

**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	Chyba! Záložka nie je definovaná.
3.1	Slovník pojmov, skratky	Chyba! Záložka nie je definovaná.
3.2	Používateľská časť	Chyba! Záložka nie je definovaná.
3.2.1	Načítaj tému	Chyba! Záložka nie je definovaná.
3.2.2	Vyskladaj avatara	Chyba! Záložka nie je definovaná.
3.2.3	Ulož avatara	Chyba! Záložka nie je definovaná.
3.2.4	Načítaj avatara	Chyba! Záložka nie je definovaná.
3.3	Administrátorská časť	Chyba! Záložka nie je definovaná.
3.3.1	Vlož tému	Chyba! Záložka nie je definovaná.
3.4	Výkonnostné požiadavky	Chyba! Záložka nie je definovaná.
3.5	Dostupnosť	Chyba! Záložka nie je definovaná.
3.6	Bezpečnostné požiadavky	Chyba! Záložka nie je definovaná.
4	Analýza používateľov	10
5	Diagramy	11
5.1	Use-case diagram pre jednotlivých používateľov	11
5.2	Stavový diagram	11
5.3	Entitno-relačný diagram	12
6	Používateľské rozhranie	14
6.1	Rozhranie pre používateľov (deti)	14
6.1.1	Hlavné menu	14

6.1.2	Vytvorenie avatara	15
6.2	Rozhranie pre správcu aplikácie	18
6.2.1	Pridanie témy.....	18
7	Analýza technológií.....	19
7.1	Výber programovacích jazykov pre serverom vykonávané skripty	19
7.2	Výber programovacieho jazyka pre klientom vykonávané skripty	19
7.3	Výber frameworkov	19
7.3.1	Framework pre backend	19
7.3.2	Framework pre frontend	19
7.4	Výber pomocných knižníc a pluginov	20
7.4.1	Pre frontend.....	20
7.4.2	Pre backend	20
8	Dátový model.....	21
9	Komponenty aplikácie.....	22
9.1	Schéma zapojenia komponentov	22
9.2	Databázový uzol.....	23
9.3	Frontendový uzol	24
9.4	Backendový uzol	25
10	Návrh	26
10.1	Úvod k návrhu.....	26
10.2	Návrh objektov	26
10.2.1	Návrh tabuľkových modelov.....	26
10.2.2	Návrh funkcií plochy na skladanie avatara	27
10.3	Stavový diagram	29
10.4	Sekvenčný diagram	30
11	Testovanie.....	31
11.1	Testovacie scenáre – User interface.....	31
11.2	Testovacie scenáre – admin interface	32
12	Záznam z odovzdávania a predvedenia výslednej aplikácie zadávateľovi	33
12.1	Plán stretnutia	33
12.2	Priebeh stretnutia.....	33
13	Zhodnotenie.....	34

13.1	Spokojnosť s výsledným dielom, ťažkosti počas vývoja	34
13.2	Zmeny do ďalších verzií	34
13.3	Odlišnosti od pôvodného plánu.....	34
13.4	Tímová práca, rozdelenie úloh a komunikácia	34
13.5	Záver	35
14	Zdroje.....	36

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 objednávatelom 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

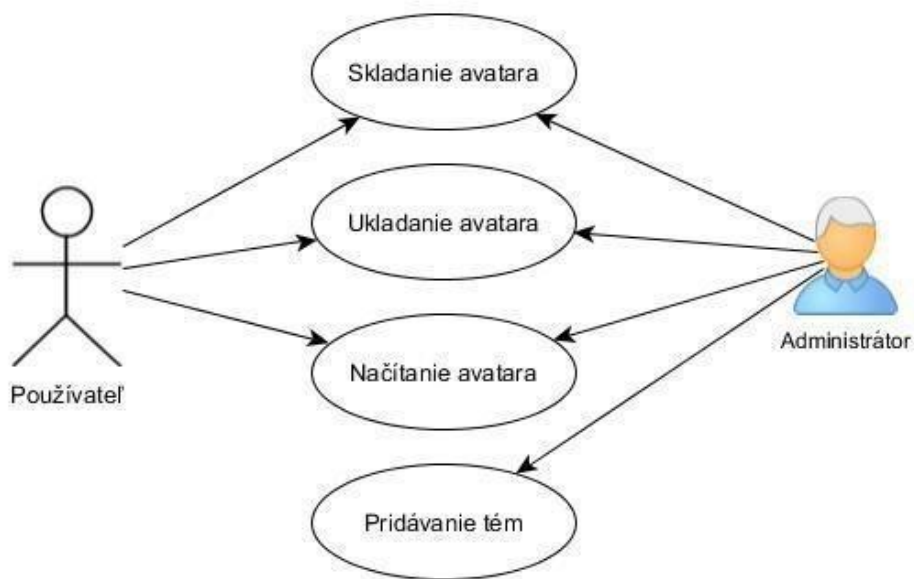
SSR-1	Aplikácia by mala byť spustiteľná na linuxovom serveri s nainštalovaným NodeJS.
SSR-2	Na serveri musí byť nainštalovaná databáza MariaDB.

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

2.6 Obmedzenia

- Aplikácia bude spustiteľná na operačnom systéme Linux s nainštalovaným NodeJS a databázou MariaDB.
- Nebude možné importovať objekty z iných formátov ako svg, JSON.
- Aplikáciu nebude možné spustiť bez funkčného prehliadača

2.7 Predpoklady a závislosti

2.7.1 Softvérové závislosti

Celá aplikácia bude postavená na NodeJS a komunikovať bude s databázou MariaDB pomocou SQL dotazov. Na vykreslenie a prácu s objektmi bude použitá javascriptová knižnica Fabric.js. Všetky ostatné vizuálne prvky budú vytvorené v jazyku html s použitím kaskádových štýlov CSS.

2.7.2 Hardvérové závislosti

Na bezproblémový chod celej aplikácie bude nutné vlastniť zariadenie, na ktoré je možné nainštalovať a nakonfigurovať operačný systém Linux so všetkými závislosťami spomenutými vyššie.

3 Špecifikácia požiadaviek

3.1 Vyskladaj avatara

Používateľ bude môcť skladať avatara na grafickej ploche kliknutím na ikony objektov tém, kliknutím na tlačidlá uvedených nástrojov a presunom, škálovaním objektov.

3.1.1 Vstupná požiadavka

Musí byť načítaná aspoň jedna téma.

3.1.2 Názov avatara

Možnosť zadať názov avatara.

3.1.3 Načítaj témy

Systém načíta zoznam dostupných tém. Používateľ si bude môcť vybrať tému kliknutím na ikonku témy.

3.1.4 Lišta s nástrojmi

Používateľ bude môcť pracovať s viacerými nástrojmi:

- Zmena z-tovej súradnice grafického objektu.
- Zmazanie objektu na grafickej ploche.
- Zmazanie grafickej plochy.
- Uloženie avatara.

3.1.5 Grafická plocha

Objekty v grafickej ploche sa budú dať presúvať, zmenšovať a zväčšovať.

3.1.6 Paleta objektov

Paleta objektov sa bude nachádzať vedľa grafickej plochy. Používateľ kliknutím na daný objekt v palete nástrojov, pridá objekt do grafickej plochy.

3.2 Uprav avatara

Možnosť vybrať si vyskladaného avatara zo zoznamu vyskladaných avatarov kliknutím na ikonku avatara a po načítaní upraviť avatara.

3.2.1 Vstupná požiadavka

Musí byť načítaný aspoň jeden avatar.

3.2.2 Načítaj avatara

Systém načíta zoznam vyskladaných avatarov.

3.3 Ulož avatara

Používateľ bude môcť uložiť vyskladaného avatara.

