

Tímový projekt  
Projektová dokumentácia

Míľnik č. 1

**Členovia tímu:** Jakub Hlavačka, Dominik Horváth, Jakub Müller, Táňa Poláková, Dávid Silady, Adam Šípka

**Vedúci tímu:** Ing. Richard Marko, PhD.

**Študijný program:** Inteligentné softvérové systémy

**Ročník:** 1.

**Kontakt na tím:** tim8.fiit.stuba@gmail.com

Ak. Rok: 2021/2022

Inžinierske dielo

<b>Big Picture</b>	<b>4</b>
Úvod	4
Globálne ciele projektu na zimný semester	4
Vzorka dát	4
Čistenie dát	4
Uloženie dát	4
Webová stránka tímu	5
Indexovanie dát	5
Funkčné webové rozhranie pre AMS	5
Funkčná API	5
Prototyp so statickým sťahovaním a indexovaním	5
Celkový pohľad na systém	6
Vysoká architektúra	6
Sekvenčný diagram indexovania	7
Sekvenčný diagram fungovania Webovej aplikácie	8
Sekvenčný diagram fungovania API serveru	9
Sekvenčný diagram fungovania Scraperu	10
<b>Moduly systému</b>	<b>11</b>
Modul Elasticsearch a indexovanie	11
Analýza	11
Návrh	11
Implementácia	12
Testovanie	12
Modul Webová Aplikácia	13
Analýza	13
Základný prípad použitia	13
Implementácia	13
Testovanie	14
Modul API server	15
Analýza	15
Návrh	15
Implementácia	15
Testovanie	16
<b>Prílohy</b>	<b>19</b>
Modul na indexovanie	19
Modul webovej aplikácie	19
Modul na server API	19

# Big Picture

## Úvod

Táto časť dokumentu slúži na predstavenie Inžinierskeho diela, ktoré je aktuálne výstupom našej spoločnej práce na tímovom projekte.

V prvej kapitole je opísaný vývoj globálnych cieľov pre tento systém počas zimného semestra. Ďalšia kapitola je venovaná celkovému pohľadu na systém, ktorý sme sa používateľovi snažili priblížiť najmä prostredníctvom diagramov.

## Globálne ciele projektu na zimný semester

S tým ako plynuli týždne a šprinty v zimnom semestri sa vyvíjali aj globálne ciele tohto tímového projektu. Jednotlivé globálne ciele sú opísané v kapitolách tejto podkapitoly, pričom je dodržaný chronologický postup akým sme ich napĺňali.

## Vzorka dát

Dôležitou časťou pred začiatkom implementácie projektov, ktoré pracujú s veľkým množstvom dát je potreba tieto dáta spoznať.

Náplňou tohto cieľa bolo vymyslieť spôsobom akým budeme získavať potrebné dáta pre náš systém, následne zistiť ich štruktúru veľkosť a podobne...

Vzorkou dát boli články o zločinoch, ktoré sme sťahovali pomocou Google RSS. Veľkosť vzorky bola ~30GB, bolo v nej zahrnutých cez 100 zločinov, časová perióda bola jeden mesiac, samotné články pochádzali z regiónu USA a štruktúra článkov bola v HTML formáte.

## Čistenie dát

Vzhľadom na veľkosť samotnej vzorky dát za časové obdobie jedného mesiaca sme museli nájsť spôsob akým tieto dáta zredukovať. Rozhodli sme sa preto vyberať iba potrebný text z nadpisov a paragrafov článku.

Úlohou systému má byť indexovanie obsahu samotných článkov a preto je pre neho text z nadpisov a paragrafov článku absolútne postačujúci.

Ďalej sme sa dohodli aplikovať kompresiu Huffmanovho kódovania na ešte menšie vyťaženie úložného priestoru našich dát.

Výsledkom bolo zredukovanie ~30GB na ~1.5GB bez aplikovania Huffmanovho kódovania a ~350MB po aplikovaní kompresie.

## Uloženie dát

V analýze pre tento cieľ sme zistili, že jednou z vhodných databáz pre veľké dáta je MongoDB.

Túto databázu sme si vybrali z nasledovných dôvodov:

- Jednoduché integrácia s Scrapy systémom, ktorý používame na sťahovanie dát.
- Zabudovaná zlíb kompresia.
- Do budúca jednoduchá integrácia s Elasticsearch, vďaka JSON formátu samotných dokumentov.

## Webová stránka tímu

Po udelení SSH kódu na prístup do tímovej virtuálnej mašiny bolo sa hlavným cieľom stalo vytvoriť webovú stránku. Táto webová stránka má umožňovať jednoduchý prístup k zápisnicam zo stretnutí a v prvom rade predstaviť tím č.8.

## Indexovanie dát

Hlavnou funkcionalitou systému Adverse Media Screening má byť možnosť vyhľadávania fyzických a právnických osôb spojených so zločinmi, ktoré údajne vykonali. Aby sme tieto osoby a zločiny dokázali jednoducho vyhľadávať je potrebné indexovať obsah stiahnutých článkov.

## Funkčné webové rozhranie pre AMS

Predpokladáme, že používatelia, ktorí budú náš systém používať nemajú znalosti ako jeho vývojári alebo iný kolegovia z IT oblasti. Z tohto dôvodu bolo úlohou ďalšieho cieľa vytvoriť užívateľsky príjemné rozhranie, ktoré je používateľom ľahko prístupné, vďaka svojej webovej podstate.

## Funkčná API

Námet na vytvorenie aplikačno programovacieho rozhrania prišiel od nášho vedúceho. Cieľom tejto API je sprístupniť možnosť vyhľadávania kriminálnych priestupkov fyzických a právnických osôb aj používateľom, ktorí nebudú používať systém Adverse Media Screening ale budú chcieť využívať túto funkcionalitu napríklad vo svojom vlastnom systéme.

## Prototyp so statickým sťahovaním a indexovaním

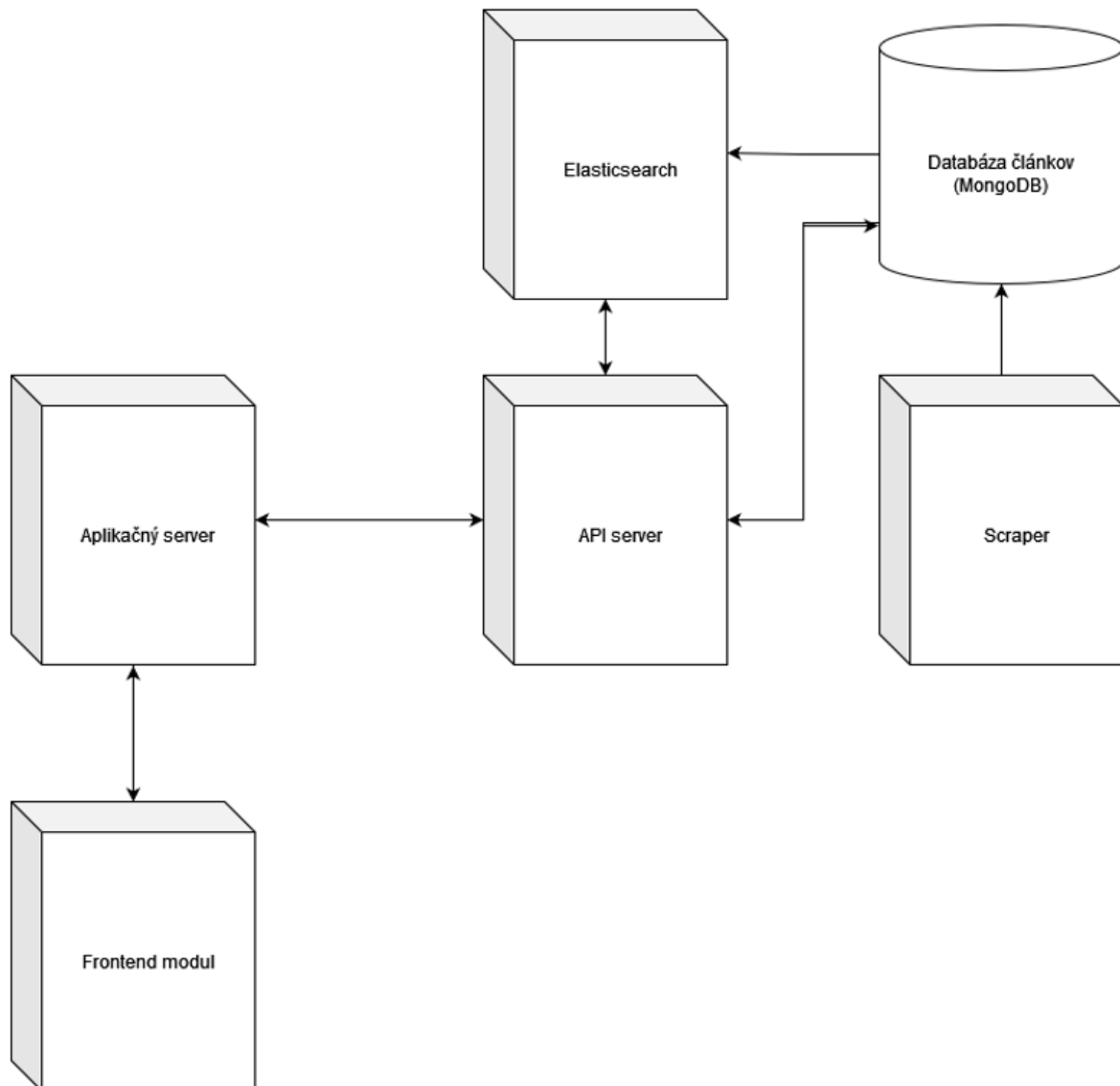
Aktuálne nám bežia všetky funkčné moduly samostatne a úlohou tohto globálneho cieľa je ich úspešne integrovanie na vytvorenie celistvého systému, ktorý na konci zimného semestra môžeme považovať za prvotný prototyp systému Adverse Media Screening.

