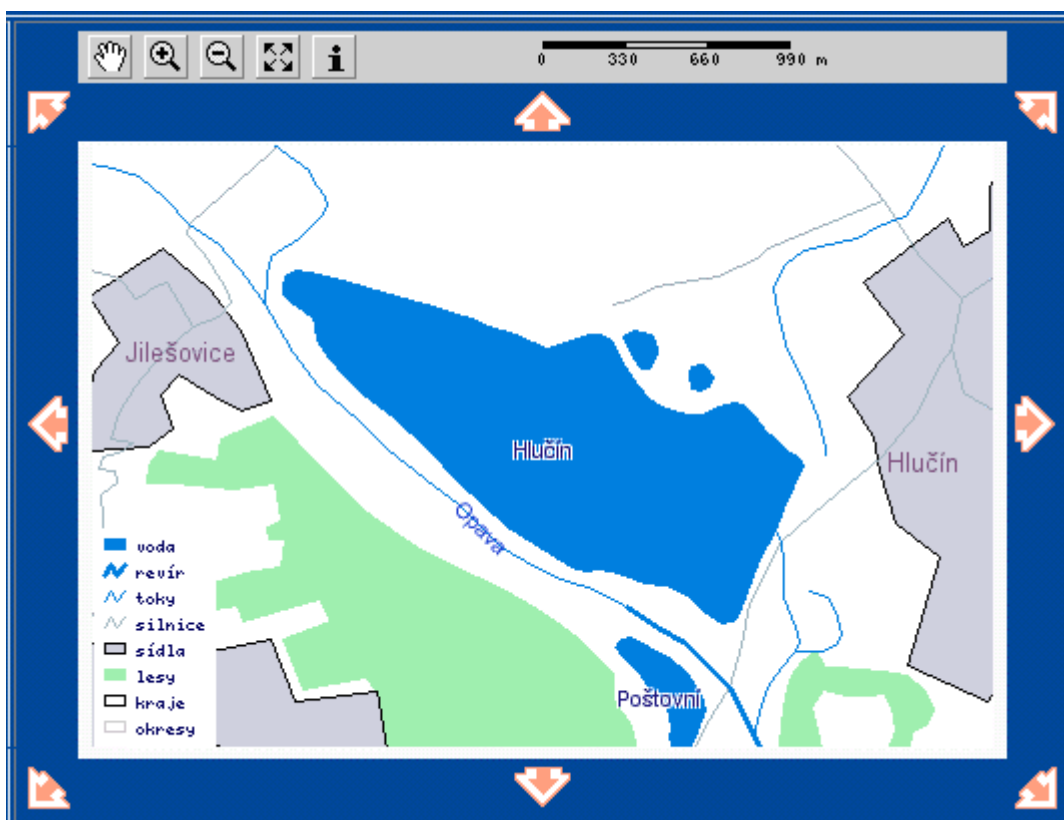
Na výpisu této stránky se můžeme dozvědět, kolik revírů bylo nalezeno dle stanoveného vyhledávacího klíče; v tomto případě se jedná o dva záznamy. Pod číslem revíru „473002“ se nachází dva záznamy, tok Rožnovské Bečvy a Nádrž u Macečků na Horní Bečvě. Kromě čísla revíru a názvu revíru se zde ještě dozvíme lokalitu na revíru, pokud nějaká existuje, a dále katastrální území, ve kterém je lokalita soustředěna. Další informací je délka revírů u toků a rozloha u uzavřených lokalit. Je zde taktéž informace



o revíru a omezení na tomto revíru. Vybraný revír je možno nechat vyhledat na mapě, k čemuž slouží tlačítko -  v kolonce *mapa*.

Tlačítko předá parametr hlavnímu skriptu. Skript prohledá vrstvu *toky* a *voda* a nalezne objekt, jenž má v poli „*Id\_pod*“ identickou hodnotu jako hodnota předaná HTTP protokolem. Skript přebere index daného objektu a pomocí knihovny PHP/Mapscriptu na daný objekt *Map* nastaví EXTENT pro zobrazení.

Jakmile se nastaví nový rozsah zobrazovaného území, vygeneruje se obrázek, který je načten do aplikace. Viz *Příloha 1* – „vyhledej revír dle vyberu“.

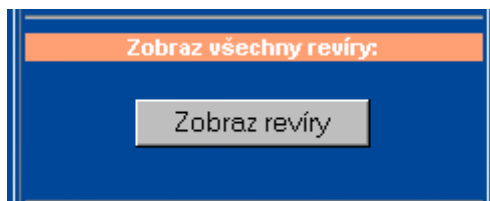


Obr. 20 Mapové okno s nalezeným revírem

### 7.2.2 Vyhledávání informací o všech revírech

Princip této funkce je obdobný jako u funkce pro vyhledání specificky zadaného revíru. Při kliknutí na tlačítko „Zobraz revíry“ se předá skriptu *soupis.php* parametr pro výpis všech revírů. Otevře se stejná stránka jako na *Obr. 19* s tím rozdílem, že jsou zobrazeny všechny revíry nalezené v databázi, tedy 444 revírů a to včetně jejich lokalit. Revíry jsou zobrazovány po 30 záznamech na stránku. Ve spodní části jsou odkazy

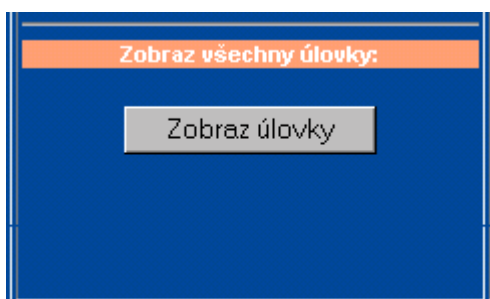
na následující revíry. Způsob zobrazení námi vybraného revíru je stejný jako u předchozí funkce.



Obr. 21 Formulář pro vyhledání všech revírů


### 7.2.3 Vyhledávání informací o všech úlovcích

Tato funkce umožní vypsat data o všech úlovcích na všech revírech. Ke spuštění






slouží tlačítko (Obr. 22) umístěné na hlavní stránce aplikace. Tlačítko předá parametry skriptu *celkul.php* (Příloha 2). Ten otevře novou stránku (Obr. 23) s chronologicky uspořádanými úlovky na všech revírech. Zde je možné prohlédnout si úlovky a lokalitu vybraného úlovku zobrazit na mapě.

Obr. 22 Formulář pro vyhledání všech úlovků

K zobrazení úlovku na mapě slouží opět tlačítko -  v kolonce *mapa*.

V tabulce jsou vyhledány 5 záznamů

| id | jmeno | email        | datum      | revir  | popis   | mapa  |
|----|-------|--------------|------------|--|---|---|
| 1  | vita  | vita@emal.cz | 2014-03-24 | MAPSA 1 A - VOZ ČR3 Dobruva -                            | Bábo od 6 do 14 hod., námo jsem chytl na rybičku a všecko má. Ve 12 jsem začal vstávat a na druhý námo to přišlo - 18 kg šáem. Pakem ještě 16 mála + 80. Nechá chytlas. |  |
| 2  | panka | panka.cz     | 2014-03-22 | MAPSA 1 A - VOZ ČR3 Dobruva -                            | Dyhal jsem od 6 - 10 hodin. Za celou dobu jen jedno šáem - 34cm.  |  |
| 3  | hula  | hula.cz      | 2014-03-28 | MAPSA 1 A - VOZ ČR3 Dobruva -                            | Dyhal jsem od 6-13 hod. Jedna 38 kg šáem.   |  |
| 4  | vita  | vita@emal.cz | 2014-03-24 | MAPSA 1 A - VOZ ČR3 Dobruva -                            | Dyhal jsem od 6 - 10 hodin. Chytl jsem jedno 38 kg šáem.  |  |
| 5  | vita  | vita         | 2014-03-28 | BAŘTICE 1 A - MO Rybní. Mlýns - 81 - vodní nádrž Bařtice | kapr  |  |

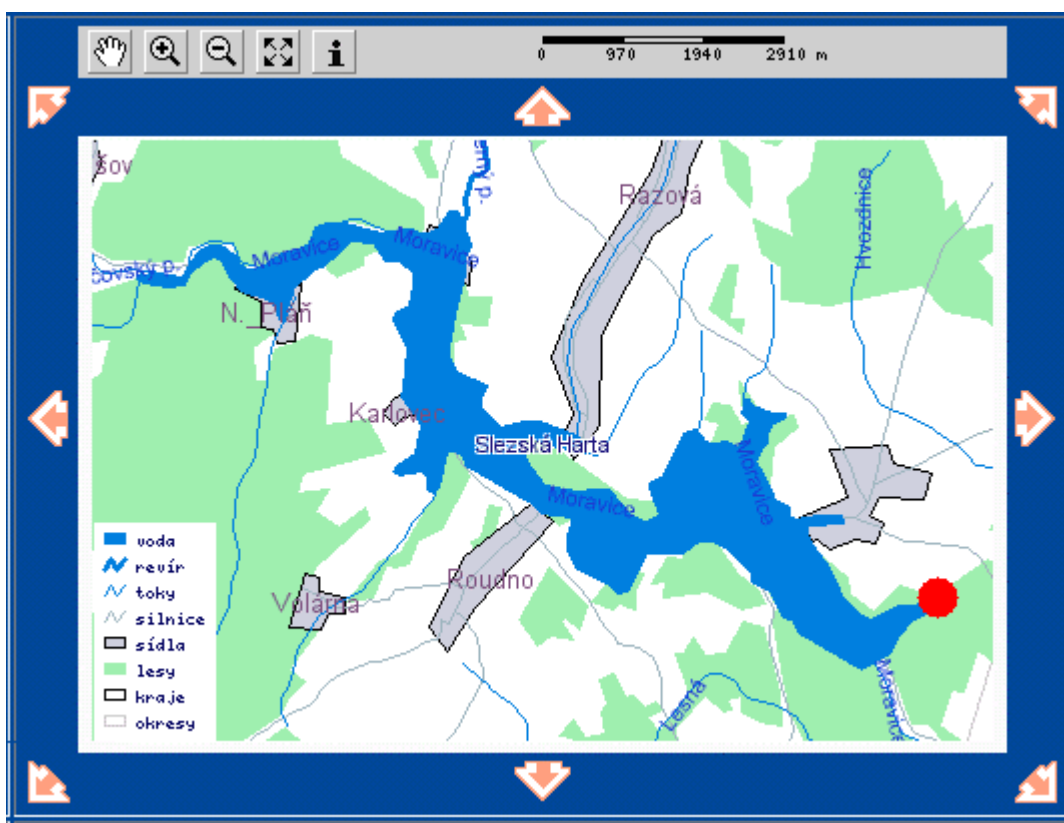
Obr. 23 Výpis informací s úlovky na všech revírech

Hlavnímu skriptu jsou předány tři parametry. Číslo revíru, na němž došlo k úlovku; dle něj se nastaví poloha zobrazovaného území. Dále číslo, jenž představuje identifikátor

úlovku; podle tohoto čísla se zobrazí daná příslušná data v pravé informační části aplikace (Obr. 28). Poslední předaný parametr udává identifikátor bodu, který se zobrazí na načtené mapě. Tady si můžeme všimnout, že všechny vrstvy nemusí být předem nadefinované v souboru MAPFILE. Modul PHP/Mapscript nám nabízí nástroje pro přidávání tématických vrstev přímo do mapové kompozice. Nemusí se jednat pouze o vektorové vrstvy ESRI Shapefile, ale může se jednat jako v tomto případě, o bodovou vrstvu s vykreslováním z databáze.

