

**FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO**

**Závěrečná správa**

VYSKLADAJ SI AVATARA

zimný semester 2015/2016

Michal Piják

Károly Belokostolský

Michal Rakovský

# Obsah

1	Úvod .....	5
2	Popis projektu .....	5
2.1	Predmet špecifikácie .....	5
2.2	Rozsah projektu .....	5
2.3	Kontext aplikácie .....	5
2.3.1	Systémové rozhrania .....	6
2.3.2	Používateľské rozhrania.....	6
2.3.3	Hardvérové rozhrania.....	6
2.3.4	Softvérové rozhrania .....	6
2.4	Triedy používateľov a ich vlastnosti .....	6
2.5	Funkcie aplikácie .....	6
3	Katalóg požiadaviek.....	7
3.1	Slovník pojmov, skratky.....	7
3.2	Používateľská časť .....	7
3.2.1	Načítaj tému .....	7
3.2.2	Vyskladaj avatara.....	8
3.2.3	Ulož avatara.....	10
3.2.4	Načítaj avatara.....	11
3.3	Administrátorská časť.....	12
3.3.1	Vlož tému.....	12
3.4	Výkonnostné požiadavky.....	14
3.5	Dostupnosť .....	14
3.6	Bezpečnostné požiadavky .....	14
4	Analýza používateľov.....	15
5	Diagramy .....	16
5.1	Use-case diagram pre jednotlivých používateľov.....	16
5.2	Stavový diagram .....	17
5.3	Entitno-relačný diagram .....	18
6	Používateľské rozhranie .....	19
6.1	Rozhranie pre používateľov (deti) .....	19
6.1.1	Okienko na výber témy.....	19
6.1.2	Obrazovka s plochou na vyskladanie avatara a objektmi.....	20

6.2	Rozhranie pre správcu aplikácie .....	20
6.2.1	Okienko so zoznamom tém .....	21
6.2.2	Okienko na pridanie témy .....	22
7	Analýza technológií .....	24
7.1	Výber programovacích jazykov pre serverom vykonávané skripty .....	24
7.2	Výber programovacieho jazyka pre klientom vykonávané skripty .....	24
7.3	Výber frameworkov .....	24
7.3.1	Framework pre backend .....	24
7.3.2	Framework pre frontend .....	24
7.4	Výber pomocných knižníc a pluginov .....	25
7.4.1	Pre frontend .....	25
7.4.2	Pre backend .....	25
8	Dátový model .....	26
9	Komponenty aplikácie .....	27
9.1	Schéma zapojenia komponentov .....	27
9.2	Databázový uzol .....	28
9.3	Frontendový uzol .....	29
9.4	Backendový uzol .....	30
10	Návrh .....	31
10.1	Úvod k návrhu .....	31
10.2	Návrh objektov .....	31
10.2.1	Návrh tabuľkových modelov .....	31
10.2.2	Návrh funkcií plochy na skladanie avatara .....	32
10.3	Stavový diagram .....	33
10.4	Sekvenčný diagram .....	34
11	Testovanie .....	35
11.1	Testovacie scenáre – User interface .....	35
11.2	Testovacie scenáre – admin interface .....	36
12	Záznam z odovzdávania a predvedenia výslednej aplikácie zadávateľovi .....	37
12.1	Plán stretnutia .....	37
12.2	Priebeh stretnutia .....	37
13	Zhodnotenie .....	38
13.1	Spokojnosť s výsledným dielom, ťažkosti počas vývoja .....	38

13.2	Zmeny do ďalších verzií .....	38
13.3	Odlišnosti od pôvodného plánu .....	38
13.4	Tímová práca, rozdelenie úloh a komunikácia .....	38
13.5	Záver .....	39
14	Zdroje.....	40

# 1 Úvod

Táto záverečná správa slúži ako finálny výstup nášho projektu na predmet Tvorba informačných systémov. Sú tu zahrnuté rôzne analýzy projektu vytvorené počas semestra, dokumentácie funkčnosti, funkcií, kódu a výsledky testovania a odovzdania projektu, tak ako aj zoznam nedostatkov.

## 2 Popis projektu

### 2.1 *Predmet špecifikácie*

Táto špecifikácia požiadaviek (ďalej ŠP) popisuje používateľské a funkčné požiadavky aplikácie „Vyskladaj si avatara“. ŠP je určená pre tím, ktorý bude výslednú aplikáciu implementovať. Špecifikácia je súčasťou zmluvy medzi objednávateľom a dodávateľom. Bude slúžiť ako východisko pre vyhodnocovanie správnosti softvéru.

### 2.2 *Rozsah projektu*

Aplikácia vyskladaj si avatara, bude vo svojej prvej verzii obsahovať funkcie týkajúce sa hlavne skladania avatara z objektov. Úlohou tejto verzie je vytvoriť základ pre ďalšie verzie systému, v ktorých bude možné spojiť sa s databázou a následne importovať alebo exportovať objekty. Celá aplikácia bude vyvíjaná ako doplnok už do existujúcej aplikácie, ktorá je reálne nasadená v produkcii.

### 2.3 *Kontext aplikácie*

Aplikácia „Vyskladaj si avatara“ bude vyvíjaná ako doplnok do existujúceho projektu, ktorý sa v súčasnosti používa na základnej škole pri výučbe detí. S aplikáciou bude pracovať administrátor, ktorý bude môcť pridávať nové témy a deti, ktoré si budú vedieť z objektov vyskladať avatara. Takto vyskladaný avatar sa potom bude zobrazovať vedľa ich mena v profile. Celá aplikácia bude komunikovať s databázou, v ktorej budú uložené témy, objekty a vyskladaní avatari.

### 2.3.1 Systémové rozhrania

SR-1	Rozhranie na komunikáciu s DB
SR-1.1	Aplikácia z DB načíta uložené objekty a témy
SR-1.2	Aplikácia uloží vyskladané dáta do databázy
SR-2	Rozhranie na ukladanie objektov
SR-2.1	Zápis objektov vo formáte json
SR-2.2	Vytvorenie obrázka vo formáte png ktorý sa bude zobrazovať v profile

### 2.3.2 Používateľské rozhrania

PR-1	Používateľské rozhranie musí byť formou webovej aplikácie
PR-2	Používateľské rozhranie musí byť zladené vo farbách súčasného projektu.

### 2.3.3 Hardvérové rozhrania

Systém neobsahuje žiadne hardvérové rozhrania.

### 2.3.4 Softvérové rozhrania

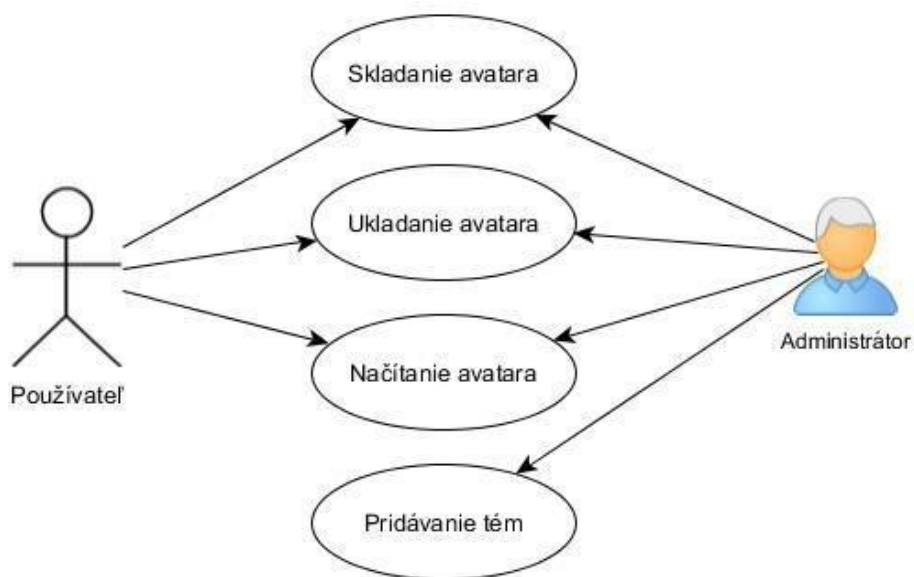
SSR-1	Aplikácia by mala byť spustiteľná na linuxovom serveri s nainštalovaným NodeJS.
SSR-2	Na serveri musí byť nainštalovaná databáza MariaDB.

## 2.4 Triedy používateľov a ich vlastnosti

Trieda	Popis
Používateľ (dieťa)	Je základný typ používateľa, ktorý môže skladať a ukladať avatara.
Administrátor	Je špeciálny používateľ, ktorý môže pridávať témy.

## 2.5 Funkcie aplikácie

Prehľad funkcií, ktoré systém poskytuje je znázornený diagramom prípadov použitia na Obrázku 1.



Obrázok 1: diagram prípadov použitia

## 3 Katalóg požiadaviek

### 3.1 Slovník pojmov, skratky

Avatar	Postavička ktorá sa bude dať vyskladať z objektov.
DB	Databáza
ŠP	Špecifikácia požiadaviek
Importovať	Uložiť dáta databázy
Exportovať	Získať dáta z databázy

### 3.2 Používateľská časť

#### 3.2.1 Načítaj tému

##### 3.2.1.1 Popis prípadu použitia

Označenie:	UC01	Názov:	Načítaj tému	Priorita:	vysoká
Popis:	Používateľ načíta tému z DB tém, následne bude môcť vybrať prvky témy pre vyskladanie Avatara na grafickej ploche				
Vstupné požiadavky	Formát témy musí byť kompatibilný s aplikačnou časťou programu				
Výstupné požiadavky:	-				

Opakovanosť	neobmedzené	
	Krok	Činnosť
Bežná cesta:	0.1	Používateľ si načíta dostupné témy
	0.2	Systém zobrazí dostupné témy
	0.3	Systém čaká na výber témy
	0.4	Zákazník označí vybranú tému
	0.5	Systém tému pridá do panela tém
Alternatívna cesta:	1.0	Po kroku 0.5 si môže používateľ odomknúť kombinovanie viacerých tém (Avatar môže byť výsledkom spojenia viacerých tém)
Výnimky:	V1	Používateľ sa pokúša potvrdiť výber pridanej témy
	V1.1	Systém zobrazí varovanie o výbere rovnakej témy
	V1.2a	Používateľ výber ukončí. Prípád použitia končí
	V1.2b	Používateľ si vyberie tému. Prípád použitia pokračuje bodom 0.5

### 3.2.1.2 Funkčné požiadavky

Označenie	Popis
načítanieTémy.téma.výberTémy	Kontrola, či používateľ vybral jednu z tém a nepokúša sa potvrdiť neoznačený výber
načítanieTémy.téma.kontrolaDuplicity	Ak používateľ vyberá tému ktorú už má načítanú, systém ho na to upozorní a tému nepovolí načítať znovu

### 3.2.2 Vyskladaj avatara

#### 3.2.2.1 Popis prípadu použitia

Označenie:	UC02	Názov:	Vyskladaj Avatara	Priorita:	vysoká
Popis:	Používateľ bude môcť skladať Avatara na grafickej ploche presunom objektov tém z panela ovládaním myšou.				