## Celkový pohľad na systém

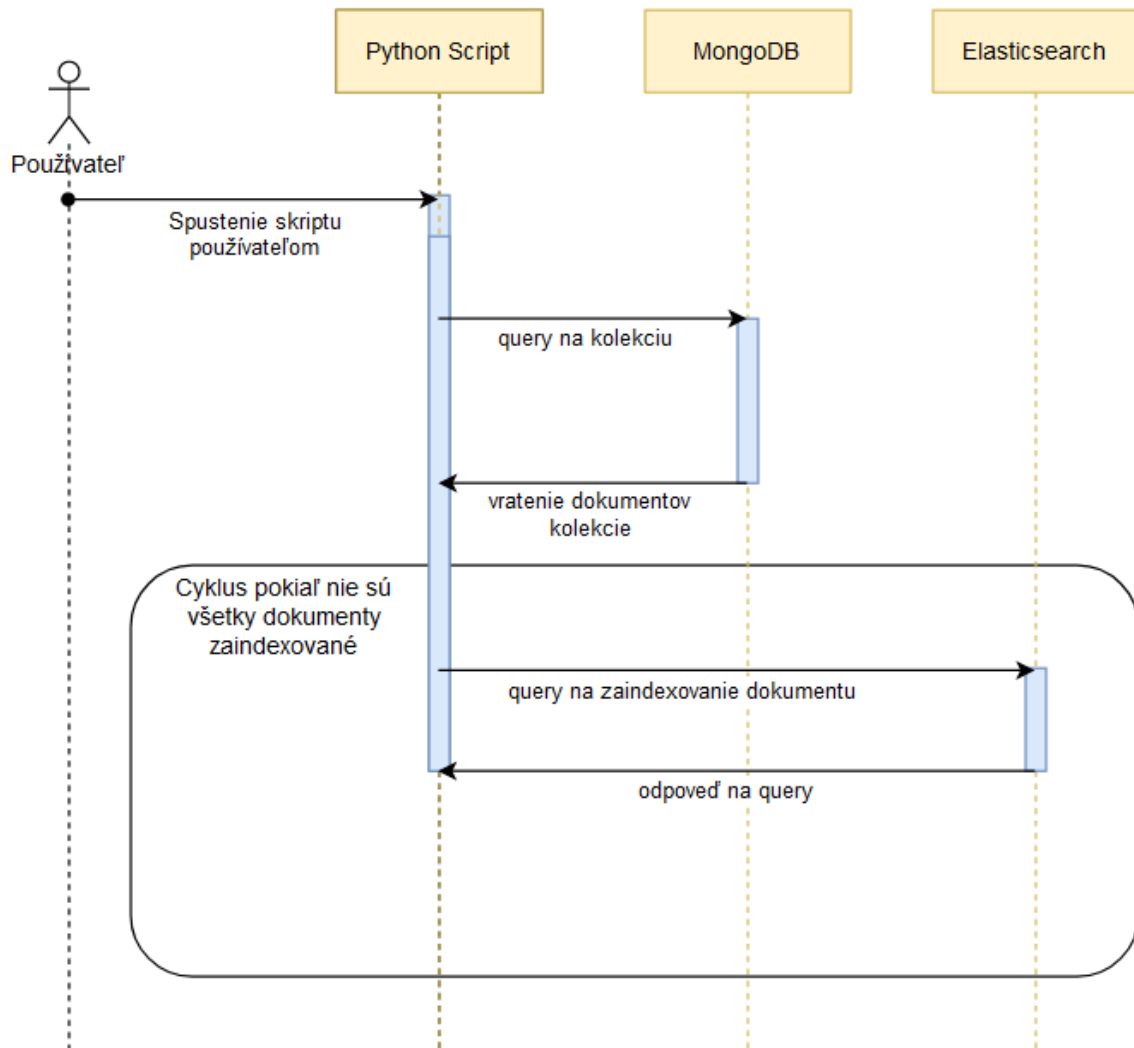
V tejto kapitole sú zobrazené diagramy, ktorými sme sa riadili počas implementácie tohto systému alebo vznikli neskôr na zlepšenie komunikácie v tíme. Samotné diagramy by mali čitateľovi priblížiť fungovanie aktuálnej verzie systému Adverse Media Screening.

### Vysoká architektúra

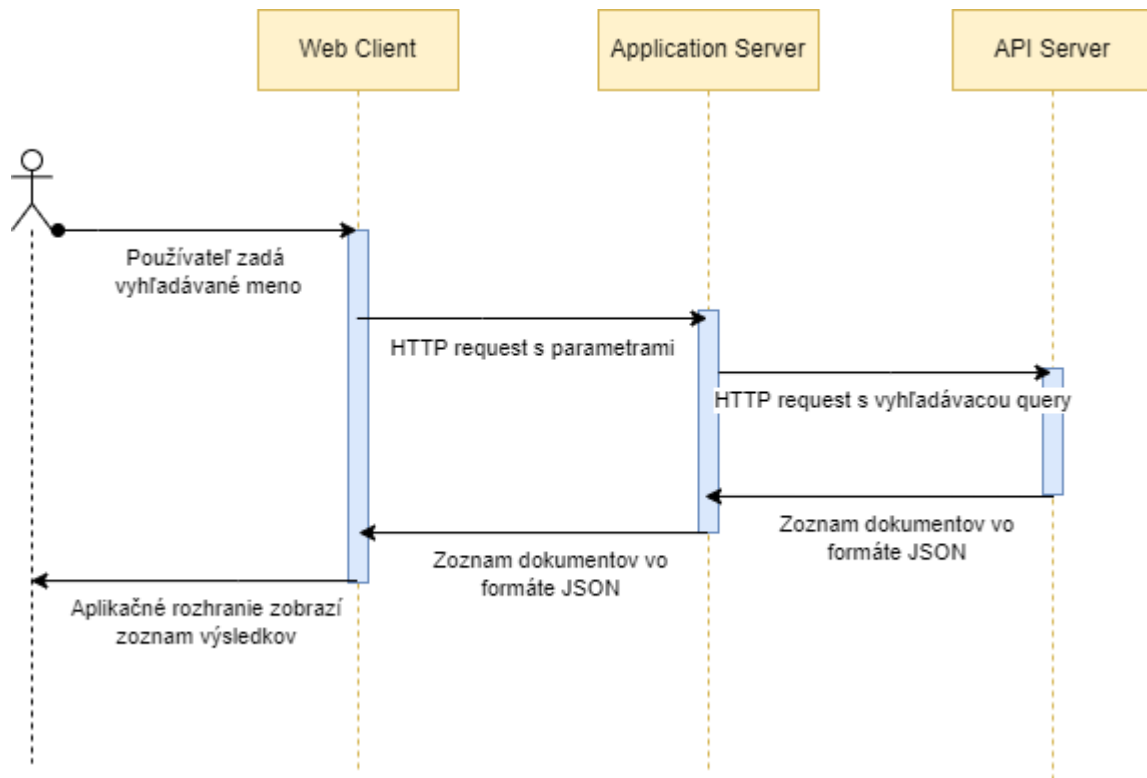
Na obrázku nižšie je zobrazená vysoká architektúra s modulmi, nástrojmi a databázami, ktoré teraz využívame.