3.4 Vlož tému

Administrátor bude môcť vložiť tému do systému, táto téma bude potom prístupná používateľom. Pri vložení témy s nevyplneným názvom, bez náhľadového obrázku alebo s nesprávnym počtom nahratých objektov, systém vyhlási chybu.

3.4.1 Názov témy

Administrátor bude musieť zadať názov témy.

3.4.2 Náhľadový obrázok

Administrátor si bude musieť vybrať vhodný náhľadový obrázok pre tému veľkosti 100x100 pixelov.

3.4.3 Objekty témy

Administrátor bude môcť nahráť objekty k téme. Minimálny počet nahratých objektov potrebných pre vloženie témy bude 1. Maximálny počet objektov bude neobmedzený.

4 Ďalšie požiadavky

4.1 Výkonnostné požiadavky

- Načítanie témy nesmie trvať dlhšie ako 5 sekúnd.
- Pohyb grafických objektov musí byť plynulý.
- Načítanie Avatara nesmie trvať dlhšie ako 10 sekúnd.

4.2 Dostupnosť

- V pracovné dni musí byť dostupnosť aplikácie 98.5%.
- Počas víkendov a sviatkov musí byť dostupnosť 95%.

4.3 Bezpečnostné požiadavky

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

- Ochrana proti SQL Injection.
- Ochrana proti XSS.

5 Analýza používateľov

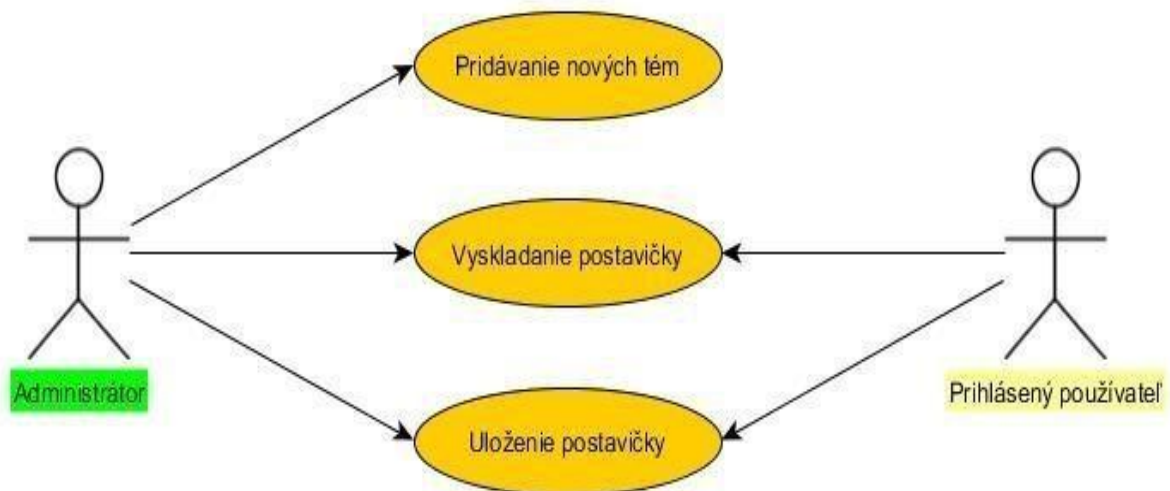
Na základe požiadaviek stanovených zadávateľom, sme stanovili nasledovných používateľov, ktorí budú pracovať s aplikáciou:

- 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 (avatare). K takto vyskladanej postavičke sa bude môcť kedykoľvek vrátiť a upraviť.

6 Diagramy

6.1 Use-case diagram pre jednotlivých používateľov

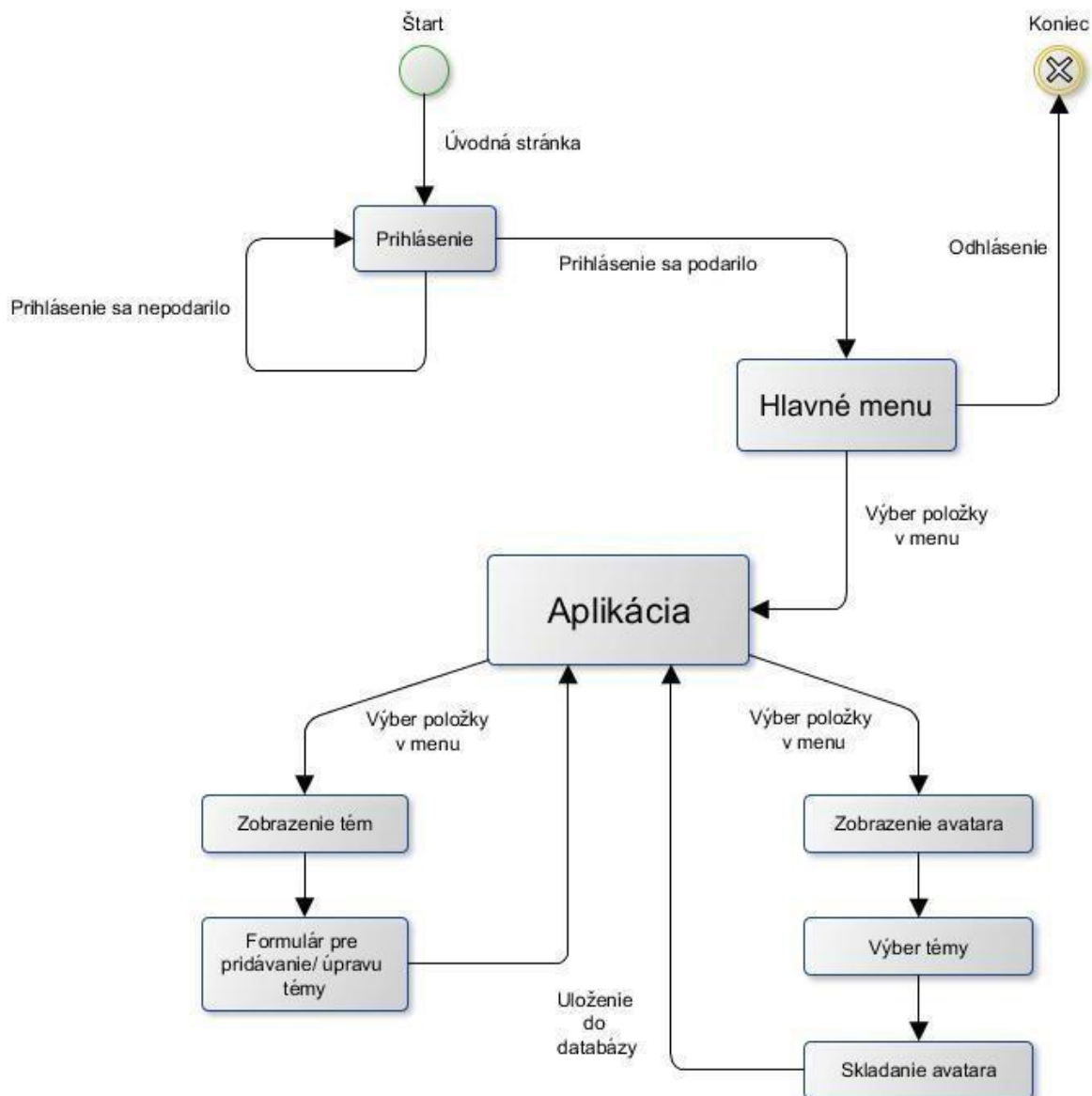
Nasledujúci use-case diagram zobrazuje množinu činností, ktoré môže vykonávať administrátor systému a prihlásený používateľ.



