

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

NÁVRH

WEBOVÁ APLIKÁCIA PRE TURISTICKÚ SKUPINU

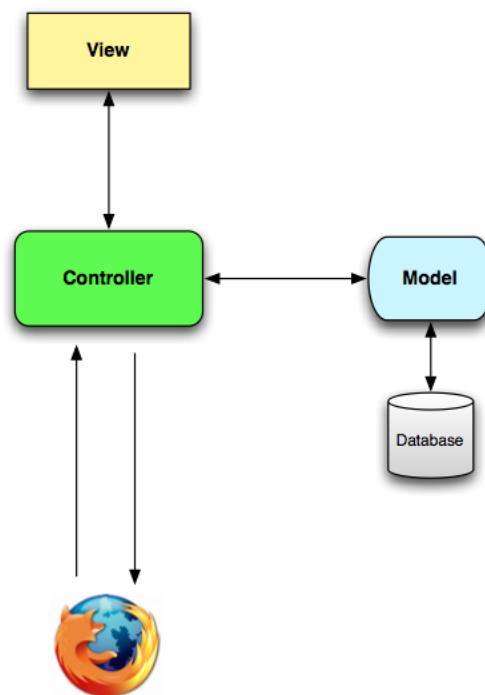
2015/2016
Kristína Miklošová
Juraj Mašlej
Juraj Košťál
Roman Brojo

1. Úvod

Na naprogramovanie použijeme framework Django. Django podporuje štandardnú MVC architektúru. Návrh tried bude preto ovplyvnený touto architektúrou.

2. Model-View-Controller (MVC)

- **Model** - reprezentuje dátovú štruktúru stránok. Triedy vrstvy Modelu obsahujú zvyčajne metódy pre prístup k dátam, ktoré sú uložené v databáze.
- **View** - sú informácie, ktoré dostáva z controllera a sú prezentované užívateľovi. Nemá priamy prístup k dátam.
- **Controller** - slúži ako prostredník a sprostredkúva premostenie medzi Modelom a View a tiež medzi ostatnými zdrojmi potrebnými.



Obr.1 diagram architektúry MVC

2.1. Model

Prvým krokom pri vytváraní databázovej Django aplikácie je definícia modelov – t.j. štruktúry databázy s dodatočnými metadátami. V našom projekte bude preto jeden model pre každú tabuľku. Napríklad definujeme triedu Zápis, ktorá bude dediť z triedy model. Model bude predstavovať jednu tabuľku v databáze. V tejto triede sa ďalej zdefinujú triedne premenné, ktoré budú predstavovať jeden riadok v tabuľke. Tieto triedne premenné budú

odkazovať na jednotlivé triedy typu Field napr. CharField (znakové pole) alebo DateField (datum a čas).

Pomenovaním jednotlivých polí Field chceme definovať tzv. “strojové” (machine-friendly) meno. Cez ne sa bude v Pythone pristupovať k hodnote poľa a v databáze bude využité pre pomenovanie stĺpca. Pokiaľ by sme chceli, môžeme pomocou prvého argumentu vo Field vytvoriť “ľudský čitateľný” (human-readable) meno. Ak ho ne zadáme, Django si ho odvodí zo strojového mena. Niektoré Field triedy majú povinné parametre, napr. CharField vyžaduje špecifikáciu max_length (maximálna dĺžka reťazca). Django cez Triedu podporuje takisto všetky obvyklé databázové relácie: N (many-to-many), 1:N (many-to-one) a 1:1 (one-to-one).

Čiže premenná Zapis bude obsahovať triedne premenné:

1. id – vygeneruje sa automaticky, toto správanie sa dá zmeniť
2. nazov – typu CharField
3. datum – typu DateField
4. kraj – typu CharField
5. ucastnici – typu OneToManyField, kde ako parameter bude trieda Zapis_uzivatel
6. autor – typu OneToOneField, kde ako parameter bude trieda Uzivatel
7. vrstvy – typu OneToManyField, kde ako parameter bude trieda Zapis_vrstva
8. textovy_popis – typu TextField
9. gps_poloha - typu PointField

Zvyšné triedy budú definované podobným spôsobom. V Triede je vhodné dodefinovať rôzne metódy, ktoré využijeme pri tvorbe views. Napr. pri pomocnom vypisovaní funkciou print (objekt) nám bude funkcia vracať výraz <Zapis: Zapis object>. Preto je vhodné dodefinovať špeciálnu metódu __str__, ktorá bude vracať vhodnejšiu reprezentáciu objektu. Napr. pre Triedu Zapis, by funkcia return mohla vracať: nazov+“ „+ datum.

