



**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Název:** Databáze zvířat pro neziskovou organizaci  
**Student:** Tomáš Taro  
**Vedoucí:** Ing. Lukáš Maleček  
**Studijní program:** Informatika  
**Studijní obor:** Webové a softwarové inženýrství  
**Katedra:** Katedra softwarového inženýrství  
**Platnost zadání:** Do konce zimního semestru 2020/21

### Pokyny pro vypracování

Cílem práce je vytvořit novou verzi webové aplikace Plemenná kniha (databázi zvířat), která slouží k evidenci informací o zvířatech v aktivním chovu, evidenci odchovů, schvalování vrhů, pomáhá členům neziskové organizace v plánování chovu a umožňuje jednoduché generování průkazů původu pro odchovy.

- \* Analyzujte současný stav aplikace, seznamte se s její vnitřní implementací, současnou strukturou databáze a již implementovanými funkcemi.
- \* Na základě analýzy navrhnete optimálnější strukturu databáze (zejména vyřešte současné duplicitní ukládání dat).
- \* Vytvořte zcela novou aplikaci pro správu plemenné knihy v PHP s použitím MySQL databáze. V aplikaci využijte současné návrhové a architektonické vzory a postupy.
- \* Implementujte nové funkce na základě jejich specifikace dodaných vedoucím práce.
- \* Ověřte správnou funkci aplikace pomocí unit a případně i integračních testů.
- \* Migrujte data z původní aplikace do nové a navrhnete postup, jak zajistit kontrolu a ověření správnosti importu.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 21. února 2019





**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

Bakalárska práca

## **Databáza zvierat pre neziskovú organizáciu**

*Tomáš Taro*

Katedra softwarového inžénýrství  
Vedúci práce: Ing. Lukáš Maleček

5. mája 2020



---

## Pod'akovanie

Touto cestou by som sa chcel poďakovať pánovi Ing. Lukášovi Malečkovi za vedenie a trpezlivosť pri riešení problematiky týkajúcej sa tejto bakalárskej práce. Taktiež by som sa chcel poďakovať rodine za neustálu podporu počas celého štúdia.



---

## Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 5. mája 2020

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2020 Tomáš Taro. Všetky práva vyhrazené.

*Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.*

### **Odkaz na túto prácu**

Taro, Tomáš. *Databáza zvierat pre neziskovú organizáciu*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.



---

## Abstrakt

V niekoľkých vetách zhrňte obsah a prínos tejto práce v slovenčine. Po prečítaní abstraktu by mal čitateľ mať dosť informácií pre rozhodnutie, či Vašu prácu chce čítať.

**Kľúčová slova** Nahradte zoznamom kľúčových slov v slovenčine oddelených čiarkou.

---

## Abstract

Sem doplňte ekvivalent abstraktu Vašej práce v angličtině.

**Keywords** Nahradte zoznamom kľúčových slov v angličtine oddelených čiarkou.



---

# Obsah

Úvod	1
<b>1 Cieľ práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Analýza súčasnej aplikácie . . . . .	5
2.2 Analýza požiadaviek . . . . .	5
2.2.1 Funkčné požiadavky . . . . .	5
2.2.2 Nefunkčné požiadavky . . . . .	9
2.3 Používateľské role . . . . .	10
2.3.1 Bežný používateľ . . . . .	10
2.3.2 Registrátor zvierat . . . . .	11
2.3.3 Schvalovateľ vrhov . . . . .	11
2.3.4 Administrátor . . . . .	11
2.4 Prípady použitia . . . . .	12
2.4.1 Prípady použitia zvierat . . . . .	12
2.4.2 Prípady použitia vrhov . . . . .	15
<b>3 Návrh</b>	<b>17</b>
3.1 Návrh architektúry . . . . .	17
3.1.1 Architektúra MPA aplikácie . . . . .	17
3.1.2 Architektúra SPA aplikácie . . . . .	19
3.1.3 Porovnanie a voľba architektúry . . . . .	21
3.2 Návrh technológií . . . . .	21
3.2.1 Back-end technológie . . . . .	21
3.2.2 Front-end technológie . . . . .	21
3.3 Návrh modelových tried . . . . .	21
3.4 Návrh REST API . . . . .	21



---

## Zoznam obrázkov

3.1	Diagram znázorňujúci architektúru MPA aplikácií . . . . .	18
3.2	Diagram znázorňujúci architektúru SPA aplikácií . . . . .	20



---

# Úvod





## Cieľ práce

Teoretická časť práce sa zameria na analýzu súčasnej webovej aplikácie, ktorá pomáha členom organizácie s evidenciou zvierat a ich odchovov. Ďalej budú špecifikované funkčné a nefunkčné požiadavky spolu s používateľskými rolami a prípadmi použitia, na základe ktorých bude navrhnutá architektúra a technológie, ktoré budú použité v novej webovej aplikácii.

Praktická časť práce sa bude zaoberať implementáciou aplikácie na základe výstupov z teoretickej časti práce a migráciou pôvodných dát do novej aplikácie. Nakoniec bude aplikácia otestovaná pomocou integračných testov.