## Sekvenčný diagram indexovania

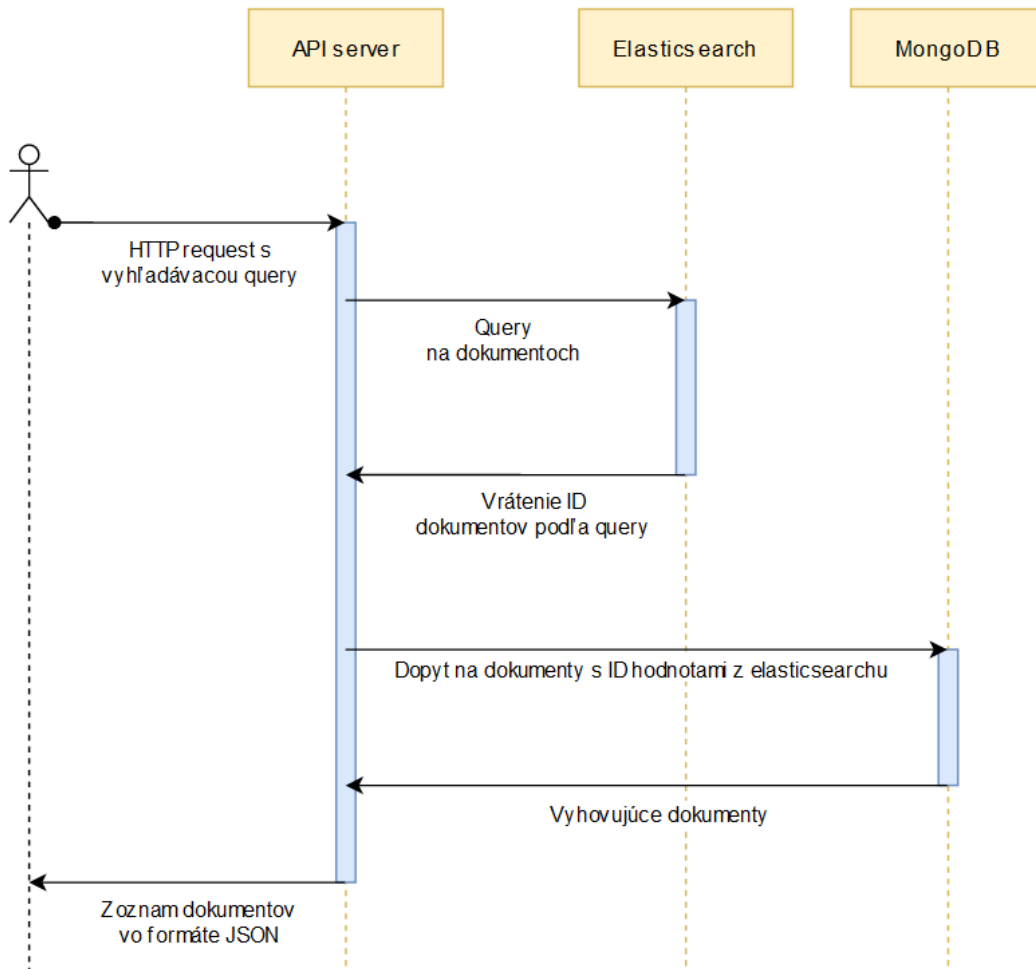


## Sekvenčný diagram fungovania Webovej aplikácie

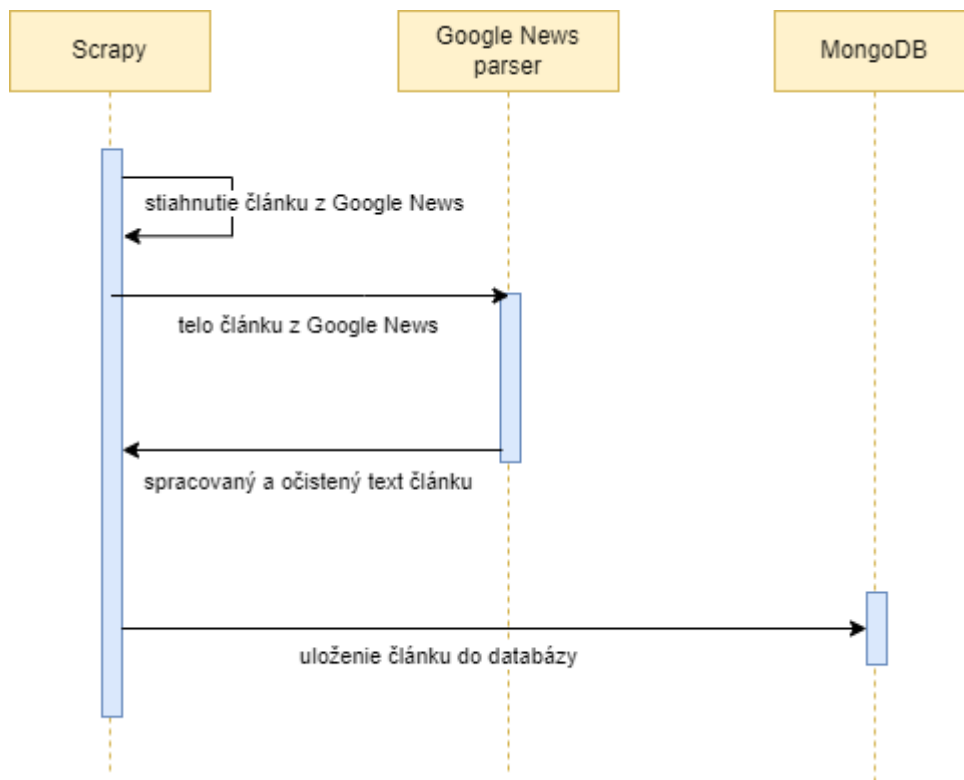




## Sekvenčný diagram fungovania API serveru



## Sekvenčný diagram fungovania Scrapera



# Moduly systému

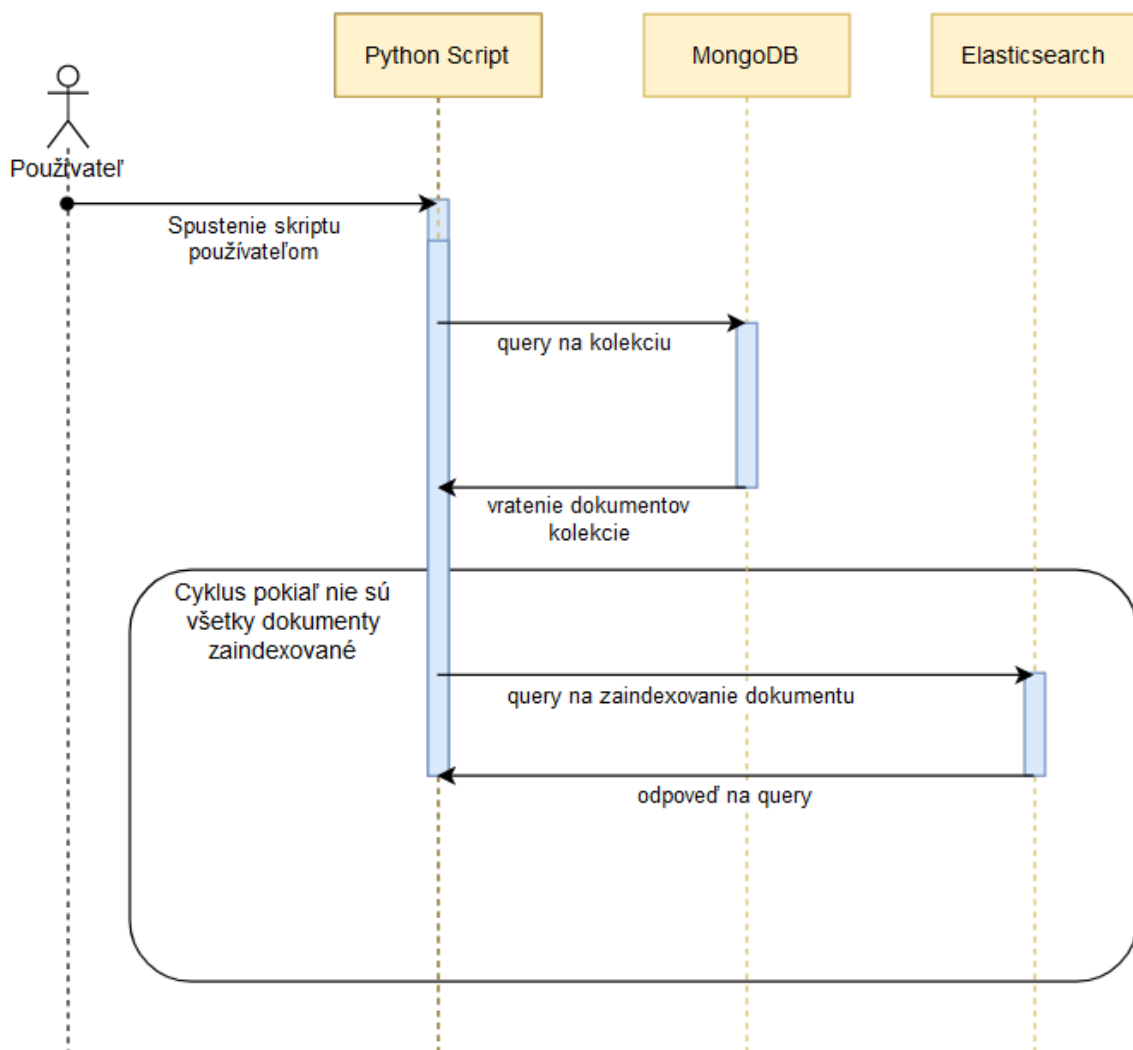
## Modul Elasticsearch a indexovanie

V tejto podkapitole sa bližšie pozrieme na modul systému, ktorý súvisí s indexovaním dát.

### Analýza

Samotné dáta boli stiahnuté prostredníctvom nášho scraperu, ktorý ich aj očistil a uložil do distribuovanej Mongo databázy. Aktuálne fungujeme na vlastnom aktualizovaní indexu v Elasticsearch prostredníctvom Python skriptu. Úlohou tohto skriptu je vytiahnuť dáta z úložiska a poslať ich nástroju Elasticsearch, ktorý vykoná potrebné indexovanie.

### Návrh



## Implementácia

Elasticsearch je nasadený prostredníctvom technológie Docker. Aktuálne fungujeme na jednom node, ktorý má jeden shard a jednu repliku. V samotnom sharde a aj replike je uložený Lucene index.

Elasticsearch využíva pri indexovaní nových dokumentov a uskutočňovaní používateľských dopytov štandardný analyzátor s filtrovaním anglických stop slov.

Každý dokument je zložený z jedného poľa (angl. field) text, ktorého hodnota je obsah jedného článku z databázy.

Elasticsearch si hodnotu poľa neukladá, pretože to v rámci našej funkcionality nie je potrebné a bolo by to skôr zbytočné a pri vyhľadávaní zaťažujúce.

Jednotlivým dokumentom je priradená id z Mongo databázy, vďaka čomu sa dokážeme z vyhľadávania v indexe jednoducho dostať k obsahu a metadátam článku, ktoré sú uložené v databáze.

MongoDB je tiež nasadená na virtuálnej mašine prostredníctvom technológie Docker. V databáze admin sa nachádza kolekcia articles, ktorej obsah je komprimovaných Huffmanovým kódovaním za účelom šetrenia úložiska.

Jeden element kolekcie obsahuje id, title (názov článku), published (dátum vydania článku), link (link na článok) a html (obsah článku) pole.

Python script využíva knižnice ako pymongo, prostredníctvom ktorej sa pripája ku Mongo databáze. Ďalšou dôležitou knižnicou je elasticsearch, vďaka ktorej dokážeme poslať údaje samotnému nástroju Elasticsearch, ktorý ich následne indexuje.

## Testovanie

Hlavným cieľom tohto modulu bolo zistenie pamäťovej náročnosti indexu Elasticsearch. Na poslednú chvíľu sme sa však rozhodli tento modul využívať pri prvotnom prototype nášho systému. Vzhľadom na to ešte neboli napísané žiadne jednotkové, integračné alebo end-to-end testovania a samotný modul sme testovali iba ručne.

Tento spôsob rúčného testovania fungoval na princípe vytiahnutia údaju z databázy, zaindexovania v Elasticsearch a následne overenia správneho indexovania prostredníctvom nástroja Kibana, ktorý nie je súčasťou tohto modulu v produkčnej časti.

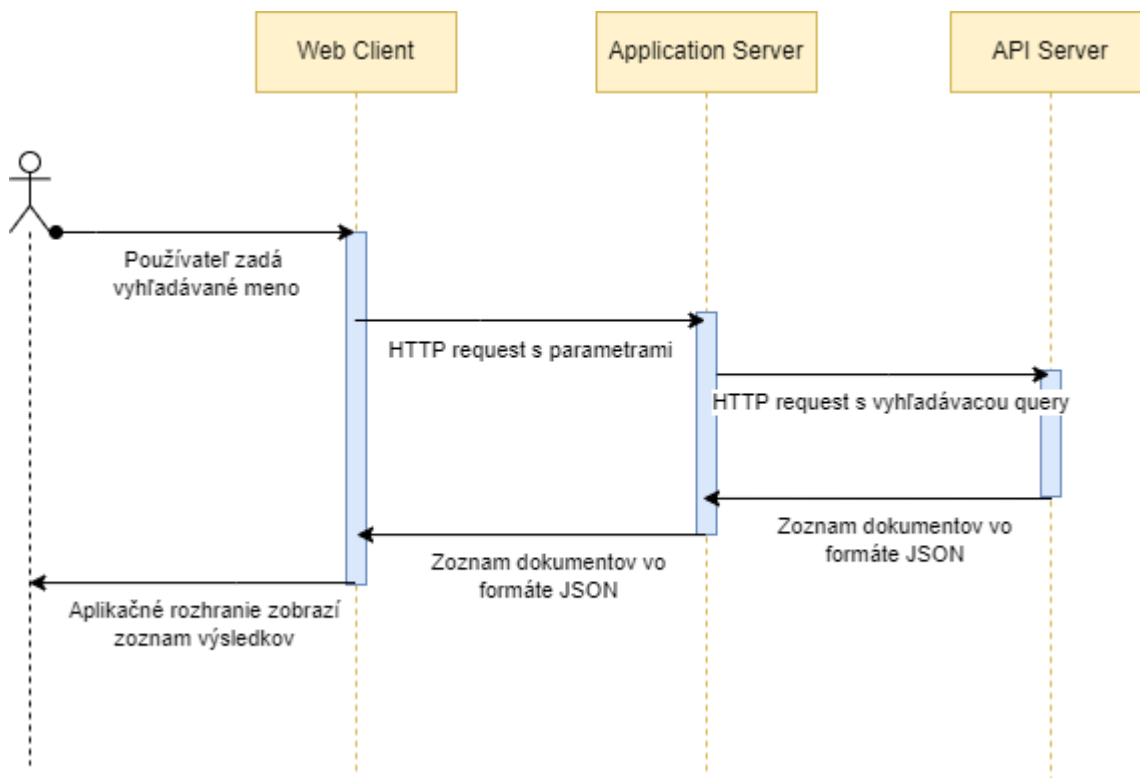
# Modul Webová Aplikácia

V tejto podkapitole sa bližšie pozrieme na aplikačnú časť nášho riešenia.