#### **Ukázka zdrojového kódu – vykreslení bodu načteného z databáze:**

```
$connection = mysql_connect($server,$uzivatel,$heslo) or die ("Spojení k
      databázi se nezdařilo.");
$databaze = mysql_select_db("rybydb");
$vypis_bodu = mysql_query("SELECT * FROM bod WHERE id_ul LIKE '$bodka'");
$x_s = MySQL_Result($vypis_bodu, 0,"x_sour");
$y_s = MySQL_Result($vypis_bodu, 0,"y_sour");
    $pointobj = ms_newPointObj();
    $pointobj->setXY($x_s,$y_s);
    $lineobj = ms_newLineObj();
    $lineobj->add($pointobj);
    $pointobj->free();
    $shapeobj = ms_newShapeObj(MS_SHAPE_POINT);
    $shapeobj->add($lineobj);
    $lineobj->free();
    $layerobj = ms_newLayerObj($map);
    $layerobj->set('type',MS_LAYER_POINT);
    $layerobj->set('status',MS_ON);
    $layerobj->addFeature($shapeobj);
    $classobj = ms_newClassObj($layerobj);
    $classobj->set('symbol',1);
    $classobj->set('size',20);
    $newColor = $map->addColor(255,0,0);
    $classobj->set("color", $newColor);
```



Obr. 24 Zobrazení úlovku v mapě

#### 7.2.4 Zobrazovací operace

Pro práci s mapou je určena nástrojová lišta, která je umístěna nad mapovým oknem. Na nástrojové liště je umístěno 5 tlačítek pro obsluhu aplikace a měřítko pro aktuální mapový výřez.



Obr. 25 Nástrojová lišta pro práci s mapou

První tlačítko je pro posun mapy „Pan“. Uživatel vybere bod kliknutím do mapy. Na tento bod se nastaví střed nového zobrazeného mapového okna.

Druhé tlačítko je pro zvětšení výřezu v mapě „Zoom-in“. Uživatel kliknutím a tažením vytvoří výběrový obdélník. Na tento obdélník se nastaví nový zozsah zobrazovaného území, jenž se zobrazí v mapovém okně.

Třetím tlačítkem je zmenšení mapy „Zoom-out“, po jeho potvrzení a kliknutí do mapy se aktuální výřez na mapě trojnásobně oddálí.

Po vybrání čtvrtého tlačítka se nastaví plný plošný rozsah zobrazovaného území „Full-extent“, jenž je nastavena v MAPFILE souboru. Zobrazí se celá Česká republika.


Páté tlačítko „Information“ slouží ke zjišťování atributů u jednotlivých revírů. Klikne-li uživatel na revír určený k lovu, vypíše se informace o tomto revíru (Obr. 26). Pro výpis informací slouží část aplikace, kde se jednak vypisují informace o revírech a informace o úlovku na revíru.

Při výpisu informace o revíru vidí uživatel tyto údaje: číslo revíru, název revíru, lokalitu, v k. ú. (katastrální území), délku (revíru), rozlohu (revíru), informace o revíru, omezení na revíru, úlovky na tomto revíru.

| Informace o revíru:     |   |
|-------------------------|---|
| číslo revíru:           | 471200  |
| název revíru:           | HARTA 1 A- VÚS ČRS Ostrava  |
| lokalita:               |   |
| v. k. ú.:               |   |
| délka                   |   |
| rozloha:                | 900   |
| informace:              | Revír tvoří nádrž na Moravici, přítoku Opavy, Odry. Od tělesa hráze v ř.km 57,5 až po stupně na konci vzdutí v ř.km 74,4 v k.ú. Valšov a v ř.km 3,8 na Čemém potoce v k.ú. Dlouhá Stráž.  |
| omezení:                | Přítoky nádrže Moravice, Kočovský potok a Čemý potok nejsou součástí revíru. Ostatní přítoky jsou chráněnou rybní oblastí, určenou k chovu lososovitých ryb a organismů Červeného seznamu ČR. Hranice revíru označeny tabulemi. Zákaz rybolovu v úseku od hráze po obou březích - vyznačeno tabulemi na obou březích a u přítoků Moravice a Čemý potok - vyznačeno tabulemi. Nepořádek na místě lovu bude postihován dle rybářského řádu. Cejn velký, plotice a perlní nemají stanovenou lovnou míru. Na revíru je povolena hospodářská těžba dle vývoje skladby rybního společenství. Míra kapra je 40 cm. |
| úlovky na tomto revíru: | <input type="button" value="zobraz"/>   |

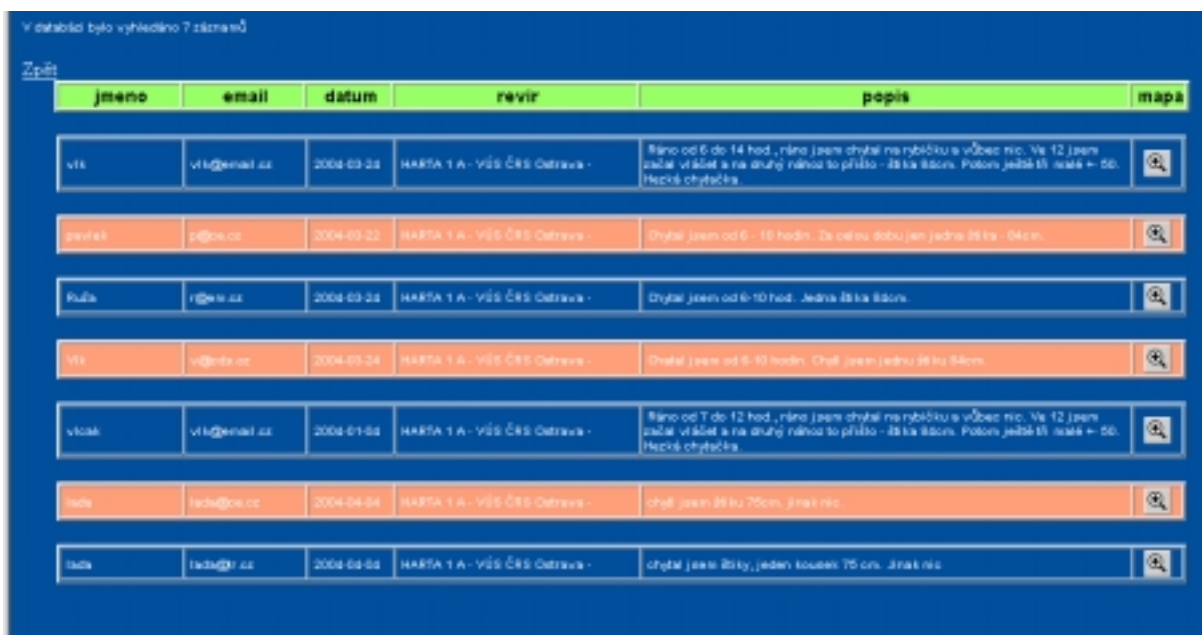
Obr. 26 Formulář s výpisem informací o revíru








Posledním prvkem na nástrojové liště je měřítko. Jeho definice zobrazení je v souboru MAPFILE v sekci *Scalebar* (Příloha 5).

Kolem *Mapového okna* byly vytvořeny šipky , jenž umožňují pohyb s mapovým výstupem.


## 7.2.5 Vyhledávat informace o úlovcích na revírech

Postup zjištění úlovku na revíru začíná stejnými kroky, jaké jsou popsány v předcházejících kapitolách (vyhledávání informací o revíru). Jakmile má uživatel před sebou vypsanou tabulku s informacemi o revíru, má možnost kliknout na tlačítko „Zobraz“. Tímto tlačítkem je vyvolán PHP skript (*Příloha 3*), kterému je předáno číslo revíru, na němž chceme vidět úlovky. Otevře se okno s výpisem úlovku na daném revíru (*Obr. 27*).



| jmeno   | email       | datum      | revir                         | popis   | mapa  |
|---------|-------------|------------|-------------------------------|---|---|
| vtk     | vt@denat.cz | 2004-03-24 | HARTA 1 A - VŠS ČES Ostrava - | Řano od 6 do 14 hod., rano jsem chytal na rybíčku a vůbec nic. Ve 12 jsem začal vláčet a na druhý nához to přišlo - 28 kg šačra. Potom ještě tři - 24 kg + 50. Hezčí chytáčka |    |
| jevíček | je@ce.cz    | 2004-03-22 | HARTA 1 A - VŠS ČES Ostrava - | Chytal jsem od 6 - 10 hodin. Za celou dobu jen jedna 26 kg - 04 kg.   |    |
| Kuža    | r@e.cz      | 2004-03-24 | HARTA 1 A - VŠS ČES Ostrava - | Chytal jsem od 6-10 hod. Jedna 28 kg šačra.   |    |
| Vik     | v@ota.cz    | 2004-03-24 | HARTA 1 A - VŠS ČES Ostrava - | Chytal jsem od 6-10 hodin. Chytal jsem jednu 28 kg šačru.   |    |
| vstok   | vt@denat.cz | 2004-01-04 | HARTA 1 A - VŠS ČES Ostrava - | Řano od 7 do 12 hod., rano jsem chytal na rybíčku a vůbec nic. Ve 12 jsem začal vláčet a na druhý nához to přišlo - 28 kg šačra. Potom ještě tři - 24 kg + 50. Hezčí chytáčka |    |
| lada    | lada@ce.cz  | 2004-04-04 | HARTA 1 A - VŠS ČES Ostrava - | chytal jsem 20 kg, jeden kousek 75 cm. Jinak nic.   |  |
| lada    | lada@ce.cz  | 2004-04-04 | HARTA 1 A - VŠS ČES Ostrava - | chytal jsem 20 kg, jeden kousek 75 cm. Jinak nic.   |  |

Obr. 27 Výpis informací s úlovky na jednom revíru

Pro lokalizaci úlovku v mapě slouží tlačítko  v kolonce *mapa*. Po jeho potvrzení se postupuje stejně jako v kapitole 6.2.3 *Vyhledávání informací o všech úlovcích*. V informační části aplikace se místo informace o revíru zobrazí informace o úlovku (*Obr. 28*).

| Informace o úlovku: |  |
|---------------------|--|
| Jméno:              | vlk  |
| e-mail:             | vlk@email.cz   |
| datum:              | 2004-03-24   |
| revír:              | HARTA 1 A - VÚS ČRS Ostrava  |
| popis:              | Ráno od 6 do 14 hod., ráno jsem chytil na rybičku a vůbec nic. Ve 12 jsem začal vláčet a na druhý nához to přišlo - štika 84cm. Potom ještě tři malé + 50. Hezká chytáčka. |
| Fotografie úlovku:  | zobraz foto  |

Obr. 28 Výpis informací o jednom úlovku

Na tomto formuláři máme k dispozici tlačítko „Zobraz foto“, po jeho potvrzení se otevře nové okno (*Příloha 8*) s fotografií úlovku, tedy pokud byla přiložena. Nové okno je otevíráno ve specifickém rozměru, jenž je dán výškou a šířkou fotografie. Tuto funkci nám poskytuje PHP knihovna *php\_gd.dll*, která je určena pro práci s obrázky. Viz *Příloha 1 (zjištění velikosti fotografie)*.