Konečný výstup praktickej časti bude prospešný pre členov organizácie, ktorým sa výrazne zjednoduší a prehľadní evidencia informácií o zvieratách.



---

# Analýza

Táto kapitola sa venuje analýze súčasnej webovej aplikácie, spolu s jej architektúrou. Následne sú vymedzené jednotlivé funkčné a nefunkčné požiadavky novej webovej aplikácie, na ktoré nadväzuje sekcia používateľských rolí. Záver kapitoly je určený pre analýzu prípadov použitia, ktoré vyplývajú z funkčných požiadaviek.

## 2.1 Analýza súčasnej aplikácie

== TO BE ADDED ==

## 2.2 Analýza požiadaviek

V tejto sekcii sa nachádza popis všetkých požiadaviek kladených na vznikajúcu webovú aplikáciu. Tieto požiadavky delíme na funkčné a nefunkčné.

### 2.2.1 Funkčné požiadavky

Funkčné požiadavky vo všeobecnosti vymedzujú hranice aplikácie v kontexte jej funkcionality, ktorú používateľ od aplikácie očakáva. Ich úlohou je taktiež spresniť odhad pracnosti na vznikajúcej aplikácii [1].

#### 2.2.1.1 Evidencia zvierat

Aplikácia musí umožniť jednoduchú správu zvierat, od vytvorenia nového zvierata, cez zobrazenie jeho detailov, až po následnú editáciu, prípadné jeho zmazanie.

V evidencii zvierat je pre každé zviera potrebné ukladať nasledujúce údaje:

- meno a prezývku

## 2. ANALÝZA

---

- dátum narodenia
- majiteľa a chovateľa zvierata
- vrh, v ktorom sa zviera narodilo
- matku a otca zvierata
- pohlavie
- farbu očí
- typ uší
- farbu a typ srsti
- znaky
- dátum a dôvod úmrtia

Pre uľahčenie vytvorenia a editovania zvierata budú textové polia v maximálnej možnej miere interaktívne.

To znamená, že polia majiteľa a chovateľa zvierata vrátia na základe vstupu zoznam majiteľov, resp. chovateľov, ktorí už sú evidovaní v aplikácii. Následne si z tohto zoznamu používateľ zvolí požadovaného človeka. V prípade, že aplikácia požadovaného človeka nevráti, používateľovi bude ponúknutá možnosť ho dodatočne vytvoriť.

Na podobnom princípe bude fungovať aj textové pole pre vrh, s tým rozdielom, že podľa zvoleného vrhu aplikácia predvyplní matku a otca zvierata.

### 2.2.1.2 Evidencia registrácií zvierata

Je potrebné, aby aplikácia ponúkala možnosť zaregistrovať evidované zviera pod záujmový klub. Pri takejto registrácii sa budú zbierať nasledovné údaje:

- klub<sup>1</sup>, pod ktorým bude zviera zaregistrované
- typ registrácie<sup>2</sup>
- registračné číslo
- dátum registrácie
- informácia, či je nasledovný chov povolený
- informácia o obmedzení chovu

---

<sup>1</sup>Na výber z možností: ČKP, SOCHP alebo Ostatné

<sup>2</sup>Vyžadované iba v prípade registrácie zvierata pod klubom ČKP

### 2.2.1.3 Evidencia vrhov

Medzi nevyhnutnú funkcionálnosť aplikácie sa radí aj evidencia vrhov. Tak ako pri evidencii zvierat, tak aj v tomto prípade používateľské rozhranie umožní vytvoriť nový vrh, zobraziť ho, respektíve ho editovať alebo vymazať.

Pre účel evidencie vrhov sa budú v aplikácii ukladať nasledujúce údaje:

- typ a označenie vrhu
- majiteľ vrhu
- meno a kontakt na chovateľa
- dátum narodenia
- matka a otec vrhu
- línia
- genetické informácie
- počet narodených a odchovaných mláďat
- počet odchovaných samčekov a samičiek
- počet mláďat určených pre maznanie a následný chov

Rovnako ako pri evidencii zvierat, tak aj tu umožní aplikácia interaktívne zvoliť matku, otca a majiteľa vrhu zobrazením zoznamu uložených zvierat/ludí.

### 2.2.1.4 Správa žiadostí o schválenie vrhov

Pre organizáciu je žiaduce, aby aplikácia obsahovala správu žiadostí o schválenie vrhu.

V rámci žiadosti o schválenie vrhu rozlišujeme dva typy osôb – žiadateľa o schválenie vrhu<sup>3</sup>, a registrátora vrhu.

Aplikácia žiadateľovi umožní poslať žiadosť o schválenie vrhu s možnosťou zanechania poznámky pre registrátora. Po odoslaní a úspešnom spracovaní tejto žiadosti serverom sa odošle e-mail všetkým registrátorom s informáciou o vytvorení novej žiadosti o schválenie vrhu. Na tento e-mail bude môcť zareagovať akýkoľvek registrátor, ktorý sa na základe dostupných informácií o vrhu rozhodne, či danú žiadosť schváli alebo zamietne. O zmene stavu žiadosti bude žiadateľ informovaný e-mailom.