Obrázok 9

6.2 Stavový diagram

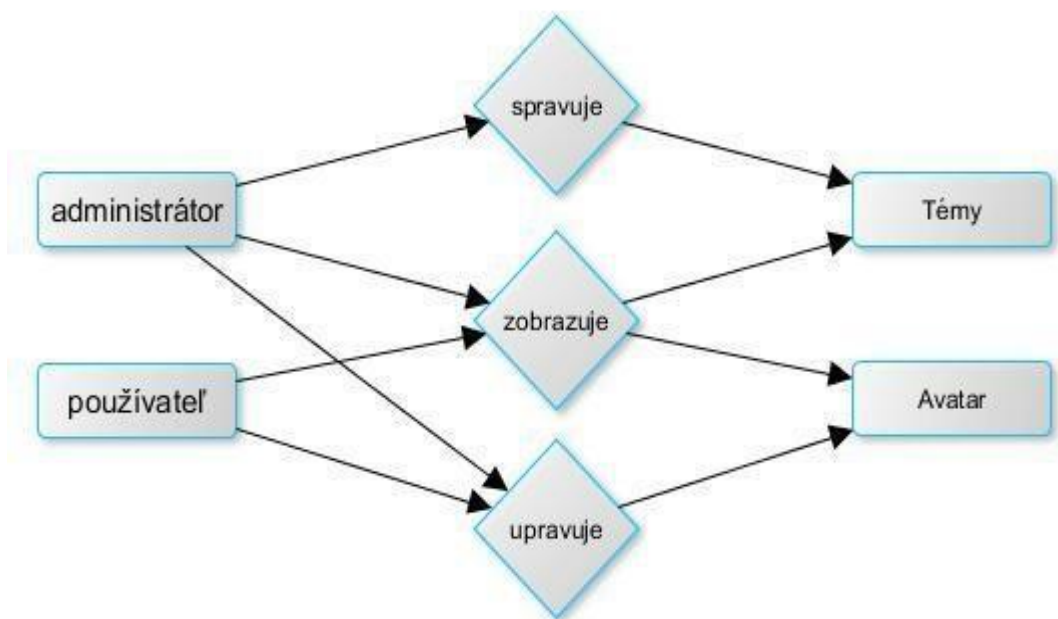
Nasledujúci stavový diagram zachytáva všetky kroky a stavy práce používateľa v aplikácii od jeho prihlásenia až po odhlásenie.



Obrázok 10

6.3 Entitno-relačný diagram

Diagram znázorňuje vzťahy medzi používateľom a jednotlivými fázami behu aplikácie.



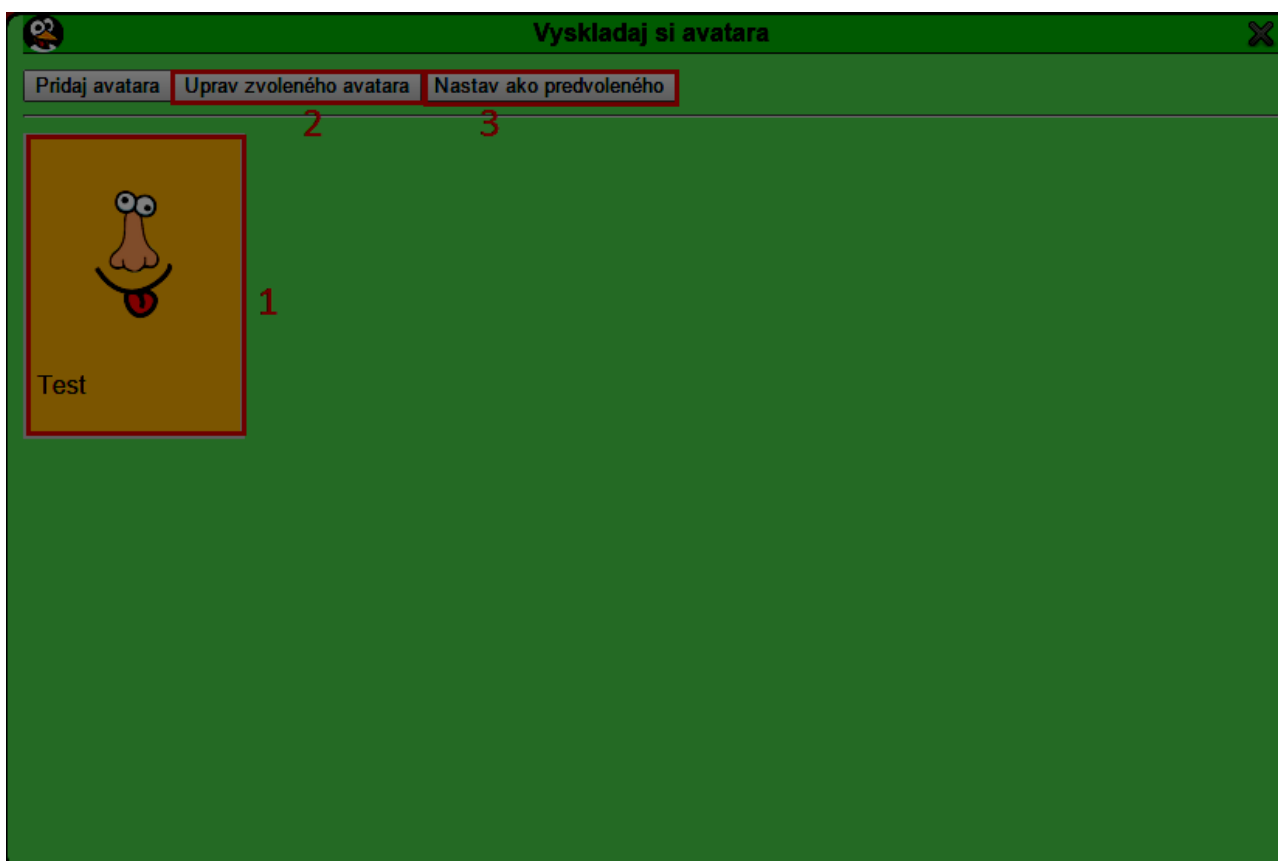
Obrázok 11

7 Používateľské rozhranie

7.1 Rozhranie pre používateľov (deti)

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

7.1.1 Hlavné menu



1. – Používateľ si vyberie avatára.
2. – Po výbere používateľ si môže upraviť avatára. Po kliknutí na tlačidlo sa otvorí rovnaký okno ako pri vytváraní.
3. – Po kliknutí na tlačidlo používateľ si nastaví predvoleného avatara (profilový obrazok používateľa).