## Analýza

Webová aplikácia je potrebná pre zabezpečenie užívateľského rozhrania pre jednoduchší prístup pre prácu s našimi dátami. Aplikačná časť je tvorená webovým klientom a aplikačným serverom. Aplikačný server zabezpečuje komunikáciu s ostatnými modulmi. Webový klient slúži používateľovi ako grafické rozhranie.

## Základný prípad použitia



## Implementácia

Aktuálne je zabezpečená pomocou klient aplikácie napísanej v ReactJS a NodeJS aplikačného servera, ktorý funguje pomocou knižnice ExpressJS. Zatiaľ sme neimplementovali žiadne prípady použitia, ktoré by vyžadovali aplikačnú databázu, plánujeme však použiť PostgreSQL pre zabezpečenie možnosti prihlasovania a rozšírenej funkcionality pre prihlásených používateľov.

## Testovanie

Zhodnotenie funkcionality zatiaľ spočíva najmä z integračných testov, resp. kontroly komunikácie medzi aplikačnými modulmi (web client, application server) a vonkajšími modulmi (API server). Jednotkové testovanie zatiaľ nebolo zabezpečené.

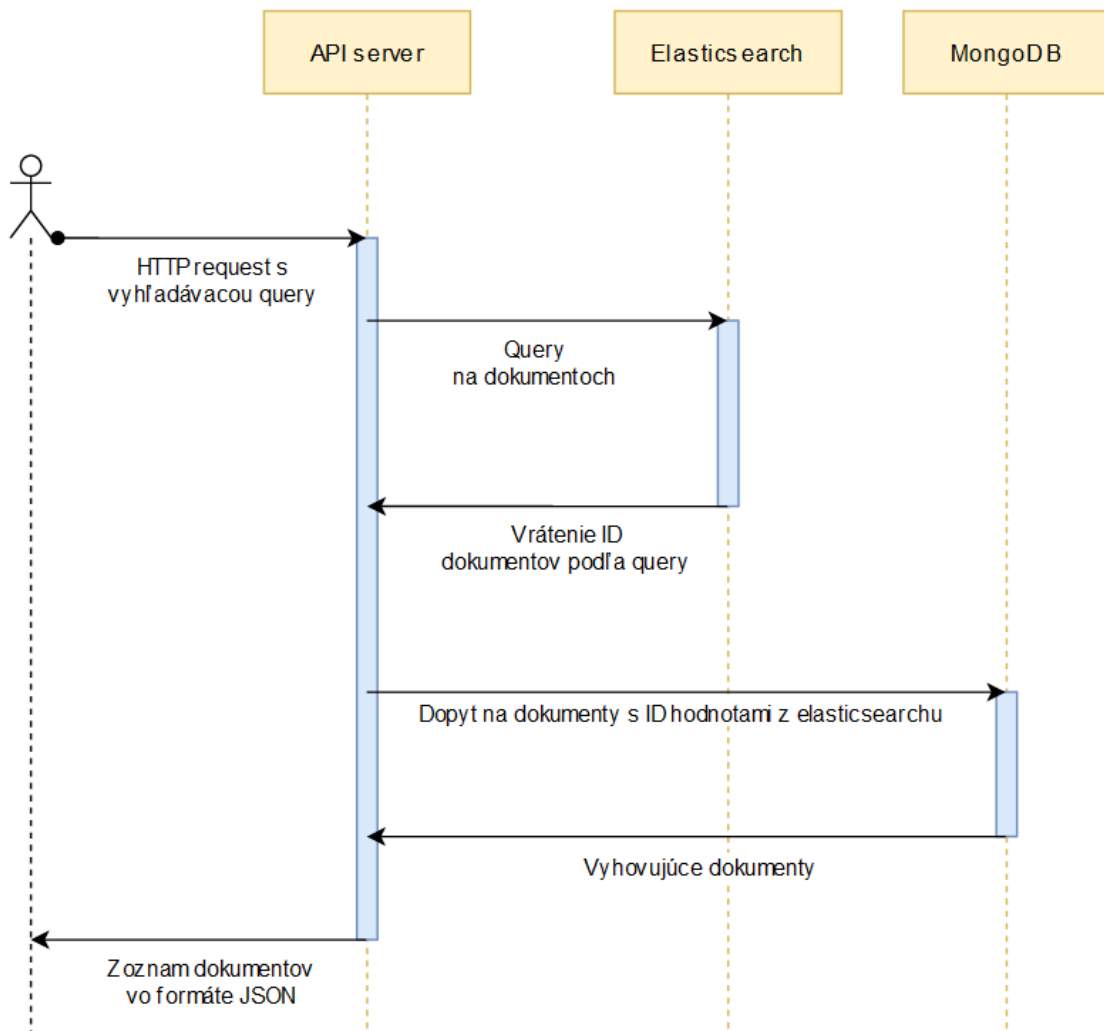
# Modul API server

V tejto sekcii predstavíme API server nášho projektu.

## Analýza

API server je potrebné implementovať pre oddelenie používateľského prostredia a hlavných biznis funkcií. API server spracováva vstupy z používateľského aplikačného serveru a vracia dáta do hlavného prostredia. Získava údaje z externých dátových zdrojov ako MongoDB a vyhľadáva v indexe dát prostredníctvom ElasticSearch. Komunikácia na tento server je taktiež možná mimo aplikačného serveru, pre externé aplikácie a rýchle experimentovanie s dátami.

## Návrh



## Implementácia

API server je implementovaný pomocou jazyku Python a Flask frameworkom. Pomocou Flask triedy BluePrints sme implementovali verzovanie našej API. Pre vyhľadávanie bola

vytvorená prvá API route `/search`, ktorá akceptuje jeden povinný parameter `q`, ktorý obsahuje vyhľadávaný reťazec. Ďalšie parametre sú smerované na redukciu vyhľadávania, ako napríklad časové rozpätie alebo lokalizácia (jazyk a krajina pôvodu).

## Testovanie

API sa momentálne testuje manuálne, prostredníctvom HTTP requestov priamo na koncové body definované v zdrojovom kóde. Najjednoduchší spôsob tohoto testovania môže byť prostredníctvom jednoduchých command-line nástrojov ako `curl` alebo `wget`.



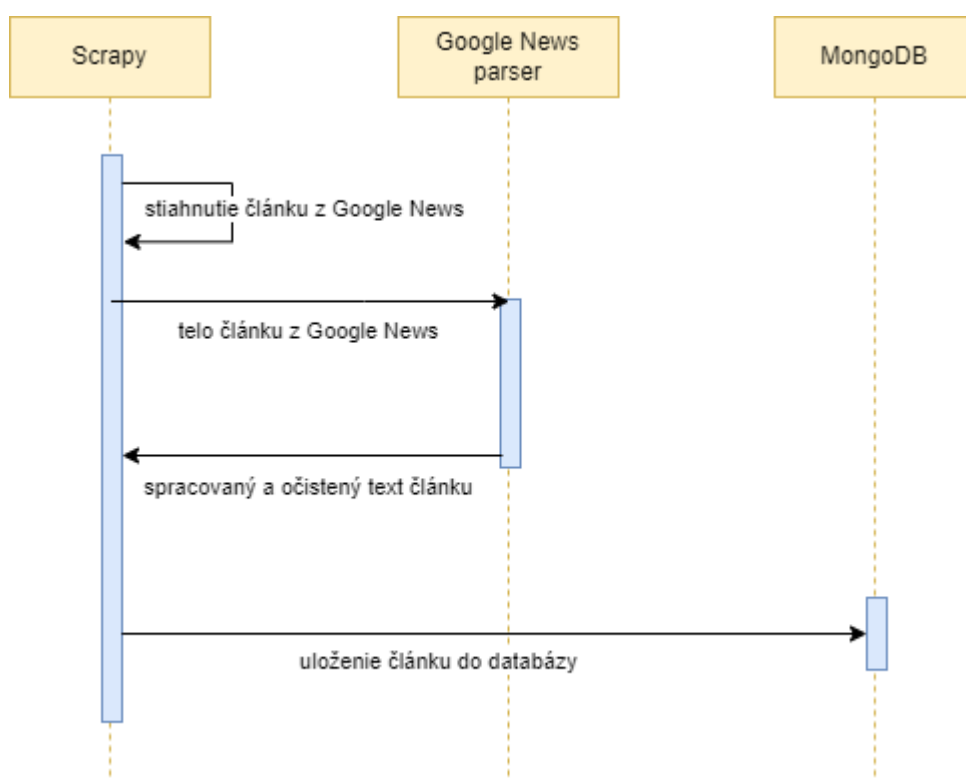
## Modul Scrapper

V tejto kapitole sa pozrieme na fungovanie modulu Scrapper.

### Analýza

Hlavnou úlohou scraperu bolo získať potrebné dáta z Google News - teda spravodajské články, ktoré nejako súvisia s trestnou činnosťou. Po získaní týchto článkov je potrebné ich vhodným spôsobom očistiť, aby zaberali čo najmenej miesta, ale zároveň aby nestratili svoju výpovednú hodnotu. Nakoniec naše dáta scraper ukladá do Mongo databázy.

### Návrh



### Implementácia

Scraper bol implementovaný v jazyku Python s využitím knižnice Scrapy. Pomocou tejto knižnice sme dokázali získať články a potrebné informácie o nich z portálu Google News. Na získavanie pre nás relevantných článkov (teda článkov, ktoré súvisia s nejakou trestnou činnosťou) sme použili nami vytvorený zoznam trestných činov, ktorý obsahoval 96 názvov zločinov.

Telá získaných článkov, teda HTML súbory, sú následne sparované. Parsovanie prebieha tak, že z tela článku - resp. z body tagu sa vyseletovali iba pre nás potrebné tagy, konkrétne paragrafy a headingy, ktoré boli ešte dodatočne očistené od nepotrebných atribútov.

O získanom článku si nakoniec ukladáme tieto informácie:

- názov článku
- dátum uverejnenia článku
- odkaz na daný článok
- sparované telo článku

Nakoniec scraper uloží daný článok do našej MongoDB.