---

<sup>3</sup>Zväčša majiteľ daného vrhu

### 2.2.1.5 Zobrazenie rodokmeňu zvieráťa a vrhu

Pre jednoduchšiu vizualizáciu predkov konkrétneho zvieráťa aplikácia ponúkne zobrazenie rodokmeňu zvieráťa vo forme jednoduchej tabuľky.

V tejto tabuľke budú okrem mien zvierat zobrazené aj dodatočné informácie o zvierati, definované v sekcii 2.2.1.1. Rodokmeň bude zobrazený v rámci jednotlivých zvierat a vrhov.

V prípade vrhu bude rodokmeň zobrazovať predkov matky a otca daného vrhu.

### 2.2.1.6 Tvorba poznámok pre zviera a vrh

V rámci aplikácie bude potrebné implementovať poznámky, ktoré budú môcť byť priradené jednotlivým zvieratám a vrhom. Takáto poznámka by mala mať nastaviteľnú viditeľnosť<sup>4</sup> a typ<sup>5</sup>. Taktiež musí poskytnúť informáciu, kedy bola vytvorená, respektíve editovaná.

Nakoľko sa jedná o poznámky pre zviera a vrh, budú zobrazené u príslušných zvieratách, resp. vrhoch, ku ktorým sa vzťahujú.

### 2.2.1.7 Zobrazenie histórie zmien zvierat a vrhov

Medzi želanú funkcionálnosť novej webovej aplikácie patrí sledovanie a následné zobrazenie histórie zmien pre všetky zvieratá a vrhy

Aplikácia bude zaznamenávať nasledujúce zmeny v systéme:

- vytvorenie zvieráťa/vrhu
- úprava údajov zvieráťa/vrhu
- zmazanie zvieráťa/vrhu
- obnova<sup>6</sup> zmazaného zvieráťa/vrhu

Pri každej zmene popísanej vyššie je taktiež nutné ukladať, kto a kedy danú zmenu vykonal. V prípade úpravy údajov budú navyše zaznamenané tie údaje, ktoré boli zmenené používateľom. Túto históriu zmien bude možné vidieť vo forme tabuľky u každého zvieráťa, resp. vrhu.

### 2.2.1.8 Generovanie preukazov

Pre potreby organizácie je nevyhnutné implementovať generovanie preukazov (osvedčení) vo forme PDF súboru. V tomto súbore bude prvá strana vyplnená informáciami o danom zvierati 2.2.1.1, vrátane jeho registrácie zo sekcie

---

<sup>4</sup>Poznámka môže byť buď verejná alebo súkromná

<sup>5</sup>Typ poznámky je jeden z nasledujúcich: všeobecná, upozornenie alebo výstraha

<sup>6</sup>Obnoviť zviera/vrh bude môcť iba administrátor aplikácie

2.2.1.2. V niektorých prípadoch bude taktiež zobrazená registrácia vrhu v ktorom sa zviera narodilo, ktorej obsah je definovaný v 2.2.1.4. Druhá strana súboru bude vyplnená rodokmeňom zvieraťa (2.2.1.5).

Preukaz bude možné vygenerovať tlačidlom na stránke konkrétneho zvieraťa.

### 2.2.1.9 Filtrovanie a radenie

Pre vylepšenie používateľskej skúsenosti s aplikáciou bude potrebné implementovať filtrovanie a radenie zvierat a ako aj vrhov na príslušných stránkach so zoznamom zvierat, resp. verhov. Implementácia tejto funkcionality umožní jednoduchšiu prácu s aplikáciou a rýchlejšie nájdenie potrebných informácií.

### 2.2.1.10 Správa používateľov a rolí

Pre administrátorov aplikácie je nutné spravovať jednotlivých používateľov aplikácie, priradovať im príslušné role, alebo im ich naopak odobrať. Na základe tejto skutočnosti je žiaduce vytvoriť pohľad so zoznamom používateľov a ich rolami, s následnou možnosťou im danú rolu zmeniť, prípadne daných používateľov odobrať.

### 2.2.1.11 Lokalizácia aplikácie

Nakoľko budú aplikáciu používať nie len českí ale aj zahraniční používatelia, je nutné, aby aplikácia poskytovala obsah lokalizovaný do anglického jazyka. Jazyk bude možné jednoducho zmeniť v menu na paneli webovej aplikácie.

## 2.2.2 Nefunkčné požiadavky

Na rozdiel od funkčných požiadaviek, nefunkčné požiadavky umožňujú určiť obmedzenia kladené na aplikáciu. V neposlednom rade majú taktiež zásadný dopad na návrh architektúry webovej aplikácie [1].

### 2.2.2.1 Webová aplikácia

Nakoľko je požadovaný systém navrhnutý ako webová aplikácia, bude potrebné, aby bola prístupná z internetu pomocou moderných webových prehliadačov.

### 2.2.2.2 Používateľské rozhranie

Webová aplikácia bude musieť obsahovať používateľské rozhranie, s ktorým budú môcť používatelia interagovať. Prostredie bude navyše responzívne, čo uľahčí prípadný prístup do systému z mobilného prehliadača.

### 2.2.2.3 Technológie

Po konzultácii s vedúcim práce boli vymedzené nasledovné technológie, ktoré budú použité na strane servera.