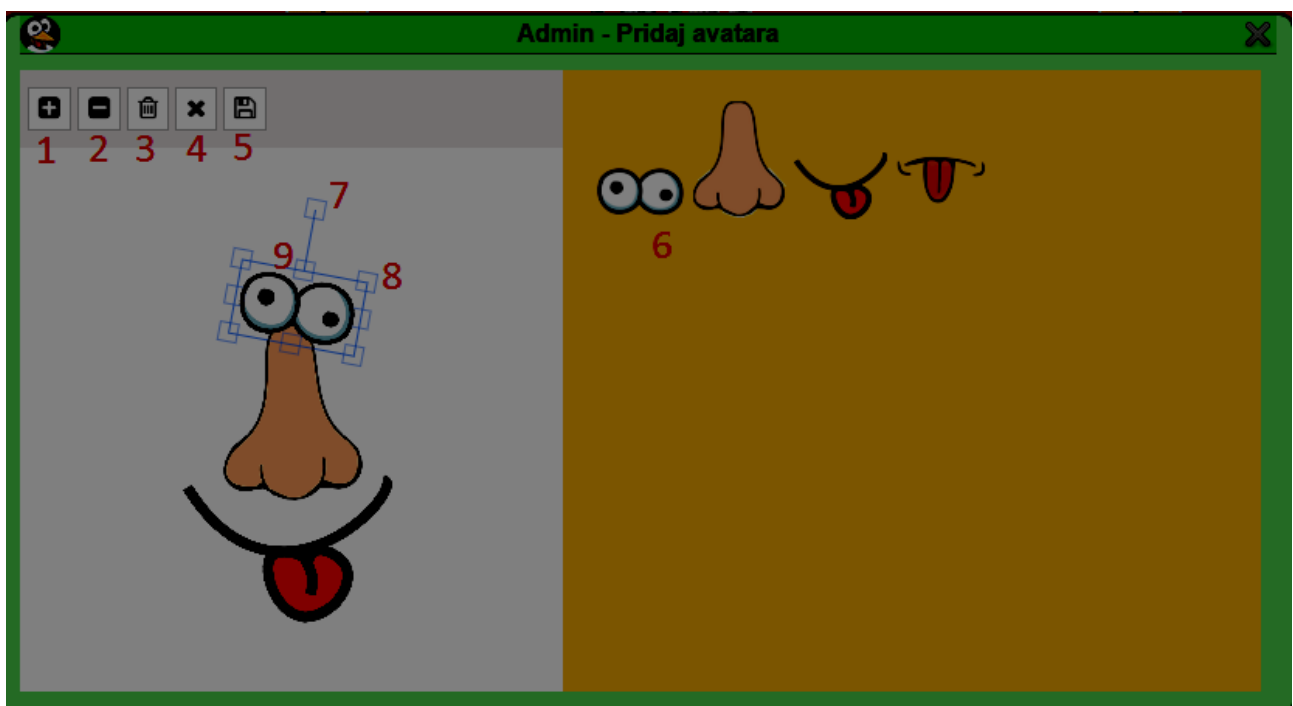
7.1.2 Vytvorenie avatara



Po kliknutí na tlačidlo "Pridaj avatara" sa otvorí okno (viď dole).



1. – Používateľ (dieťa) si zvolí názov avatara.
2. – Používateľ si vyberie jednu z tém zo zoznamu, ktorú bude predstavovať obrázok.
3. – Po kliknutí na tlačidlo “Pokračovať” otvorí sa nový okno s plochou na vyskladanie avatara (editor).



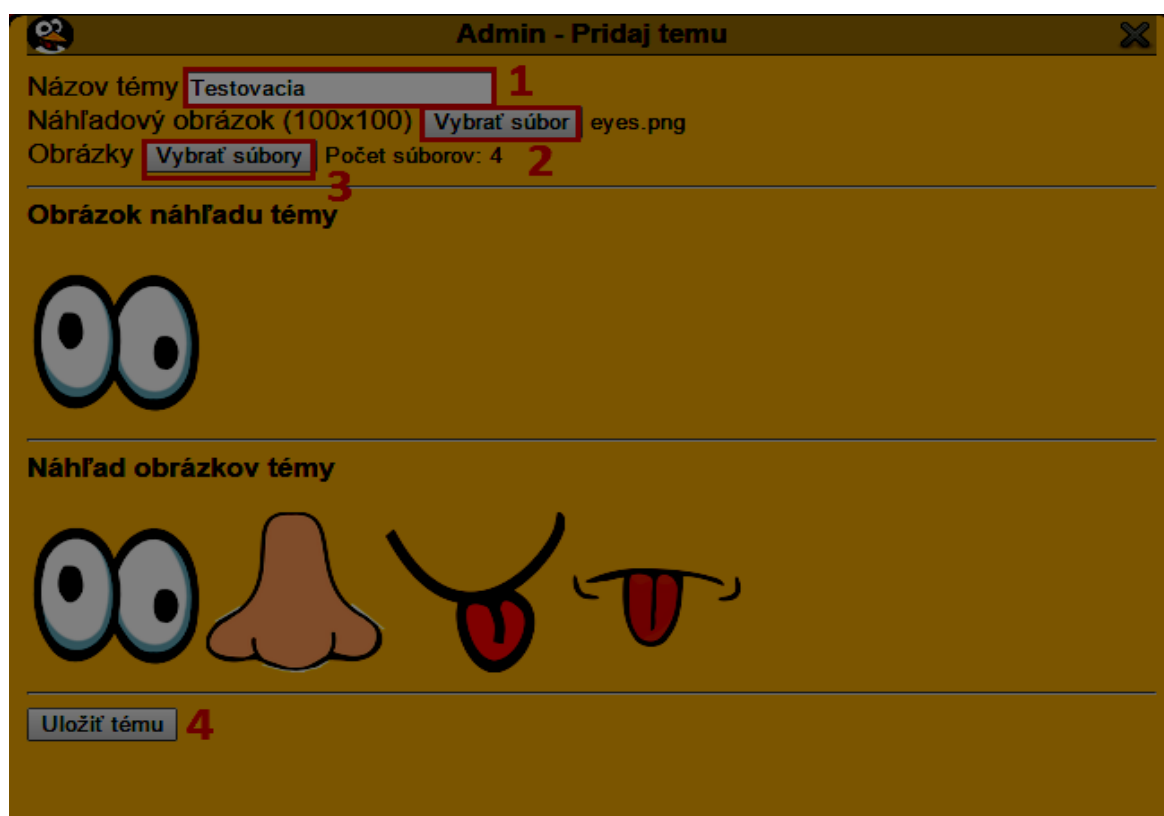
Lišta s nástrojmi na pracovanie s objektmi a grafickou plochou:

1. – Po kliknutí na tlačidlo “+” používateľ si zmení z-tovú súradnicu grafického objektu.
2. – Po kliknutí na tlačidlo “-” používateľ si zmení z-tovú súradnicu grafického objektu.
3. – Po kliknutí na tlačidlo používateľ si zmení z-tovú súradnicu grafického objektu.
4. – Tlačidlo na zmazanie vybratého objektu.
5. – Tlačidlo na zmazanie celého plochu.
6. – Používateľ si uloží avatara.
7. – Zoznam objektov. Po kliknutí na ikonku objektu object sa ukladáa na plochu.
8. – Tlačidlo na škálovanie objektu.
9. – Tlačidlo na posunutie objektu.

7.2 Rozhranie pre správcu aplikácie

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

7.2.1 Pridanie témy



1.–
Admi
nistr
átor
zadá
názo
v
témy
.
2. –
Po
klikn
utí
na
tlači
dlo

“Vybrať súbor” sa zobrazí okno so zoznamom tém.Vyberie vhodný náhľadový obrázok pre tému.

3. – Po kliknutí na tlačidlo “Vybrať súbor” sa zobrazí okienko na pridávanie objektov.Následne nahrá objekty k téme.

4. – Administrátor uloží tému.

8 Analýza technológií

8.1 *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.

8.2 *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

8.3 *Výber frameworkov*

8.3.1 *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

8.3.2 *Framework pre frontend*

Pre frontend nebude použitý žiaden framework.