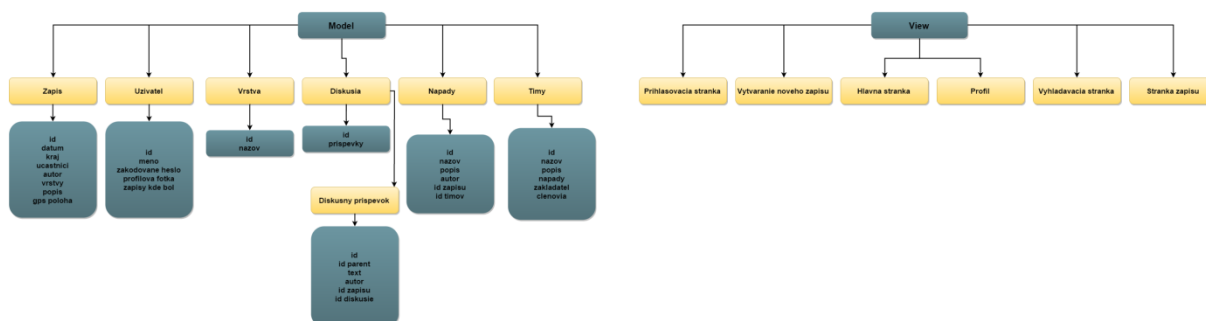
2.2. Controller

V prípade Django „controller“ je frameworkom ako takým: mechanizmus, ktorý posielá požiadavky určenému view, podľa Django URL konfigurácie. Preto sa triedy Controller-u v projekte nebudú vytvárať.

2.3. Views

View je “druh” webovej stránky, ktorá má v našej aplikácii určitú úlohu a výsledok svojej činnosti prezentuje prostredníctvom šablóny. V Django je každé view zrealizované jednoduchou funkciou v Pythone. V našej aplikácii bude pre prehľadnosť každé view osobitná trieda (dediť bude z View). V tejto triede bude zadaná špeciálna funkcia get, ktorá ako parameter bude dostávať HTTP request a bude vracať HTTP response. Django obsahuje takisto šablónový systém. Vo views odovzdá kontext, čo je slovník, ktorý mapuje šablónové premenné na objekty Pythonu. Vo views sa takisto môžu objaviť základné prvky Pythonu ako for loop alebo if statement. V našej aplikácii bude každá url adresa so stránkou predstavovať 1 triedu typu View.

2.4. Triedny diagram



2.5. Dizajn šablón

Vďaka Django šablónovému systému bude vytvorená len základná stránka base.html. Ostatné stránky následne budú „dediť“ jej jednotlivé časti pričom v prípade potreby budú prepisovať tieto časti (platí pre všetky šablóny okrem login.html a logout.html).

3. Funkčné celky systému

3.1. LogIn/ LogOut

Budú využívať Django’s authentication system. Je to systém, ktorý slúži na overenie prihlasovania a odhlasovania užívateľov. Takisto registráciu užívateľov. Tento systém budú využívať aj iné triedy, napr. trieda Mapa zavola svoj template a v prípade ak je používateľ prihlásený.

3.2. Mapa

(Mapa na ktorej budú zaznamenané jednotlivé zápisky, označovať ich v nej budú samotní používatelia).

Na zobrazenie jednotlivých geografických dát využijeme Mapy.cz od spoločnosti Seznam.cz. Každý zápis na mape si bude pamätať id jednotlivých zápiskov. Na prepojenie mapy aplikácie využijeme voľne dostupné Mapy Api.

3.3. ZapisDetail

Po kliknutí na zápis sa vygeneruje šablóna na základe id hodnoty. Natiahnu sa príslušné údaje z databázy. Každý zápis bude takisto mať prepojenie na diskusiu pod zápisom (prepojenie s databázou DiskusnyPrispevok). Fotoalbum, videoalbum a slide show budú vytvorené v programovacom jazyku PHP.

3.4. ZapisVytvorenie

Pomocou html formulára bude možné pridávať údaje (zápis) do databázy. Vstupy budú vhodne ošetrované a v prípade nevhodného vstupu sa vypíše príslušná správa.

3.5. Napad,Team

Rovnako sa bude postupovať aj pri vytvorení Nápadu a Tímu. Formuláre pre vyplnenie Nápadu a Tímu nebudú povinné.