Ako programovací jazyk bude použitý jazyk PHP, ktorý bol navrhnutý v roku 1994 Rasmusom Lerdorfom. Tento jazyk je univerzálny programovací jazyk na strane servera a je primárne určený na vývoj webových stránok [2].

Použitý jazyk bude vo verzii 7.3, hoci momentálne je najnovšia verzia jazyku 7.4 [3]. Dôvod výberu nižšej verzie jazyka je daný PHP podporou webhostingu<sup>7</sup>, na ktorom bude aplikácia nasadená.

Pre pridávanie, spracovanie a získavanie dát nám bude slúžiť open-source SQL databázový systém nazývaný MySQL, ktorý je vyvíjaný, distribuovaný a podporovaný spoločnosťou Oracle Corporation [4]. Tento databázový systém budeme používať vo verzii 5.6.

Táto verzia MySQL taktiež nie je najnovšia (v skutočnosti bola prvýkrát vydaná v roku 2013, avšak je stále oficiálne podporovaná [5]), opäť z dôvodu neexistujúcej podpory novšieho databázového systému webhostingom.

## 2.3 Používateľské role

Novovznikajúca webová aplikácia bude prístupná iba zaregistrovaným a prihláseným používateľom. Navyše, niektoré akcie budú obmedzené iba pre určitý okruh používateľov.

K tomu, aby sme umožnili prístup k vybraným akciám iba vybraným používateľom, bude potrebné do aplikácie implementovať používateľské role a práva. Následne bude aplikácia riadiť prístup používateľa k jednotlivým akciám na základe príslušnosti k vybranej roli.

V nasledujúcich podsekciach budú priblížené jednotlivé role a k nim príslušné práva, ktoré sa budú vyskytovať v aplikácii.

### 2.3.1 Bežný používateľ

Túto rolu bude mať každý používateľ automaticky po registrácii do webovej aplikácie. Bežný používateľ bude môcť v aplikácii:

- Vytvoriť a zobraziť svoje zvieratá
- Upraviť svoje zvieratá v prípade, že nie sú zaregistrované pod klubom ČKP
- Vytvoriť a upraviť registrácie svojich vlastných zvierat, ktoré nespadajú pod klub ČKP

---

<sup>7</sup>Server poskytujúci webovú aplikáciu používateľom na internete, v tomto prípade sa jedná o webhosting Endora – [www.endora.cz](http://www.endora.cz)



- Zobrazíť zoznam všetkých cudzích zvierat spolu s ich detailmi
- Vytvoriť a zobrazíť vrhy, u ktorých je používateľ ich majiteľom
- Upraviť vrhy, ktorých je majiteľom, pokiaľ neboli tieto vrhy schválené žiadosťou
- Zobrazíť všetky vrhy typu VP alebo schválené vrhy typu PP a NV spolu s ich detailmi
- Pridať poznámku k zvieratám a vrhom, ktoré vlastní

### 2.3.2 Registrátor zvierat

Registrátor zvierat je v hierarchii rolí postavený nad bežným používateľom. Tým pádom má všetky práva bežného používateľa a navyše nasledujúce práva:

- Možnosť pridať poznámku k akýmkoľvek zvieratám
- Možnosť pridať, editovať a vymazať akúkoľvek registráciu u každého zvierata

### 2.3.3 Schvalovateľ vrhov

Schvalovateľ vrhov je taktiež postavený v hierarchii rolí nad bežným používateľom podobne ako registrátor zvierat, s tým rozdielom, že schvalovateľ vrhu môže v aplikácii:

- Vidieť a odpovedať na žiadosti o schválenie vrhov
- Editovať akékoľvek zviera a vrh
- Pridať poznámky k akémukoľvek vrhu
- Vygenerovať preukaz zvierata

### 2.3.4 Administrátor

Ako obvykle, pre administrátora neplatia žiadne reštrikcie, čo znamená, že bude mať prístup ku všetkej funkcionalite definovanej vo funkčných požiadavkách v sekcii 2.2.1.

### 2.4 Prípady použitia

Prípady použitia sú špecifikácie rôznych činností, ktoré môžu používatelia s aplikáciou vykonávať [6]. Tieto prípady použitia budú zachytené vo forme scenáru a budú vychádzať nielen zo známych funkčných požiadaviek popísaných v sekcii 2.2.1, ale aj z jednotlivých používateľských rolí, ktoré boli definované v sekcii 2.3.

V tejto práci som sa rozhodol venovať iba takým prípadom použitia, ktoré sú z môjho pohľadu pre čitateľa prínosnejšie. Tie som následne rozdelil podľa príslušnosti k jednotlivým prvkom aplikácie. Všetky nasledujúce prípady použitia predpokladajú, že používateľ bude v systéme prihlásený.

Poznámka: Prvé štyri scenáre použitia sú aplikovateľné aj na vrhy, avšak z dôvodu neuvádzania duplicitných informácií nebudú ďalej rozvinuté v príslušnej podsekcii. Zároveň pri následnej aplikácii prvých štyroch scenárov na vrhy, sa výskyt slova „zvíra“ automaticky nahrádza slovom „vrh“ v príslušnej podobe.