## 7.2.6 Vkládání nových údajů o lovu a úlovku

Pro vkládání nových záznamů do informačního systému slouží spodní část aplikace.

| Zadávání nového úlovku |                                  |              |                      |
|------------------------|----------------------------------|--------------|----------------------|
| Jméno:                 | <input type="text"/>             | Email:       | <input type="text"/> |
| Datum:                 | <input type="text"/>             | (YYYY-MM-DD) |                      |
| Revír:                 | Vyber revír <input type="text"/> |              |                      |
| Popis:                 | <input type="text"/>             |              |                      |
| Obrázek:               | <input type="text"/>             | Browse...    |                      |
| Odeslat                |                                  |              |                      |

Obr. 29 Formulář pro zadávání nových úlovků

Jedná se o formulář, který vyplňuje uživatel. Formulář se skládá z pěti povinných položek, které musí uživatel vyplnit. Jedná se o jméno lovce, e-mailovou adresu lovce, datum

lovu, revír, na kterém uživatel lovil a popis lovu. Dále má uživatel možnost zadat nepovinný údaj, což je fotografie úlovku. Povinné údaje jsou ověřovány, a pokud uživatel nezadá jednu z těchto pěti povinných podmínek je mu tato skutečnost sdělena a musí údaj doplnit. O validaci a ukládání údajů do databáze se stará PHP skript *uloz.php* (Příloha 7). Pokud uživatel chce zadat fotografii k záznamu, musí kliknout na tlačítko „*Procházet*“ čímž vyvolá dialogové okno, které jej vybízí k vybrání souboru s fotografií. u fotografie je ověřováno zda se jedná o soubor typu JPG či GIF, žádné jiné soubory nám skript nevezme. Dále je zde ověřována podmínka, jestli je soubor menší než 100 kB a zároveň nemá nulovou velikost. Jsou-li tyto podmínky dodrženy, soubor je nahrán na server do adresáře foto. Originální názvy fotografií jsou měněny a nahrazovány číselným názvem, jenž zaručuje jedinečnost, a tudíž nemůže dojít k přepisování souborů. Proběhne-li vše korektně, uživatel je informován výpisem informací. Pokud byla vložena fotografie, skript vypíše původní název fotografie, velikost souboru, typ souboru (zda se jedná o JPG či GIF), a především informuje uživatele, že fotografie byla v pořádku uložena na server. Dalším údajem, jenž se zobrazí uživateli je informace, že pro zakreslení místa lovu má kliknout do mapy. Posledním údajem je informace o úspěšném uložení do databáze. Pokud uživatel nevloží fotografii, je v tomto okně vypsána informace: „*Nebyla vložena fotografie*“. Veškeré ostatní informace jsou stejné jako v předchozím případě. Uživatel v tomto okně musí kliknout na tlačítko „*Zakresli bod*“. Po jeho stisknutí se volá hlavní skript. Před jeho zobrazením se ale uživateli opět připomene, že kliknutím do mapového okna se provede zakreslení místa lovu. Tato hláška je vyvolávána javaskriptem. V hlavním skriptu se mapové okno automaticky přiblíží na revír, jenž zadal rybář ve formuláři. Nyní uživatel klikne do mapového okna, čímž uloží do databáze x a y souřadnici místa úlovku.

### 7.2.7 Doplnkové funkce

Levá dolní část aplikace informuje uživatele o počtu připojených uživatelů, o optimalizovaném rozlišení aplikace a o tvůrci informačního systému. O výpis on-line uživatelů se stará PHP skript *online.php* (Příloha 4). Skript zjišťuje IP adresu uživatele, podle počtu různých IP adres vypíše počet uživatelů.

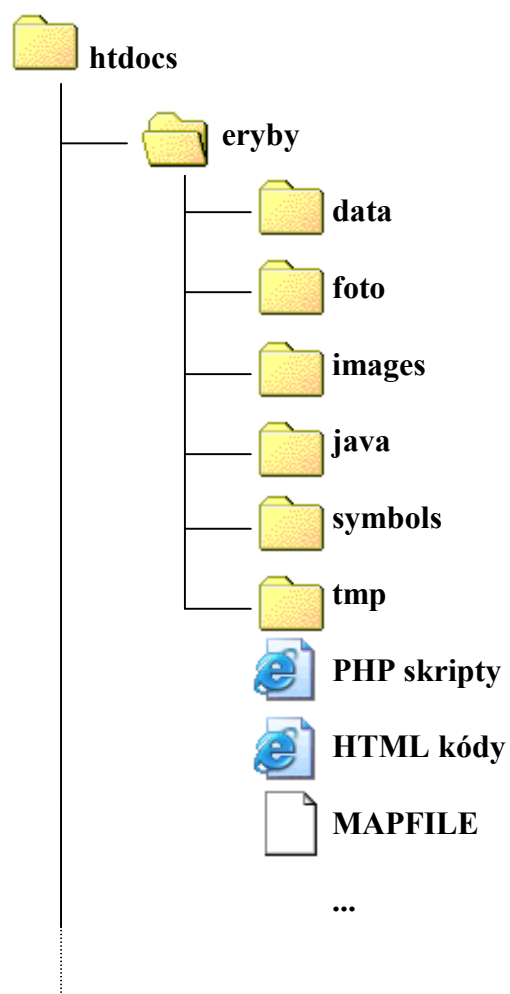
V pravé spodní části je „*mapka*“, která zachycuje celé území mapových zdrojů. Na ní je zobrazován bílý obdélník, který představuje území, jenž je přiblíženo v mapovém okně. Mapka tedy zlepšuje naši orientaci v hodně přiblíženém území v hlavním mapovém okně.



### 7.3 Adresářová struktura aplikace

Všechna data jsou uložena v adresáři „eryba“. Podrobný přehled adresářové struktury je na *Obr. 30*.

|                |   |
|----------------|---|
| <i>htdocs</i>  | adresář webového serveru Apache, kde jsou uloženy jednotlivé webové aplikace                        |
| <i>eryby</i>   | domovský adresář informačního systému, zde jsou uloženy HTML a PHP kódy, také je zde soubor MAPFILE |
| <i>data</i>    | adresář, ve kterém jsou uloženy vektorové vrstvy ve formátu ESRI shapefile                          |
| <i>foto</i>    | v tomto adresáři jsou ukládány fotografie úlovků, jež vkládají uživatelé                            |
| <i>images</i>  | zde jsou uloženy obrázky, které jsou použity v aplikaci (ikony, logo, ...)                          |
| <i>java</i>    | adresář s javovským appletem, využívaném v aplikaci   |
| <i>symbols</i> | adresář, v němž jsou definované symboly pro geoprvky  |
| <i>tmp</i>     | zde jsou ukládány obrázky vygenerované mapserverem (mapový výstup, zmenšená mapa území, měřítko)    |



Obr. 30 Adresářová struktura informačního systému

## 8 Závěr

Informační systém pro evidenci rybářských úlovků splňuje požadavky, které byly uvedeny v zadání diplomové práce. Výsledná aplikace byla testována za pomoci spolužáků a rybářů v mém blízkém okolí. Výsledkem testování bylo odladění systému a oprava drobných programových chyb. Po této drobné úpravě bylo opět provedeno testování, jehož výsledkem byly jen kladné reakce okolí na výslednou aplikaci.

Problémy při realizaci se vyskytly v případě instalace všech softwarových komponent pro provoz aplikací, jenž umožňují prezentaci prostorových dat v prostředí www. Je třeba si dávat pozor na verze všech používaných komponent. U verze modulu PHP/Mapscript je vždy uvedeno, od které verze PHP je možno jej použít. Ne vždy je však tato skutečnost pravdivá. Většinou je modul PHP/Mapscript odladěn pro danou verzi PHP, ne však pro vyšší verze.

Problém nastal také u popisu atributů v mapovém okně, které generoval mapový server. Tyto popisy neobsahovaly české znaky. Nepomohl ani převod do jiných kódovacích znakových sad. Řešení přinesla až konverze vstupních dat do **Unicode**, což je znaková sada. Světový standard kódování znaků **Unicode Worldwide Character Standard** zavedlo **Unicode Consortium** jako systém kódování znaků určený pro podporu výměny, zpracování a zobrazování psaného textu v rozličných jazykových verzích moderního světa. Po převodu již bylo vše zobrazováno s českou diakritikou.

Dalším eventuálním rozšířením aplikace by byla možnost celý informační systém rozvést o diskusní fóra, rady pro rybáře, předpověď počasí, atd. Avšak tato rozšíření již nejsou předmětem geografické složky informačního systému. Geografickým rozšířením by mohla být bodová vrstva nad každým revírem, přičemž by bod reprezentoval odkaz na fotografii z daného místa na revíru.

Pro umístění informačního systému na internet by bylo nutno zakoupit příslušná vektorová data a provést zabezpečení aplikace.

Tato diplomová práce by měla sloužit především jako šablona pro vytváření jednoduchých informačních systémů s možností prezentace vektorových dat a jejich obsluhy. Hlavní předností aplikace je možnost jejího provozování za pomoci „nekomerčního“ programového vybavení.

## SEZNAM LITERATURY

- [1]: ŠTEFÁČEK, S.: *Rybářský průvodce po nádržích a rybnících*, nakladatelství Brázda, Praha 1996, ISBN 80-209-0259-7
- [2]: WILLIAMS, H. E. A LANE, D.: *PHP a MySQL – Vytváříme webové databázové aplikace*. 1. vyd. Praha : Computer Press, 2002. 530 s., ISBN 80-7226-760-4
- [3]: VONDRÁK, I.: *Úvod do softwarového inženýrství*, VŠB – Technická univerzita Ostrava, Ostrava 2002
- [4]: *MapServer* [online], Dostupný na WWW: <<http://mapserver.gis.umn.edu/>>
- [5]: *Rybářský řád a soupis revírů platný od 1. 1. 2003*, Český rybářský svaz – výbor územního svazu pro Severní Moravu a Slezska, Ostrava 2002
- [6]: GREER, T.: *Intranety principy a praxe*, Brno, Computer Press, 1999
- [7]: *Apache* [online], Dostupný na WWW: <<http://www.apache.org>>

## SEZNAM OBRÁZKŮ

|   |    |
|---|----|
| <i>Obr. 1</i> ER diagram databáze .....   | 12 |
| <i>Obr. 2</i> Architektura komunikace klient – server .....                                   | 17 |
| <i>Obr. 3</i> Třívrstvá architektura webové databázové aplikace [2] .....                     | 18 |
| <i>Obr. 4</i> Doménový model .....  | 27 |
| <i>Obr. 5</i> Diagram aktivit .....   | 30 |
| <i>Obr. 6</i> Diagram případů užití .....   | 32 |
| <i>Obr. 7</i> Sekvenční diagram pro práci s mapou (full extent) .....                         | 33 |
| <i>Obr. 8</i> Sekvenční diagram pro práci s mapou (pan) .....                                 | 34 |
| <i>Obr. 9</i> Sekvenční diagram pro práci s mapou (zmenšení měřítka zobrazení dat) .....      | 35 |
| <i>Obr. 10</i> Sekvenční diagram pro práci s mapou (zvětšení měřítka zobrazení dat) .....     | 36 |
| <i>Obr. 11</i> Sekvenční diagram pro zadání nového úlovku .....                               | 37 |
| <i>Obr. 12</i> Sekvenční diagram pro vyhledání revíru .....                                   | 38 |
| <i>Obr. 13</i> Sekvenční diagram pro vyhledávání úlovků a zobrazení požadovaného úlovku ..... | 40 |
| <i>Obr. 14</i> Diagram tříd .....   | 42 |
| <i>Obr. 15</i> Diagram spolupráce pro zadání nového úlovku .....                              | 43 |
| <i>Obr. 16</i> Stavový diagram pro revír .....  | 44 |
| <i>Obr. 17</i> Vzhled aplikace - rybářský průvodce .....                                      | 46 |
| <i>Obr. 18</i> Formulář pro vyhledávání revíru .....  | 48 |

|  |    |
|--|----|
| <i>Obr. 19</i> Formulář s vyhledanými revíry .....               | 48 |
| <i>Obr. 20</i> Mapové okno s nalezeným revírem.....              | 49 |
| <i>Obr. 21</i> Formulář pro vyhledání všech revírů .....         | 50 |
| <i>Obr. 22</i> Formulář pro vyhledání všech úlovků.....          | 50 |
| <i>Obr. 23</i> Výpis informací s úlovkou na všech revírech ..... | 50 |
| <i>Obr. 24</i> Zobrazení úlovku v mapě .....                     | 52 |
| <i>Obr. 25</i> Nástrojová lišta pro práci s mapou .....          | 52 |
| <i>Obr. 26</i> Formulář s výpisem informací o revíru.....        | 53 |
| <i>Obr. 27</i> Výpis informací s úlovkou na jednom revíru .....  | 54 |
| <i>Obr. 28</i> Výpis informací o jednom úlovku.....              | 55 |
| <i>Obr. 29</i> Formulář pro zadávání nových úlovků.....          | 55 |
| <i>Obr. 30</i> Adresářová struktura informačního systému.....    | 58 |