Vstupné požiadavky	Musí byť načítaná aspoň jedna téma	
Výstupné požiadavky:	-	
Opakovanosť	neobmedzené	
	Krok	Činnosť
Bežná cesta:	0.1	Používateľ prejde kurzorom na grafický objekt
	0.2	Kliknutím ľavého tlačítka myši sa aktivuje posun
	0.3	Ťahaním myši je možné objekt presúvať
	0.4	Pustením myši prebehne kontrola pozície pusteného objektu a následné pripevnenie objektu ku grafickej ploche ak vyhovuje kontrole
	0.5	Systém spracuje objekt do celku
Alternatívna cesta:	1.0	V prípade načítania viacerých tém si používateľ môže odomknúť ich kombinovanie pri tvorbe Avatara
Výnimky:	V1	Používateľ sa pokúša umiestniť objekt mimo grafickú plochu
	V1.1	Systém ignoruje posun objektom a objekt vráti do panela
	V1.2	Používateľ pokračuje ďalším ťahaním
	V2	Používateľ sa pokúša skladať Avatara z viacerých tém, pričom táto možnosť nie je povolená
	V2.1	Systém zobrazí upozornenie o skladaní porušujúcom pravidlá
	V2.2	Používateľ pokračuje krokom 0.1

### 3.2.2.2 Funkčné požiadavky

Označenie	Popis
skladanieAvatara.kritéria.správnePozície	Kontrola, či používateľ pustil objekt v rámci grafickej plochy

skladanieAvatara.kritéria.kombinovanieTém	Ak sa používateľ pokúša skladať Avatara prvkami z viacerých tém, systém skontroluje, či je táto možnosť povolená. Ak nie je, upozorní ho a používateľ môže pokračovať ďalším korektným postupom. Ak je povolené skladanie z viacerých tém, kontrola prebehne v poriadku
---	---

### 3.2.3 Ulož avatara

#### 3.2.3.1 Popis prípadu použitia

Označenie:	UC03	Názov:	Ulož Avatara	Priorita:	vysoká
Popis:	Používateľ bude môcť uložiť vyskladaného Avatara do databázy				
Vstupné požiadavky	Grafická plocha nemôže byť prázdna a musí obsahovať minimálny počet prvkov (určené minimum)				
Výstupné požiadavky:	-				
Opakovanosť	neobmedzené				
	Krok	Činnosť			
Bežná cesta:	0.1	Používateľ zvolí uloženie Avatara			
	0.2	Systém skontroluje či sú splnené všetky požiadavky			
	0.3	Systém uloží Avatara do databázy			
	0.4	Používateľ môže ďalej upravovať Avatara (ďalší postup nebude uložený automaticky), alebo resetnúť plochu a začať skladať nového Avatara			
Alternatívna cesta:	-	-			
Výnimky:	V1	Používateľ ukladá Avatara, pričom na grafickej ploche nie je minimálny počet prvkov			
	V1.1	Systém vyhlási chybu o nedodržaní minimálneho počtu prvkov a Avatara neuloží			
	V1.2a	Používateľ pokračuje skladaním. Prípád použitia pokračuje bodom 0.1			
	V1.2b	Používateľ zruší skladanie. Prípád použitia končí			

	V2	Používateľ ukladá Avatara, ktorý neobsahuje nutnú triedu objektu (ak je požadované telo / hlava / ...)
	V2.1	Systém vyhlási chybu o nesprávnom zložení Avatara a Avatara neuloží
	V2.1a	Používateľ pokračuje skladaním. Prípád použitia pokračuje bodom 0.1
	V2.1b	Používateľ zruší skladanie. Prípád použitia končí

### 3.2.3.2 Funkčné požiadavky

Označenie	Popis
ulozenieAvatara.kritéria.minimálnyPočetPrvkov	Kontrola, či grafická plocha obsahuje minimum prvkov potrebné pre zostavenie Avatara
SkladanieAvatara.kritéria.obsahPovinnýchTried	Kontrola, či Avatar obsahuje povinné triedy objektov (ak sú dané)

### 3.2.4 Načítaj avatara

#### 3.2.4.1 Popis prípadu použitia

Označenie:	UC04	Názov:	Načítaj Avatara	Priorita:	stredná
Popis:	Používateľ bude môcť načítať svojho Avatara z databázy				
Vstupné požiadavky	-				
Výstupné požiadavky:	-				
Opakovanosť	neobmedzené				
	Krok	Činnosť			
Bežná cesta:	0.1	Systém skontroluje či je grafická plocha prázdna			
	0.2	Systém vykreslí Avatara pomocou informácií o objektoch uložených v databáze			
Alternatívna cesta:	-	-			
Výnimky:	V1	Používateľ načítava Avatara do rozpracovanej plochy			

	V1.1	Systém upozorní používateľa, že sa všetok aktuálny obsah na grafickej ploche prekreslí a nebude možné ho vrátiť
	V1.2a	Používateľ nepotvrdí načítanie. Prípád použitia končí
	V1.2b	Používateľ potvrdí načítanie. Prípád použitia končí
	V2	Avatar nie je uložený v databáze
	V2.1	Systém vyhlási chybu o nenájdení Avatara v databáze
	V2.1a	Používateľ pokračuje skladaním. Prípád použitia končí
	V2.1b	Používateľ zruší skladanie. Prípád použitia končí

#### 3.2.4.2 Funkčné požiadavky

Označenie	Popis
ulozenieAvatara.kritéria.minimálnyPočetPrvkov	Kontrola, či grafická plocha obsahuje minimum prvkov potrebných pre zostavenie Avatara
SkladanieAvatara.kritéria.obsahPovinnýchTried	Kontrola, či Avatar obsahuje povinné triedy objektov (ak sú dané ... napr. hlava, telo, uši, ...)

### 3.3 Administrátorská časť

#### 3.3.1 Vlož tému

##### 3.3.1.1 Popis prípadu použitia

Označenie:	UC05	Názov:	Vlož tému	Priorita:	stredná
Popis:	Administrátor vloží tému do systému, táto téma bude potom prístupná používateľom				
Vstupné požiadavky	Formát témy musí byť kompatibilný s aplikačnou časťou programu				
Výstupné požiadavky:	-				
Opakovanosť	neobmedzené				
	Krok	Činnosť			

Bežná cesta:	0.1	Administrátor vyberie tému v prehliadači súborov
	0.2	Systém skontroluje kompletnosť a kompatibilitu témy
	0.3	Systém uloží súbory na disk
	0.4	Systém uloží informácie, konfiguráciu a cieľové cesty k súborom do databázy
Alternatívna cesta:	-	-
Výnimky:	V1	Administrátor sa pokúša načítať nekompatibilnú tému
	V1.1	Systém zobrazí upozornenie o nekompatibilite témy
	V1.2a	Administrátor pridávanie ukončí. Prípád použitia končí
	V1.2b	Administrátor vyberie inú tému. Prípád použitia pokračuje bodom 0.1

### 3.3.1.2 Funkčné požiadavky

Označenie	Popis
vloženieTémy.kritéria.vloženieTémy	Kontrola, či administrátor vybral zdroj témy. Ak áno, potrebné súbory sa uložia na disk a informácie do databázy. Ak nie, vyhlási chybu a pokračuje výberom.
vloženieTémy.kritéria.kontrolaDuplicity	Ak administrátor vkladá do systému tému, ktorá v ňom už je, systém ho na to upozorní
vloženieTémy.kritéria.kontrolaKompatibility	Systém skontroluje či téma vyhovuje aplikačnej vrstvy programu, teda formát je kompatibilný použitým knižniciam a podobne.
vloženieTémy.kritéria.kontrolaKompletnosti	Systém skontroluje, či téma je kompletne pridaná aj so zdrojovými súbormi na disk.

### **3.4 Výkonnostné požiadavky**

VP-1	Načítanie témy nesmie trvať dlhšie ako 5 sekúnd
VP-2	Pohyb grafických objektov musí byť plynulý
VP-2	Načítanie Avatara nesmie trvať dlhšie ako 20 sekúnd

### **3.5 Dostupnosť**

D-1	V pracovné dni musí byť dostupnosť 98.5%
D-2	Počas detských aktivít musí byť dostupnosť 99.99%
D-3	Počas víkendov a sviatkov musí byť dostupnosť 95%

### **3.6 Bezpečnostné požiadavky**

Treba zabezpečiť nasledovné typy útokov, aby sa predišlo k poškodeniu, spomaleniu systému, alebo strate údajov v databáze.