#### 2.4.1 Prípady použitia zvierat

V tejto podsekcii budú popísané jednotlivé prípady použitia, ktoré sa týkajú zvierat.

##### UC1 – Vyhľadanie zvieraťa

Vyhľadanie zvieraťa je jeden zo základných prípadov použitia, kedy používateľ chce vyhľadať dané zviera.

##### Scenár:

1. Používateľ zvolí možnosť „Zvieratá“ z navigačného menu aplikácie.
2. Aplikácia následne zobrazí tabuľku so zvieratami a filtrom, na základe ktorého si používateľ môže nastaviť dodatočné kritéria filtrovania zvierat.
3. Používateľ nastaví filter podľa kritérií hľadaného zvieraťa a klikne na tlačidlo „Filtrovať“.
4. Aplikácia zobrazí všetky zvieratá vyhovujúce zadaným kritériám.

##### Alternatívny scenár:

3. Aplikácia zobrazí hľadané zviera už po načítaní tabuľky so zvieratami. V tomto prípade scenár vyhľadávania zvieraťa končí.

Tento prípad použitia realizuje viaceré funkčné požiadavky na aplikáciu, konkrétne evidenciu a filtrovanie a radenie zvierat.

### UC2 – Zobrazenie detailu zvierťa

Tento prípad použitia popisuje zobrazenie detailu zvierťa a zahŕňa prípad UC1 – Vyhľadanie zvierťa.

#### Scenár:

1. UC1 – Vyhľadanie zvierťa
2. Používateľ klikne na meno zvierťa, ktorého detail si želá vidieť.
3. Aplikácia zobrazí detail zvierťa.

Popísaný prípad taktiež realizuje funkčnú požiadavku na aplikáciu, a to evidenciu zvierat.

### UC3 – Vytvorenie zvierťa

Medzi ďalší základný prípad použitia patrí pridanie zvierťa používateľom do systému. Pri vytváraní zvierťa sa aplikuje dodatočné obmedzenie pre bežného používateľa, ktoré spočíva v nemožnosti zvolenia majiteľa iného ako je sám používateľ (2.3.1).

#### Scenár:

1. Používateľ zvolí možnosť „Zvieratá“ z navigačného menu aplikácie, ktorá následne načíta stránku so zoznamom uložených zvierat.
2. Následne používateľ klikne na tlačidlo „Pridať zviera“.
3. Aplikácia načíta novú stránku s formulárom pre vytvorenie nového zvierťa.
4. Používateľ vyplní formulár.
5. Po vyplnení všetkých potrebných údajov pre vytvorenie zvierťa aplikácia umožní odoslať formulár kliknutím na tlačidlo „Uložiť“.
6. Používateľ klikne na tlačidlo „Uložiť“.
7. Po úspešnom vytvorení zvierťa na základe vložených údajov systém presmeruje používateľa na stránku so zoznamom uložených zvierat.

Tento prípad použitia realizuje funkčnú požiadavku evidencie zvierat.

### UC4 – Úprava zvieráťa

V tomto prípade sa používateľ aplikácie snaží upraviť existujúce zviera v systéme. Tento prípad použitia taktiež zahŕňa prípady UC1 – Vyhľadanie zvieráťa alebo UC2 – Zobrazenie detailu zvieráťa, v závislosti od spôsobu úpravy zvieráťa, ktorý si používateľ zvolí.

#### Scenár:

1. UC1 – Vyhľadanie zvieráťa
2. Používateľ klikne na rozbalovacie menu u zvieráťa, ktoré chce editovať.
3. Po tomto kroku aplikácia ponúkne možnosť editácie alebo zmazania zvieráťa.
4. Používateľ zvolí možnosť editácie.
5. Aplikácia následne presmeruje používateľa na stránku s editáciou zvieráťa, ktorá bude obsahovať formulár s predvyplnenými údajmi zvieráťa.
6. Používateľ upraví údaje zvieráťa.
7. Kliknutím na tlačidlo „Odoslať“ sa odošle vyplnený formulár na server.
8. Po úspešnom spracovaní formuláru je používateľ presmerovaný na stránku s detailmi upravovaného zvieráťa.

#### Alternatívny scenár:

1. UC2 – Zobrazenie detailu zvieráťa
2. Používateľ klikne na tlačidlo „Upraviť“ na pravom postrannom paneli.
3. Následne scenár pokračuje bodom 5 v pôvodnom scenári.

Popísaný prípad použitia realizuje funkčnú požiadavku evidencie zvieráťa.

### UC5 – Vytvorenie registrácie zvieráťa

Tento prípad použitia popisuje vytvorenie registrácie zvieráťa. Pri samotnom vytváraní novej registrácie sa aplikujú dodatočné obmedzenia pre bežného používateľa, tak ako je popísané v 2.3.1.

Tento prípad použitia zahŕňa prípad zobrazenia detailov zvieráťa.