## Testovanie

Náš modul scrapera sa testoval manuálne a na menšej dátovej vzorke. Konkrétne sme scrapli spravodajské články z Google News za posledný mesiac a z jedného regiónu - konkrétne US. Každý člen tímu si zobral jednu časť zo zoznamu zločinov a spustil scraper lokálne. Následne sme testovali aj parser, ktorý bol testovaný na jednej z týchto častí článkov, ktorá mala cca 3.5GB.

# Prílohy

Všetky docker containery bežia na Docker verzii 20.10.10

Python programy fungujú na verzii 3.9

NodeJS verzia: 14

ExpressJS verzia: 4.7.1

ReactJS verzia: 17.0.1

## Modul na indexovanie

Návod na spustenie tohto modulu je opísaný v README.md tohto [repozitáru](#).

## Modul webovej aplikácie

Návod na spustenie frontendovej a backedovej časti modulu Webovej aplikácie je v README tohto [repozitáru](#).

## Modul na server API

Krátka dokumentácia k jednotlivým HTTP requestom je dostupná v README [repozitáru](#).  
Obsahuje taktiež Dockerfile pre vytvorenie Docker image na testovanie API na lokálnom prostredí.

Riadenie projektu

<b>Big Picture</b>	<b>23</b>
Úvod	23
Práca v tíme	24
Jakub Hlavačka	24
Dominik Horváth	24
Jakub Müller	24
Táňa Poláková	24
Dávid Silady	25
Adam Šípka	25
Šprinty	27
Šprint č. 1	27
Šprint č. 2	29
Šprint č. 3	31
Manažment tímu	33
Každý člen tímu je zodpovedný za svoju rolu	33
Na riadenie projektu využívame metodiku Scrum	33
Sme tímoví hráči	34
<b>Motivačný dokument</b>	<b>35</b>
Tím	35
Motivácia	36
Téma č. 4 - Adverse Media Screening [AMS]	36
Téma č. 8 - Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]	37
Príloha A - Zoradenie všetkých tém podľa priority	38
Príloha B - Rozvrh tímu	39
<b>Metodiky</b>	<b>41</b>
Metodika Scrum	41
Standup	41
Šprint review	41
Plánovanie	41
Práca v nástroji Jira	42
Backlog	42
Nástenka	42
Git Metodika	43
Vetvenie	43
Commit	43
Pull requests	43
<b>Export evidencie úloh</b>	<b>44</b>

# Big Picture

## Úvod

Riadenie projektu je kľúčový aspekt, ktorý ovplyvňuje, či bude projekt úspešne ukončený, alebo nie. Zo začiatku semestra sme tejto oblasti venovali málo pozornosti, čo sa čiastočne odzrkadlilo aj na našich výsledkoch. Človek sa najlepšie učí na vlastných chybách a my sme si dali záväzok neodsúvať náležitosti agilného vývoja bokom.

Aby tím dbal na každú oblasť či už implementačnú, dokumentačnú alebo inú, v súvislosti s vývojom softvéru, určili sme role pre každého člena tímu. Role členov tímu, ako aj ich plnenie a manažment úloh sme zdokumentovali v kapitole Práca v tíme. Aby nám v tíme fungovalo všetko na 100 %, boli potrebné určité manažérske schopnosti a procesy, ktoré sme definovali v kapitole Manažment tímu. Prvky agilného prístupu – šprinty, k vývoju inžinierskeho diela sme zdokumentovali v kapitole Šprinty.

## Práca v tíme

Spočiatku sme fungovali na princípe, že všetci robia všetko. Domnievali sme sa, že ak bude každý z nás zahrnutý do všetkých prebiehajúcich procesov, či už implementačných, alebo nie, bude to pre nás veľké plus. S pribúdajúcimi týždňami semestra sme si uvedomili, že týmto prístupom sme dospeli k stavu, kedy nikto nie je zodpovedný za nič. Náš posun vpred to výrazne spomalilo. Pochopili sme, že ak človek dostane svoju rolu v rámci tímu, môže sa zdokonaľovať iba v jednej oblasti, namiesto všetkých.

V ďalších odstavcoch vám predstavíme prácu jednotlivých členov tímu, ktorá je zosumarizovaná v tabuľke pod posledným členom tímu.

### Jakub Hlavačka

Tak, ako každý z nás, na začiatku vývoja prišiel Jakub do kontaktu hlavne s prieskumnými úlohami. Z dlhodobého hľadiska sa ale zamerával najmä na prácu s dátami, čo zahŕňalo implementovanie získavania dát a následne ich indexovanie. Z manažérskej perspektívy sme sa ako tím zhodli, že je vhodný adept na dokumentovanie inžinierskeho diela. Samozrejme, jednotlivé časti softvéru budú spravidla dokumentovať práve ich vývojári, avšak Jakub dá dokumentu základnú štruktúru a bude sa oň starať. Existujúca časť dokumentácie, ktorej časť autorstva patrí práve Jakubovi, je vysoká architektúra softvéru. Okrem nej zdokumentoval Big Picture inžinierskeho diela a modul Elasticsearch a indexovanie.

### Dominik Horváth

Dominik venoval takmer všetok svoj čas práci na projekte dátam. Na začiatku mu však vystala aj úloha urobiť diagram prípadov použitia, ktorý je veľmi jednoduchý vzhľadom na povahu nášho softvéru. Diagram je súčasťou dokumentácie inžinierskeho diela. Okrem toho zdokumentoval modul API server. Spoločne s ostatnými členmi tímu sa podieľal na tom, že v momente písania tohto dokumentu konečne vieme, ako budú naše dáta vyzerieť. Má zásluhu aj na tom, ako bude vyzerieť komunikácia medzi klientom a serverom z dátovej časti (API) a komunikáciu aj zdokumentoval. Dominikova rola je manažér plánovania.

### Jakub Müller

Vďaka za našu peknú stránku patrí Jakubovi M., ktorý ju implementoval. Pomáhal aj pri mechanizme získania a čistenia dát. Jakub sa postaral aj o to, aby sme mali všetky kódy na jednom mieste na platforme GitHub. Aby sme zaručili kvalitu nášho kódu, Jakub sa podujal robiť tzv. code review a postavil sa do role manažéra kvality. V rámci dokumentácie sa venoval modulu scrapper.

### Táňa Poláková

Tánina rola – manažér dokumentácie riadenia projektu, bola takmer od začiatku jasná, keď sa podujala písať denníky z našich stretnutí. Neskôr, keď sme zistili, že náš nástroj na manažment nepoužívame efektívne, sme sa zhodli, že bude scrum master a postará sa oň. Okrem formálnych a dokumentačných záležitostí sa taktiež podieľala na čistení dát a návrhu

API. Navrhla, ako by mohla vyzerat' databáza používateľov nášho softvéru ale zatiaľ nie je súčasťou dokumentácie. V rámci dokumentácie sa venovala Big Picture riadenia projektu, export úloh upravila tak, aby bolo v nich ľahko vyhľadávateľné a taktiež upravila dokumenty, ktoré sú na stránke tímu.

## Dávid Silady

Dávidove skúsenosti s tvorbou webu sme využili pri našom vlastnom webe projektu. Svoj čas nevenoval takmer ničomu inému, než náležitosti súvisiacim s webom. Medzi ne patrí aj návrh používateľského rozhrania, komunikácia medzi klientom a serverom a konfigurácia servera. Dávid je spoluautorom dokumentácie vysokej architektúry a aj preto sme ho zvolili ako hlavného architekta. Dokumentácia modulu webová aplikácia je jeho zásluha.

## Adam Šípka

Aby sme vedeli, ako funguje knižnica, ktorá je naším zdrojom získavania údajov, Adam ju ako jednu z prvých úloh analyzoval. Neskôr sa začal venovať Mongo databáze, ktorú spravuje dodnes. Taktiež, časť implementácie získavania dát patrí práve Adamovi. Rola, ktorej sa zaviazal, že sa bude poctivo venovať a študovať potrebné náležitosti je manažér testovania. Adam zdokumentoval naše metodiky.

Nasledujúca tabuľka je zhrnutím odsekov vyššie pre rýchlejšiu orientáciu.

Meno	Rola	Hlavné zameranie	Autorstvo dokumentácie
Jakub Hlavačka	Manažér dokumentácie inžinierskeho diela	Dáta, Dokumentácia	Big picture - inžinierske dielo, Modul - Elasticsearch a indexovanie
Dominik Horváth	Manažér plánovania	Dáta, Server	Diagram prípadu použitia, Modul - API server
Jakub Müller	Code review, Manažér kvality	Stránka tímu, Dáta	Modul - scrapper
Táňa Poláková	Scrum master, Manažér komunikácie, Manažér dokumentácie riadenia projektu	Jira, Dáta, Dokumentácia	Big picture - riadenie projektu, Export úloh
Dávid Silady	Hlavný architekt	Webová aplikácia	Modul - webová aplikácia



Adam Šípka	Manažér testovania	Dáta, Databáza článkov	Dokumentácia metodík
------------	--------------------	------------------------	----------------------

Snažíme sa, aby nikto z nás nemal pocit, že niekto robí viac, prípadne menej. Našu snahu, samozrejme, podkladáme počtom hodín, ktoré za každý šprint odrobíme. Celkový podiel práce sa nachádza v tabuľke nižšie.

<b>Meno</b>	<b>Podiel práce (%)</b>
Jakub Hlavačka	16.66667
Dominik Horváth	16.66667
Jakub Müller	16.66667
Táňa Poláková	16.66667
Dávid Silady	16.66667
Adam Šípka	16.66667

# Šprinty

Kapitola je venovaná dokumentácii prvých troch šprintov. Prvý šprint sme začali o týždeň neskôr oproti odporúčanému dátumu začiatku. Dôvodom bola naša nevedomosť, ktorá nás samozrejme neospravedlňuje. Prvý šprint začal 12. 10. 2021, čo je štvrtý týždeň semestra. Naša práca na projekte však začala už v druhom týždni, avšak tieto dva týždne neboli evidované ako oficiálny šprint, a preto úlohy, ktoré sme vtedy urobili, neprislúchajú žiadnemu šprintu. Fungovali sme v podstate v týždňových iteráciách bez zhodnotenia na konci iterácie.

Pri prvom šprinte sme sa oboznámili s pridelovaním bodov za jednotlivé úlohy/príbehy. Nedá sa však povedať, že by naše odhady zodpovedali realite, čo sa aj odrazilo pri šprint review. Taktiež sme si uvedomili, že ľudské zdroje neboli naplno využité a poučili sme sa, že do budúca určite naplánujeme viac úloh/príbehov. Prvý šprint je zdokumentovaný v sekcii Šprint č. 1.