8.4 Výber pomocných knižníc a pluginov

8.4.1 Pre frontend

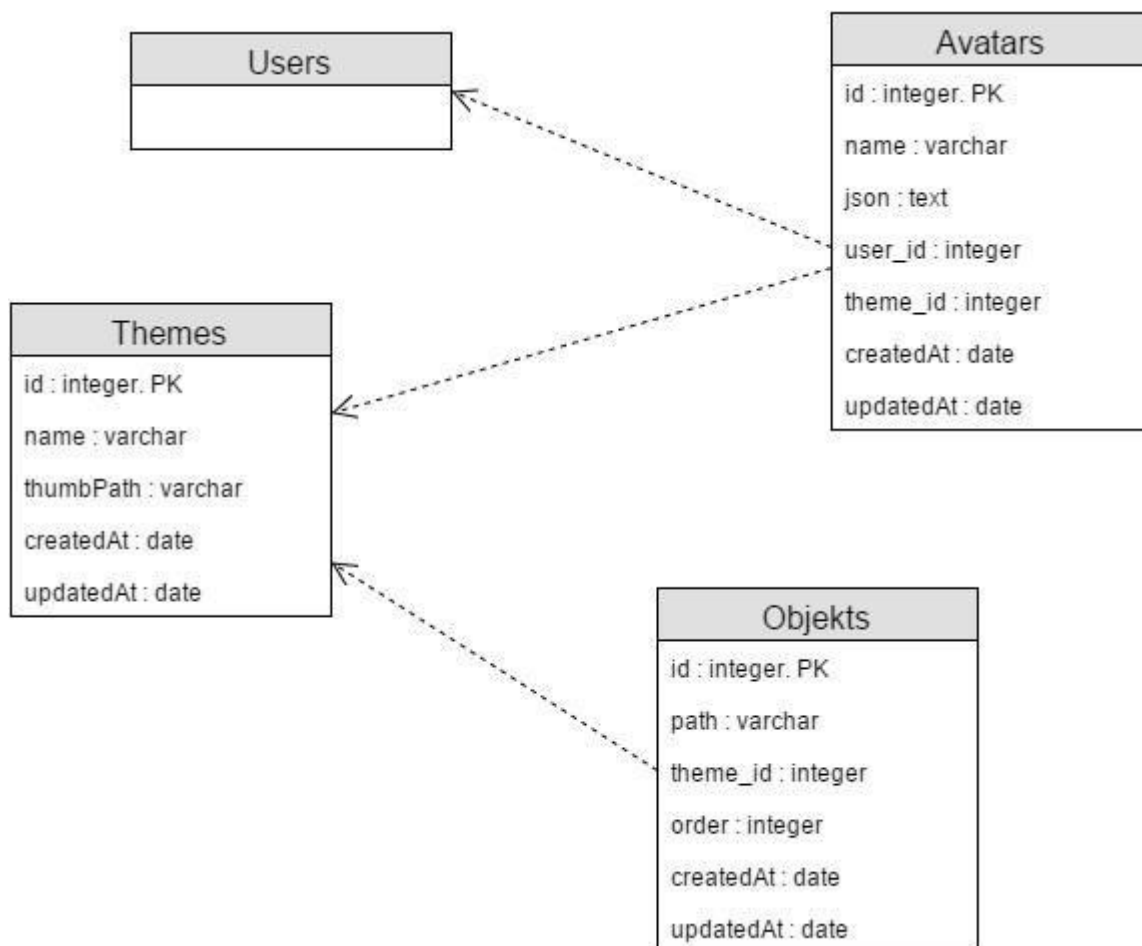
Pri tvorbe frontendu bude použitá knižnica Fabric.js, ktorá slúži na prácu s HTML5 canvasom. Hlavné výhody tejto knižnice:

- 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

8.4.2 Pre backend

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

9 Dátový model

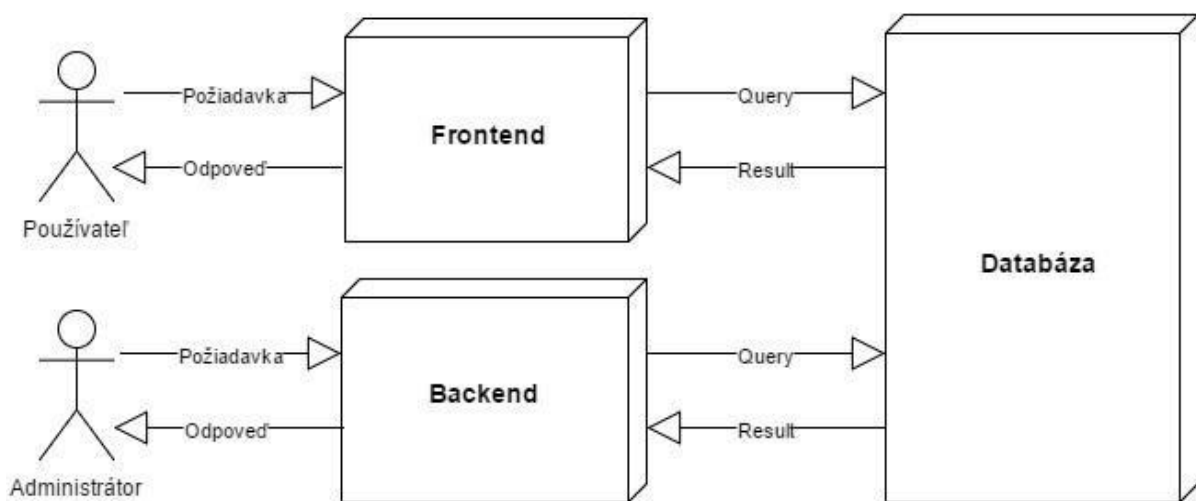


Vysvetlivky k obrázku 1:

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

10 Komponenty aplikácie

10.1 Schéma zapojenia komponentov

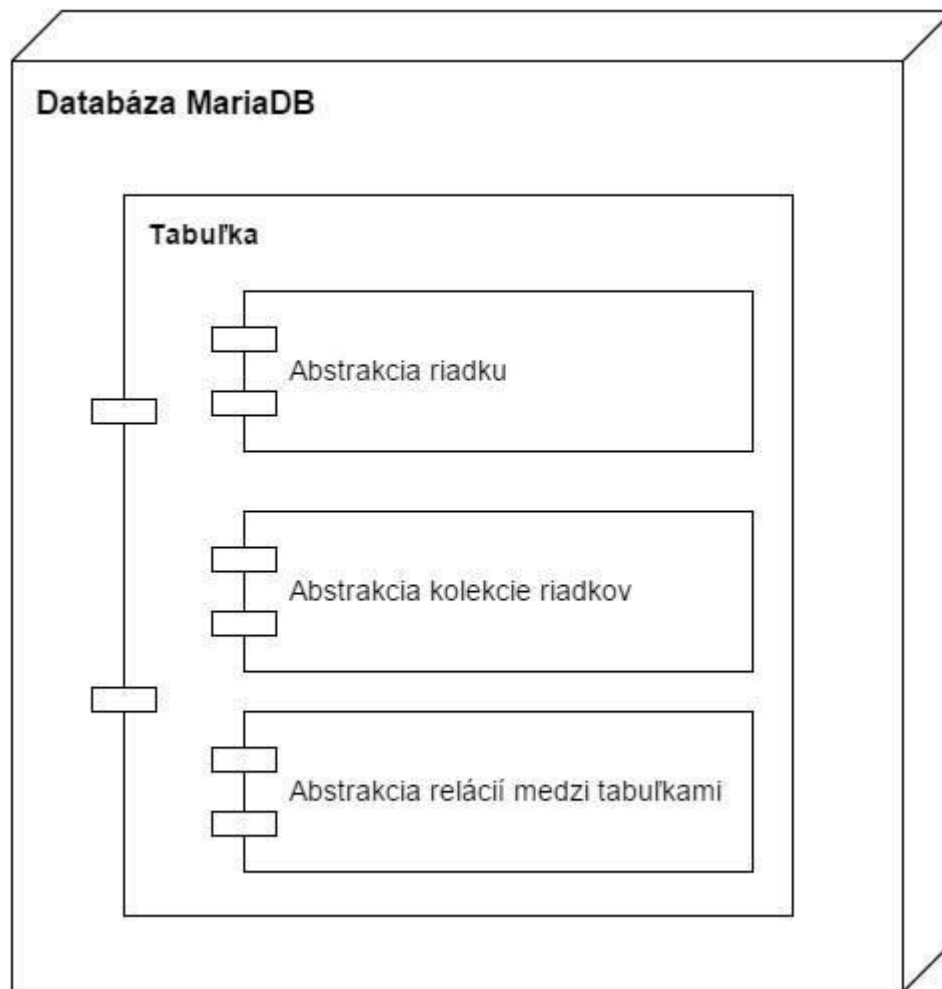


Obrázok 2: Schéma zapojenia komponentov

Používateľ vidí frontend uzol, ktorý v sebe zahŕňa ďalšie komponenty. Tento uzol komunikuje s databázovým uzlom, ktorý v sebe taktiež zahŕňa iné komponenty. Používateľ posiela 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.

