

Návrh

Headis

Tvorba informačných systémov ZS 2018/19

Tím Hájnikova žena

Matej Rychtárik
rychtovy@gmail.com

Tomáš Velich
feldatv@gmail.com

Marián Bagyanszký
bagyanszky1@uniba.sk

Balázs Horváth
hbalazs09@gmail.com

20. decembra 2018

Obsah

1	Úvod	1
1.1	Účel dokumentu	1
2	Podrobná špecifikácia vonkajších interfejsov	1
3	Používané technológie	1
3.1	HTML/CSS	1
3.2	PHP/MySQL	1
3.3	Laravel	1
3.4	JavaScript	1
3.4.1	Vue.js	1
3.4.2	Axios	1
3.5	CKEditor	2
4	Podrobný dátový model perzistentných údajov atď.	2
4.1	Databázový model	2
4.2	Komunikačné protokoly	2
5	Návrh používateľského rozhrania	3
6	Návrh implementácie	5
6.1	UML - State diagram	5
6.2	UML - Triedny diagram	6
6.3	UML - Diagram komponentov	7

1 Úvod

1.1 Účel dokumentu

Tento dokument slúži ako návrh informačného systému Headis a je určený pre vývojárov systému. Dokument dôkladne popisuje funkcie a metódy informačného systému a podáva návrh na implementáciu. Všetky obrázky v dokumente sú pridané do priečinka "prilohy".

2 Podrobná špecifikácia vonkajších interfejsov

Aplikácia bude bežať na serveri a komunikovať s MySQL databázovým serverom, kde bude uložený celý obsah aplikácie.

3 Používané technológie

3.1 HTML/CSS

Tieto technológie budeme používať na vytvorenie celého používateľského rozhrania aplikácie.

3.2 PHP/MySQL

V MySQL databáze bude uložený celý obsah aplikácie, čo sú dáta o používateľoch, zápasoch, výzvach a všetky pomocné údaje. PHP bude komunikovať s MySQL a pracovať s dátami v databáze.

3.3 Laravel

Laravel je open source PHP framework pre webové aplikácie. Aplikácia bude postavená na tomto frameworku. Bude slúžiť na vytvorenie celej aplikácie, prácu s dátami, vytvorenie testov.

3.4 JavaScript

Táto technológia umožňuje dynamicky aktualizovať obsah aplikácie na strane klienta.

3.4.1 Vue.js

Framework pre JavaScript. Pomocou Vue.js bude vykreslená "rebríčková pyramída", kalendár dostupných dní pre zápasy a pre komunikáciu vo výzvach.

3.4.2 Axios

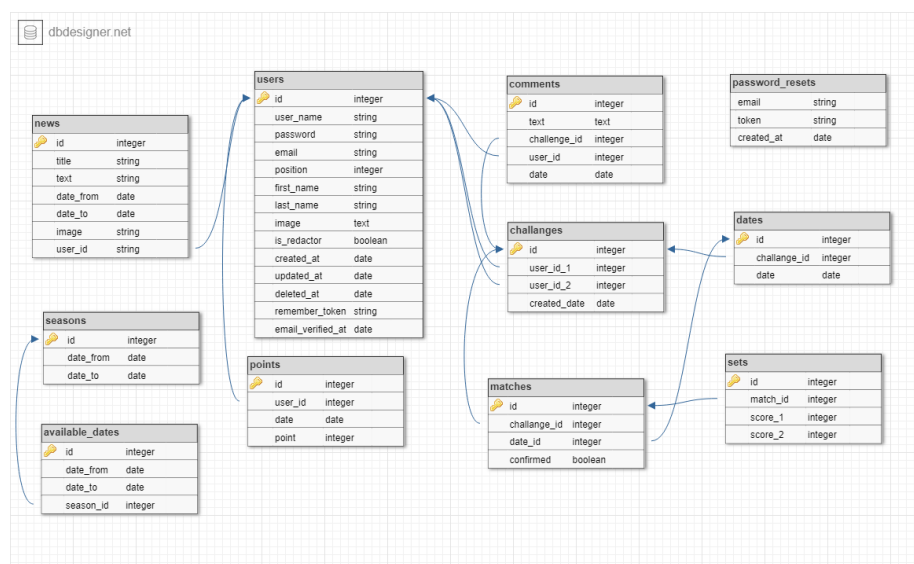
Táto technológia bude použitá na posielanie http requestov na server.

3.5 CKEditor

Bude použitá na formátovanie textu noviniek na hlavnej stránke a bude len pre redaktora.