Druhý šprint sa už javil profesionálnejšie. Odhad sa nám síce podarilo o trochu zlepšiť, ale stále je čo zlepšiť. Druhý šprint je zdokumentovaný v sekcii Šprint č. 2.

Bohužiaľ, podarilo sa nám do tretieho šprintu preniesť dva príbehy. Postihla nás neočakávaná situácia a deň pred ukončením šprintu nám pribudlo veľa nových úloh, ktoré súviseli s odovzdaním prvého kontrolného bodu. Poučili sme sa, že si veci nebudeme odkladať na poslednú chvíľu a že si určíme, kto sa bude priebežne o dokumentáciu starať. Tretí šprint je zdokumentovaný v sekcii Šprint č. 3.

## Šprint č. 1

Trvanie: 12. 10. 2021 – 26. 10. 2021

Čo hodnotíme pozitívne?

- Po zavedení šprintov máme väčší prehľad, kto čo urobí/il.

Čo chceme zlepšiť?

- Zistili sme, že máme veľa nevyužitých ľudských zdrojov. Určili sme si do budúca, že cez týždeň budeme venovať projektu viac hodín.
- Chceme sa snažiť rozdeliť úlohy tak, aby nemal niekto ťažšie a niekto ľahšie úlohy.
- Chceme sa naučiť pracovať s Jirou efektívnejšie.
- Voliť relevantnejšie príbehy, nie také „aby boli“.
- Kvôli nerelevantnosti ostala jedna úloha neukončená. Budeme sa jej venovať neskôr.
- Chceme zlepšiť svoj odhad pri pridelovaní počtu bodov.

Názov príbehu	Riešiteľ	Odhadovaný počet bodov	Je príbeh ukončený?	Pridelený počet bodov
Stránka tímu	Jakub Müller	6	áno	6
Otvorenie portov na serveri	Dávid Silady	6	áno	6
Vyriešiť "sudo su"	Táňa Poláková	2	áno	2
User DB	Táňa Poláková	6	nie	6
Prieskum DB	Jakub Müller	1	áno	1
Prihláška na TP CUP	Dávid Silady	2	áno	2
Set up Elasticsearch v docker container	Jakub Hlavačka	4	áno	4
Získanie dát pre prototyp	Jakub Hlavačka	4	áno	4
Získanie dát pre prototyp	Adam Šípka	8	áno	8
Získanie dát pre prototyp	Dominik Horváth	8	áno	8

Meno	Súčet počtu odhadovaných bodov	Súčet počtu pridelených bodov	Podiel práce (%)
Jakub Hlavačka	8	8	16.66667
Dominik Horváth	8	8	16.66667
Jakub Müller	8	8	16.66667
Táňa Poláková	8	8	16.66667
Dávid Silady	8	8	16.66667
Adam Šípka	8	8	16.66667

## Šprint č. 2

Trvanie: 26. 10. 2021 – 09. 11. 2021

Čo hodnotíme pozitívne?

- Naplánovali sme si 62 hodín, čo vychádza, že každý z nás mal 1.6 hodiny rezervu. Nakoniec sme odrobili viac, a to 68 hodín, čo je oproti predošlým 43.5 posun vpred. Úlohy sme si pridali v priebehu šprintu.
- Náročnosti jednotlivých príbehov sa celkom vyrovnali.

Čo chceme zlepšiť?

- Chceme príbehy špecifikovať menej, aby sme príbehy nemuseli rušiť, ak narazíme na nejaký problém.
- Zistili sme, že náročnosť úlohy je výrazne odlišná, ak na úlohe pracuje človek, ktorý s danou technológiou má skúsenosti a človek, ktorý má minimálne skúsenosti.
- Ak príbeh pridáme človeku s minimálnymi skúsenosťami, chceme daný príbeh radšej rozdeliť na menšie.

Názov príbehu	Riešiteľ	Odhadovaný počet bodov	Je príbeh ukončený?	Pridelený počet bodov
Parsovanie tagov pomocou CSS selector	Jakub Müller	5	áno	13
Parsovanie tagov pomocou regex	Táňa Poláková	8	zrušený	3
Upratať Jiru	Táňa Poláková	2	áno	3
Analýza bezstratovej kompresie textu	Jakub Hlavačka	8	áno	8
Lokálna MongoDB so získanými dátami	Adam Šípka	8	áno	5
Request URL s odpoveďou inou ako 200 uložiť do súboru	Adam Šípka	3	áno	1
Pridať info o projekte a opísať členov tímu	Jakub Müller	3	áno	1
Rozšíriť docker compose	Dominik Horváth	13	áno	8
Zavesenie klienta	Dávid Silady	13	áno	13
Parsovanie tagov pomocou lxml, readability, trafiletura	Táňa Poláková	3	áno	8

Pamäťová zložitosť Elasticsearch na vzorke dát	Jakub Hlavačka	5	nie	5
Rozšírenie google news scrapperu na lokáciu UK	Dominik Horváth	1	áno	1
Vyhľadanie RSS dvojčaťa k HTML článku - všeobecný postup	Dominik Horváth	8	áno	5
Napĺňanie MongoDB priamo pri scrapovaní	Adam Šípka	8	nie	8

Meno	Súčet počtu odhadovaných bodov	Súčet počtu pridelených bodov	Podiel práce (%)
Jakub Hlavačka	16	13	16.66667
Dominik Horváth	22	13	16.66667
Jakub Müller	9	13	16.66667
Táňa Poláková	13	13	16.66667
Dávid Silady	13	13	16.66667
Adam Šípka	19	13	16.66667

## Šprint č. 3

Trvanie: 09. 11. 2021 – 23. 11. 2021

Čo hodnotíme pozitívne?

- Máme rozšírené komponenty, majú viditeľnejšiu funkcionálnosť.
- Prelúskali sme už väčšinu technológií.
- Aklimatizovali sme sa na metodiku scrum.

Čo chceme zlepšiť?

- Chceme, aby sa jednotliví členovia tímu venovali viac svojim rolám.
- Chceme zabezpečiť, aby bol každý príbeh viac zdokumentovaný.
- Chceme, aby sa každý staral o svoj príbeh a v komentároch v nástroji Jira popísal, čo urobil. Taktiež do komentáru napíše komentár od vedúceho ako záznam zo stretnutia.
- Mali by sme uvažovať o pár týždňov dopredu pri plánovaní príbehov aby sa nestalo, že na poslednú chvíľu nám pribudne veľa nových – ako tomu bolo v tomto šprinte.

Názov príbehu	Riešiteľ	Odhadovaný počet bodov	Je príbeh ukončený?	Pridelený počet bodov
Napĺňanie MongoDB priamo pri scrapovaní	Adam Šípka	2	áno	3
Zabrániť zacykleným buildom	Dominik Horváth	5	áno	3
Rozšíriť úložisko pomocou nepriradeného disku	Dominik Horváth	5	áno	3
Integrácia scraperu a vylepšeného parseru	Jakub Müller	5	áno	8
Integrácia MongoDB a Elasticsearch	Jakub Hlavačka	8	áno	8
Úprava dokumentácie zo stretnutí a šprintov tak, aby mohli ísť na stránku tímu	Táňa Poláková	8	áno	8
Zabezpečenie komunikácie medzi klientom a serverom	David Silady	8	áno	8
Analýza možností implementácie testovania	Adam Šípka	5	áno	5

Návrh Flask API	Táňa Poláková	5	áno	5
Implementácia dummy Flask API	Dominik Horváth	5	áno	5
Rozšírenie docker compose o MongoDB a Elasticsearch	Jakub Hlavačka	5	áno	3
Dokumentácia k inžinierskemu dielu	Jakub Hlavačka	5	áno	5
Dokumentácia k inžinierskemu dielu	Dominik Horváth	3	áno	3
Dokumentácia k inžinierskemu dielu	Jakub Müller	5	áno	5
Dokumentácia k inžinierskemu dielu	David Silady	5	áno	5
Dokumentácia k riadeniu projektu	Táňa Poláková	5	áno	5
Dokumentácia k riadeniu projektu	Adam Šípka	5	áno	5

Meno	Súčet počtu odhadovaných bodov	Súčet počtu pridelených bodov	Podiel práce (%)
Jakub Hlavačka	18	13	16.66667
Dominik Horváth	18	13	16.66667
Jakub Müller	10	13	16.66667
Táňa Poláková	18	13	16.66667
Dávid Silady	13	13	16.66667
Adam Šípka	12	13	16.66667

## Manažment tímu

Ako sme už vyššie uviedli, nedá sa povedať, že v začiatkoch semestra by sme mali nejakú formu manažmentu tímu. V tejto chvíli už však fungujeme nasledovne:

### Každý člen tímu je zodpovedný za svoju rolu

Čo to znamená? Napríklad, že scrum master sa postará o to, aby mal každý čo robiť. Je potrebné, aby kontroloval, či sa členovia tímu starajú o svoje príbehy v nástroji Jira. Určili sme si, že scrum master nebude stáť nad členmi tímu a kontrolovať, či niekto urobil menej, alebo viac. Za toto už berie každý osobnú zodpovednosť. Ak niekto vidí, že odrobil príbehy rýchlejšie, ako sme predpokladali, ohlási to ostatným členom tímu a spoločne sa vymyslí ďalší príbeh.

Manažér plánovania je zodpovedný za to, aby sme nevybočovali s pôvodného plánu a aby sme boli schopní dosiahnuť malé ciele, ktoré sme si vopred definovali. Mal by sledovať, či jednotlivé príbehy vedú k naplneniu plánu, alebo nie.

Manažér kvality dohliada na každý vyprodukovaný kus kódu a snaží sa z autora vyťažiť čo najvyššiu kvalitu. Domnievame sa, že je to istou formou aj psychologický efekt. Človek, ktorý si je istý, že jeho kód nikto iný okrem neho nebude vidieť, ho s veľkou pravdepodobnosťou odflákne. Ak ale kód bude posudzovať niekto ďalší, pôvodný autor sa bude viac snažiť.

Manažér komunikácie predchádza nedorozumeniam zapríčineným zlou komunikáciou. Ak je potrebné niečo odkomunikovať s vedúcim tímu na diaľku, postará sa o to.

Hlavný architekt má posledné slovo, pokiaľ sa jedná o architektúru systému. Je odborníkom vo svojej oblasti a pomáha ostatným členom tímu navrhnuť architektúru čo najlepšie.

Manažér testovania rozhoduje, ktoré časti systému sa dajú testovať. Odporúča, aké typy testov je vhodné implementovať a dohliadne, aby „doručené“ časti kódu boli vhodne otestované.