BP-1	Ochrana proti SQL Injection
BP-2	Ochrana proti XSS
BP-3	Ochrana proti DoS a DDoS

## 4 Analýza používateľov

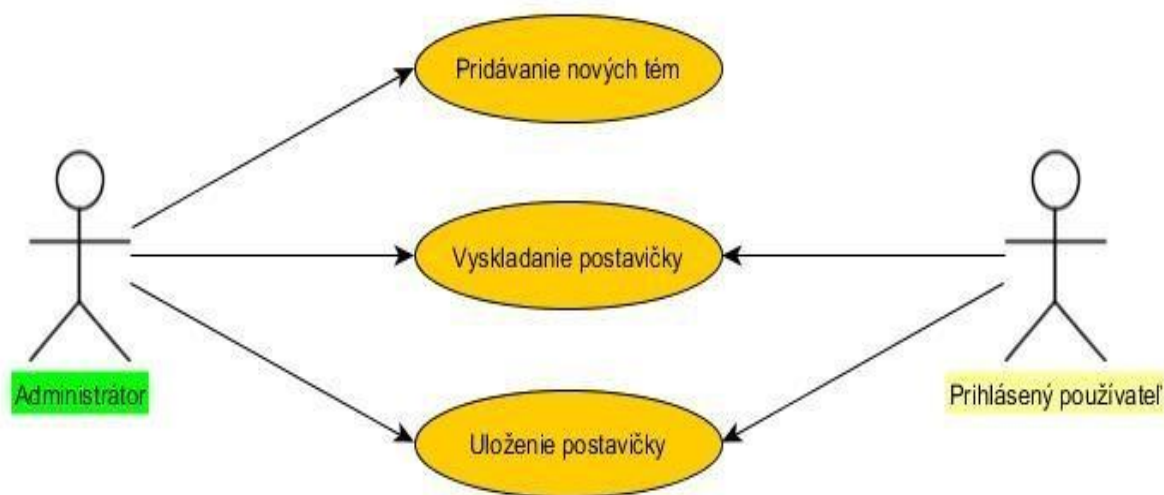
Na základe požiadaviek stanovených zadávateľom, sme stanovili nasledovných používateľov, ktorí budú pracovať s aplikáciou:

- Administrátor – bude sa prihlasovať do aplikácie cez rozhranie prístupné cez webový prehliadač. Toto rozhranie už je naprogramované, keďže aplikácia do ktorej bude tento model implementovaný už reálne funguje. Systém bude administrátorovi umožňovať pridávať nové témy avatarov, ich modifikáciu a odstraňovanie.
- Prihlásený používateľ – bude sa taktiež prihlasovať do aplikácie cez rozhranie prístupné cez webový prehliadač. Aplikácia mu bude umožňovať vybrať si jednu z prednastavených tém a následne si vyskladať postavičku (avatara). K takto vyskladanej postavičke sa bude môcť kedykoľvek vrátiť a upraviť.

## 5 Diagramy

### 5.1 Use-case diagram pre jednotlivých používateľov

Nasledujúci use-case diagram zobrazuje množinu činností, ktoré môže vykonávať administrátor systému a prihlásený používateľ.

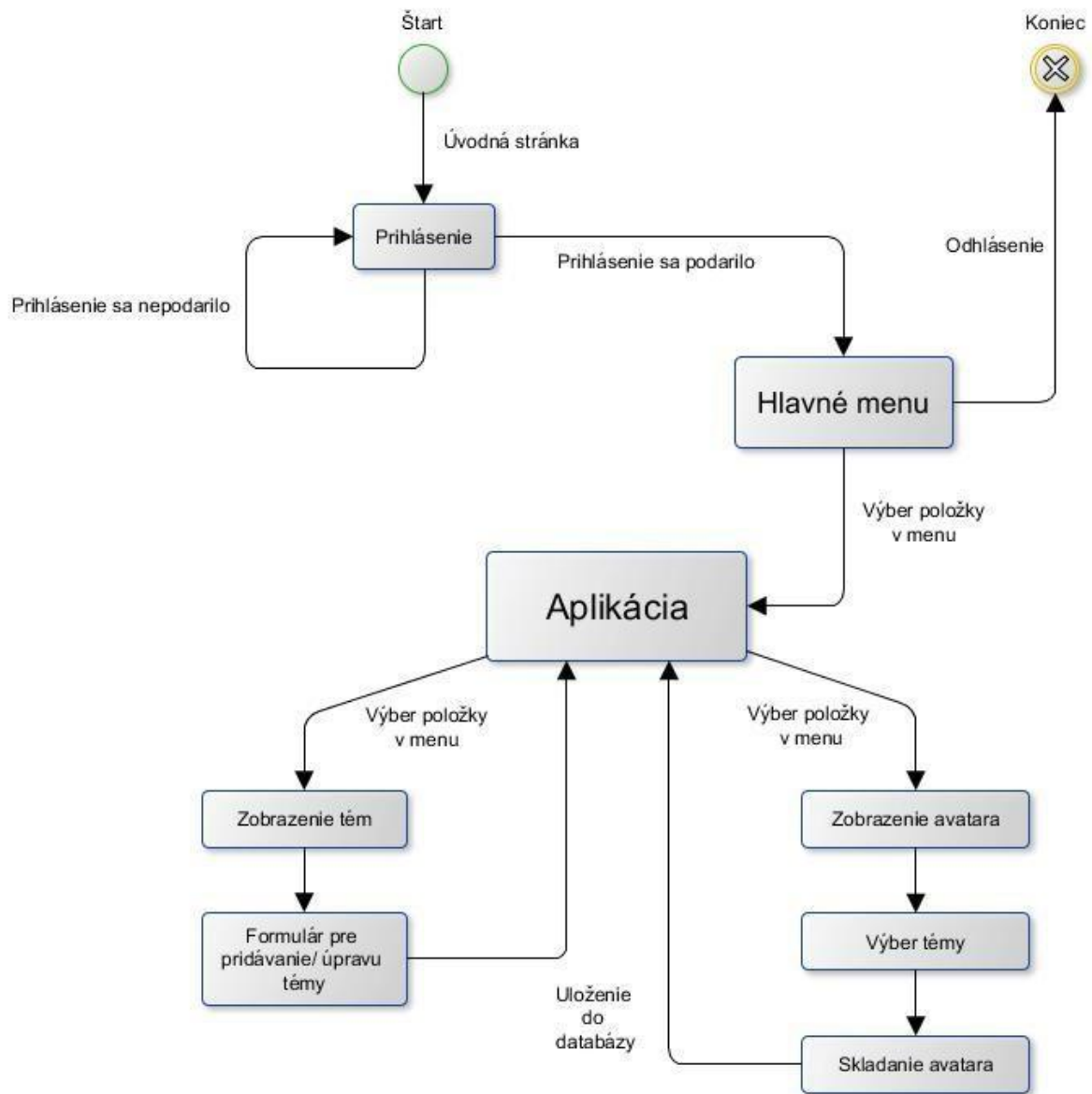


Obrázok 9



## 5.2 Stavový diagram

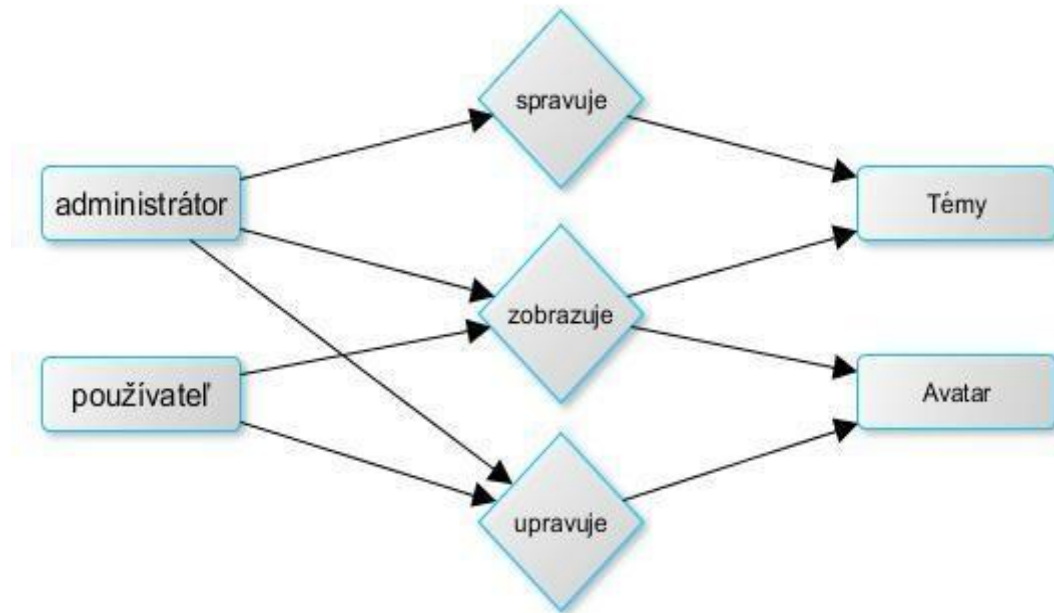
Nasledujúci stavový diagram zachytáva všetky kroky a stavy práce používateľa v aplikácii od jeho prihlásenia až po odhlásenie.



Obrázok 10

### 5.3 Entitno-relačný diagram

Diagram znázorňuje vzťahy medzi používateľom a jednotlivými fázami behu aplikácie.



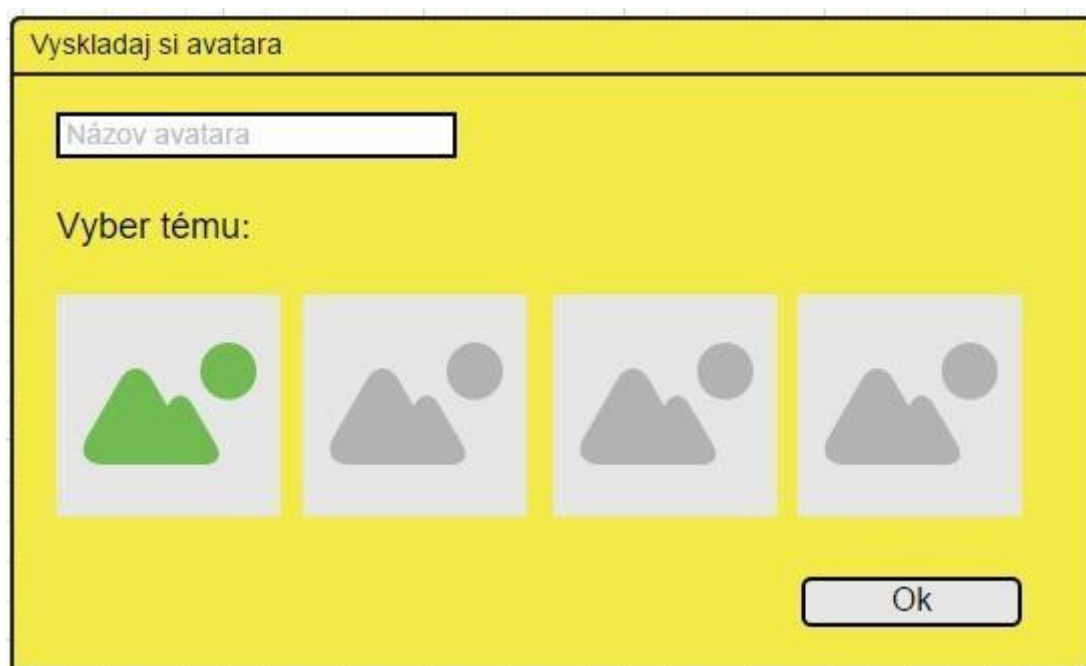
Obrázok 11

## 6 Používateľské rozhranie

### 6.1 Rozhranie pre používateľov (deti)

Táto časť dokumentu pojednáva o používateľskom rozhraní, ktoré bude vidieť používateľ žiak.