### Scenár:

1. UC2 – Zobrazenie detailu zvieraťa
2. Kliknutím na tlačidlo „Pridať novú registráciu“ aplikácia zobrazí registračný formulár v modálnom okne<sup>8</sup>.
3. Používateľ vyplní potrebné polia.
4. Po stlačení tlačidla „Uložiť“ sa vyplnený formulár odošle na server.
5. Po úspešnom spracovaní formuláru serverom sa modálne okno zavrie.

Tento prípad použitia realizuje funkčnú požiadavku kladenú na aplikáciu, konkrétne evidenciu registrácií zvieraťa.

### 2.4.2 Prípady použitia vrhov

Táto podsekcia popisuje jednotlivé prípady použitia, ktoré sa týkajú vrhov.

#### UC6 – Vytvorenie žiadosti o schválenie vrhu

Tento prípad použitia reálne zahŕňa prípad použitia „zobrazenia detailov vrhu“, avšak pre neuvádzanie duplicitných informácií môže čitateľ vychádzať z prípadu použitia zobrazenia detailov zvieraťa. Scenáre sa v oboch prípadoch obsahovo nelíšia, tak ako bolo uvedené poznámke v sekcii 2.4.

Na tento prípad sa taktiež vzťahujú dodatočné obmedzenia pre bežného používateľa vyplývajúce z 2.3.1.

### Scenár:

1. UC2 – Zobrazenie detailu zvieraťa
2. Používateľ klikne na tlačidlo „Vytvoriť žiadosť“.
3. Aplikácia následne zobrazí modálne okno s formulárom obsahujúcim textové pole slúžiace pre poznámku registrátorovi.
4. Po stlačení tlačidla „Uložiť“ sa formulár odošle na server.
5. Po úspešnom spracovaní formuláru serverom sa vytvorí nová žiadosť o schválenie vrhu.
6. Modálne okno sa automaticky zavrie.

Popísaný prípad použitia realizuje funkčnú požiadavku správy žiadostí o schválenie vrhu.

---

<sup>8</sup>Menšie okno aplikácie prekrývajúcce jej hlavný obsah

### UC7 – Odpoveď na žiadosť o schválenie vrhu

Tak ako v predchádzajúcom prípade, tento prípad použitia bude zahŕňať „zobrazenie detailov vrhu“.

Pre tento prípad použitia sa vzťahujú dodatočné obmedzenia – na žiadosť o schválenie vrhu bude môcť odpovedať používateľ majúci rolu schvalovateľa žiadostí vrhov alebo administrátora.

#### Scenár:

1. UC2 – Zobrazenie detailu zvieraťa
2. Používateľ stlačí tlačidlo „Odpovedať“ pri príslušnej žiadosti o schválenie vrhu.
3. Aplikácia zobrazí modálne okno s formulárom pre schválenie vrhu.
4. Používateľ vyplní formulár a kliknutím na tlačidlo „Uložiť“ sa odošle na server.
5. Server spracuje odoslaný formulár a ihneď odošle e-mail žiadateľovi o stave jeho žiadosti.

#### Alternatívny scenár:

1. Používateľ klikne na link v e-maile, ktorý mu prišiel po odoslaní novej žiadosti o schválenie vrhu žiadateľom.
2. Ako reakcia na kliknutie na link, operačný systém otvorí predvolený webový prehliadač s adresou vedúcou na detail vrhu.
3. Následne scenár pokračuje bodom 2 v pôvodnom scenári.

Tak ako v predchádzajúcom prípade použitia, aj v tomto prípade použitia je realizovaná funkčná požiadavka správy žiadostí o schválenie vrhu.

## Návrh

Táto kapitola sa venuje návrhu architektúry budúcej aplikácie, s priblížením dvoch hlavných architektúr používaných pri tvorbe webových aplikácií. Následne sú porovnané a podľa výsledkov porovnania je vybraná taká, ktorá bude použitá pri vývoji aplikácie. V nasledujúcej sekcii sú vymedzené použité technológie budúcou aplikáciou. V neposlednom rade táto kapitola obsahuje návrh modelových tried a končí návrhom REST API, ktoré bude aplikácia používať pri komunikácii so serverom.

### 3.1 Návrh architektúry

Dôležitý aspekt pri návrhu aplikácie spočíva vo výbere architektúry, na ktorej bude aplikácia postavená. Na základe jej výberu sa odvíjajú technológie, ktorými bude daná aplikácia disponovať.

V nasledujúcich sekciách sú zdokumentované dva typy architektúr, z ktorých bude na základe porovnania a vlastných skúseností s tvorbou webových aplikácií vybraná jedna, ktorá bude definovať podobu budúcej webovej aplikácie.

#### 3.1.1 Architektúra MPA aplikácie

MPA je skratka pre viac-stránkovú webovú aplikáciu. Takáto aplikácia vždy načíta celú stránku a zobrazí novú v prípade interakcie s aplikáciou – zvyčajne po odoslaní formuláru používateľom [7].