4 Podrobný dátový model perzistentných údajov atď.

4.1 Databázový model



Obr. 1: Databázový model

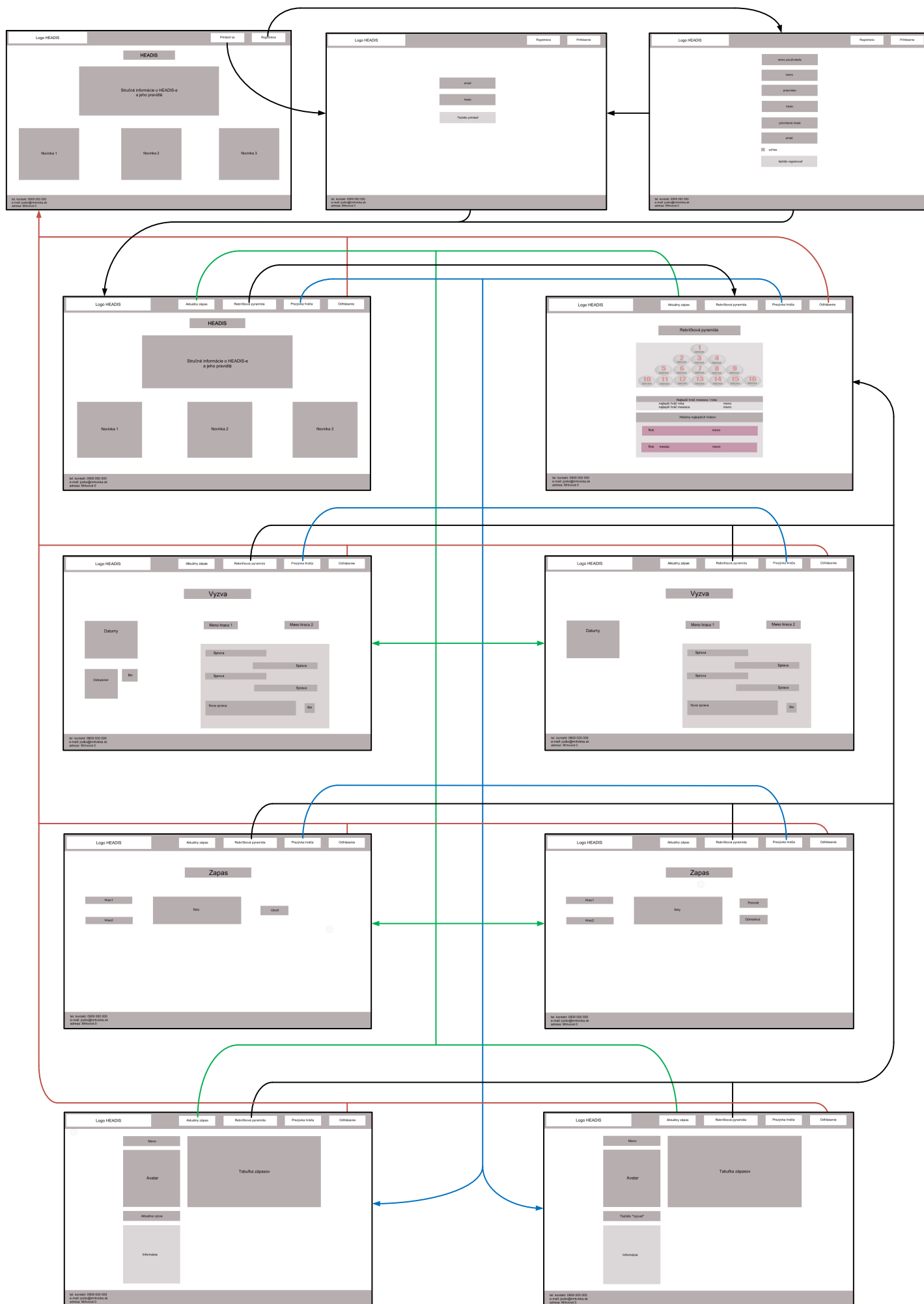
4.2 Komunikačné protokoly

Aplikácia využíva http requesty so základnými metódami GET a POST na prechod medzi jednotlivými podstránkami, taktiež využíva Ajax requesty s metódami GET a POST, pri ktorých nie je nutné znova načítať stránku.