3.6. ZapisVyhľadavanie

Táto trieda bude slúžiť na vyhľadávanie dát (zápisov) z databázy pomocou SQL jazyka na základe html formulárov. Takisto bude zobrazovať výsledky s možnosťou prepojenia na konkrétny zápis. Jej funkcionality bude využívať aj trieda Mapa.

3.7. Profil

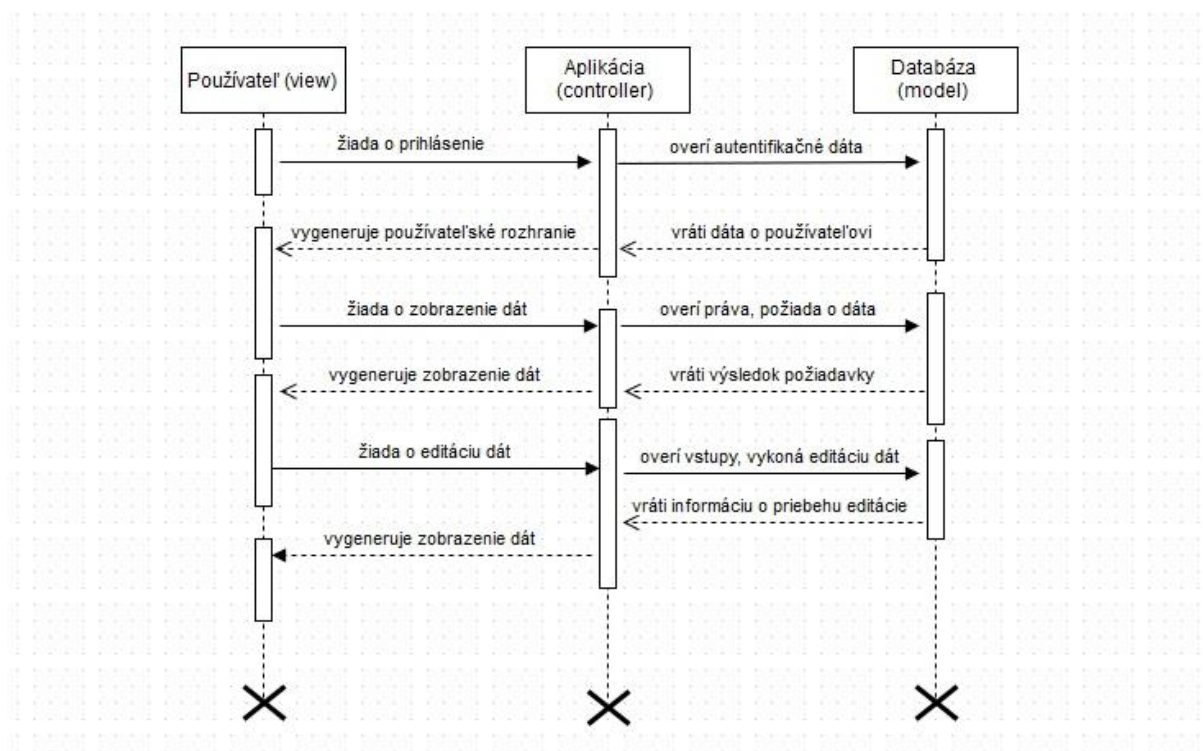
Cela funkcionality tejto triedy bude zabezpečená Django's authentication system.

3.8. Prehľad_timov

Pomocou SQL jazyka budú vybrané a zobrazené príslušné údaje z databázy (tími).

4. Sekvenčný diagram

Nasledujúci diagram zobrazuje časový priebeh operácií a prepojenie požiadaviek jednotlivých častí systému (architektúra MVC).



Obr.2 Sekvenčný diagram zobrazenia stránky

5. Používateľské rozhranie

5.1. Používateľská hierarchia

Zadávateľ nevyžaduje admin – user hierarchiu. Musí byť vyriešené prvé vytváranie konta, keďže má ísť o webovú aplikáciu používanú obmedzeným okruhom používateľov.

Vytváranie konta – nutná konzultácia so zadávateľom

Nato, aby používateľ mohol prezerať a editovať obsah musí prejsť autentifikačným procesom. To zahŕňa vyplnenie mena a hesla na prihlasovacej stránke.