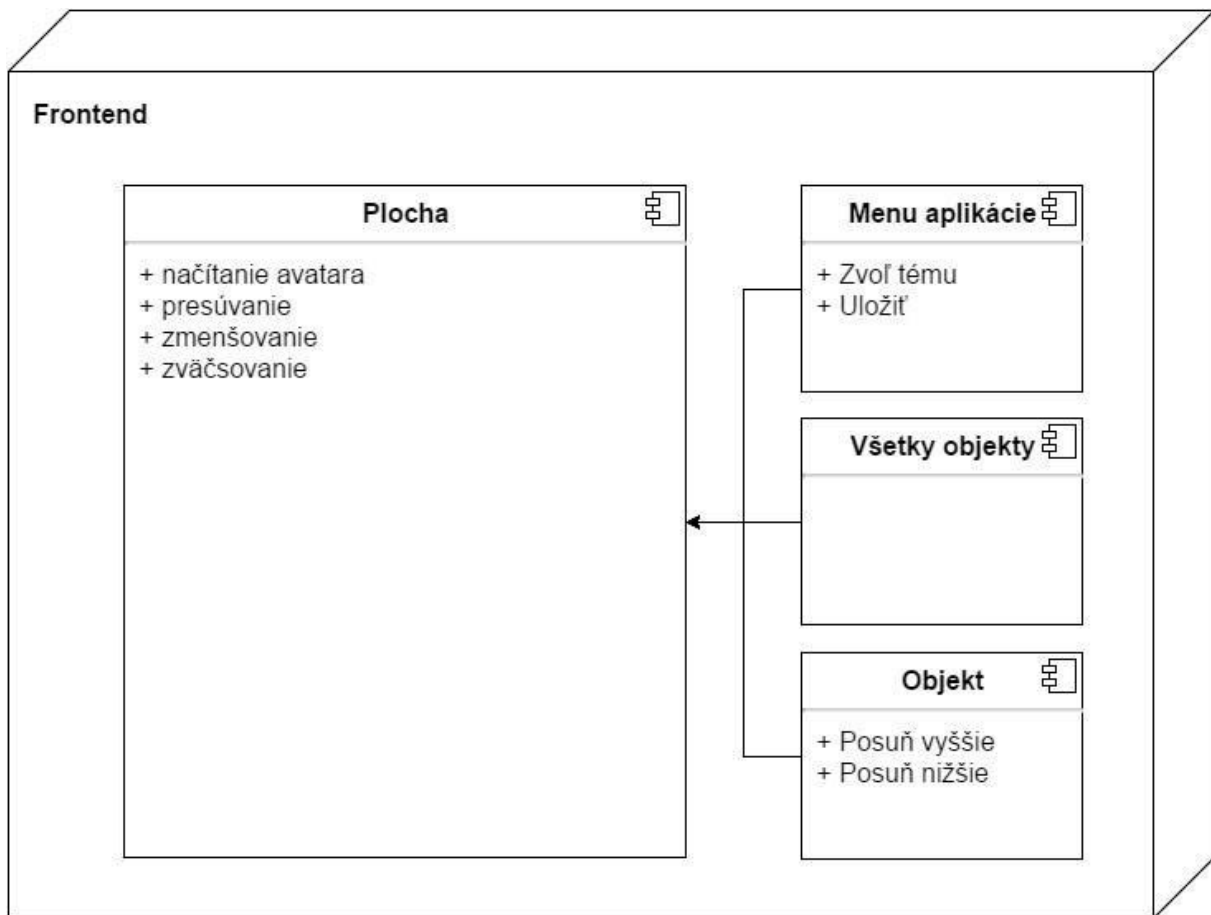
10.2 Databázový uzol



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

10.3 Frontendový uzol

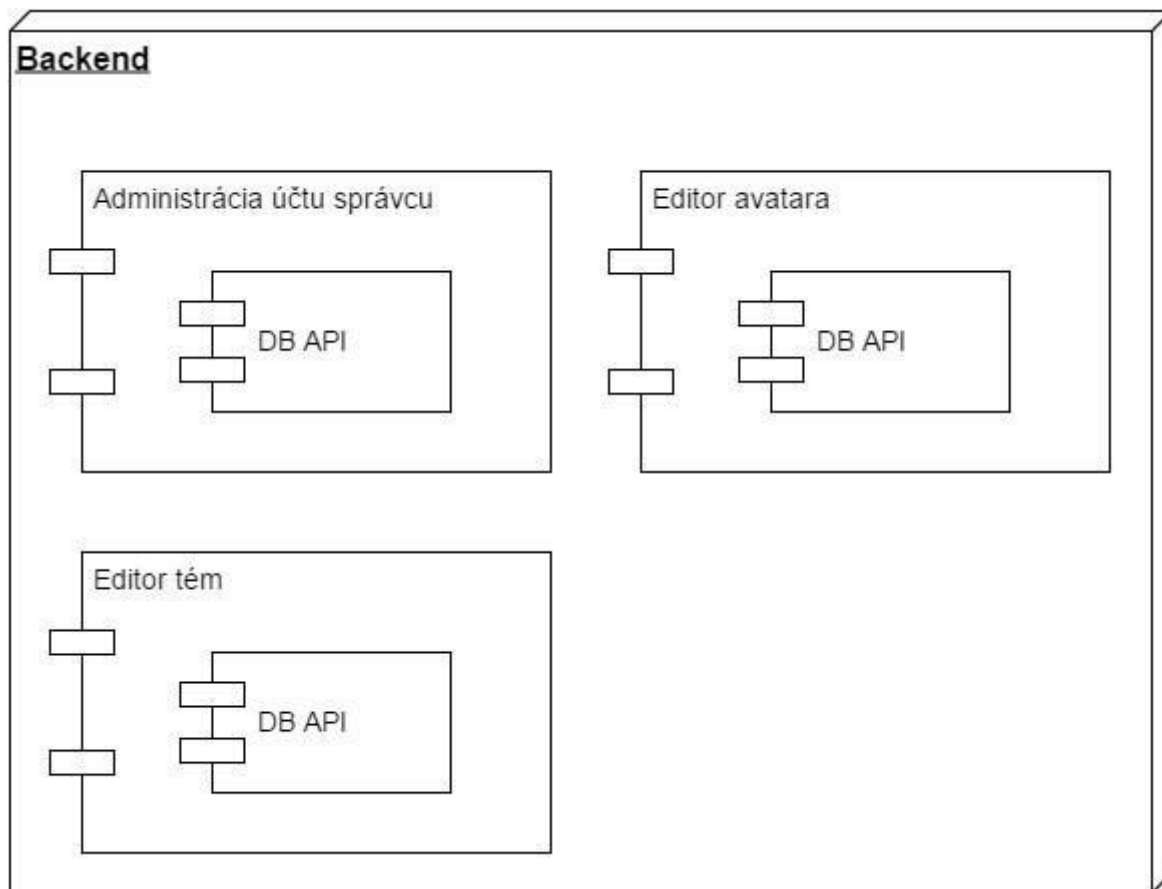


Obrázok zobrazuje uzol frontendových komponentov (komponent pre stránku zobrazenú návštevníkom)

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

10.4 Backendový uzol



Obrázok zobrazuje uzol komponentov pre backend (správčovskú 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

11 Návrh

11.1 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.