## SEZNAM PŘÍLOH

|                                       |    |
|---------------------------------------|----|
| Příloha 1 – <i>evidence.php</i> ..... | 1  |
| Příloha 2 – <i>celkul.php</i> .....   | 17 |
| Příloha 3 – <i>jedul.php</i> .....    | 20 |
| Příloha 4 – <i>online.php</i> .....   | 23 |
| Příloha 5 – <i>index.html</i> .....   | 24 |
| Příloha 6 – <i>soupis.php</i> .....   | 25 |
| Příloha 7 – <i>uloz.php</i> .....     | 28 |
| Příloha 8 – <i>foto.php</i> .....     | 33 |
| Příloha 10 – <i>mapa.map</i> .....    | 34 |

## Příloha 1

Zdrojový kód uživatelského rozhraní (evidence.php):

```
<?php
dl('php_mapscript_35.dll');
dl('php_dbase.dll');
dl('php_gd.dll');

echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .opt {');
echo('        font-family: Arial;');
echo('    }');
echo('    .tlac {');
echo(' color : black');
echo('} ');
echo('    -->');
echo('</STYLE>');

//-----funkce pro vyplnění comba-----
function selectDistinct ($connection,
                        $tableName,
                        $columnName1,
                        $columnName2,
                        $columnName3,
                        $pullDownName,
                        $additionalOption,
                        $defaultValue)
{
    $defaultWithinResultSet = false;
    $distinctQuery = "SELECT DISTINCT $columnName1,$columnName2,$columnName3
        FROM $tableName";
    $distinctQuery = "SELECT $columnName1,$columnName2,$columnName3 FROM
        $tableName";
    if (!($resultId = @ mysql_query ($distinctQuery, $connection)))
        $i = 0;
        while ($row = @ mysql_fetch_array($resultId))
            $resultBuffer[$i++] = $row[$columnName1];
            $distinctQuery1 = "SELECT DISTINCT $columnName2 FROM
                $tableName";
            $distinctQuery1 = "SELECT $columnName2 FROM $tableName";
    if (!($resultId1 = @ mysql_query ($distinctQuery1, $connection)))
        $j = 0;
        while ($row1 = @ mysql_fetch_array($resultId1))
            $resultBuffer1[$j++] = $row1[$columnName2];
            $distinctQuery2 = "SELECT DISTINCT $columnName3 FROM
                $tableName";
            $distinctQuery2 = "SELECT $columnName3 FROM $tableName";
    if (!($resultId2 = @ mysql_query ($distinctQuery2, $connection)))
        $l = 0;
        while ($row2 = @ mysql_fetch_array($resultId2))
            $resultBuffer2[$l++] = $row2[$columnName3];
    echo "\n<select name=nrevir class=opt onchange=nrevir.selectedIndex>";
    if ($nrevir == "2")
```

```

{
echo "-----2-----";
}
if (isset($additionalOption))
    if ($defaultValue == $additionalOption)
        echo "\n\t<option selected size=20>$additionalOption";
    else
        echo "\n\t<option>$additionalOption";

if (isset($defaultvalue))
    {
        foreach ($resultBuffer2 as $result2)
            if ($result2 == $defaultValue)
                echo "\n\t<option selected size=30>$result2";
            else
                echo "\n\t<option>$result2";
        foreach ($resultBuffer1 as $result1)
            if ($result1 == $defaultvalue1)
                echo "\n\t<option selected size=30>$result1";
            else
                echo "\n\t<option>$result1";
    }
else
    {
        foreach ($resultBuffer as $result)
            {
                $idre = $k + 1;
                $resultBuffer1[$k] = substr($resultBuffer1[$k],0,22);
                $resultBuffer2[$m] = substr($resultBuffer2[$m],0,30);
                echo "\n\t<option value=$idre name=nopt
class=opt>$result $resultBuffer1[$k]
$resultBuffer2[$m]</option>";
                $k++;
                $m++;
            }
    }

    echo "\n</select>";

}

} //konec fce
//-----funkce pro vyplnění comba-----
-----

$val_zsize=3;
$check_pan="CHECKED";
$map_path="c:/apache/htdocs/vlk/";
$map_file="mapa.map";
$map = ms_newMapObj($map_path.$map_file);
function getFieldNames($record){
    $value = reset($record);
    $keys = array();
    while($value){
        array_push($keys,key($record));
        $value = next($record);
    }
    return $keys;
}
//-----vyhledej revír dle vyberu-----
-----

//----pro toky -----
$hled_m = $HTTP_POST_VARS["hled_m"];
if (isset($hled_m)){

```

```

$layer = $map->getLayerByName("toky");
$dbfFile = $map_path."data/". $layer->data." .dbf";
$layer->open($map_path."data/". $layer->data." .shp");

$db = dbase_open($dbfFile,0);
$fields = getFieldNames(dbase_get_record_with_names($db,1));
for($i=1;$i<dbase_numrecords($db)+1;$i++){
    $rec = dbase_get_record($db,$i);
    $recName= dbase_get_record_with_names($db,$i);
    $wasFound = false;
    for($j=0;$j<dbase_numfields($db);$j++){
        $rp = $rec[2];
        if(($rec[2] == $hled_m)!=false){$wasFound = true;}
    }
    if($wasFound){
        $show = $layer->getShape(-1,$i-1);
        $r = $show->bounds;
            $P0 = (($r->minx)-500);
            $P1 = (($r->miny)-500);
            $P2 = (($r->maxx)+500);
            $P3 = (($r->maxy)+500);
        $map->setextent($P0,$P1,$P2,$P3);
        break;
    }
}
//-----pro vody-----
$layerv = $map->getLayerByName("voda");
$dbfFilev = $map_path."data/". $layerv->data." .dbf";
$layerv->open($map_path."data/". $layerv->data." .shp");
$dbv = dbase_open($dbfFilev,0);
$fieldsv = getFieldNames(dbase_get_record_with_names($dbv,1));
for($iv=1;$iv<dbase_numrecords($dbv)+1;$iv++){
    $recv = dbase_get_record($dbv,$iv);
    $recNamev= dbase_get_record_with_names($dbv,$iv);
    $wasFoundv = false;
    for($jv=0;$jv<dbase_numfields($dbv);$jv++){
        $rpv = $recv[2];
        if(($recv[2] == $hled_m)!=false){$wasFoundv = true;}
    }
    if($wasFoundv){
        $showv = $layerv->getShape(-1,$iv-1);
        $rv = $showv->bounds;
            $P0v = (($rv->minx)-500);
            $P1v = (($rv->miny)-500);
            $P2v = (($rv->maxx)+500);
            $P3v = (($rv->maxy)+500);
        $map->setextent($P0v,$P1v,$P2v,$P3v);
        break;
    }
}
//if($rec[2] != $hled_m){echo neeevvvvv;}
//-----konec pro toky-----
}
//-----konec vyhledej revír dle vyberu-----
//-----
$tlacitko = $HTTP_POST_VARS["tlacitko"];
$necoNoveho = $HTTP_GET_VARS["nn"];

```



```

//-----zobrazení bodu nacteného z databáze ulovku-----
-----
$bodka = $HTTP_POST_VARS["bodka"];
$id_ulov = $HTTP_POST_VARS["id_ulov"];
if (isset($bodka)){
//echo $id_ulov;
$server="localhost";
    $uzivatel="";
    $heslo="";
    $connection = mysql_connect($server,$uzivatel,$heslo) or die
    ("Spojení k databázi se nezdařilo.");
    $database=mysql_select_db("rybydb");
    $vypis_bodu = mysql_query("SELECT * FROM bod WHERE id_ul LIKE
    '$bodka'");
    $x_s = MySQL_Result($vypis_bodu, 0,"x_sour");
    $y_s = MySQL_Result($vypis_bodu, 0,"y_sour");
    $pointobj = ms_newPointObj();
    $pointobj->setXY($x_s,$y_s);
    $lineobj = ms_newLineObj();
    $lineobj->add($pointobj);
    $pointobj->free();
    $shapeobj = ms_newShapeObj(MS_SHAPE_POINT);
    $shapeobj->add($lineobj);
    $lineobj->free();
    $layerobj = ms_newLayerObj($map);
    $layerobj->set('type',MS_LAYER_POINT);
    $layerobj->set('status',MS_ON);
    $layerobj->addFeature($shapeobj);
    $classobj = ms_newClassObj($layerobj);
    $classobj->set('symbol',1);
    $classobj->set('size',20);
    $newColor = $map->addColor(255,0,0);
    $classobj->set("color", $newColor);
    $Image = $map->prepareImage();
//*****
//-----vypsání informací o ulovku-----
-----

//*****

echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .prvky {');
echo('        font-family: Arial;');
echo('        font-size: 9px;');
echo('        font-weight: bold;');
echo('        color: White;');
echo('    }');
echo('    .vypis {');
echo('        font-family: Arial;');
echo('        font-size: 9px;');
echo('        color: White;');
echo('    }');
echo('    -->');
echo('</STYLE>');
//*****
//-----vypsání informací o ulovku-----
-----

//*****
}

```

```

//-----zobrazení bodu nacteného z databaze ulovku-----
-----
//Zoom a pan
$id_bod:
    if(!isset($HTTP_POST_VARS["sel_but"])){
    $selected_but= "0";
    }else{
    $selected_but= $HTTP_POST_VARS["sel_but"];
    }
if ($selected_but != 3 or isset($HTTP_POST_VARS["Redraw"]) or
$selected_but != 4){
    if (isset($HTTP_POST_VARS["imgx"])) {
        $extent_to_set = explode(" ",$HTTP_POST_VARS["extent"]);
        $map-
>setextent($extent_to_set[0],$extent_to_set[1],$extent_to_set[2]
,$extent_to_set[3]);
$my_extent = ms_newrectObj();
        $my_extent-
>setextent($extent_to_set[0],$extent_to_set[1],$extent_to_set[2]
,$extent_to_set[3]);
if ($selected_but == 0){
        $my_point = ms_newpointObj();
        $my_point-
>setXY($HTTP_POST_VARS["imgx"],$HTTP_POST_VARS["imgy"]);
        $map->zoompoint(1,$my_point,$map->width,$map-
>height,$my_extent);
        $s_x = $HTTP_POST_VARS["imgx"];
    }
if ($selected_but == 1){
        if ($HTTP_POST_VARS["imgx"] != $HTTP_POST_VARS["imaxx"]){
            $z_box = ms_newrectObj();
            $z_box->setextent($HTTP_POST_VARS["imgx"],
$HTTP_POST_VARS["imgy"], $HTTP_POST_VARS["imaxx"],
$HTTP_POST_VARS["imaxy"]);
            $map->zoomrectangle($z_box, $map->width,$map-
>height,$my_extent);
        }
    }
if ($selected_but == 2){
        $my_point = ms_newpointObj();
        $my_point-
>setXY($HTTP_POST_VARS["imgx"],$HTTP_POST_VARS["imgy"]);
        $map->zoompoint("-3",$my_point,$map->width,$map-
>height,$my_extent);
    }
    if ($selected_but == 3){
//---full extent-----
-----
        $map = ms_newMapObj($map_path.$map_file);
    }

////---vykresleni bodu-----
-----
    if ($selected_but == 5){
        $hled_m = $HTTP_POST_VARS["hled_m"];
//Zapsani bodu do databaze - save point to database -----
-----
        $delkaX =($extent_to_set[0] - $extent_to_set[2]);
        $delkaY =($extent_to_set[1] - $extent_to_set[3]);
        $usekX = ($delkaX / 450)*($HTTP_POST_VARS["imgx"]);
        $usekY = ($delkaY / 300)*($HTTP_POST_VARS["imgy"]);
    }
}