5.2. Popis funkčnosti aplikácie pre prihláseného používateľa:

5.2.1. Prihlasovacia stránka

5.2.1.1. Zadávanie mena a hesla – prístupne pri prvom otvorení stránky

5.2.1.2. Prípád zabudnutého hesla – užívateľovi sa pošle heslo na vopred zadaný mail

5.2.2. Odhlasovanie

Používateľ sa odhlási, môže sa zase prihlásiť.

5.2.3. Interaktívna mapa

5.2.3.1. Mapa – užívateľ interakcia

Užívateľ má možnosť použiť zoom a presúvať sa v rámci mapy. Na mape sa zobrazujú už pridané zápisky ktoré môže prezerať kliknutím na ne. Po kliknutí sa zobrazí stránka daného zápisu.

5.2.3.2. Pridávanie zápisu

Po kliknutí na pridanie zápisu sa užívateľovi otvorí okno s pridávaním zápisu. Musí vyplniť všetky položky (opísane v špecifikácii požiadaviek). Položka „Nápady“ je dobrovoľná. V okne sa bude zobrazovať aj mapa kde používateľ zaznačí polohu zápisu. Alternatívne pridanie na mapu by mohlo prebehnúť na základe užívateľom zadanej GPS pozície. Zápis môže pridať ktorýkoľvek prihlásený používateľ. Pri vytváraní zápisu sa môže vytvoriť nová vrstva, alebo sa zápis priradí k niektorej z už existujúcich.

5.2.4. Vymazávanie dát

5.2.4.1. Zmazanie zápisu

Na stránke zápisu je možnosť zmazať ho. Táto možnosť je dostupná iba pre používateľa, ktorý daný zápis vytvoril.

5.2.4.2. Zmazanie vrstvy

Vrstvu môže zmazať iba užívateľ, ktorý do nej priradil prvý zápis a teda ju vytvoril.

5.2.4.3. Mazanie nápadu

Užívateľ ktorý daný nápad vytvoril má možnosť ho zmazať.

5.2.5. Vyhľadávanie v zápisoch –mimo mapy

Všetci užívatelia majú možnosť vyhľadávať v zápisoch aj bez použitia interaktívnej mapy. Kritéria vyhľadávania sú popísané v špecifikácii požiadaviek.

5.2.6. Pridávanie nápadu

Pri vytváraní akéhokoľvek zápisu má užívateľ možnosť vytvoriť „Nápad“ – čo obsahuje je opísané v špecifikácii požiadaviek.

5.3. Návrh rozloženia komponentov

5.3.1. Prihlasovacia stránka 5.3.

Názov turistickej skupiny

Prihlasovacie meno

Email Address

Heslo

Email Address

Prihlas ma

Zabudol som heslo

5.3.2. Hlavná stránka

Názov turistickej skupiny

Hlavná stránka
Vytvoriť nový zápis
Vyhľadávať v zápisoch
Moj profil
Prehľad tímov
Odhlásiť

Nazov zapisu zadaj...
Vrstvy ▶
Datum ▶
Region ▶
Ucastnici zapisu zadaj...

5.3.3. Vytvorenie zápisu

Názov turistickej skupiny

Hlavná stránka
Vytvoriť nový zápis
Vyhľadávať v zápisoch
Moj profil
Prehľad tímov
Odhlásiť

Zadaj nazov zapisu

Všetky údaje
(datum, region,
vrstvy.....)

Vs. údaje

Zadaj textový popis

adresar

Nahraj fotografie

adresar

Nahraj videozaznam

Nazov, textový popis napadu

Pridaj ďalší napad

5.3.4. Zobrazenie zápisu

Názov turistickej skupiny

Hlavná stránka
Vytvoriť nový zpis
Vyhľadávať v zpisoch
Moj profil
Prehľad tímov
Odhlásiť

Názov zpisu

Všetky údaje (datum, region, vrstvy...)