#### 6.1.1 Okienko na výber témy



Vyskladaj si avatara

Názov avatara

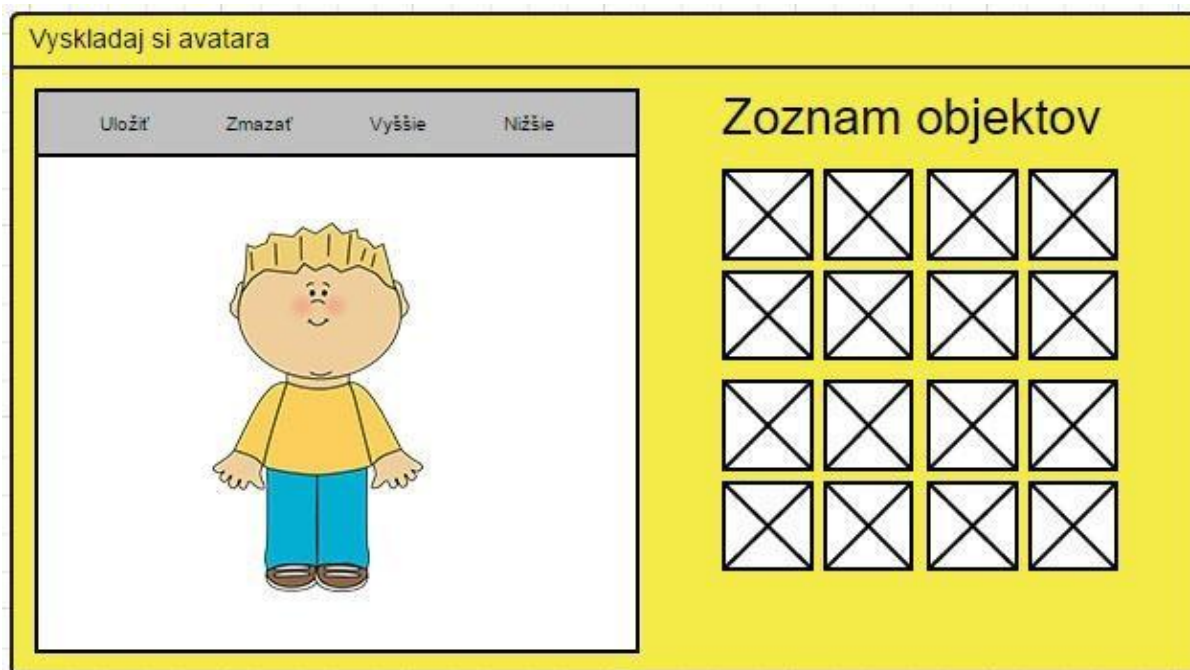
Vyber tému:

Four square buttons with landscape icons (mountains and a sun/moon). The first button is green, and the others are grey.

Ok

Po kliknutí na ikonku avatara v menu aplikácie sa zobrazí okno vyššie. Používateľ dieťa si zvolí názov avatara a následne si vyberie jednu z tém zo zoznamu, ktorú bude predstavovať obrázok.

### 6.1.2 Obrazovka s plochou na vyskladanie avatara a objektmi

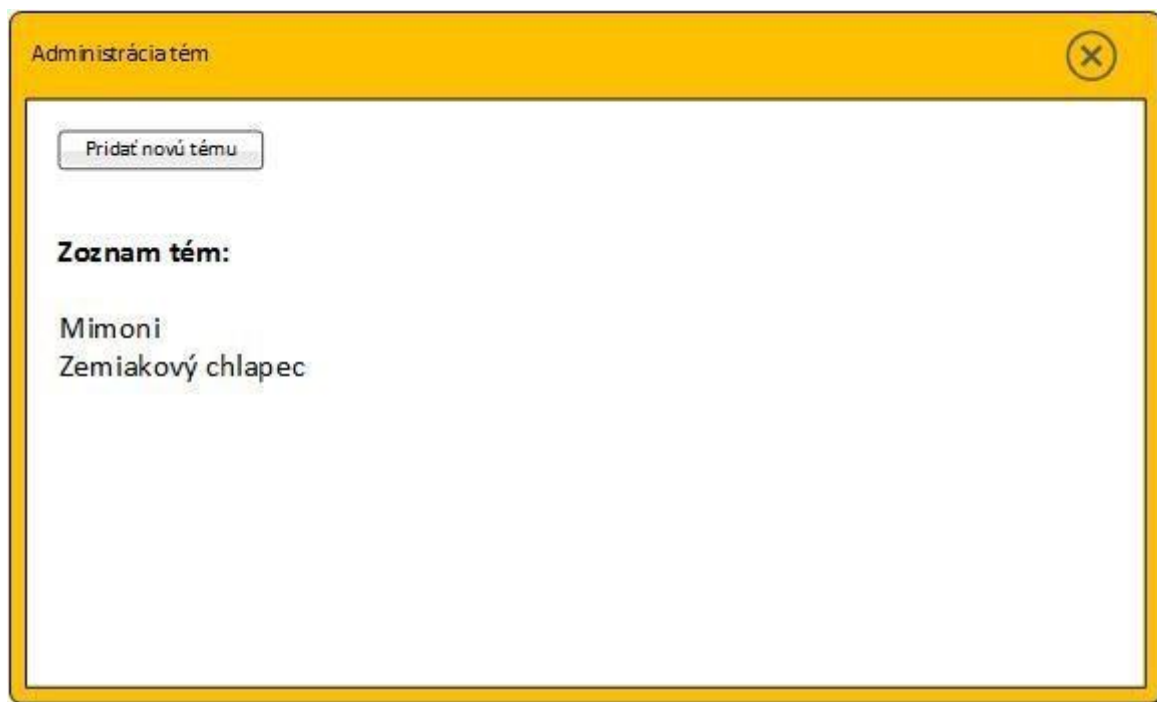


Ako náhle používateľ potvrdí tému, zobrazí sa mu okienko s plochou a objektmi. V ploche bude lišta s nástrojmi na pracovanie s objektmi a grafickou plochou ako napríklad zmazanie objektu, posunutie vyššie/nížšie a uloženie avatara.

## 6.2 Rozhranie pre správcu aplikácie

Táto časť dokumentu pojednáva o používateľskom rozhraní pre správcov aplikácie (administrátorov).

### 6.2.1 Okienko so zoznamom tém



Po kliknutí na tlačidlo administrácie tém v menu aplikácie sa zobrazí okno so zoznamom tém. Po kliknutí na tlačidlo pridať novú tému sa zobrazí ďalšie okienko na pridávanie objektov. Po kliknutí na jednu z tém zo zoznamu sa zobrazí okienko na upravovanie témy.

### 6.2.2 Okienko na pridanie témy

Administrácia tém

Názov témy

Náhľadový obrázok

Zoznam objektov


Presuňte obrázky do tejto plochy

Uložiť tému















V tomto okienku administrátor zadá názov témy. Vyberie vhodný náhľadový obrázok pre tému a následne nahrá objekty k téme.

Administrácia tém

Náhľadový obrázok



Zoznam objektov

Objekty v zozname objektov sa medzi sebou budú dať presúvať. V databáze je to pre tabuľku objekt kolónka order.

## **7 Analýza technológií**

### **7.1 Výber programovacích jazykov pre serverom vykonávané skripty**

Kedže vytvárame modul do už hotovej aplikácie, môžeme použiť len ten istý programovací jazyk a tým je JavaScript.

### **7.2 Výber programovacieho jazyka pre klientom vykonávané skripty**

Na klientom vykonávané skripty sa bude používať jazyk JavaScript, pretože:

- je to najznámejší a najpoužívanější client-side programovací jazyk
- je už použitý v aplikácii, do ktorej vytvárame tento modul
- nie je nutnosť používať dva rozdielne jazyky pre frontend a backend
- existuje množstvo kvalitných knižníc a frameworkov

### **7.3 Výber frameworkov**

#### **7.3.1 Framework pre backend**

Pre backend vyberáme framework NodeJS, pretože už je použitý v hotovej aplikácii, do ktorej vytvárame tento modul. Výhody frameworku NodeJS:

- nízke nároky na pamäť servera
- neblokujúce I/O
- veľké množstvo knižníc a frameworkov
- veľmi dobrá dokumentácia

#### **7.3.2 Framework pre frontend**

Pre frontend nebude použitý žiaden framewok.



## **7.4 Výber pomocných knižníc a pluginov**

### **7.4.1 Pre frontend**

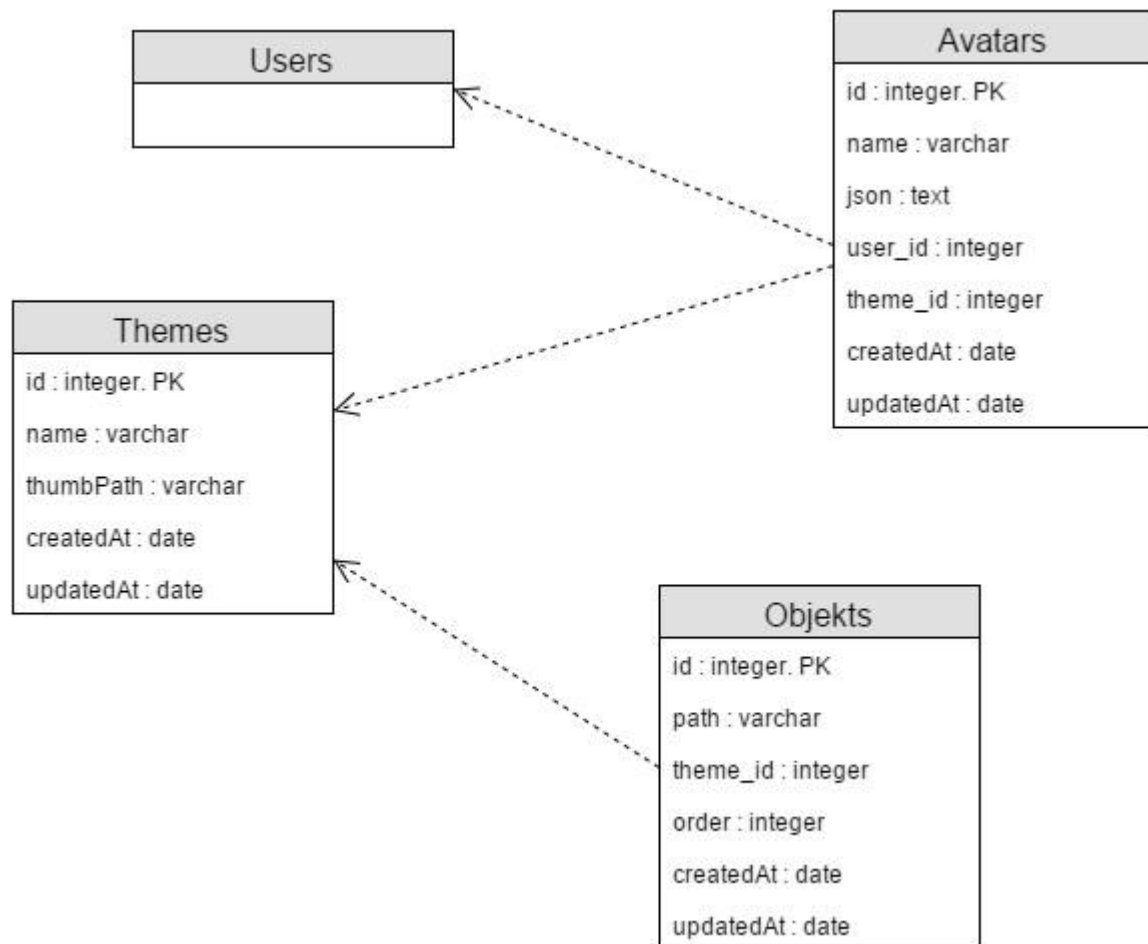
Pri tvorbe frontendu bude použitá knižnica Fabric.js, ktorá slúži na prácu s HTML5 canvasom. Hlavné výhody tejto knižnice:

- veľmi dobrá práca s canvasom
- objekty sa dajú zmenšovať zväčšovať
- celá plocha canvasu sa dá serializovať do formátu JSON alebo SVG a môže byť znovu obnovená
- podpora v NodeJS

### **7.4.2 Pre backend**

Pri tvorbe backendu bude použitá knižnica Underscore.js. Táto knižnica obsahuje sadu užitočných funkcií na prácu s poliami, mapami, atď...

## 8 Dátový model

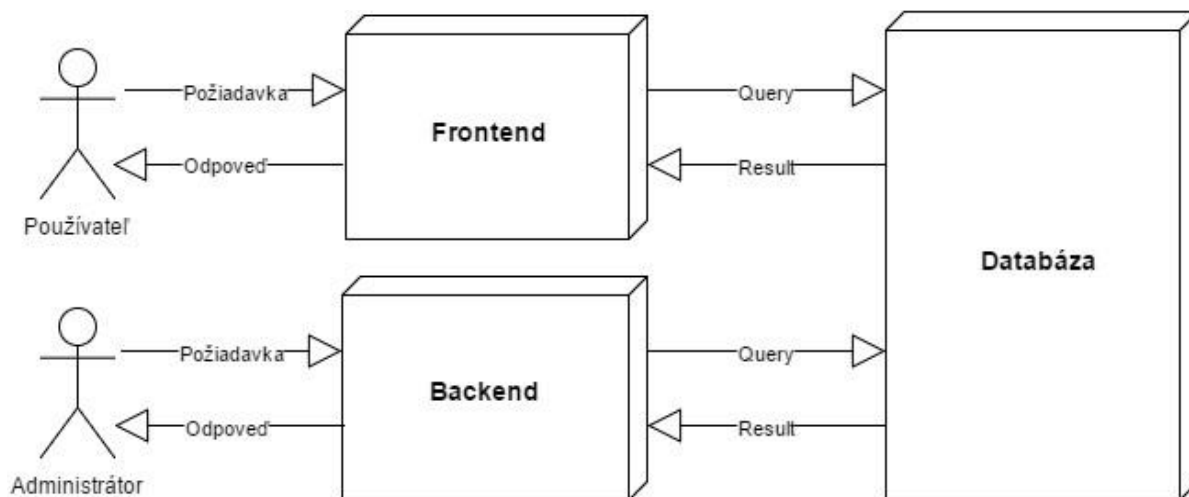


Vysvetlivky k obrázku 1:

- PK označuje primárny kľúč tabuľky
- Šípky označujú tabuľky ktoré tvoria medzi sebou relácie

## 9 Komponenty aplikácie

### 9.1 Schéma zapojenia komponentov

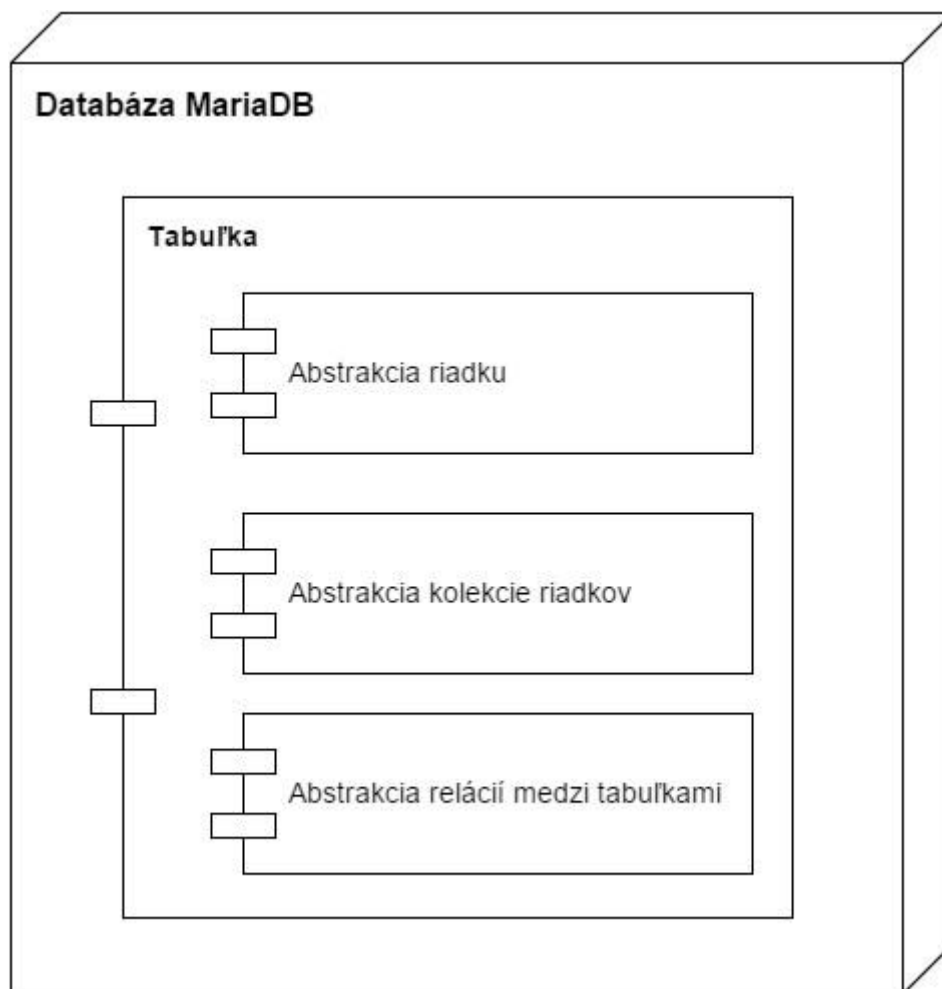


Obrázok 2: Schéma zapojenia komponentov

Používateľ vidí frontend uzol, ktorý v sebe zahŕňa ďalšie komponenty. Tento uzol komunikuje s databázovým uzlom, ktorý v sebe taktiež zahŕňa iné komponenty. Používateľ posielá požiadavky na frontend uzol a ten následne query na databázový uzol. Databáza odošle výsledok (result) na frontend uzol, ktorý sa zobrazí používateľovi.

Administrátor vidí backend uzol, pričom schéma funguje rovnako ako pri používateľovi.

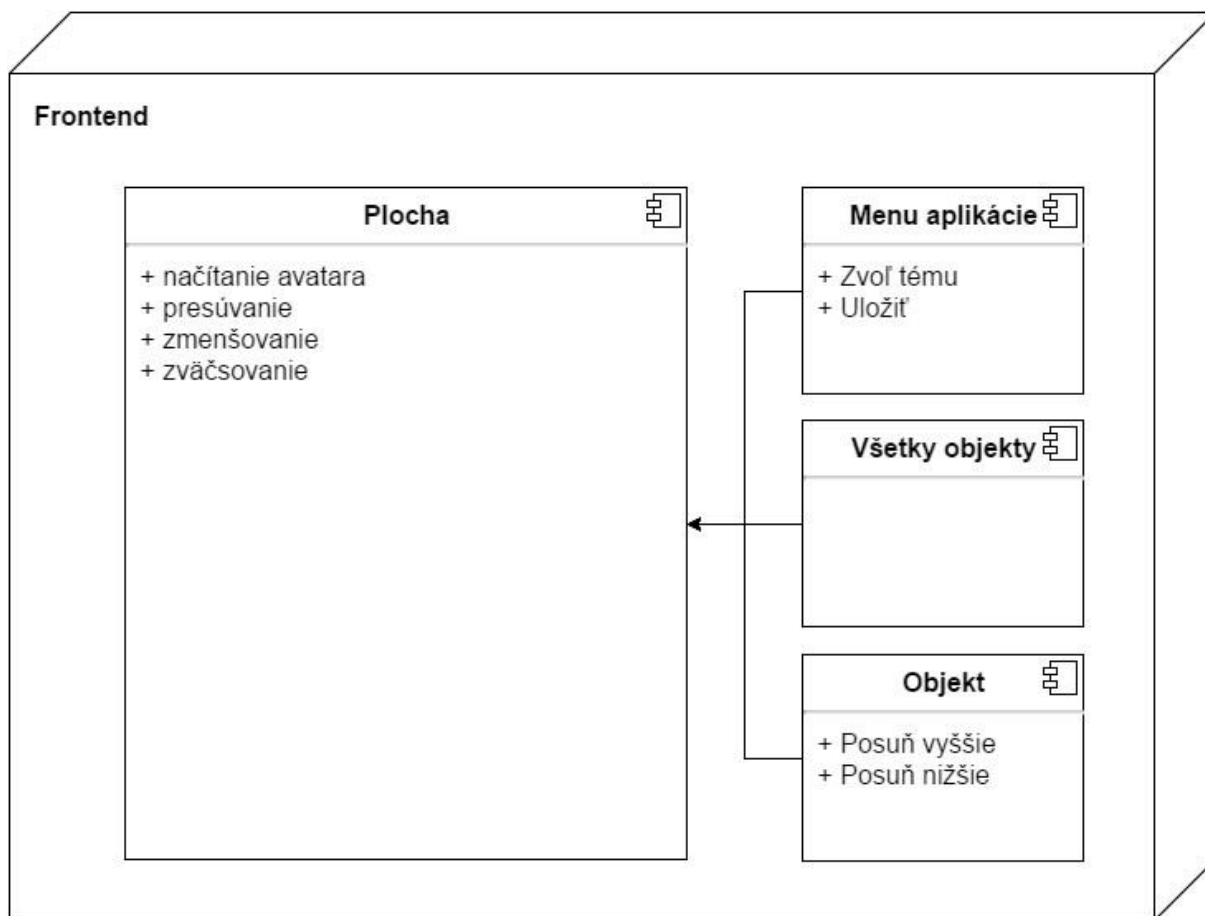
## 9.2 Databázový uzol



Databázový uzol MariaDB obsahuje komponenty pre jednotlivé tabuľky, ktoré sa skladajú z troch ďalších komponentov:

- abstrakcia riadku – mapuje jeden riadok tabuľky ako objekt, ktorý uchováva dáta tohto riadku a dokáže ich upravovať, mazať a vytvárať
- abstrakcia kolekcie riadkov – mapuje celý result do objektu obsahujúceho riadky tabuľky, používa rôzne filtre na upresnenie výberu a zoradenie výsledných riadkov
- abstrakcia relácií medzi tabuľkami – vytvára prepojenie jednej tabuľky s inou

### 9.3 Frontendový uzol

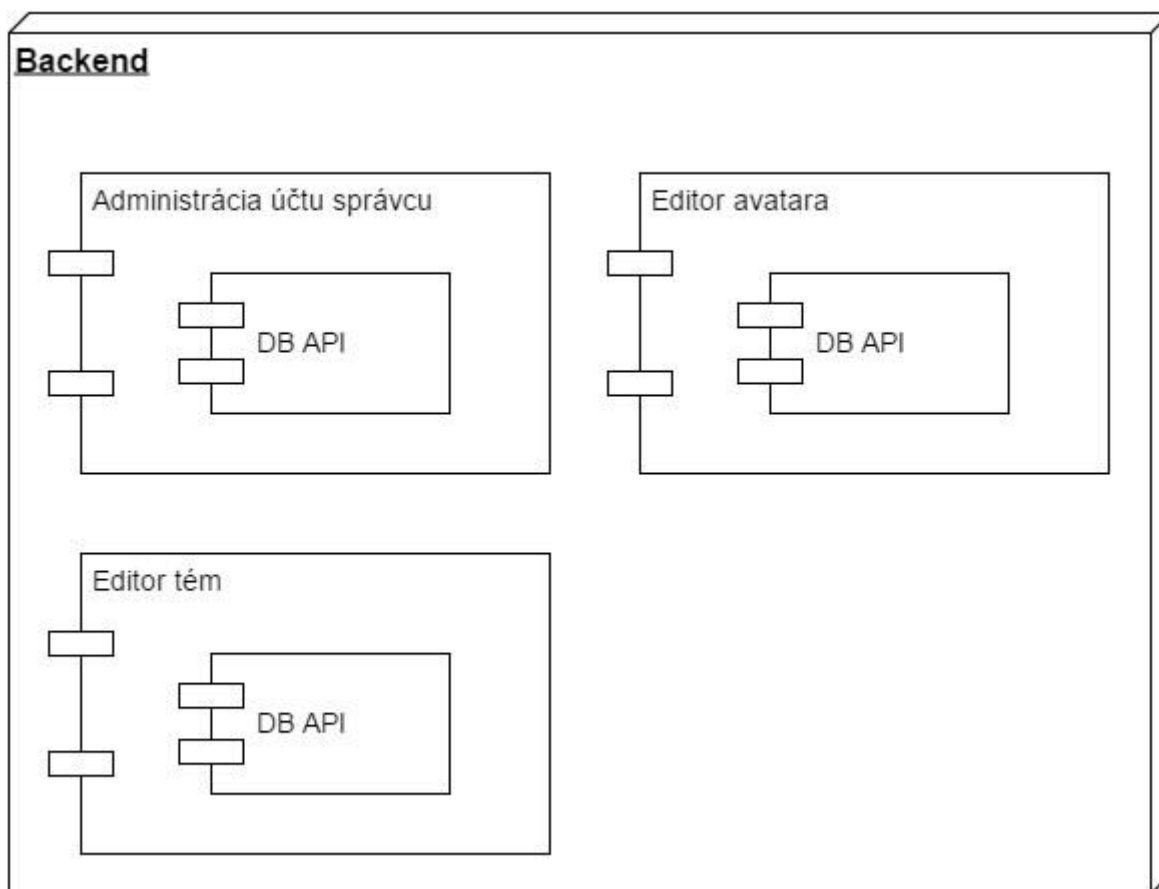


Obrázok zobrazuje uzol frontendových komponentov (komponent pre stránku zobrazenú návštevníkom)

Zahŕňa komponenty plochy, menu aplikácie, objektu (grafickej plochy) a ostatných objektov, pričom:

- plocha – je komponent, ktorý zobrazuje používateľom grafické objekty a umožňuje manipuláciu s nimi
- menu aplikácie – je komponent, ktorý zobrazuje používateľom ovládacie prvky aplikácie, pomocou ktorých komunikuje so systémom a tým riadi chod aplikácie
- objekt (grafickej plochy) - je komponent, ktorý zahŕňa grafický objekt zložený z ďalších objektov
- ostatné objekty – je komponent, ktorý zobrazuje používateľom všetky dostupné objekty, ktoré sú zobrazované na stránke s aktuálnou konfiguráciou

## 9.4 Backendový uzol



Obrázok zobrazuje uzol komponentov pre backend (správcovskú časť stránky), kde sa nachádzajú tieto komponenty:

- Administrácia účtu správcu – je komponent, ktorý zobrazuje pre správcu stránkovanú mriežku ku konkrétnej tabuľke, ktorá má definované operácie, ako napríklad upraviť záznam v tabuľke, zmazať záznam, overiť záznam (kde záznamom je účet registrovaného používateľa) pomocou databázového API
- Editor avatara – je komponent, ktorý zobrazuje všetky objekty z ktorých sa Avatar skladá a povoľuje upravovať ich parametre. Pri načítaní a ukladaní využíva databázové API
- Editor tém – je komponent, ktorý dovoľuje nahrávať a spravovať grafické témy pre vytváranie avatara. Tento komponent môže modifikovať a mazať existujúce témy, alebo pridávať nové témy kompatibilné so systémom. Komponent pritom využíva databázové API. Jeho hlavnou úlohou je správa tém v systéme, určenie tém ktoré budú zobrazované používateľom

## 10 Návrh

### 10.1 Úvod k návrhu

Cieľom dokumentu je podrobne špecifikovať hlavné časti a komponenty systému WebIDE, jednotlivé rozhrania a funkčnosti takým spôsobom, aby sme mohli začať prácu na samotnej implementácii projektu.

### 10.2 Návrh objektov

#### 10.2.1 Návrh tabuľkových modelov

Na návrh tabuľkových modelov bude použitá knižnica Sequelize.js

##### 10.2.1.1 Objekt Theme

Objekt Theme bude predstavovať tabuľku Themes v databáze MariaDB. Tabuľka Themes bude mať tieto atribúty:

- id : integer, autoincrement: true, PK
- name : string (názov témy)
- thumbPath : string (cesta k obrázku náhľadu témy)

##### 10.2.1.2 Objekt Avatar

Objekt Avatar bude predstavovať tabuľku Avatars v databáze MariaDB. Tabuľka Avatars bude mať tieto atribúty:

- id : integer, autoincrement: true, PK
- name : string (názov avatara)
- json : string (pozície objektov v avatarovi vo formáte json)
- user\_id : integer (ku ktorému užívateľovi avatar patrí)
- theme\_id : integer (téma z ktorej je avatar poskladaný)

### 10.2.1.3 Objekt Objekt

Objekt Objekt bude predstavovať tabuľku Objekts v databáze MariaDB. Tabuľka Objekts bude mať tieto atribúty:

- id : integer, autoincrement: true, PK
- path : string (cesta k obrázku objektu)
- order : integer (poradie v zozname pri skladaní avatara)
- theme\_id : integer (téma do ktorej patrí objekt)

PK označuje primárny kľúč.

### 10.2.2 Návrh funkcií plochy na skladanie avatara

**function getRandomNum(min, max)** – Funkcia dostane ako parameter dve čísla min a max. Funkcia vráti náhodné číslo v intervale min, max.

**function getRandomLeftTop()** – Funkcia vráti náhodnú pozíciu v ploche (canvas).

**function pad(str, length)** – Funkcia dostane ako parameter farbu v hexadecimálnom zápise. Funkcia vráti hexadecimálny zápis farby doplnený nulami.

**function getRandomColor()** – Funkcia vráti náhodnú farbu v hexadecimálnom zápise.

**function addTriangle()** – Funkcia pridá náhodný trojuholník do plochy.

**function addImage(imageName, minScale, maxScale)** – Funkcia dostane parametre názov obrázka, minimálne škálovanie, maximálne škálovanie. Potom obrázok vloží na náhodnú pozíciu do plochy.