Manažéri dokumentácie sa postarajú o to, aby sa dokumentácia písala priebežne (!). Toto je naozaj oblasť, kde si chceme dať obzvlášť veľký pozor. Písať dokumentáciu na poslednú chvíľu je veľmi nepríjemné.

### Na riadenie projektu využívame metodiku Scrum

Táto metodika je formálne opísaná v prílohe Metodiky. Scrum sme si pre potreby projektu prispôbili. Naše šprinty trvajú dva týždne, pričom sa snažíme pri plánovaní šprintu o to, aby každý člen odrobil približne 13 hodín. Je nám jasné, že budeme čeliť aj istým odchýlkam, avšak tieto rozdiely si navzájom akceptujeme. Body, ktoré jednotlivým príbehom prideliujeme zodpovedajú náročnosti úlohy. V našom prípade sa náročnosť rovná počtu predpokladaných hodín, ktoré človek odpracuje. Ekvivalenciu bodov a hodín sme si prispôbili kvôli lepšiemu plánovaniu. Na konci šprintu každý člen tímu zodpovie na otázky, čo hodnotí na šprinte ako pozitívne a naopak, čo by chcel zlepšiť. V rámci zhodnotenia šprintu taktiež uvádzame percentuálny podiel členov na výsledku šprintu. Podiel počítame ako celkový počet získaných bodov za príbehy a celkový počet bodov, ktoré tím mohol



získať, a to je šesťnásobok čísla 13. Zároveň, podiel je aplikovaný aj z pohľadu všetkých šprintov, a tak vieme zhodnotiť, či človek, ktorý počas jedného šprintu odrobil menej, dobehol hodiny v inom šprinte.

## Sme tímoví hráči

Motivujeme a rešpektujeme sa. Veríme, že príjemný kolektív je základ dobre odvedenej práce. Ak niečo nefunguje, nezvaľujeme na seba vinu navzájom. Namiesto toho sa snažíme pochopiť, odkiaľ problém prišiel a nájsť spoločne riešenie, ktoré v rámci možností vyhovuje každému z nás.

# Motivačný dokument

## Tím

Náš tím tvoria perspektívni ľudia, ktorých záujmy pokrývajú rôzne oblasti informatiky. Poznáme sa už od strednej školy. Počas štúdia sme spolupracovali na rôznych projektoch, čo nás naučilo hľadať efektívne kompromisy. Zameriavame sa najmä na dátovú vedu, webové technológie a umelú inteligenciu, čomu zodpovedajú aj naše absolvované predmety na bakalárskom stupni štúdia (Vývoj aplikácií v jazyku JavaScript, Webové technológie, Inteligentná analýza údajov). Niektorí z nás si tieto vedomosti plánujú rozšíriť na predmetoch Vývoj webových aplikácií v prostredí cloud a Neurónové siete.

Väčšina členov nášho tímu aktuálne pôsobí v zamestnaní, ktoré im dáva cennú informatickú prax. Dávid Silady a Dominik Horváth sú kolegovia vo firme IBM, kde pracujú na pozíciách Package specialist a DevOps inžinier. Automatizácia procesov, Docker, Kubernetes, Bash alebo Python sú častou súčasťou ich práce. Táňa Poláková má hlavnú náplň práce zber a predspracovanie veľkého množstva dát (využíva MySQL, numpy), vďaka ktorým bude môcť navrhnúť model neurónovej siete, ktorý nahradí aktuálne riešenie. Jakub Hlavačka v práci často využíva technológie Scrapy a Selenium na scrapovanie údajov z internetu.

Naše bakalárske práce sa zaoberali dátovou analýzou, umelou inteligenciou, technológiou blockchain a webom. Veľkou hodnotou (tak, ako aj ostatní členovia) v našom tíme je Dominik Horváth, ktorého bakalársky projekt bol zameraný na dátovú analýzu falošných správ súvisiacich s pandemiou COVID-19, a jeho práca bola jednou z prvých, ktoré sa venovali tejto oblasti. Daná tematika sa priamo dotýka témy na tímový projekt, o ktorý máme najväčší záujem. Táňa Poláková a Dávid Silady sa na svojich bakalárkach zaoberali neurónovými sieťami (pracovali s knižnicami TensorFlow a PyTorch). Dávid odhadoval hĺbku z fotografií pomocou neurónových sietí, kde sa zoznámil najmä s generovaním a spracovaním dát, rovnako ako so základmi konvolučných sietí. Táňa trénovala neurónovú sieť, ktorej výstup nezávisel od usporiadania prvkov vo vstupe, pričom vstup boli medicínske dáta. Práca Jakuba Hlavačku bola zameraná na využitie technológie blockchain v IoT zariadeniach, kde nadobudol prvé väčšie znalosti so systémami ako Docker a Hyperledger Fabric. Jakub Müller má zo svojej bakalárskej práce skúsenosti z oblasti tvorby webových stránok a UX testovania, keďže jeho práca bola zameraná na používateľmi vnímanú dôveryhodnosť v online spravodajstve. Adam Šípka v bakalárke vytvoril webový crawler (použil framework Selenium), ktorý zaznamenával a ukladal metadáta o YouTube videách. Vďaka tomu získal skúsenosti so získavaním dát a následne aj s ich prespracovaním a analýzou.

Každý z nás má vášeň pre web. Dávid obľubuje prácu s webovými technológiami ako React, Nextjs a Express(Nodejs). Dominik preferuje frameworky ako Flask alebo Django a baví ho bezpečnosť webových technológií. Táňa sa rada kreatívne vyžije pri tvorbe webového front-endu (pomocou frameworku React), avšak má skúsenosti aj s back-endom (Nodejs). Jakub H. sa najviac zaujíma o vytváranie webových aplikácií s prístupom Fullstack za pomoci technológií ako Express, React, Flask, Bootstrap, SQL. Jakub M. sa najradšej angažuje v oblasti front-endu a má skúsenosti s tvorbou webových aplikácií. Adam sa v jednom zo svojich projektov venoval parsovaniu dát z rôznych webových stránok.

# Motivácia

## Téma č. 4 - Adverse Media Screening [AMS]

Tému Adverse Media Screening [AMS] sme si vybrali najmä kvôli zaujímavosti a rôznorodosti infromatických úloh, ktoré budú pri jej vývoji potrebné. Vďaka väčšinovému zameraniu tímu na webové technológie nám nebude robiť problém tvorba front-end a back-end rozhrania. Niektorí členovia tímu majú taktiež skúsenosti so "scrapingom" rôznych webových portálov (pomocou technológií Scrapy a Selenium) na ďalšie spracovanie dát z nich, čo je pri tejto téme dôležitá skúsenosť.

Každý člen tímu má v prvom semestri inžinierskeho ročníka predmety Vyhľadávanie informácií a Neurónové siete, čo nám výrazne pomôže pri problematike vybranej témy. Svoju kvalifikáciu doplníme absolvovanými predmetmi ako Inteligentná analýza údajov, Webové technológie a Vývoj aplikácií v JavaScript.

Zameranie bakalárskej práce dvoch členov nášho tímu bolo na neurónové siete a jeden z členov pracoval na podobnej téme ako je téma AMS vo svojej bakalárskej práci, kde využil technológiu Elasticsearch pre vyhľadávanie informácií vo veľkej dátovej sade.

## Téma č. 8 - Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]

Táto téma sa taktiež z veľkej časti opiera o technológie, s ktorými sme všetci nielen stotožnení, ale naozaj nás bavia. Tiež sa jedná o oblasť, ktorá nám je ako študentom veľmi blízka, a bolo by zaujímavé prispieť do procesu testovania svojou prácou, ktorá potenciálne môže priniesť niečo nové. Skúsenosti tímu s webovými technológiami, DevOps a taktiež poznatky z prostredia veľkých projektov v pracovnom svete budú v tomto projekte veľmi cenné.

## Príloha A - Zoradenie všetkých tém podľa priority

V tabuľke nižšie sa nachádza zoradenie 12-ich tém podľa priority.

Priorita témy	Číslo témy	Názov témy
1.	4.	Adverse Media Screening [AMS]
2.	8.	Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]
3.	16.	FIIT WIX
4.	17.	Document Wizard
5.	3.	DataHub pre rôzne typy zariadení, ich spracovanie / analýzu / vizualizáciu
6.	11.	(Q)SAR analýza fototoxických látok
7.	15.	Ion Mobility Spectrometry for Rapid HEMP Potency Testing
8.	14.	IoT platforma na priemyselnú automatizáciu - malý pivovar
9.	2.	Transformácia priestorov na bezpečné a inteligentné miesta na prácu [space2]
10.	9.	Monitorovanie a správa systému pre výrobný areál [LOMON]
11.	18.	Webové IDE pre ASIC [ASICDE]
12.	19.	Automatizácia procesov KYC (Know your client) a AML [Anti-money laundering]

## Príloha B - Rozvrh tímu

V prílohe sa nachádzajú povinnosti jednotlivých členov nášho tímu. Stretnutia tímu s vedúcim a spoločnú prácu tímu sme stanovili nasledovne:

- Stretnutie tímu s vedúcim: Utorok 11:00 - 14:00
- Spoločná práca tímu: Pondelok 15:00 - 18:00

Sme otvorení iným možnostiam na stretnutie s vedúcim v prípade, že by náš potenciálny vedúci nebol v náš preferovaný čas k dispozícii.