Textový popis

Prehľad fotografií

Prehľad nápadov k zpisu

Samotný videozáznam, alebo odkazy na youtube

Diskusia

5.3.5. Vyhľadávanie zpisu

Názov turistickej skupiny

Hlavná stránka
Vytvoriť nový zpis
Vyhľadávať v zpisoch
Moj profil
Prehľad tímov
Odhlásiť

Názov zpisu :

Vrstvy ▶

Datum ▶

Region ▶

Učastníci zpisu :

Vylisťovanie zpisov vyhovujúcich podmienkam

5.3.6. Profil používateľa

Názov turistikej skupiny

Hlavná stránka


Vytvoriť nový zápis

Vyhľadávať v zápisoch

Moj profil

Prehľad tímov

Odhlásiť



Meno

Zmena emailu:

nový email

Zmen email

Email

Zmena hesla:

staré heslo

nové heslo

nové heslo

Zmen heslo

Zoznam tímov v ktorých je členom

Prehľad nápadov podľa tímov do ktorých je zaradený

Vypis výletov ktorých sa zúčastnil

Názov nového tímu

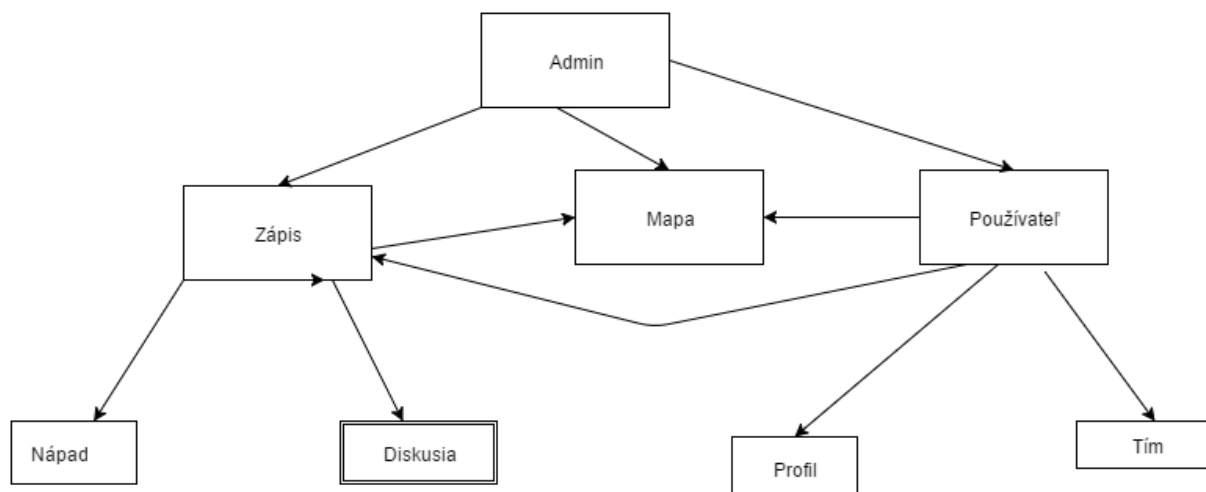
Popis nového tímu

Vytvor nový tím

5.3.7. Prehľad tímov

Názov turistikej skupiny		
Názov tímu		
popis		
členovia		
xxx	yyy	zzz
aaa	bbb	ccc

6. Návrh komponentov



Obr. 3 Diagram komponentov

6.1. Komponent mapa

(Mapa na ktorej budú zaznamenané jednotlivé zápisky, označovať ich v nej budú samotní používatelia).

Komponent mapa bude centrálnym komponentom projektu. Na zobrazenie jednotlivých geografických dát využijeme open source OpenStreetMap alebo Mapy.cz od spoločnosti Seznam.cz. Každý zápis na mape si bude pamätať id jednotlivých zápiskov. V prípade OSM využijeme na prepojenie mapy a aplikácie open source django-osm-field package.

6.2. Komponent Zápis

Po kliknutí na zápis sa vygeneruje stránka na základe id hodnoty. Natiahnu sa príslušné údaje z databázy. Každý zápis bude takisto mať prepojenie na diskusiu pod zápisom (prepojenie s databázou). Tento komponent sa bude dať vytvoriť adminom (v administratívnom prostredí), alebo používateľom pomocou formulárou na stránke. Pri vytvorení komponentu je takisto možné vytvoriť komponent Nápad. (nápad na ďalšie zaujímavosti v blízkosti daného zápisu). Bude sa dať vytvoriť adminom v administratívnom rozhraní alebo používateľom formou formulárov.