```

```

        $geoX = $extent_to_set[2] + ($delkaX-$usekX);
        $geoY = $extent_to_set[3] + $usekY;
    $X=$geoX;
        $Y=$geoY;
        $server="localhost";
        $uzivatel="";
        $heslo="";
        $connection = mysql_connect($server,$uzivatel,$heslo) or die
        ("Spojení k databázi se nezdařilo.");
        $database=mysql_select_db("rybydb");
        $query = "INSERT INTO bod
        set ID_bod = null, " .
        "x_sour = \"\" . $X . "\", " .
        "y_sour = \"\" . $Y . "\", " .
        "id_rev = \"\" . $hled_m . "\", " .
        "id_ul = \"\" . $id_bod . "\"";
        if (!(@ mysql_query ($query, $connection)))
        $IDul = mysql_insert_id();
        //body -----
        }
    }

    $pointobj = ms_newPointObj();
    $pointobj->setXY($X,$Y);
        $lineobj = ms_newLineObj();
    $lineobj->add($pointobj);
    $pointobj->free();
    $shapeobj = ms_newShapeObj(MS_SHAPE_POINT);
    $shapeobj->add($lineobj);
    $lineobj->free();
    $layerobj = ms_newLayerObj($map);
    $layerobj->set('type',MS_LAYER_POINT);
    $layerobj->set('status',MS_ON);
    $layerobj->addFeature($shapeobj);
    $classobj = ms_newClassObj($layerobj);
    $classobj->set('symbol',1);
    $classobj->set('size',20);
        $newColor = $map->addColor(255,0,0);
    $classobj->set("color", $newColor);
    $Image = $map->prepareImage();
}

#####
#####
// bodovy dotaz do mapy
#####
    if ($selected_but == 4){
        $delkaX =($extent_to_set[0] - $extent_to_set[2]);
        $delkaY =($extent_to_set[1] - $extent_to_set[3]);
        $usekX = ($delkaX / 450)*($HTTP_POST_VARS["imgx"]);
        $usekY = ($delkaY / 300)*($HTTP_POST_VARS["imgy"]);
        $geoX = $extent_to_set[2] + ($delkaX-$usekX);
        $geoY = $extent_to_set[3] + $usekY;
    $X=$geoX;
        $Y=$geoY;
        $click = ms_newPointObj();
        $click->setXY($X,$Y);
        @$map->queryByPoint($click, MS_MULTIPLE, 20);
    $gbShowQueryResults = TRUE;
}

```

```

//funkce pro nacteni hodnoty ze SHAPEFILE
function GMapDumpQueryResults()
{
    GLOBAL $map,$gbShowQueryResults;

    if (!$gbShowQueryResults )
    {
    printf("&nbsp;");
    return;
    }
$numResultsTotal = 0;
for($iLayer=0; $iLayer < $map->numlayers; $iLayer++)
{
    $oLayer = $map->GetLayer($iLayer);
    $numResults = $oLayer->getNumResults();
    if ($numResults == 0)
        continue;
    $oLayer->open($map->shapepath);
    $selFields = explode(" ", $oLayer->getMetaData("RESULT_FIELDS"));
    // printf("</TR>\n");
    for ($iRes=0; $iRes < $numResults; $iRes++)
    {
        $oRes = $oLayer->getResult($iRes);
        $oShape = $oLayer->getShape($oRes->tileindex,$oRes->shapeindex);
        $oShape->free();
        $numResultsTotal++;
    }
    for($iField=0; $iField < sizeof($selFields); $iField++)
    {
        $crevvyp = $oShape->values[$selFields[$iField]];
    }
// _____ pripojení k databazi a vypis reviru

if (($crevvyp != 0) && ($crevvyp != 0)){
echo('<STYLE TYPE="TEXT/CSS"><!--');
echo('    .prvky {');
echo('        font-family: Arial;');
echo('        font-size: 9px;');
echo('        font-weight: bold;');
echo('        color: White;');
echo('    }');
echo('    .vypis {');
echo('        font-family: Arial;');
echo('        font-size: 9px;');
echo('        color: White;');
echo('    }');
echo('    .screvf {');
//echo(' height: 300px;');
echo(' overflow-y:scroll;');
//echo(' scrollbar-face-color:SILVER;');
//echo(' scrollbar-highlight-color:SILVER;');
//echo(' scrollbar-3dlight-color: BLACK;');
//echo(' scrollbar-darkshadow-color:BLACK;');
//echo(' scrollbar-shadow-color:SILVER;');
//echo(' scrollbar-track-color: #F5F5F5;');
//echo(' scrollbar-arrow-color: BLACK;');
echo('    display : inline;');
    echo('} ');
echo('    -->');
echo('</STYLE>');
$server="localhost";

```

```

        $uzivatel="";
        $heslo="";
$spojeni = mysql_connect($server,$uzivatel,$heslo) or die ("Spojení k
        databázi se nezdařilo.");
$databse=mysql_select_db("rybydb");
$vypis = mysql_query("SELECT * FROM revir WHERE id_r LIKE '$crevvyp'");
$radkovani=5;
if (!isset($max))
{
$max=mysql_num_rows($vypis);
$odkud=0;
}
if(isset($odkud))
{
$kam=$radkovani+$odkud;
if ($kam>$max) {$kam=$max;}
echo "<div class=scrinf align=center><table border=1 cellspacing=1
        cellpadding=1 align=left >";
$id_rev = MySQL_Result($vypis, "$i","id_r");
echo "<td width=70 class=prvky bgcolor=#FFA07A>číslo revíru:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","c_rev")." </td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>název revíru:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","naz_r")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>lokalita:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","lokal")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>v. k. ú.:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","v_k_u")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>delka</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","delka")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>rozloha:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","rozl")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>inforamce:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","info")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>omezení:</td>";
echo "<td class=vypis>".MySQL_Result($vypis, "$i","omez")."</td>";
echo "<tr>";
echo "<form action=jedul.php method=get>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>úlovky na tomto
        revíru:</td>";
echo "<INPUT TYPE=HIDDEN value=$id_rev name=revir>";
echo "<td class=vypis><div align=center><INPUT TYPE=SUBMIT NAME=tlacitko
        VALUE='zobraz' ALT=Digitalizace bodu class=tlac> </div></td>";
echo "</form>";
echo "</select>";
echo "</table>";
echo "</div>";
}
}
}
if ($crevvyp == 0){
        echo "<font face=Arial size=1 color=#FFFFFF><div align=center>Revír
        nenalezen</div></font>";
}
}

```

```

//pohyb v mape pomoci sipek
    if (isset($HTTP_POST_VARS[sz_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] - (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2] - (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[s_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0];
        $P1 = $extent_to_set[1] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2];
        $P3 = $extent_to_set[3] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[sv_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] + (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2] + (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3] + (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[j_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0];
        $P1 = $extent_to_set[1] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2];
        $P3 = $extent_to_set[3] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[v_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] + (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1];
        $P2 = $extent_to_set[2] +
(abs($extent_to_set[2] - $extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3];
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[z_x])){

```

```

        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] - (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1];
        $P2 = $extent_to_set[2] -
(abs($extent_to_set[2] - $extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3];
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[jv_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] + (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2] + (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
    if (isset($HTTP_POST_VARS[jz_x])){
        $extent_to_set = explode("
", $HTTP_POST_VARS["extent"]);
        $P0 = $extent_to_set[0] - (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P1 = $extent_to_set[1] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $P2 = $extent_to_set[2] - (abs($extent_to_set[2] -
$extent_to_set[0]))*0.3;
        $P3 = $extent_to_set[3] - (abs($extent_to_set[1] -
$extent_to_set[3]))*0.3;
        $map->setextent($P0,$P1,$P2,$P3);
    }
}

$image=$map->draw();
$image_url=$image->saveWebImage(MS_GIF,0,1,0);
$extent_to_html = $map->extent->minx." ".$map->extent->miny."
".$map->extent->maxx." ".$map->extent->maxy;
//$s_x = $map->extent->minx;
$refer=$map->drawReferenceMap();
$ref_url=$refer->saveWebImage(MS_GIF,0,1,0);
$scalebar=$map->drawScaleBar();
$scalebar_url=$scalebar->saveWebImage(MS_PNG,1,1,1);

```

```

?>
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function but_onload(but){
if (but != null){
for (i=0; i<=4; i++){
document.images[i].border=0;
}
document.images[but].border=1;

document.Mapa.sel_but.value = but;

```

```

    }
    }
function buttons(nazbut){
document.Mapa.sel_but.value = nazbut;
if (nazbut == 3){
    document.Mapa.submit();
}
if (nazbut == 4){
}
for (i=0;i<=4;i++){
    document.images[i].border=0;
}
document.images[nazbut].border=1;
}
function mapplet_apply(name, minx, miny, maxx, maxy, redraw) {
document.Mapa.imgx.value = minx;
    document.Mapa.imgy.value = miny;
    document.Mapa.imaxx.value = maxx;
    document.Mapa.imaxy.value = maxy;
    document.Mapa.submit();
}
function mapplet_error(message) {
alert(message);
}
}
</SCRIPT>

<TITLE>vlkova prohlížečka</TITLE>
<STYLE TYPE="TEXT/CSS"> <!--
    h1 {
        font-family: Arial;
        font-size: 16;
        font-weight: bold;
        color: #FEFEFE;
        text-align: center;
        background-color : #FFA07A;
    }
    .prvky {
        font-family: Arial;
        font-size: 10px;
        font-weight: bold;
        color: White;
    }
    .okraj { background-color: #FFA07A; border-left-width: 2px;
border-right-width: 2px; border-top-width: 2px; border-bottom-
width: 2px; border-color: #FEFEFE;}
    .nazvy {
        font-family: Arial;
        font-size: 11;
        font-weight: bold;
        color: #FEFEFE;
        text-align: center;
        background-color : #FFA07A;
        border-top : 0;
    }
}
-->
</STYLE>
</HEAD>
<div align="center"><BODY bgcolor="#004C9D" onload="but_onload(<?php echo
$HTTP_POST_VARS["sel_but"]?>)"></div>