11.2 Návrh objektov

11.2.1 Návrh tabuľkových modelov

Na návrh tabuľkových modelov bude použitá knižnica Sequelize.js

11.2.1.1 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)

11.2.1.2 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)

11.2.1.3 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.

11.2.2 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.

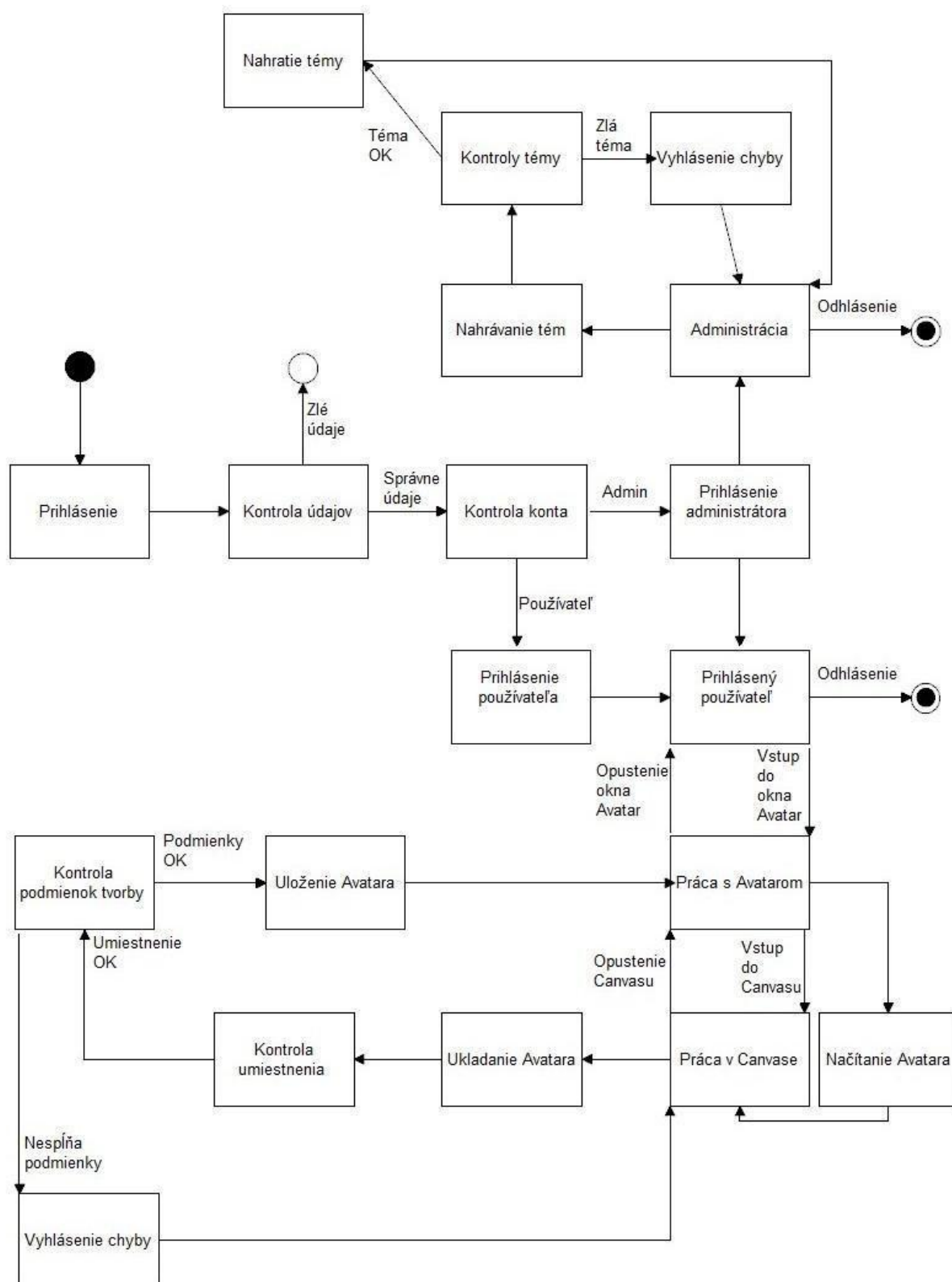
function getRandomColor() – Funkcia vráti náhodnú farbu v hexadecimálnom zápise. **function addTriangle()** – Funkcia pridá náhodný trojuholník do plochy.

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.

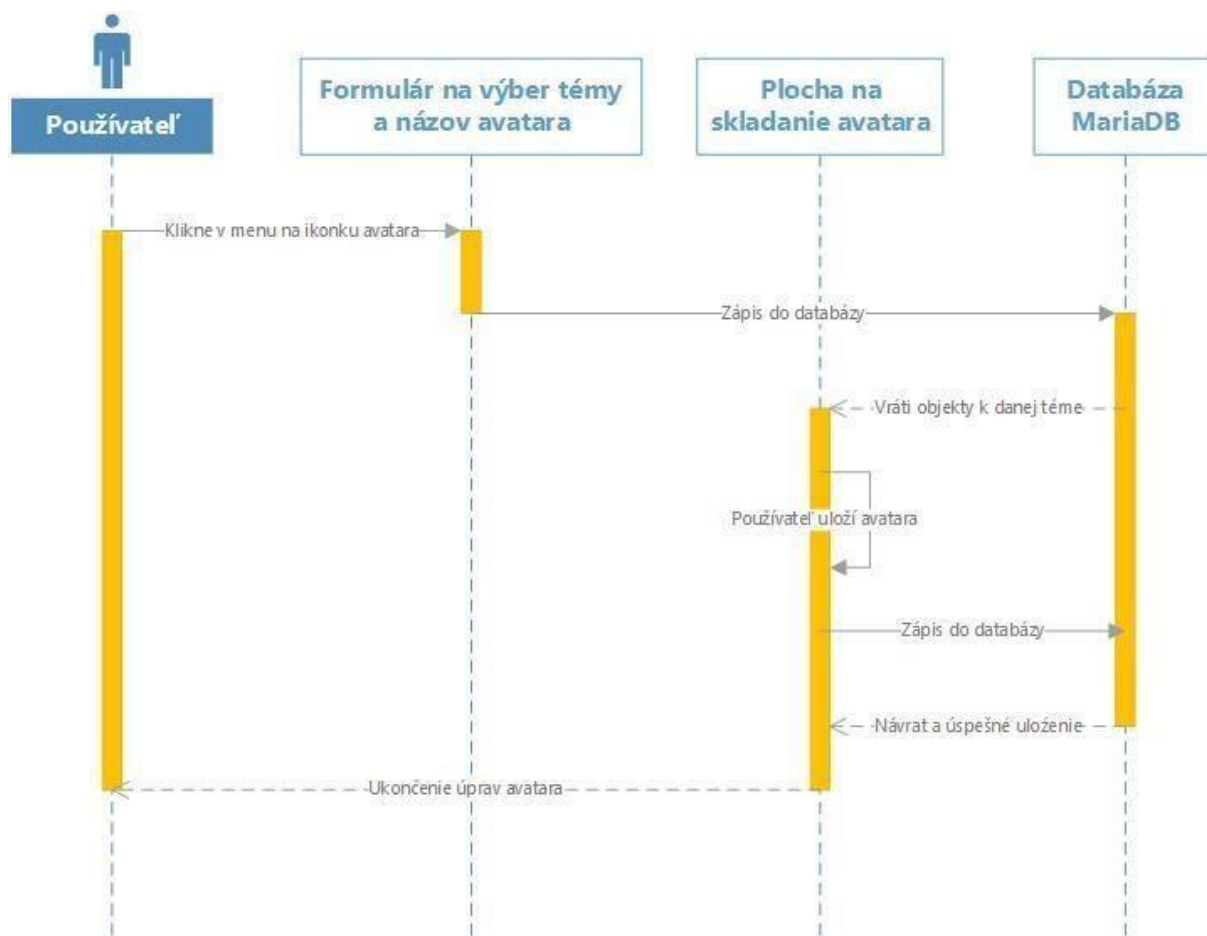
function sendBackwards() – Funkcia posunie označený objekt v ploche o úroveň nižšie.