5 Návrh používateľského rozhrania

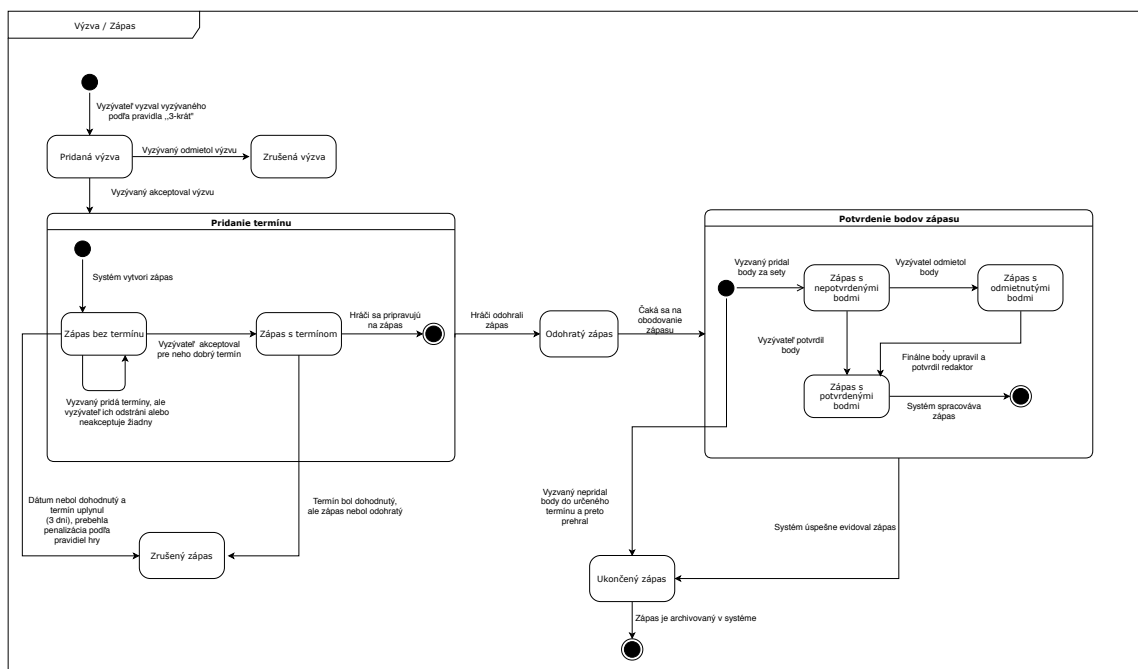
Návrh používateľského rozhrania sa nachádza na nasledujúcej strane.