```







```

        <input type="hidden" name="imaxy" value="">
        <input type="hidden" name="hled_m" value="<? echo
        $HTTP_POST_VARS["hled_m"]; ?>">
        <input type="hidden" name="id_bod" value="<? echo
        $HTTP_POST_VARS["id_bod"]; ?>">
        <INPUT TYPE=HIDDEN NAME="extent" VALUE="<?php echo
        $extent_to_html?>">
        <INPUT TYPE=HIDDEN NAME="ix" VALUE="<?php echo $X?>">
        <INPUT TYPE=HIDDEN NAME="iy" VALUE="<?php echo $Y?>">
    </form>
</td>
<td width="400" valign="top" bgcolor="#004C9D">
    <?
    if (isset($id_ulov))
{
echo "<div align=center class=nazvy>Informace o úlovku:</div>";
echo "<hr>";
$server="localhost";
    $uzivatel="";
    $heslo="";
    $connection = mysql_connect($server,$uzivatel,$heslo) or die
    ("Spojení k databázi se nezdařilo.");
    $database=mysql_select_db("rybydb");
//$vypis_in = mysql_query("SELECT * FROM ulovek WHERE ID_ul LIKE
    '$id_ulov'");
$vypis_in = mysql_query("SELECT * FROM ulovek,uzivatel WHERE ulovek.ID_ul
    LIKE '$id_ulov' and ulovek.id_uz=uzivatel.id_uz ");
$pokus = MySQL_Result($vypis_in, 0, "jmeno");
echo "<table border=1 cellpadding=1 cellspacing=1 align=left>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>Jméno:</td>";
echo "<td class=vypis>".MySQL_Result($vypis_in, 0,"jmeno")." </td>";
$ulovil = MySQL_Result($vypis_in, 0,"jmeno");
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>e-mail:</td>";
echo "<td class=vypis>".MySQL_Result($vypis_in, 0,"email")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>datum:</td>";
echo "<td class=vypis>".MySQL_Result($vypis_in, 0,"datum")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>revír:</td>";
$id_rev = MySQL_Result($vypis_in, 0,"id_rev");
$vypis_ret = mysql_query("SELECT * FROM revir WHERE id_r LIKE '$id_rev'");
echo "<td class=vypis>".MySQL_Result($vypis_ret, "id_rev","naz_r")."</td>";
echo "<tr>";
echo "<td width=70 class=prvky bgcolor=#FFA07A>popis:</td>";
echo "<td class=vypis>".MySQL_Result($vypis_in, 0,"popis")."</td>";
echo "<tr>";
// zjištění velikosti fotografie
    $pripona=strtok($foto, ".");
        $pripona=strtok(".");
        // echo $pripona;
        if($pripona == "jpg")
        {
$fn = "foto/" . $foto;
if (imagecreatefromjpeg($fn) == true)
{
$im = imagecreatefromjpeg($fn);
$width = imagesx($im);
$heig = imagesy($im);
$width = $width + 50;
$heig = $heig + 80;

```





## Příloha 2

Zdrojový kód výpisu všech úlovek (celkul.php):

```
<?
echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .prvky {');
echo('        font-family: Arial;');
echo('        font-size: 10px;');
echo('        font-weight: bold;');
echo('        color: White;');
echo('    }');
echo('    .hla {');
echo('        font-family: Arial;');
echo('        font-size: 14px;');
echo('        font-weight: bold;');
echo('        color: black;');
echo('        text-align: center;');
echo('    }');
echo('    -->');
echo('</STYLE>');
echo('<body bgcolor="#004C9D" link="#FFFFFF" vlink="#FFFF00">');

$server="localhost";
$uzivatel="";
$heslo="";
$spojeni = mysql_connect($server,$uzivatel,$heslo) or die ("Spojení k
databázi se nezdařilo.");
$databaze=mysql_select_db("rybydb");

if ($ulall == "Zobraz úlovky") {
    $vypis = mysql_query("SELECT * FROM ulovek,uzivatel WHERE
ulovek.id_uz=uzivatel.id_uz");}

else
    if ($tlacitko == "zobraz"){
        $vypis = mysql_query("SELECT * FROM ulovek WHERE id_rev LIKE
'$revir'");
    }

$radkovani=30;
if (!isset($max))
{
    $max=mysql_num_rows($vypis);
    $odkud=0;
}
if(isset($odkud))
{
    $kam=$radkovani+$odkud;
    if ($kam>$max) {$kam=$max;}

echo "<p class=prvky>";
echo "V databázi bylo vyhledáno <b>$max</b> záznamů";
echo "</p>";
echo "<a name=Zpět id=Zpět href=evidence.php title=Aplikace>Zpět</a>";

echo "<br />";
```

```

echo "<table width=870 height=18 border=2 cellpadding=1 align=center
      bgcolor=#99FF66 class=hla cellpadding=1>";
echo "<tr>";
// echo "<td width=20 >&nbsp;id</td>";
echo "<td width=100 nowrap>&nbsp;jmeno</td>";
echo "<td width=100 nowrap>&nbsp;email</td>";
echo "<td width=70 nowrap>&nbsp;datum</td>";
echo "<td width=200 nowrap>&nbsp;revir</td>";
echo "<td width=400 nowrap>&nbsp;popis</td>";
echo "<td width=40 nowrap>&nbsp;mapa</td></tr>";
echo "</table>";
echo "<p class=prvky>";

for ($i=$odkud ; $i<$kam ; $i++)
{
$rev_sl = "$id_revir, $lokalita";
echo "<form METHOD=post ACTION=http://localhost/vlk/evidence.php>";
$ID_R = MySQL_Result($vypis, "$i","ID_ul"); if (($i%2)==1)
echo "<table width=870 height=18 border=2 cellpadding=1 align=center
      bgcolor=#FFA07A class=prvky cellpadding=1>";
else
echo "<table width=870 height=18 border=2 cellpadding=1 align=center
      bgcolor=#004C9D class=prvky cellpadding=1>";
echo "<tr>";
$j = $i - 1;
$id_rev = MySQL_Result($vypis, "$i","id_rev");
$id_bod = MySQL_Result($vypis, "$i","ID_ul");

$id_ulov = MySQL_Result($vypis, "$i","ID_ul");

echo "<td width=100 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","jmeno")."</td>";
echo "<td width=100 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","email")."</td>";
echo "<td width=70 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","datum")."</td>";
      $vypis_revir = mysql_query("SELECT * FROM revir WHERE id_r LIKE
      '$id_rev'");
if ($id_rev != 0){
echo "<td width=200 nowrap>&nbsp;".MySQL_Result($vypis_revir,
      "id_rev","naz_r") . " - ". MySQL_Result($vypis_revir,
      "id_rev","lokal")."</td>";
$re = MySQL_Result($vypis_revir, "id_rev","naz_r");
}
else
{
echo "<td width=300 nowrap>&nbsp;</td>";
}
echo "<td width=400 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","popis")."</td>";
echo "<INPUT TYPE=HIDDEN value=$id_bod name=bodka>";
echo "<INPUT TYPE=HIDDEN value=$id_ulov name=id_ulov>";
echo "<input type=hidden name=hled_m id=hled_m value=$id_rev>";
echo "<td width=40 align=middle nowrap><INPUT TYPE=image
      SRC=Images/zoom_in.gif width=22 height=22 BORDER=0></td></tr>";
echo "</form>";
echo "</table>";
}
}
echo "</p>";

```

```

$pocet_tlacitek=intval($max/$radkovani);
if ($pocet_tlacitek < ($max/$radkovani)) { $pocet_tlacitek++; }

for ($i = 1; $i <= $pocet_tlacitek; $i++)
{ $y=($i*$radkovani)-$radkovani;
if ($y==$odkud){$tlacitko.= "<B>";}
$tlacitko.= "<font face=\"arial\" size=\"1\">&nbsp;";
<a href=\"celkull.php?ulall=Zobraz
        úlovy&revir=$id_rev&kam=$kam&max=$max&odkud=$y&\">$i</a>&nbsp; <
        /font>";
if ($y==$odkud){$tlacitko.= "</B>";}
}

echo "<br><br>";
echo "$tlacitko<br>";
echo "<a name=Zpět id=Zpět href=evidence.php title=Aplikace>Zpět</a>";
echo('</body>');

?>

```





```

echo "<td width=70 nowrap>&nbsp;datum</td>";
echo "<td width=200 nowrap>&nbsp;revir</td>";
echo "<td width=400 nowrap>&nbsp;popis</td>";
echo "<td width=40 nowrap>&nbsp;mapa</td></tr>";
echo "</table>";
echo "<p class=prvky>";

for ($i=$odkud ; $i<$kam ; $i++)
{
$rev_sl = "$id_revir, $lokalita";
echo "<form METHOD=post ACTION=http://localhost/vlk/evidence.php>";
$ID_R = MySQL_Result($vypis, "$i","ID_ul"); if (($i%2)==1)
echo "<table width=870 height=18 border=2 cellspacing=1 align=center
      bgcolor=#FFA07A class=prvky cellpadding=1>";
else
echo "<table width=870 height=18 border=2 cellspacing=1 align=center
      bgcolor=#004C9D class=prvky cellpadding=1>";
echo "<tr>";
$j = $i - 1;
$id_rev = MySQL_Result($vypis, "$i","id_rev");
$id_bod = MySQL_Result($vypis, "$i","ID_ul");

$id_ulov = MySQL_Result($vypis, "$i","ID_ul");

echo "<td width=100 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","jmeno")."</td>";
echo "<td width=100 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","email")."</td>";
echo "<td width=70 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","datum")."</td>";
      $vypis_revir = mysql_query("SELECT * FROM revir WHERE id_r LIKE
      '$id_rev'");
if ($id_rev != 0){
echo "<td width=200 nowrap>&nbsp;".MySQL_Result($vypis_revir,
      "id_rev","naz_r") . " - ". MySQL_Result($vypis_revir,
      "id_rev","lokal")."</td>";
$re = MySQL_Result($vypis_revir, "id_rev","naz_r");
}
else
{
echo "<td width=300 nowrap>&nbsp;</td>";
}
echo "<td width=400 nowrap>&nbsp;".MySQL_Result($vypis,
      "$i","popis")."</td>";
echo "<INPUT TYPE=HIDDEN value=$id_bod name=bodka>";
echo "<INPUT TYPE=HIDDEN value=$id_ulov name=id_ulov>";
echo "<input type=hidden name=hled_m id=hled_m value=$id_rev>";
echo "<td width=40 align=middle nowrap><INPUT TYPE=image
      SRC=Images/zoom_in.gif width=22 height=22 BORDER=0></td></tr>";
echo "</form>";
echo "</table>";
}
}
echo "</p>";
$pocet_tlacitek=intval($max/$radkovani);
if ($pocet_tlacitek < ($max/$radkovani)) { $pocet_tlacitek++; }
for ($i = 1; $i <= $pocet_tlacitek; $i++)
{ $y=($i*$radkovani)-$radkovani;
if ($y==$odkud){$tlacitko.= "<B>";}
$tlacitko.= "<font face=\"arial\" size=\"1\">&nbsp;";

```

```
<a
    href=\"jedull.php?tlacitko=zobraz&revir=$id_rev&kam=$kam&max=$max&odkud=$y&\">$i</a>&nbsp;</font>";
if ($y==$odkud){$tlacitko.= "</B>";}
}
echo "<br><br>";
echo "$tlacitko<br>";
echo "<a name=Zpět id=Zpět href=evidence.php title=Aplikace>Zpět</a>";
echo('</body>');
?>
```