**function rasterizeJSON()** – Funkcia rasterizuje canvas do formátu JSON

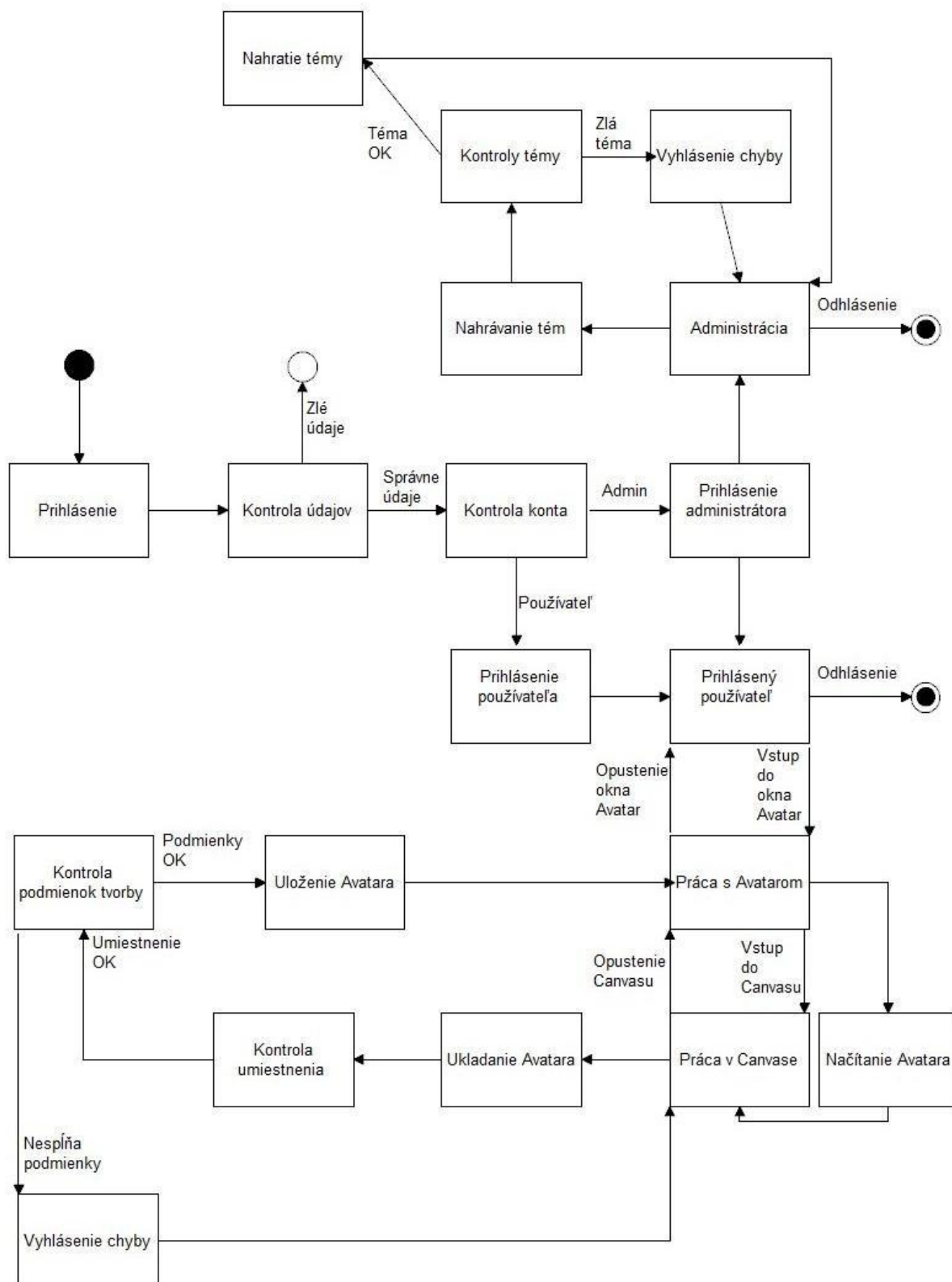
**function loadJSON(json)** – Funkcia načíta objekty zo vstupného stringu vo formáte json do plochy.

**function sendBackwards()** – Funkcia posunie označený objekt v ploche o úroveň nižšie.

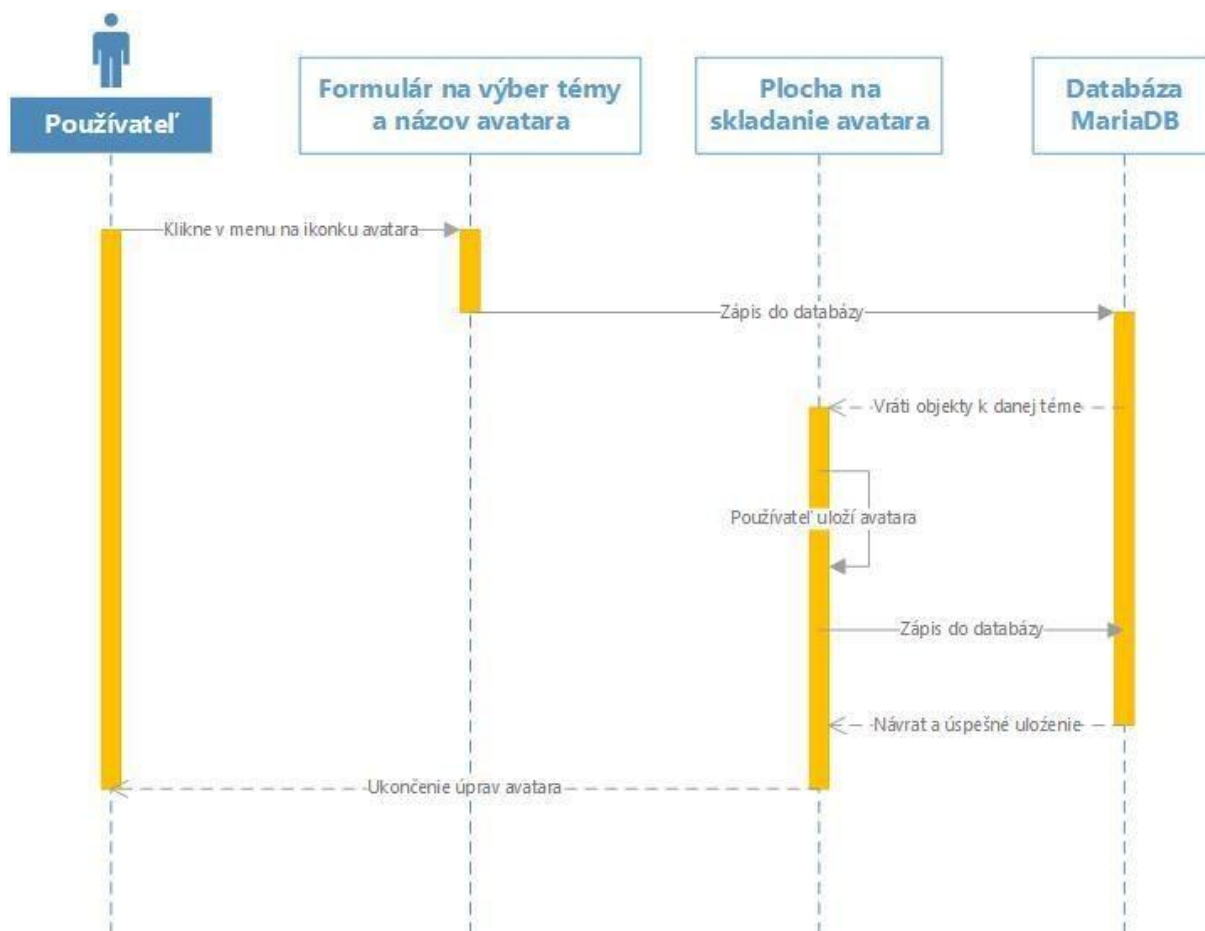
**function bringForward()** – Funkcia posunie označený objekt v ploche o úroveň vyššie.  
**function deleteSelected()** – Funkcia zmaže označený objekt z plochy.



## 10.3 Stavový diagram



## 10.4 Sekvenčný diagram



Po prihlásení používateľ klikne v menu na ikonku avatara. Následne sa mu zobrazí formulár kde zadá názov avatara a vyberie si jednu z tém. Po potvrdení sa v databáze uloží záznam o avatarovi. Následne sa používateľovi zobrazí okienko na vyskladanie avatara. Po dokončení skladania, používateľ klikne na tlačidlo uložiť a avatar sa uloží v databáze.

# 11 Testovanie

## 11.1 Testovacie scenáre – User interface

Test ID	Popis	Očakávaný výsledok	Predpoklady	Kroky k vykonaniu požiadavky	Status	Dátum ukončenia	Tester	Komentáre
1	Vytvorenie kompletného okna pre prácu s Avataram	Systém po stlačení tlačidla grafickej plochy pre Avatara otvorí nové okno s prvkami grafická plocha, na ktorú sa dajú premiestňovať grafické objekty z listy, v ktorej sú zobrazené. Ďalej sa bude zobrazovať ovládací panel obsahujúci tlačidlá potrebné pre ovládanie udalostí objektov na grafickej ploche	Systém korektné načítal všetky potrebné moduly	Stlačením tlačidla Avatar v paneli aplikácií sa otvorí okno s požadovanými funkčnými prvkami na tvorbu Avatara	Passed	20.11.2015	Michal Rakovský	
2	Umiestňovanie grafických objektov na plátno	Systém umiestni grafické objekty z listy objektov po interakcii užívateľom na grafickú plochu na náhodné pozície	Je k dispozícii grafická téma, obsahujúca grafické objekty a okno Avatar sa kompletne načítalo	Keď užívateľ klikne na požadovaný objekt v liste grafických objektov, tento objekt sa vytvorí ako nový objekt a systém mu nastaví súradnice z intervalu rozmerov grafickej plochy. Tento objekt sa zobrazí na plátno	Passed	42328	Michal Rakovský	
3	Posun, rotácia a zmena veľkosti grafických objektov na plátno	Systém umožňuje interaktívnu prácu z vytvorenými objektami na grafickej ploche. Zmení ich atribútov, ako sú súradnice, uhol, veľkosť	Na grafickej ploche sa nachádza grafický objekt a okno Avatar sa kompletne načítalo	Používateľ klikne na požadovaný grafický objekt na plátno, tento systém určí za označený, následne sa objektu sprístupní interaktívne rozhranie, ktoré zabezpečí prácu s jeho súradnicami, uhlom a veľkosťou pomocou udalostí riadených myšou, alebo iným zariadením umožňujúcim riadenie kurzora	Passed	42328	Michal Rakovský	
4	Vyčistenie grafickej plochy, odstraňovanie jednotlivých objektov	Systém pomocou prvkov ovládacieho panela grafickej plochy zabezpečí odstraňovanie grafických objektov jednotlivito a taktiež hromadne každý objekt, ktorý plátno obsahuje	Odstraňovanie objektov grafickej plochy / knižnice jabra.js zabezpečuje vhodné použitie tejto knižnice	Používateľ označí interakciou myši objekt grafickej plochy. Stlačením tlačidla pre jednotlivé mazanie objektov sa zmaže objekt, ktorý je systémom určený ako označený objekt. Ak používateľ zvolí tlačidlo pre vyčistenie plátna, systém vymaže akýkoľvek grafický objekt nachádzajúci sa na danom plátno	Passed	42328	Michal Rakovský	
5	Zmena úrovni vrstiev objektov na plátno	Systém umožňuje zmenu vrstvy daného objektu pomocou prvkov na ovládacom paneli grafickej plochy		Používateľ kliknutím myši označí grafický objekt na plátno, tento sa nastaví systémom za označený. Pomocou tlačidiel je možné upravovať úroveň jeho vrstvy a teda je možné ním prekryvať iné objekty, alebo ho iným objektom prekryť	Passed	42328	Michal Rakovský	
6	Uloženie grafických objektov Avatara do databázy	Systém uloží všetky grafické prvky v podobe serializácie objektov typu JSON do databázy	Grafické objekty splnili požiadavky na uloženie	Používateľ interakciou na tlačidlo uloženia Avatara pošle systému požiadavku. Systém následne prekontroluje plochu a jednotlivé objekty, ktoré po splnení požiadaviek serializuje do podoby JSON a následne odošle ORM modulom Sequelize na databázový server.	Passed	42356	Michal Rakovský	
7	Načítanie uloženého Avatara do grafickej plochy	Systém z databázy načíta serializované objekty, preloží ich do objektov grafickej plochy a následne zobrazí na grafickej ploche	V databáze sa nachádza aspoň jeden objekt Avatara pre daného používateľa	Používateľ zvolí zo zoznamu Avatarov kliknutím tlačidla myši, systém ho určí ako označený. Následne používateľ stlačením tlačidla pre načítanie Avatara odošle systému požiadavku, systém ju spracuje, deserializuje objekty z JSON do grafických objektov, ktoré následne vykreslí na plátno	Passed	42356	Michal Rakovský	
8	Úprava atribútov grafických objektov načítaného Avatara	Systém umožňuje upravovať uloženého Avatara rovnakými prvkami, ktoré využíva pri vytváraní Avatara	Avatar bol úspešne načítaný a grafické objekty sú vykreslené na plátno	Používateľ klikne na požadovaný grafický objekt na plátno, tento systém určí za označený, následne sa objektu sprístupní interaktívne rozhranie, ktoré zabezpečí prácu s jeho súradnicami, uhlom a veľkosťou pomocou udalostí riadených myšou, alebo iným zariadením umožňujúcim riadenie kurzora	Passed	42356	Michal Rakovský	
9	Načítanie grafickej témy pri tvorbe nového Avatara	Systém umožňuje tvorbu Avatara s využitím tém z databázy	V databáze sa nachádza aspoň jedna grafická téma	Používateľ si kliknutím myši zvolí danú tému, ktorú bude používať na vytváranie Avatara, systém ju určí ako označenú. Následným kliknutím na tlačidlo načítania témy ju systém načíta a otvorí okno s grafickou plochou, a ďalšími prvkami tohto okna a v liste grafickej témy sa nachádzajú objekty zvolenej témy	Passed	42356	Michal Rakovský	