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

11.3 Stavový diagram



11.4 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.

12 Testovanie

12.1 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 Avatarom	Systém po stlačení tlačidla graickej plochy pre Avatara otvorí nové okno s prvkami graická plocha, na ktorú sa dajú premiestňovať graické objekty z lišty, v ktorej sú zobrazené. Ďalej sa bude zobrazovať ovládací panel obsahujúci tlačidlá potrebné pre ovládanie udalosí objektov na graickej ploche	Systém korektné načítal všetky potrebné moduly	Stlačením tlačidla Avatar v panely aplikácii sa otvorí okno s požadovanými funkčnými prvkami na tvorbu Avatara	Passed	20.11.2015	Michal Rakovský	
2	Umiestňovanie graických objektov na plátno	Systém umiestni graické objekty z lišty objektov po interakcii užívateľom na graickú plochu na náhodné pozície	Je k dispozícii graická téma, obsahujúca graické objekty a okno Avatar sa kompletne načítalo	Keď užívateľ klikne na požadovaný objekt v lište graických objektov, tento objekt sa vytvorí ako nový objekt a systém mu nastaví súradnice z intervalu rozmerov graickej plochy. Tento objekt sa zobrazí na plátno	Passed	20.11.2015	Michal Rakovský	
3	Posun, rotácia a zmena veľkosti graických objektov na plátno	Systém umožňuje interaktívnu prácu z vytvorenými objektami na graickej ploche. Zmení ich atribúty, ako sú súradnice, uhol, veľkosť	Na graickej ploche sa nachádza graický objekt a okno Avatar sa kompletne načítalo	Používateľ klikne na požadovaný graický objekt na plátno, tento systém určí za označený, následne sa objektu sprístupní interaktívne rozhranie, ktoré zabezpečí prácu s jeho súradnicami, uhlom a veľkosťou pomocou udalosí riadených myšou, alebo iným zariadením umožňujúcim riadenie kurzora	Passed	20.11.2015	Michal Rakovský	
4	Výčistenie graickej plochy, odstraňovanie jednotlivých objektov	Systém pomocou prvkov ovládacieho panela graickej plochy zabezpečí odstraňovanie graických objektov jednotlivito a tiež hromadne každý objekt, ktorý plátno obsahuje	Odstraňovanie objektov graickej plochy z knižnice jabric.js zabezpečuje vhodné použitie tejto knižnice	Používateľ označí interakciou myši objekt graickej polohy. Stlačením tlačidla pre jednotlivé mazanie objektov sa zmaže objekt, ktorý je systémom určený ako označený objekt. Ak používateľ zvolí tlačidlo pre výčistenie plátna, systém vymaže akýkoľvek graický objekt nachádzajúci sa na danom plátno	Passed	20.11.2015	Michal Rakovský	
5	Zmena úrovni vrstiev objektov na plátno	Systém umožňuje zmenu vrstvy daného objektu pomocou prvkov na ovládacom paneli graickej plochy		Používateľ kliknúm myši označí graický objekt na plátno, tento sa nastaví systémom za označený. Pomocou tlačidiel je možné upravovať úrovne jeho vrstvy a teda je možné ním prekryvať iné objekty, alebo ho iným objektom prekryť	Passed	20.11.2015	Michal Rakovský	
6	Uloženie graických objektov Avatara do databázy	Systém uloží všetky graické prvky v podobe serializácie objektov typu JSON do databázy	Graické 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	18.12.2015	Michal Rakovský	
7	Načítanie uloženého Avatara do graickej plochy	Systém z databázy načíta serializované objekty, preloží ich do objektov graickej plochy a následne zobrazí na graickej ploche	V databáze sa nachádza aspoň jeden objekt Avatar pre daného používateľa	Používateľ zvolí zo zoznamu Avatarov kliknúm 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 graických objektov, ktoré následne vykreslí na plátno	Passed	18.12.2015	Michal Rakovský	
8	Úprava atribútov graických objektov načítaného Avatara	Systém umožňuje upravovať uloženého Avatara rovnakými prvkami, ktoré využíva pri vytváraní Avatara	Avatar bol úspešne načítaný a graické objekty sú vykreslené na plátno	Používateľ klikne na požadovaný graický objekt na plátno, tento systém určí za označený, následne sa objektu sprístupní interaktívne rozhranie, ktoré zabezpečí prácu s jeho súradnicami, uhlom a veľkosťou pomocou udalosí riadených myšou, alebo iným zariadením umožňujúcim riadenie kurzora	Passed	18.12.2015	Michal Rakovský	

9	Načítanie graickej témy pri tvorbe nového Avatara	Systém umožňuje tvorbu Avatara s využitím tém z databázy	V databáze sa nachádza aspoň jedna graická téma	Používateľ si kliknúm myši zvolí danú tému, ktorú bude používať na vytváranie Avatara, systém ju určí ako označenú. Následným kliknúm na tlačidlo načítania témy ju systém načíta a otvorí okno s graickou plochou, a ďalšími prvkami tohto okna a v lište graickej témy sa nachádzajú objekty zvolenej témy	Passed	18.12.2015	Michal Rakovský	
---	---	--	---	--	--------	------------	-----------------	--

12.2

12.3 Testovacie scenáre – admin interface

Test ID	Popis	Očakávaný výsledok	Predpoklady	Kroky k vykonaniu požiadavky	Status	Dátum ukončenia	Tester	Komentáre
1	Načítanie okna pre administráciu - vytvorenie témy	Systém po stlačení tlačidla graickej 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	
2	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	
3	Uloženie témy do databázy	Systém ukladá objekty témy z pamäte do databázy	V pamäi systému sa nachádza dostatočný počet obrázkov	Administrátor kliknúm na tlačidlo nahraia 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	
4	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 graická 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	
5	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 graická 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, kliknú tlačidla na odstránenie systém túto tému odstráni z databázy a takiež odstráni celý jej súborový priečinok na disku	Passed	20.1.2016	Michal Rakovský	

13 Záznam z odovzdávania a predvedenia výslednej aplikácie zadávateľovi

13.1 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.

13.2 Priebeh stretnutia

Prezentáciu viedol Michal Piják, a prezentoval to na jednom z našich počítačov. Zadávatelia 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.

14 Zhodnotenie

14.1 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 problém pri ukladaní už vyskladaného postavičku. Tieto problémy sa podarilo vyriešiť pomocou materiálov nájdených na internete a tímovej spolupráce vyriešiť.

14.2 Zmeny do ďalších verzií

14.3 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.

14.4 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.

14.5 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.

15 Zdroje

[1] KTuberling –www.kde.org/applications/games/ktuberling/

[2] NodeJS – webový server napísaný v JavaScripte www.nodejs.org

[3] MariaDB – SQL relačná databáza www.mariadb.org

[4] FabricJS – JavaScriptová knižnica na prácu s html5 canvasom www.fabricjs.com