## Příloha 4

Zdrojový kód výpisu on-line uživatelů (online.php):

```
<STYLE TYPE="TEXT/CSS"> <!--
    .prvky {
        font-family: Arial;
        font-size: 10px;
        font-weight: bold;
        color: White;
    }
-->
</STYLE>

<?php

$fcrr="online.txt";

$IPadresa=$REMOTE_ADDR;
if ($HTTP_X_FORWARDED_FOR)
    $IPadresa.="@".$HTTP_X_FORWARDED_FOR;
if ($HTTP_FORWARDED)
    $IPadresa.="@".$HTTP_FORWARDED;
if ($HTTP_CLIENT_IP)
    $IPadresa.="@".$HTTP_CLIENT_IP;
if ($X_HTTP_FORWARDED_FOR)
    $IPadresa.="@".$X_HTTP_FORWARDED_FOR;
if ($X_FORWARDED_FOR)
    $IPadresa.="@".$X_FORWARDED_FOR;
$aktual=time();
$pole=file($fcrr);
for($i=0; $i<Count($pole); $i++):
    $adresa=StrTok($pole[$i],":");
    $cas=StrTok(":");
    if ($adresa==$IPadresa):
        $pole[$i]="";
    endif;
    if ($cas<($aktual-360)):
        $pole[$i]="";
    endif;
endfor;
$Fp=fopen($fcrr,"w");
$celkem=0;
for($i=0; $i<Count($pole); $i++):
    if ($pole[$i]!=""):
        FPutS($Fp,$pole[$i]);
        $celkem++;
    endif;
endfor;
FPutS($Fp,$IPadresa." : ".$aktual."\n");
$celkem++;
FClose($Fp);
echo "<p CLASS=prvky>";
echo "on-line uživatelů:" . $celkem ;
echo "</p>";
?>
```

## Příloha 5

Zdrojový kód sloužící ke spuštění aplikace (index.html):

```
<html>
<head>
<title>Spuštění aplikace</title>
</head>
<body>
<SCRIPT language=JavaScript>
<!--

function OpenJSLWindow(url, jmeno, w, h)
{
    Fokus = window.open(url, jmeno,'resizable=1,
        scrollbars=1,top=0,left=0,menubar=0,width='+ (screen.availWidth
        - 10)+' , height=' + screen.availHeight);
    Fokus.focus()
}
// -->
</SCRIPT>
<p align="center"><font face="Verdana, Arial" size="2">
<a href="javascript:
    OpenJSLWindow('evidence.php', 'open',1024,768);">Otevřít
    aplikaci</a>
</font></p>
</body>
</html>
```

## Příloha 6

Zdrojový kód sloužící k výpisu všech revírů (souis.php):

```
<?
echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .prvky {');
echo('        font-family: Arial;');
echo('        font-size: 10px;');
echo('        font-weight: bold;');
echo('        color: White;');
echo('    }');
echo('    .hla {');
echo('        font-family: Arial;');
echo('        font-size: 14px;');
echo('        font-weight: bold;');
echo('        color: black;');
echo('        text-align: center;');
echo('    }');
echo('    .sub {');
echo('        font-family: Arial;');
echo('        font-size: 14px;');
echo('        font-weight: bold;');
echo('        color: black;');
echo('        background-image: c:/apache/htdocs/vlk/Images/zoom_out.gif;');
echo('    }');
echo('    -->');
echo('</STYLE>');

echo('<body bgcolor="#004C9D" link="#FFFFFF" vlink="#FFFF00">');

$server="localhost";
    $uzivatel="";
    $heslo="";
$spojeni = mysql_connect($server,$uzivatel,$heslo) or die ("Spojení k
    databázi se nezdařilo.");
$database=mysql_select_db("rybydb");

$sall = $_HTTP_GET_VARS["sall"];
if ($sall == "Zobraz revíry") {
    $vypis = mysql_query("SELECT * FROM revir");}
else
{
if ($rev == 1) {
    $vypis = mysql_query("SELECT * FROM
    revir WHERE c_rev LIKE '$hlrev'");}
else
{$vypis = mysql_query("SELECT * FROM revir WHERE MATCH (naz_r) AGAINST
    ('$hlrev')");}
}

$radkovani=30;
if (!isset($max))
{
$max=mysql_num_rows($vypis);
$odkud=0;
}
if(isset($odkud))
{
$kam=$radkovani+$odkud;
```

```

if ($kam>$max) {$kam=$max;}

echo "<p class=prvky>";
echo "V databázi bylo vyhledáno <b>$max</b> záznamů";
echo "</p>";
echo "<br />";
echo "<table width=950 height=18 border=2 cellspacing=1 align=center
      bgcolor=#99FF66 class=hla cellpadding=1>";
echo "<tr>";
echo "<td width=50 >&nbsp;číslo revíru</td>";
echo "<td width=100 >&nbsp;název revíru</td>";
echo "<td width=80 >&nbsp;lokalita</td>";
echo "<td width=70 >&nbsp;v. k. ú.</td>";
echo "<td width=30 >&nbsp;delka</td>";
echo "<td width=30 >&nbsp;rozloha</td>";
echo "<td width=200 >&nbsp;inforamce</td>";
echo "<td width=300 >&nbsp;omezení</td>";
echo "<td width=40 >&nbsp;mapa</td></tr>";
echo "</table>";
echo "<p class=prvky>";
for ($i=$odkud ; $i<$kam ; $i++)
{
echo "<form name=$i METHOD=post ACTION=http://localhost/vlk/evidence.php>";
$ID_R = MySQL_Result($vypis, "$i","id_r"); if (($i%2)==1)
echo "<table width=950 height=18 border=2 cellspacing=1 align=center
      bgcolor=#FFA07A class=prvky cellpadding=1>";
else
echo "<table width=950 height=18 border=2 cellspacing=1 align=center
      bgcolor=#004C9D class=prvky cellpadding=1>";
echo "<tr>";
$id_reviru = MySQL_Result($vypis, "$i","id_r");
echo "<td width=50 >&nbsp;".MySQL_Result($vypis, "$i","c_rev")."</td>";
echo "<td width=100 >&nbsp;".MySQL_Result($vypis, "$i","naz_r")."</td>";
echo "<td width=80 >&nbsp;".MySQL_Result($vypis, "$i","lokal")."</td>";
echo "<td width=70 >&nbsp;".MySQL_Result($vypis, "$i","v_k_u")."</td>";
echo "<td width=30 >&nbsp;".MySQL_Result($vypis, "$i","delka")."</td>";
echo "<td width=30 >&nbsp;".MySQL_Result($vypis, "$i","rozl")."</td>";
echo "<td width=200 >&nbsp;".MySQL_Result($vypis, "$i","info")."</td>";
echo "<td width=300 >&nbsp;".MySQL_Result($vypis, "$i","omez")."</td>";
echo "<input type=hidden name=hled_m id=hled_m value=$id_reviru>";
echo "<td width=40 align=middle><INPUT TYPE=image SRC=Images/zoom_in.gif
      width=22 height=22 BORDER=0></td></tr>";
echo "</form>";
echo "</table>";
}
}

echo "</p>";

$pocet_tlacitek=intval($max/$radkovani);
if ($pocet_tlacitek < ($max/$radkovani)) { $pocet_tlacitek++; }

for ($i = 1; $i <= $pocet_tlacitek; $i++)
{ $y=($i*$radkovani)-$radkovani;
if ($y==$odkud){$tlacitko.= "<B>";}
$tlacitko.= "<font face=\"arial\" size=\"1\">&nbsp;";
<a href=\"soupis1.php?kam=$kam&max=$max&odkud=$y&\">$i</a>&nbsp;</font>";
if ($y==$odkud){$tlacitko.= "</B>";}
}
echo "<br><br>";
echo "$tlacitko<br>";

```

```
echo "<a name=Zpět id=Zpět href=evidence.php title=Aplikace>Zpět</a>";  
echo('</body>');
```

```
?>
```



## Příloha 7

Zdrojový kód sloužící k ukládání informací o úlovku (uloz.php):

```
<?php

$njmeno = $HTTP_POST_VARS["njmeno"];
$ndatum = $HTTP_POST_VARS["ndatum"];
$nrevir = $HTTP_POST_VARS["nrevir"];
$npopis = $HTTP_POST_VARS["npopis"];
$nemail = $HTTP_POST_VARS["nemail"];

$errorString = "";

if (empty($njmeno))
    $errorString .=
        "\n<br>Pole pro jméno nemůže být prázdné";
if (empty($ndatum))
    $errorString .=
        "\n<br>Pole pro datum nemůže obsahovat
        písmena";
if (empty($nrevir))
    $errorString .=
        "\n<br>Pole pro revír nemůže být prázdné";

if (empty($npopis))
    $errorString .=
        "\n<br>Pole pro popis nemůže být prázdné";

if (!empty($errorString))
{
?>

<html>
<head>
    <title>Uložení úlovku</title>
</head>

<body bgcolor="#004C9D" link="#FFFFFF" vlink="#FFFF00">
<h1><font color="#FFFFFF">Chyby v nových údajích</font></h1>
<font color="#FFFFFF"><?=$errorString?></font>
<br><a href="evidence.php">Návrat na formulář údajů o novém úlovku</a>
</body>
</html>
<?php

exit;
}
echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .prvky {'');
echo('        font-family: Arial;');
echo('        font-size: 14px;');
echo('        font-weight: bold;');
echo('        color: White;');
echo('        text-align: center;');
echo('    }');
echo('    .upl {'');
echo('        font-family: Arial;');
echo('        font-size: 10px;');
echo('        font-weight: bold;');
```

```

echo('      color: White;');
echo('      text-align: center;');
echo('      }');

echo('      -->');
echo('</STYLE>');
$server="localhost";
    $uzivatel="";
    $heslo="";

    $connection = mysql_connect($server,$uzivatel,$heslo) or die
("Spojení k databázi se nezdařilo.");
    $database=mysql_select_db("rybydb");

echo "<table width=450 height=250 border=2 cellspacing=1 align=center
bordercolor=#FFA07A bgcolor=#004C9D class=hla cellpadding=1>";
echo "<tr>";
echo "<td class=prvky>";

if($soubor != "none")
{
    if($soubor_size>100000)
    {
        echo 'Nastala chyba vetsi nez 100kB!!!';
    }
    else
    {
        if($soubor_size == 0)
        {
            echo 'Nastala chyba ma nul delku !!!';
        }
        else
        {
            $pripona=strtok($soubor_name, ".");
            $pripona=strtok(".");

            if($pripona == "jpg")
            {
                $vypzaz = mysql_query("SELECT * FROM ulovek ");
                $max=mysql_num_rows($vypzaz);
                $plus = $max + 1;
                if (copy ($soubor,
"foto/$plus"."_"."$plus"."_"."$pripona))
                {
                    $cesta = ($plus"."_"."$plus"."_"."$pripona);

                    echo "<p class=upl>";
                    echo 'Soubor : ', $soubor_name, ' (', $soubor_size, 'B -
', $soubor_type, ') byl úspěšně nahrán na server...';
                    echo "</p>";
                    $query_uz = "INSERT INTO uzivatel
                    set id_uz = null, " .
                    "jmeno = \"\" . $jmeno . "\", " .
                    "email = \"\" . $email . "\"";
                    if (!(@ mysql_query ($query_uz, $connection))){$id_uz =
mysql_insert_id();}
                    $server="localhost";
                    $uzivatel="";
                    $heslo="";
                    $connection = mysql_connect($server,$uzivatel,$heslo) or
die ("Spojení k databázi se nezdařilo.");