## 11.2 Testovacie scenáre – admin interface

Test ID	Popis	Očakávaný výsledok	Predpoklady	Kroky k vykonaniu požiadavky	Status	Dátum ukončenia	Tester	Komentáre
31.12.1899	Načítanie okna pre administráciu - vytvorenie témy	Systém po stlačení tlačidla grafickej plochy pre administráciu otvorí nové okno s prvkami ktoré umožňujú nahrávanie obrázkových súborov, ďalej textové polia pre zvolenie názvu témy a i.	Systém korektne načítal všetky potrebné moduly	Administrátor po interakcii s položkou administrácia odošle systému požiadavku na vytvorenie okna pre administráciu. Toto okno sa zobrazí podľa požiadaviek	Passed	6.1.2016	Michal Rakovský	
1.1.1900	Načítanie .png obrázkov k novej téme	Systém nahráva obrázky do pamäte, následne ich zobrazuje v zozname	Administrátorské okno bolo korektne načítané	Administrátor vloží obrázok do komponentu pre načítanie obrázkov. Systém tento obrázok spracuje a uloží do pamäte	Passed	6.1.2016	Michal Rakovský	
2.1.1900	Uloženie témy do databázy	Systém ukladá objekty témy z pamäte do databázy	V pamäti systému sa nachádza dostatočný počet obrázkov	Administrátor kliknutím na tlačidlo nahratia témy odošle systému požiadavku na uloženie témy. Systém spracuje obrázky, uloží ich na disk a do databázy uloží cestu k téme a potrebné informácie ako názov, počet prvkov a i.	Passed	6.1.2016	Michal Rakovský	
3.1.1900	Načítanie dostupných tém z databázy a ich zobrazenie v okne	Systém načíta z databázy témy, v administrátorskom okne zobrazí zoznam dostupných tém	V databáze sa nachádza aspoň jedna grafická téma	Administrátor po kliknutí myšou na tlačidlo načítania témy odošle systému požiadavku na načítanie tém, systém ich vypíše v novom okne	Passed	20.1.2016	Michal Rakovský	
4.1.1900	Úprava a mazanie uložených tém v databáze	Systém načíta z databázy témy, v administrátorskom okne zobrazí informácie o téme a náhľady k obrázkom témy. Všetky tieto zobrazenia sú upravovateľné. V zozname tém pribudne možnosť mazania témy	V databáze sa nachádza aspoň jedna grafická téma, ktorá bola zvolá administrátorom na úpravu	Systém odpovie na výber témy otvorením okna, v ktorom sú zobrazené obrázky a informácie o téme, ktoré možno meniť. Obrázky je možné vymazávať a nahrávať nové. Tieto zmeny sa po stlačení tlačidla uložia aplikujú aj na disk, aj v databáze. Ak administrátor v zozname tém zvolí tému, kliknutí tlačidla na odstránenie systém túto tému odstráni z databázy a taktiež odstráni celý jej súborový priečinok na disku	Passed	20.1.2016	Michal Rakovský	

## **12 Záznam z odovzdávania a predvedenia výslednej aplikácie zadávateľovi**

### ***12.1 Plán stretnutia***

Odovzdanie a predvedenie funkcionlít projektu zadávateľovi práce - pánovi profesorovi Marekovi Nagyovi - sa uskutočnilo 28.01.2016, streda o 10 hodine v kancelárii F248.

### ***12.2 Priebeh stretnutia***

Prezentáciu viedol Michal Rakovský, a prezentoval to na jednom z našich počítačov. Zadávatel'a sme v skratke informovali o priebehu našej práce. Určili sme požiadavky ktoré boli splnené, ktoré zatiaľ neboli splnené, prečo sa tak stalo. Predviedli sme aplikáciu a jej funkcionlitu pre rôznych užívateľov(prihlasený používateľ, administrátor). Zadávatel' zhodnotil prvý dojem ako veľmi dobrý.

Po predvedení zadávateľ napriek tomu že mal niekoľko pripomienok vyzeral že bol spokojný s projektom.

## **13 Zhodnotenie**

### ***13.1 Spokojnosť s výsledným dielom, ťažkosti počas vývoja***

Všetci členovia tímu sú s výsledným dielom spokojný. Podarilo sa nám implementovať skoro všetky požiadavky zadávateľa. Počas vývoja sa vyskytlo niekoľko problémov, ktoré sa však podarilo úspešne vyriešiť.

Jedným z problémov bola neskúsenosť členov tímu s NodeJS frameworkom. Vyskytli sa menšie problémy pri vložení viacerých tím. Tiež vyskytli menšie problémy pri ukladaní už vyskladaného postavičku. Tieto problémy sa podarilo vyriešiť pomocou materiálov nájdených na internete a tímovej spolupráce vyriešiť.

### ***13.2 Zmeny do ďalších verzií***

### ***13.3 Odlišnosti od pôvodného plánu***

Počas implementácie neprišlo k zásadným zmenám oproti plánu. Najväčším problémom bolo časové dodržanie plánu. Časový plán sa podarilo plniť počas špecifikácie a návrhu práce. Počas vývoja došlo k sklzu, úlohy sa nepodarila vždy splniť dohodnutých časových intervaloch. Napriek tomu sme žiadnu závažnú chybu neobjavili ani po odovzdaní hotovej aplikácie.

### ***13.4 Tímová práca, rozdelenie úloh a komunikácia***

Počas celej dĺžky trvania kurzu sme komunikovali na pravidelných skupinových stretnutiach na portáli Facebook. Tento spôsob sa ukázal ako dostatočný a umožnil nám riešiť problémy s implementáciou, či dohodnúť stretnutia tímu. Pri komunikácii nedochádzalo k výraznejším problémom. S tímovou spoluprácou boli všetci členovia spokojný, keďže na rozdelení úloh aj ostatných záležitostiach týkajúcich sa spolupráce sme sa vždy dohodli rýchlo a bez problémov.

Možnosť tímovej práce bola taktiež zabezpečená, po dohodnutí termínu sa členovia tímu stretávali, kde diskutovali vzniknuté problémy, nápady. Tímové stretnutia boli organizované počas pracovných dní. Na stretnutiach sa prediskutovali najsledujúce úlohy a spôsob akým sa budú riešiť, ujasnili sa problémy z minulého stretnutia.

### **13.5 Záver**

Táto správa ponúka komplementný prehľad vývoja aplikácie. Od prvých krokov, tvorby tímu, úvah o fungovaní, naplánovania funkčnosti, prepojení jednotlivých komponentov až po realizáciu a zhodnotenie výsledného produktu našej práce. Už od začiatku sme sa snažili, aby naše výsledné dielo zodpovedalo požiadavkam zadávateľa a aby bolo v budúcnosti dobre využiteľné. Myslíme, že tento cieľ sa podarilo splniť.

Práca na tomto projekte nám dala presnú predstavu o práci v kolektíve, rozdeľovaní úloh a dodržiavaní stanových termínov. Preto tento projekt hodnotíme ako cennú skúsenosť. Dúfame, že tento dokument výrazne pomôže pri prípadnom zlepšovaní alebo rozširovaní tejto aplikácie.

## 14 Zdroje

- [ 1 ] KTuberling – [www.kde.org/applications/games/ktuberling/](http://www.kde.org/applications/games/ktuberling/)
- [ 2 ] NodeJS – webový server napísaný v JavaScripte [www.nodejs.org](http://www.nodejs.org)
- [ 3 ] MariaDB – SQL relačná databáza [www.mariadb.org](http://www.mariadb.org)
- [ 4 ] FabricJS – JavaScriptová knižnica na prácu s html5 canvasom [www.fabricjs.com](http://www.fabricjs.com)