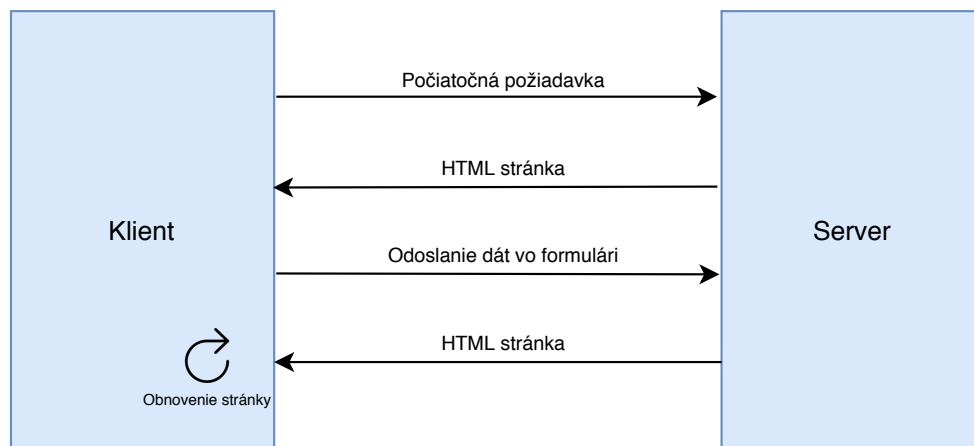
#### Priebeh komunikácie medzi serverom a klientom

Celkový priebeh komunikácie medzi klientom a serverom zachytáva nasledujúci diagram s jeho popisom.

1. Klient požiada server o stránku

### 3. NÁVRH

---



Obr. 3.1: Diagram znázorňujúci architektúru MPA aplikácií

2. Server vráti klientovi požadovanú stránku
3. Klient vrátenú stránku serverom vykreslí
4. Klient vyplní formulár, ktorý sa následne odošle na server
5. Server prijme údaje od klienta, ktoré následne spracuje
6. Po spracovaní údajov klientovi vráti späť stránku s aktualizovanými údajmi
7. Klient túto stránku načíta, čím príde k obnoveniu stránky

Technológia AJAX čiastočne rieši problém znovunačítavania celej stránky dynamickým aktualizovaním tých častí aplikácie, ktoré boli používateľom zmenené. Avšak zakomponovanie tejto technológie do MPA sťažuje a komplikuje celý vývojový proces aplikácie [8].

#### Výhody použitia MPA architektúry

- Jednoduchá optimalizácia stránok pre webové vyhľadávače [7]
- Umožňuje jednoduchú správu informácií na jednotlivých stránkach [9]
- Je potrebný menší rozsah nástrojov a znalostí narozdiel od vývoja SPA aplikácie [10]
- Veľa dostupných riešení pre vývoj MPA aplikácií [10]



### Nevýhody použitia MPA architektúry

- Zvýšený čas načítavania stránok kvôli neustálemu obnovovaniu stránok (neplatí pri použití technológie AJAX) [7]
- Znížený výkon aplikácie, kvôli neustálemu načítavaniu väčšieho množstva informácií naraz pred následným poslaním stránky používateľovi [7]
- Úzka previazanosť vývoja aplikácie na strane servera a klienta, čo znemožňuje prípadné neskoršie nasadenie rôznych technológií [10]

### 3.1.2 Architektúra SPA aplikácie

SPA je skratka pre single-page aplikáciu. Single-page aplikácia je aplikácia, ktorá nepotrebuje znovunačítanie stránky počas jej používania. Na tomto type architektúry sú postavené aplikácie ako Gmail<sup>9</sup>, Google Mapy<sup>10</sup>, Facebook<sup>11</sup> či GitHub<sup>12</sup>. V tomto prípade sa o aktualizáciu obsahu na stránke nestará server, ale klient, zväčša pomocou frameworku<sup>13</sup> bežiacom v prostredí webového prehliadača [9].

### Priebeh komunikácie medzi serverom a klientom

Celkový priebeh komunikácie medzi klientom a serverom zachytáva nasledujúci diagram s jeho popisom.

1. Klient požiada server o stránku
2. Server vráti klientovi požadovanú HTML stránku so všetkým obsahom aplikácie
3. Klient vrátenú stránku serverom vykreslí
4. Klient vyplní formulár, ktorý sa následne odošle na server pomocou AJAXu
5. Server prijme údaje od klienta, ktoré následne spracuje
6. Po spracovaní údajov server vráti späť klientovi aktualizované údaje (v JSON<sup>14</sup> formáte)
7. Klient zahodí staré údaje na stránke a nahradí ich novými, čím príde k obnoveniu údajov ale nie stránky

