

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

**ANALÝZA TECHNOLOGIÍ, DEKOMPOZÍCIA  
A DÁTOVÝ MODEL**  
VYSKLADAJ SI AVATARA

zimný semester 2015/2016

Michal Piják

Károly Belokostolský

Michal Rakovský

János Rosztovics

# Obsah

1	Analýza technológií .....	3
1.1	Výber programovacích jazykov pre serverom vykonávané skripty.....	3
1.2	Výber programovacieho jazyka pre klientom vykonávané skripty .....	3
1.3	Výber frameworkov.....	3
1.3.1	Framework pre backend.....	3
1.3.2	Framework pre frontend.....	3
1.4	Výber pomocných knižníc a pluginov .....	4
1.4.1	Pre frontend .....	4
1.4.2	Pre backend .....	4
2	Dátový model .....	4
3	Komponenty aplikácie .....	5
3.1	Schéma zapojenia komponentov .....	5
3.2	Databázový uzol .....	5
3.3	Frontendový uzol.....	6
3.4	Backendový uzol.....	7

# **1 Analýza technológií**

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

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

## **1.2 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

## **1.3 Výber frameworkov**

### **1.3.1 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

### **1.3.2 Framework pre frontend**

Pre frontend nebude použitý žiaden framewok.

## 1.4 Výber pomocných knižníc a pluginov

### 1.4.1 Pre frontend

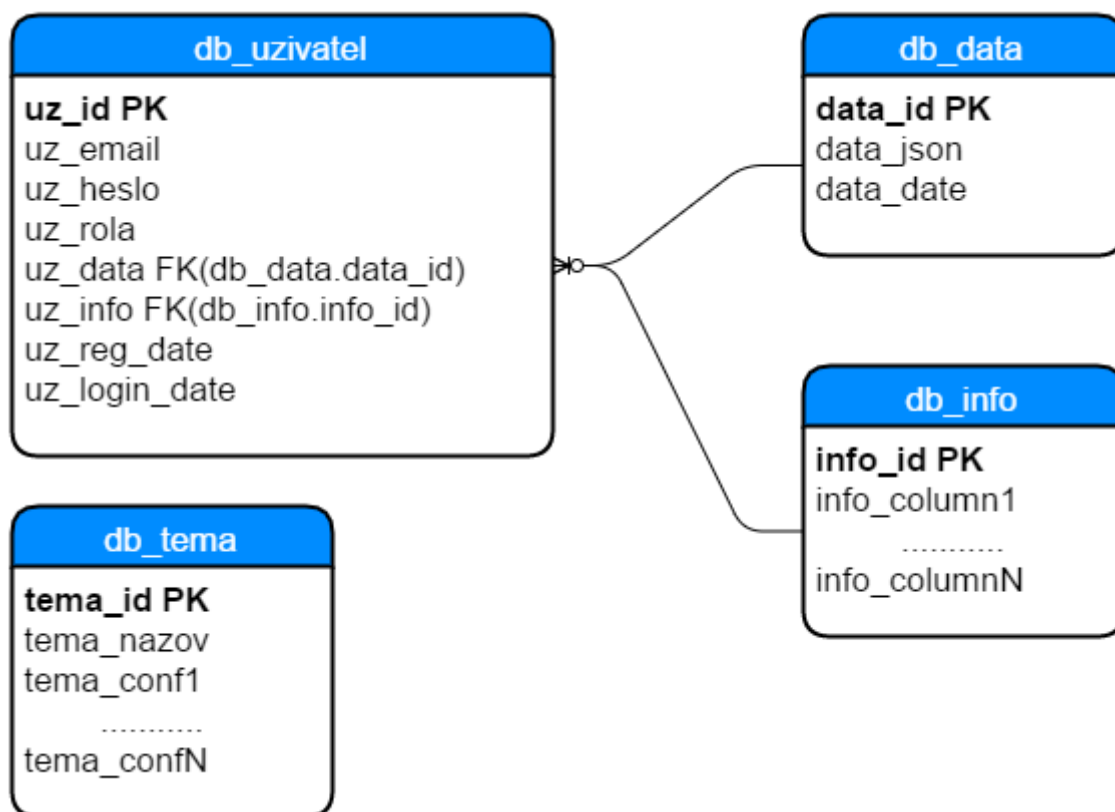
Pri tvorbe frontendu bude použitá knižnica Fabric.js, ktorá slúži na prácu s HTML5 canvasom. Hlavné výhody tejto knižnice:

- 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

### 1.4.2 Pre backend

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

## 2 Dátový model



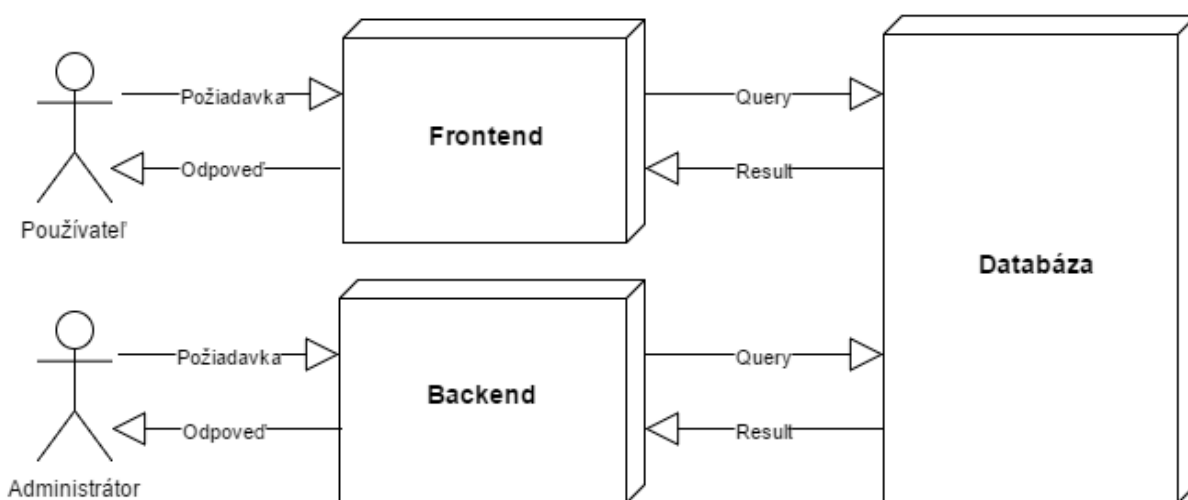
Obrázok 1: databázový model

Vysvetlivky k obrázku 1:

- PK označuje primárny kľúč tabuľky
- V zátvorkách sú označené tabuľky ktoré tvoria medzi sebou relácie