```

```

$database=mysql_select_db("rybydb");
$vypis_idi = mysql_query("SELECT * FROM uzivatel");
$maxi=mysql_num_rows($vypis_idi);
$maxi = $maxi - 1;
$id_uziv = MySQL_Result($vypis_idi, $maxi,"id_uz");

$query = "INSERT INTO ulovek
set ID_ul = null, " .
"id_uz = \"\" . $id_uziv . "\", " .
"datum = \"\" . $ndatum . "\", " .
"popis = \"\" . $npopis . "\", " .
"foto = \"\" . $cesta . "\", " .
"id_rev = \"\" . $nrevir . "\"";
}
}
else
{
if($pripona == "gif")
{
$vypzaz = mysql_query("SELECT * FROM ulovek ");
$max=mysql_num_rows($vypzaz);
$plus = $max + 1;
if (copy ($soubor,
"foto/$plus"."_"."$plus"."_"."$pripona))
{
$cesta = ($plus"."_"."$plus"."_"."$pripona);
echo "<p class=upl>";
echo 'Soubor : ', $soubor_name,
(',$soubor_size,'B - ', $soubor_type,') byl úspěšně nahrán na
server...';
echo "</p>";

$query_uz = "INSERT INTO uzivatel
set id_uz = null, " .
"jmeno = \"\" . $njmeno . "\", " .
"email = \"\" . $nemail . "\"";
if (!(@ mysql_query ($query_uz,
$connection))){$id_uz = mysql_insert_id();}
$server="localhost";
$uzivatel="";
$heslo="";
$connection = mysql_connect($server,$uzivatel,$heslo) or
die ("Spojení k databázi se nezdařilo.");
$database=mysql_select_db("rybydb");
$vypis_idi = mysql_query("SELECT * FROM uzivatel");
$maxi=mysql_num_rows($vypis_idi);
$maxi = $maxi - 1;
$id_uziv = MySQL_Result($vypis_idi, $maxi,"id_uz");

$query = "INSERT INTO ulovek
set ID_ul = null, " .
"id_uz = \"\" . $id_uziv . "\", " .
"datum = \"\" . $ndatum . "\", " .
"popis = \"\" . $npopis . "\", " .
"foto = \"\" . $cesta . "\", " .
"id_rev = \"\" . $nrevir . "\"";
}
}
else
{

```

```

        echo 'Nastala chyba s priponou!!!';
    }
}
}
else
{
$vypzaz = mysql_query("SELECT * FROM ulovek ");
    $max=mysql_num_rows($vypzaz);
    $plus = $max + 1;

echo "Nebyla vložena fotografie.";

    $query_uz = "INSERT INTO uzivatel
    set id_uz = null, " .
    "jmeno = \" . $njmeno . "\", " .
    "email = \" . $nemail . \"";
    if (!(@ mysql_query ($query_uz,
    $connection))){$id_uz = mysql_insert_id();}
    $server="localhost";
    $uzivatel="";
    $heslo="";
    $connection = mysql_connect($server,$uzivatel,$heslo) or
die ("Spojení k databázi se nezdařilo.");
    $database=mysql_select_db("rybydb");
    $vypis_idi = mysql_query("SELECT * FROM uzivatel");
    $maxi=mysql_num_rows($vypis_idi);
    $maxi = $maxi - 1;
    $id_uziv = MySQL_Result($vypis_idi, $maxi,"id_uz");
    $query = "INSERT INTO ulovek
    set ID_ul = null, " .
    "id_uz = \" . $id_uziv . "\", " .
    "datum = \" . $ndatum . "\", " .
    "popis = \" . $npopis . "\", " .
    //"foto = \" . $cesta . "\", " .
    "id_rev = \" . $nrevir . \"";

}

if (!(@ mysql_query ($query, $connection))){$IDul = mysql_insert_id();}
echo "<form action=evidence.php method=POST enctype='multipart/form-
data'>";
echo "<p class=prvky>";
echo "Pro zakreslení místa úlovku klikněte do mapy.";
echo "</p>";
$vypid = mysql_query("SELECT * FROM ulovek ");
    $max=mysql_num_rows($vypid);
    $server="localhost";
    $uzivatel="";
    $heslo="";

    $connection = mysql_connect($server,$uzivatel,$heslo) or die
("Spojení k databázi se nezdařilo.");
    $database=mysql_select_db("rybydb");
    $vypis_id = mysql_query("SELECT * FROM ulovek");
    $max=mysql_num_rows($vypis_id);
    $max = $max - 1;
    $id_ulovek = MySQL_Result($vypis_id, $max,"ID_ul");
echo "<INPUT TYPE=HIDDEN NAME=id_bod VALUE=$id_ulovek>";
echo "<INPUT TYPE=HIDDEN NAME=hled_m VALUE=$nrevir ALT=Digitalizace bodu
>";

```

```
echo "<INPUT TYPE=HIDDEN NAME=sel_but VALUE=5 ALT=Digitalizace bodu >";
echo "<div align=center><INPUT TYPE=SUBMIT NAME=tlacitko VALUE='Zakresli
      bod' ALT=Digitalizace bodu ></div>";
echo "</form>";
echo('<body bgcolor="#004C9D" link="#FFFFFF" vlink="#FFFF00">');
echo "<p class=upl>Záznam byl uložen do databáze</p> ";
echo('</body>');
echo "</td>";
echo "</tr>";
echo "</table>";
?>
```

## Příloha 8

Zdrojový kód sloužící k zobrazení fotografie úlovku (foto.php):

```
<?

echo('<STYLE TYPE="TEXT/CSS"> <!--');
echo('    .vypis {'');
echo('        font-family: Arial;');
echo('        font-size: 16px;');
echo('        color: White;');
echo('        font-weight: bold;');
echo('        text-align: center;');
echo('    }');
echo('    -->');
echo('</STYLE>');

dl('php_gd.dll');

$foto = $HTTP_POST_VARS["foto"];
$ulovil = $HTTP_POST_VARS["ulovil"];

//echo $foto;

$pripona=strtok($foto, ".");
    $pripona=strtok(".");
    // echo $pripona;
    if($pripona == "jpg")
    {
$fn = "foto/" . $foto;

$im = imagecreatefromjpeg($fn);

$width = imagesx($im);
$height = imagesy($im);
//echo $width;
//echo $height;
echo "<div align=center><img src='Foto/$foto' border=1
        align=center></div>";
echo "<p class=vypis>Úlovek jenž chytl $ulovil</p>";
}
else
{
echo "<img src='Foto/$foto' >";
}
?>
<html>
<head>
        <title>Fotografie úlovku</title>
</head>

<body bgcolor="#004C9D" link="#FFFFFF" vlink="#FFFF00" >
<h1><font color="#FFFFFF"></font></h1>
</body>
</html>
```

## Příloha 9

Zdrojový kód sloužící k definici mapového okna (mapa.map):

```
NAME FIRSTMAP
SIZE 450 300
STATUS ON
EXTENT -909802 -1254593 -429836 -902721
UNITS METERS
SHAPEPATH "./data"
IMAGECOLOR 255 255 255
SYMBOLSET "./symbols/symbols.sym"
FONTSET "./font/fonts.txt"

WEB
IMAGEPATH "c:/apache/htdocs/vlk/tmp/"
IMAGEURL "./tmp/"
END

#*****
LEGEND
  KEYSIZE 11 6
  LABEL
    TYPE BITMAP
    SIZE tiny
    COLOR 0 0 89
  END
STATUS EMBED

END

#*****

REFERENCE
  STATUS ON
  IMAGE referen.gif
  SIZE 220 144
  EXTENT -909802.15 -1254593.06 -429836.43 -902721.08
  COLOR -1 -1 -1
  OUTLINECOLOR 255 255 255
END

#*****

SCALEBAR
  IMAGECOLOR 255 255 255
  LABEL

      SIZE tiny
      COLOR 0 0 0

  END
  STYLE 0
  SIZE 150 3
  COLOR 0 0 0
  OUTLINECOLOR 0 0 0
  UNITS METERS
```

```

INTERVALS 3
TRANSPARENT on
STATUS ON
END

#*****

LAYER
NAME OKRESY
TYPE polygon
STATUS ON
DATA Okresy.shp

CLASS
NAME "okresy"
OUTLINECOLOR 200 200 200

END
END # LAYER

LAYER # Kraje
NAME Kraje
DATA kraje.shp
LABELITEM "Kodre"
LABELMINSCALE 150000
TYPE POLYGON
STATUS ON
CLASSITEM "Kodre"
CLASS
NAME "kraje"
#EXPRESSION 'OLO'
OUTLINECOLOR 0 0 0
LABEL
TYPE TRUETYPE
ANTIALIAS
SIZE 8
MINSIZE 6
MAXSIZE 12
FONT arial
COLOR 27 24 122
OUTLINECOLOR 255 255 255

END

END
END

#*****

LAYER
NAME LESY
TYPE polygon
MAXSCALE 900000
STATUS ON
DATA Lesy
CLASS
NAME "lesy"
COLOR 157 236 183
OUTLINECOLOR 157 236 183

```



```

END
END # LAYER

#*****

LAYER
NAME SIDLA
TYPE polygon
MAXSCALE 900000
STATUS ON
LABELITEM "J"
LABELMAXSCALE 250000
DATA Sidla.shp
CLASS
NAME "sídlá"
COLOR 201 210 217
OUTLINECOLOR 0 0 0
LABEL
        TYPE TRUETYPE
        ANTIALIAS
        SIZE 5
        MINSIZE 6
        MAXSIZE 12
        FONT arial
        COLOR 122 80 122

END

END
END # LAYER

#*****

LAYER
NAME SILNICE
TYPE line
MAXSCALE 500000
STATUS ON
DATA Silnice.shp
CLASS
NAME "silnice"
COLOR 166 182 197
END
END # LAYER

#*****

LAYER
NAME toky
        METADATA
        "DESCRIPTION" "Vodní tok"
        "RESULT_FIELDS" "ID_POD"
        END
TYPE line
TOLERANCE 0
TOLERANCEUNITS pixels
MAXSCALE 900000

STATUS ON
DATA toky
LABELITEM "W1250"

```

```

LABELMAXSCALE 150000
CLASS
  NAME '>47'
  NAME "revír"
  TEMPLATE "ttt_query.html"
  COLOR 4 126 229
  SYMBOL 1
  SIZE 3
  EXPRESSION ([C_rev]>47)
LABEL
ANGLE AUTO
POSITION CC
BUFFER 10
  TYPE TRUETYPE
  ANTIALIAS
  SIZE 8
  MINSIZE 6
  MAXSIZE 12
  FONT arial
  COLOR 25 60 215
  END #label
END #class

CLASS
  NAME '<47'
  NAME "toky"
  TEMPLATE "ttt_query.html"
  COLOR 4 126 229
  EXPRESSION ([C_rev]<47)
LABEL
ANGLE AUTO
POSITION CC
BUFFER 10
  TYPE TRUETYPE
  ANTIALIAS
  SIZE 8
  MINSIZE 6
  MAXSIZE 12
  FONT arial
  COLOR 25 60 215
  END #label
END #class

END # LAYER

#*****

LAYER
NAME voda
  METADATA
  "DESCRIPTION" "Vodní plocha"
  "RESULT_FIELDS" "ID_POD"
  END
TYPE POLYGON
MAXSCALE 900000
STATUS ON
DATA voda
LABELITEM "j"
LABELMAXSCALE 150000
CLASS
  NAME "voda"

```

```
TEMPLATE "ttt_query.html"  
COLOR 4 126 229  
  
OUTLINECOLOR 4 126 229
```

```
LABEL
```

```
TYPE TRUETYPE  
ANTIALIAS  
SIZE 8  
MINSIZE 6  
MAXSIZE 12  
FONT arial  
COLOR 27 24 122  
OUTLINECOLOR 255 255 255
```

```
END
```

```
END
```

```
END
```

```
*****
```

```
*****
```

```
END # Konec MAP FILE
```