---

<sup>9</sup><https://gmail.com>

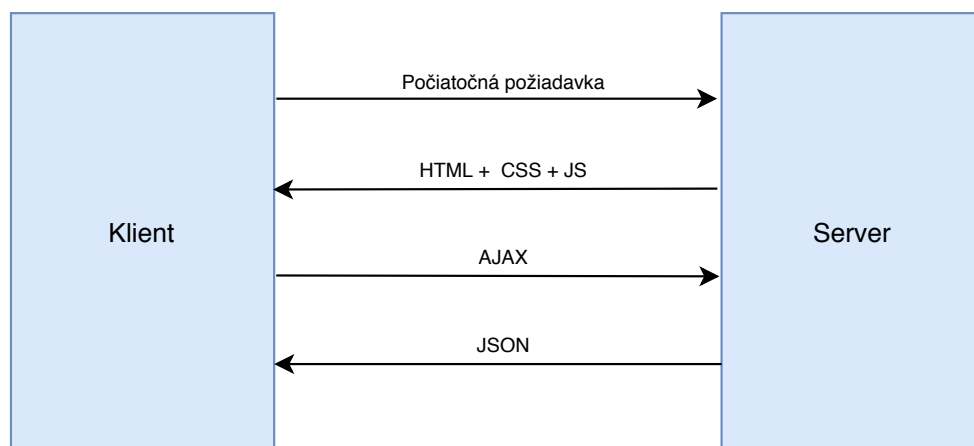
<sup>10</sup><https://maps.google.com>

<sup>11</sup><https://facebook.com>

<sup>12</sup><https://github.com>

<sup>13</sup>Sada podporných programov uľahčujúci vývoj aplikácie

<sup>14</sup>Formát na výmenu dát medzi klientom a serverom [11]



Obr. 3.2: Diagram znázorňujúci architektúru SPA aplikácií

#### Výhody použitia SPA architektúry

- Rýchlosť a responzivnosť aplikácie založenej na aktualizovaní iba tej časti aplikácie, ktorá sa zmenila [7]
- Minimálna previazanosť kódu aplikácie na strane servera a klienta, čo umožňuje jednoduchú správu používaných knižníc bez ovplyvnenia ďalších častí aplikácie [10]
- Zjednodušenie proces vývoja, nakoľko sa o vykreslenie obsahu nestará server ale klientský framework [9]

#### Nevýhody použitia SPA architektúry

- Ťažká optimalizácia aplikácie pre webové vyhľadávače, nakoľko o vykreslenie obsahu sa stará Javascript<sup>15</sup>, ktorý webové vyhľadávače neinterpretujú [7]
- Prvé načítanie aplikácie trvá dlhší čas v porovnaní s MPA architektúrou, pretože pri prvom navštívení aplikácie sa musí všetok obsah aplikácie stiahnuť do zariadenia
- Nemožnosť zobrazenia aplikácie pri vypnutom Javascripte vo webovom prehliadači [10]
- Pri komplexnejšej aplikácii bude aplikácia zaberať viac pamäte zariadenia, z ktorého je zobrazovaná, nakoľko všetok obsah aplikácie je stiahnutý v zariadení [9]

---

<sup>15</sup>Programovací jazyk bežiaci vo webovom prehliadači

**3.1.3 Porovnanie a voľba architektúry**

**3.2 Návrh technológií**

**3.2.1 Back-end technológie**

**3.2.2 Front-end technológie**

**3.3 Návrh modelových tried**

**3.4 Návrh REST API**



---

## Literatúra

- [1] Mlejnek, J.: *Analýza a sběr požadavků*. [online prezentácia][cit. 3. 5. 2020]. Dostupné z: [https://moodle-vyuka.cvut.cz/pluginfile.php/161920/mod\\_resource/content/3/03.prednaska.pdf](https://moodle-vyuka.cvut.cz/pluginfile.php/161920/mod_resource/content/3/03.prednaska.pdf)
- [2] Pros and Cons of PHP Programming Language. [webstránka][cit. 4. 5. 2020] (vlastný preklad). Dostupné z: <https://www.pros-cons.net/pros-and-cons-of-php-programming-language/>
- [3] PHP: Releases. [webstránka][cit. 4. 5. 2020]. Dostupné z: <https://www.php.net/releases/index.php>
- [4] What is MySQL? [webstránka][cit. 4. 5. 2020] (vlastný preklad). Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [5] Mysql :: mysql 5.6 release notes 2020. [webstránka][cit. 4. 5. 2020]. Dostupné z: <https://dev.mysql.com/doc/relnotes/mysql/5.6/en/>
- [6] Arlow, J.; Neustadt, I.; Kiszka, B.: *UML 2 a unifikovaný proces vývoje aplikací*. Computer Press, 2007, 95 s., [webstránka][cit. 4. 5. 2020].
- [7] Which and when to use: SPA vs MPA - JAXenter. Dostupné z: <https://jaxenter.com/spa-vs-mpa-160963.html>
- [8] Quant, M.: Single Page Application vs. Multi-page Application. 2020, [webstránka][cit. 5. 5. 2020](vlastný preklad). Dostupné z: <https://dev.to/mquant/single-page-application-vs-multi-page-application-p2m>
- [9] Kukhnavets, P.: SPA vs MPA: Key Difference Between Single-Page and Multi-Page Apps. 2020, [webstránka][cit. 5. 5. 2020](vlastný preklad). Dostupné z: <https://cuspy.io/blog/spa-vs-mpa-what-is-the-difference/>

## LITERATÚRA

---

- [10] Melnik, I.: Single Page Application (SPA) vs Multi Page Application (MPA): Pros and Cons. 2020, [webstránka][cit. 5. 5. 2020](vlastný preklad). Dostupné z: <https://merehead.com/blog/single-page-application-vs-multi-page-application/>
- [11] JSON. 2020, [webstránka][cit. 5. 5. 2020](vlastný preklad). Dostupné z: <https://www.json.org/json-en.html>