## 3 Komponenty aplikácie

### 3.1 Schéma zapojenia komponentov

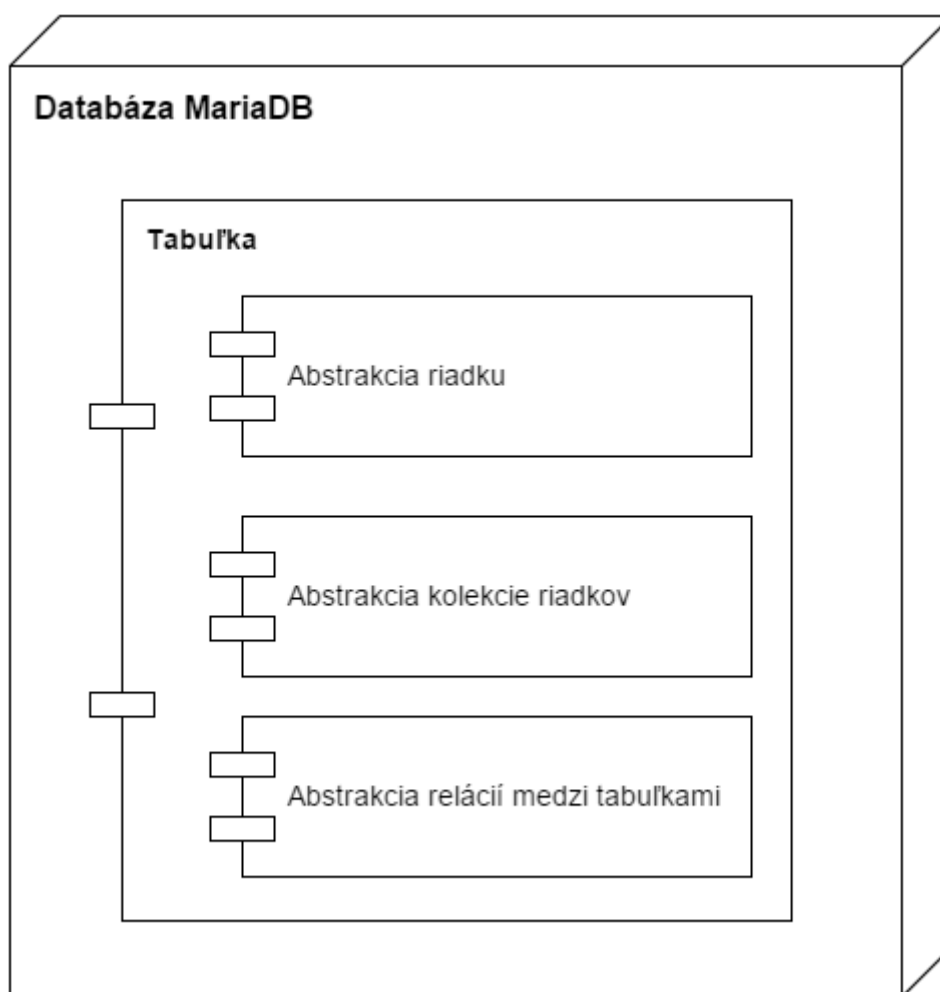


Obrázok 2: Schéma zapojenia komponentov

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.

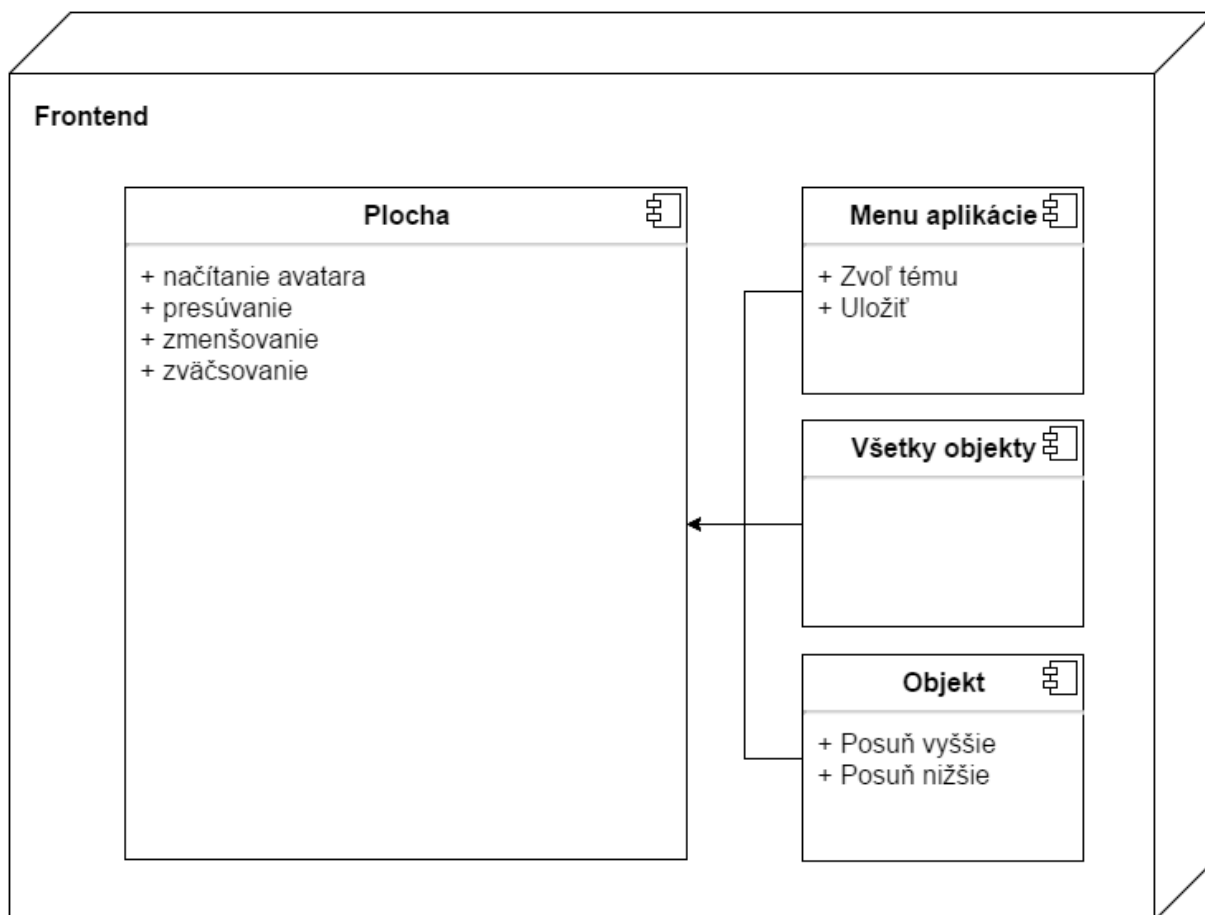
### 3.2 Databázový uzol



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

### 3.3 Frontendový uzol



### 3.4 Backendový uzol