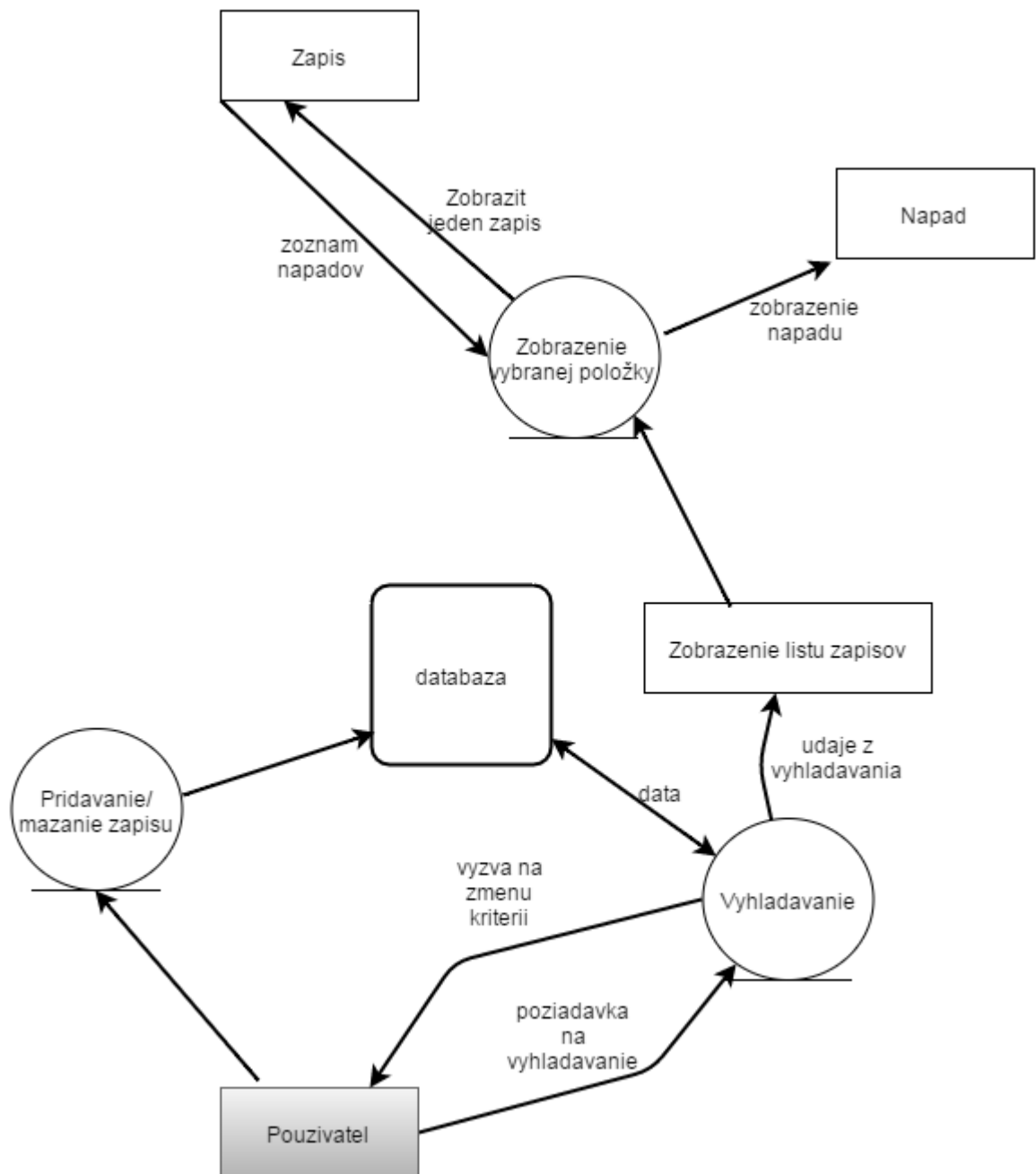
6.3. Komponent Admin

Django môže automaticky vytvoriť profesionálne administratívne rozhranie pripojené na produkčný server. Webová stránka je prístupná len autorizovanému užívateľovi. Užívateľ je schopný upravovať, mazať a vytvárať jednotlivé údaje z databáz. V prípade potreby je možné prispôsobiť administratívne prostredie užívateľovi (adminovi).

6.4. Komponent Používateľ

Používateľov vytvára admin v administratívnom rozhraní. Každé heslo bude zakódované v databáze. Používateľ bude mať obmedzený prístup k databáze, bude mať prístup k svojmu profilu kde si môže upraviť údaje formou formulárov. Takisto má prístup ku komponentu Tím, kde môže formou formulárov vytvoriť tím a pridať jednotlivých používateľov.

7. Data flow diagram



Obr. 4 Data flow diagram

Modeluje tok informácií medzi procesmi a dátovými úložiskami. Zobrazuje konkrétne kúsky informácií a procesy, ktoré si medzi sebou vymieňajú.