Návrh používateľského rozhrania



6 Návrh implementácie

6.1 UML - State diagram

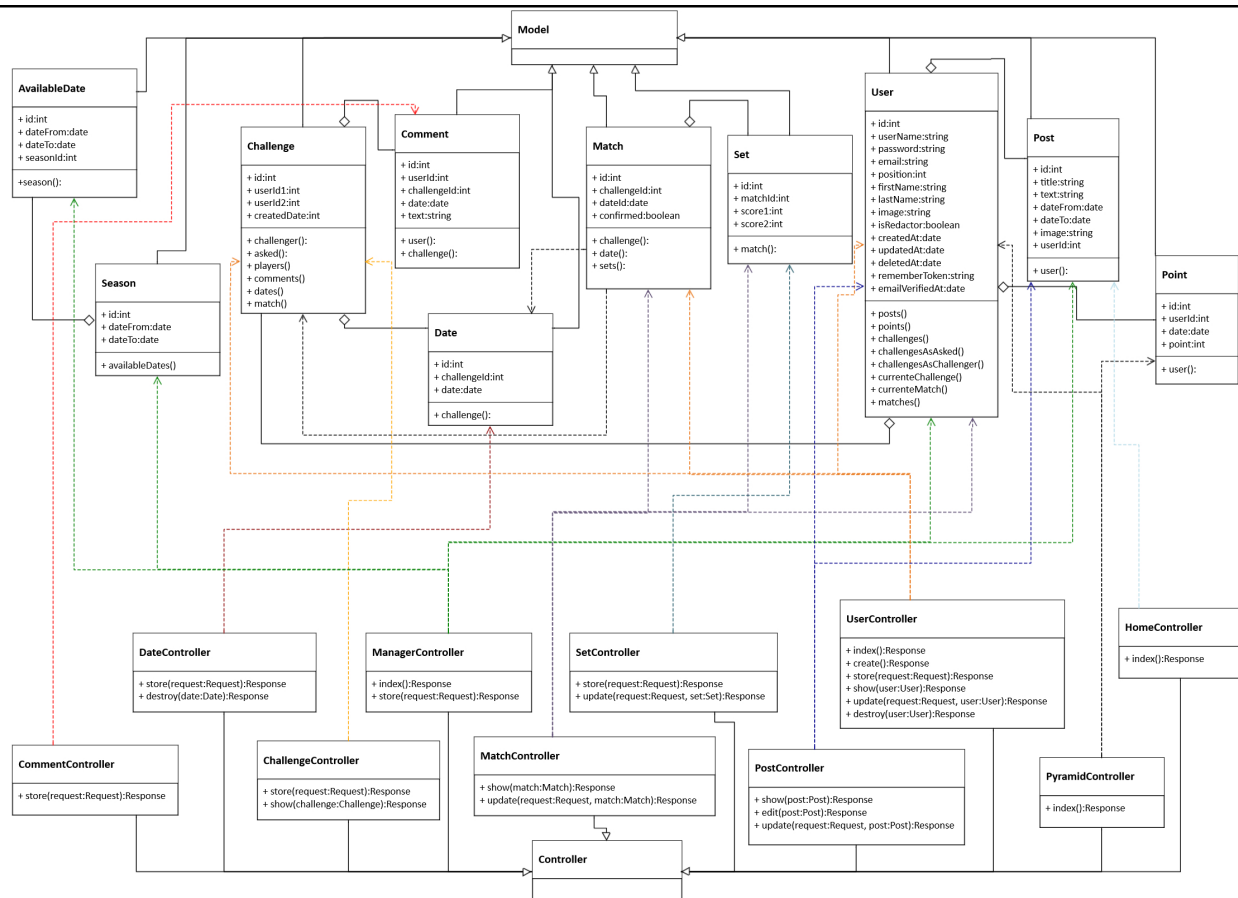


Obr. 2: Stavový diagram

Popis:

Stavový diagram reprezentuje životný cyklus entity zápas. Každý stav entity je vyjadrený oválom. Vstup entity do stavového diagramu sa realizuje cez počiatkový stav(kruh) a výstup väčšinou cez koncový stav (kruh v kružnici). Prechody medzi stavmi vyvolajú používatelia a pravidlá aplikácie.

6.2 UML - Triedny diagram



Obr. 3: Triedny diagram

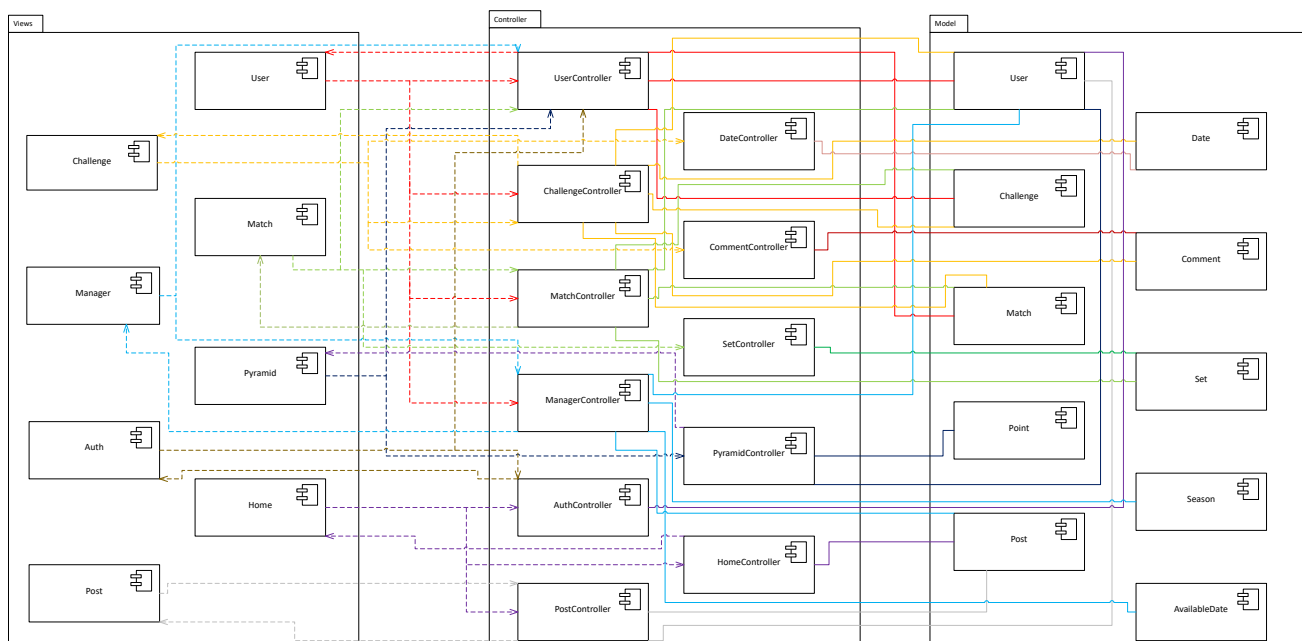
Popis:

Triedny diagram popisuje vzťahy medzi triedami.

Z abstraktnej triedy Model dedia všetky triedy v danom systéme, ktoré sú zodpovedné za správu údajov v aplikácii a prijímajú používateľský vstup od kontroleru.

Z abstraktnej triedy Controller dedia všetky triedy v danom systéme, ktoré reagujú na používateľský vstup, overujú ho a odovzdávajú vstup do modelu.

6.3 UML - Diagram komponentov



Obr. 4: Diagram komponentov

Popis:

Model je zodpovedný za správu údajov aplikácie. Dostáva používateľské vstupy od controlleru.

View reprezentuje údaje z daného modelu ako výstup vo formáte html/css.

Controller dostáva vstupy od používateľa a premieňa ich na príkazy pre model alebo view.