### Dominik Horváth

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Po	1.28 (LIS) (BA-FIIT-FIIT) Vyhľadavanie informácií (1) M. Kováč								Spoločná práca tímu			1.38 (U20b) (BA-FIIT-FIIT) Architektúra softvéru (1) L. Graf
Ut	Stretnutie tímu s vedúcim						-1.61 (Aula Magna) (BA-FIIT-FIIT) Architektúra softvéru V. Vranic	-1.61 (Aula Magna) (BA-FIIT-FIIT) Výskum inteligentných softvérových systémov V. Vranic			-1.61 (Aula Magna) (BA-FIIT-FIIT) Timový projekt I (2) K. Košťál	
St			-1.58 (U120) (BA-FIIT-FIIT) Vyhľadavanie informácií (3) M. Kováč					-1.61 (Aula Magna) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) I. Černáková				1.29 (LWT) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) I. Černáková
St	-1.58 (U120) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko		1.30a (LSS1) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko									
Pi												

### Jakub Hlavačka

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50	
Po			1.28 (LIS) (BA-FIIT-FIIT) Vyhľadavanie informácií (1) M. Kováč								Spoločná práca tímu		1.38 (U20b) (BA-FIIT-FIIT) Architektúra softvéru (1) L. Graf
Ut	Stretnutie tímu s vedúcim						-1.61 (Aula Magna) (BA-FIIT-FIIT) Architektúra softvéru V. Vranic	-1.61 (Aula Magna) (BA-FIIT-FIIT) Výskum inteligentných softvérových systémov V. Vranic			-1.61 (Aula Magna) (BA-FIIT-FIIT) Timový projekt I (2) K. Košťál		
St			-1.58 (U120) (BA-FIIT-FIIT) Vyhľadavanie informácií (3) M. Kováč					-1.61 (Aula Magna) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) I. Černáková				1.28 (LIS) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) F. Lehocki	
St	-1.58 (U120) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko		1.30a (LSS1) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko										
Pi													

### Táňa Poláková

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50	
Po									Spoločná práca tímu			1.38 (U20b) (BA-FIIT-FIIT) Architektúra softvéru (1) L. Graf	
Ut	1.28 (LIS) (BA-FIIT-FIIT) Vyhľadavanie informácií M. Kováč		Stretnutie tímu s vedúcim						-1.61 (Aula Magna) (BA-FIIT-FIIT) Architektúra softvéru V. Vranic	-1.61 (Aula Magna) (BA-FIIT-FIIT) Výskum inteligentných softvérových systémov V. Vranic			-1.61 (Aula Magna) (BA-FIIT-FIIT) Timový projekt I (2) K. Košťál
St			-1.58 (U120) (BA-FIIT-FIIT) Vyhľadavanie informácií (3) M. Kováč					-1.61 (Aula Magna) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) I. Černáková				1.29 (LWT) (BA-FIIT-FIIT) Manažment v tvorbe softvéru (3) I. Černáková	
St	-1.58 (U120) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko		1.30a (LSS1) (BA-FIIT-FIIT) Základy kryptografie V. Khylenko										
Pi													

### Jakub Müller



# Metodiky

## Metodika Scrum

Na riadenie projektu využívame metodiku SCRUM, ktorú sme si prispôsobili pre potreby nášho projektu. Dodržiavanie princípov má na starosti SCRUM master, ktorý okrem toho prideluje úlohy ostatným členom tímu v nástroji Jira a celkovo spravuje tento nástroj. Naše šprinty trvajú dva týždne a začínajú od stredy 0:00, deň po stretnutí s vedúcim projektu.

## Standup

Každý pondelok, deň pred stretnutím s vedúcim, prebehne stretnutie tímu, kde sa všetci členovia tímu vyjadria ako prebieha plnenie im pridelených úloh, s primárnym zameraním na problémy, ktoré sa vyskytli. Následne prebehne diskusia s cieľom vyriešiť identifikované problémy.

## Šprint review

V posledný deň šprintu, pri stretnutí s vedúcim (v utorok), sa vykoná šprint review, kde sa na začiatku stretnutia každý člen vyjadrí k svojim úlohám, ako sa mu ich podarilo splniť, prípadne opísať problémy, ktoré sa pri plnení úlohy vyskytli. Cieľom je vyvolať diskusiu, ktorá by nám pomohla posunúť sa ďalej a vyriešiť daný problém. Ak úloha ešte nebola vyriešená, tak sa prenesie do nasledujúceho šprintu. Okrem toho každý člen povie, koľko času na daných úlohách strávil. Strávený čas sa následne porovná s odhadovaným časom, čo nás naučí tvoriť presnejšie odhady.

Po vyjadreniach od všetkých členov sa zhodnotí samotný šprint ako celok. Členovia sa vyjadria, čo hodnotia ako pozitívne, alebo naopak, v akých oblastiach sa potrebujú zlepšiť.

## Plánovanie

V deň stretnutia, po tom, čo prebehne šprint review, sa naplánujú úlohy na nasledujúci šprint. Novým príbehom prideliujeme kartičky s hodnotami 0.5, 1, 2, 3, 5, 8, 13 a ?, ktoré reprezentujú náročnosť úlohy. V našom prípade sa náročnosť rovná počtu predpokladaných hodín, ktoré daný člen pri riešení úlohy odpracuje. Takto postupujeme kvôli ľahšiemu plánovaniu. Cieľom je, aby každý člen odpracoval za jeden šprint 12 hodín (6 na týždeň). Táto hodnota nemusí byť nutne dodržaná, je možné, že v niektorých prípadoch bude úloha vyriešená za kratší, alebo dlhší čas.

Pri pridelovaní sa uprednostňujú úlohy, ktoré nás posunú bližšie k cieľu projektu. Ak nastane situácia, kde momentálne nebudeme schopný vytvoriť úlohu a niektorý z členov bude mať pridelenú prácu s príliš nízkym počtom hodín, bude mu priradená úloha z backlogu s najvyššou prioritou.



## Práca v nástroji Jira

Pri práci používame nástroj Jira, s ktorým pracuje primárne náš SCRUM master. Pridelované a evidované tasky sú rozdelené do nasledovných kategórií:

- Biznis analýza
- Front end
- Back end - server
- Back end - data
- Softvérová architektúra
- Stránka tímu

Práca v nástroji Jira je podrobnejšie popísaná v nasledujúcich sekciách.

### Backlog

V backlogu sa nachádzajú tasky, o ktorých vieme, že v budúcnosti sa budú musieť implementovať, no v momentálnej dobe nie sú prioritné. Tvorenie nových taskov do backlogu prebieha na stretnutiach tímu, alebo na stretnutiach s vedúcim, kde počas plánovania nového šprintu môže byť niektorá z úloh presunutá do backlogu, ak sa všetci členovia zhodnú, že momentálne existujú problémy s vyššou prioritou.

### Nástenka

Na nástenke sú zobrazené tasky aktuálneho šprintu. Sú rozdelené do troch stĺpcov podľa ich aktuálneho stavu riešenia (to do, in progress a done). Pri každom tasku je v detailoch uvedený odhadovaný čas. Člen tímu, ktorému je daná úloha pridelená priebežne aktualizuje odrobený čas, ktorý sa pri šprint review bude porovnávať s odhadom. Okrem toho do komentára k danému tasku napíše krátke zhrnutie jeho výsledkov, prípadne problémov, ktoré sa vyskytli.

# Git Metodika

Na verziovanie zdrojového kódu je používaný Git. Náš tím má na GitHubu vytvorenú organizáciu FIIT-TEAM8, kde sú v jednotlivých repozitároch uložené zdrojové kódy danej časti projektu. Aktuálny zoznam repozitárov je nasledovný:

- application web
- script na prepojenie Elasticsearch s Mongo databázou
- webová stránka tímu
- scraper napojený na Mongo databázu
- flask server

Pri verziovaní kódu sú uplatnené praktiky, ktoré opisujeme v nasledujúcich sekciách.

## Vetvenie

Main branch je určená na aktuálnu verziu funkčného kódu. Nie je určená na samotný vývoj alebo implementáciu nových funkcií, tie prebiehajú v oddelených vetvách, ktoré sú vytvorené členom tímu, ktorý na dane úlohe pracuje. Meno takejto vetvy musí popisovať cieľ implementovanej funkcionality. Vďaka vetveniu môžu na rovnakej časti projektu pracovať viacerí členovia tímu súčasne a bez toho, aby došlo k veľkým konfliktom v zdrojovom kóde.

## Commit

Commity si vytvára každý člen tímu na svojej vetve, ale napriek tomu, by mali mať zmysluplný dôvod a mali by byť tvorené až po implementovaní určitej časti kódu, alebo opravení zistených chýb. Commit message je písaný v anglickom jazyku a je dostatočne deskriptívny, no bez toho, aby bol príliš rozsiahly.

## Pull requests

Po úspešnom implementovaní funkcionality, alebo odstránení problému môže člen tímu vytvoriť pull request, aby aktualizoval kód v repozitári. Merge konflikty, ktoré sa v takýchto situáciách môžu niekedy vyskytnúť, je nutné pred uzavretím pull requestu odstrániť bez toho, aby utrpela funkcionality programu.

## Export evidencie úloh

V nasledujúcej tabuľke sa nachádza evidencia jednotlivých príbehov k dňu ukončenia tretieho šprintu (23. 11. 2021). Export obsahuje všetky doposiaľ plánované, ukončené, zrušené a aktuálne plnené príbehy. Body sa nenachádzajú pri každom príbehu, nakoľko niektoré z nich sú v backlogu a niektoré z nich vznikli ešte v čase, kedy sme body nepoužívali.

Issue Type	Summary	Story points estimate	Assignee	Status	Created	Resolved	Sprint	Sprint_1
Story	Rozebnuť PostgreSQL	3	Jakub Hlavka	To Do	23-11-21 14:50		AMS Sprint 4	
Story	Komunikácia medzi aplikacným serverom a API serverom	5	David Sliady	To Do	23-11-21 14:46		AMS Sprint 4	
Story	Komunikácia medzi API serverom a Elastic Search	13	Jakub Müller	To Do	23-11-21 14:45		AMS Sprint 4	
Story	Zobrazenie výsledkov vyhľadávania	13	Táňa Poláková	To Do	23-11-21 14:43		AMS Sprint 4	
Story	Komunikácia medzi MongoDB a API serverom	13	Adam Šipka	To Do	23-11-21 14:42		AMS Sprint 4	
Story	Indexovanie stiahnutých dát	8	Jakub Hlavka	To Do	23-11-21 14:41		AMS Sprint 4	
Story	Fuzzy search			To Do	23-11-21 14:40			
Story	Analýza log sash na update Elasticsearch z MongoDB a analýza elastic-ym			To Do	22-11-21 11:55			
Story	Zabezpečiť elasticsearch a mongodb kontajner			To Do	21-11-21 19:17			
Story	Deployment scraper	13	Dominik Horvath	To Do	15-11-21 13:08		AMS Sprint 4	
Story	PDF report z nájdených článkov			To Do	02-11-21 14:50			
Story	Príhásenie sa na odber nejakeho vyhľadaneého človeka			To Do	02-11-21 14:50			
Story	Označenie článku ako vídeného (prečítaného)			To Do	02-11-21 14:49			
Story	Hodnotenie risku pri nájdených článkoch			To Do	02-11-21 14:47			
Story	Zvýraznenie hľadanych slovíčok			To Do	02-11-21 14:45			
Story	Filter na web stránke			To Do	02-11-21 14:32			
Story	Technický návrh developerského prostredia	8	David Sliady	To Do	02-11-21 13:50		AMS Sprint 4	
Story	Analýza medií, ktoré dostaneme vo výstupoch a aké search queries treba poslať, aby sme mali výsledky ako z Google			To Do	02-11-21 13:25			
Story	Pozrieť sa na zahodané články, kt. neobsahovali paragraphy			To Do	02-11-21 13:23			
Story	Porovnanie výsledkov nášho vyhľadávania s výsledkami Google News			To Do	13-10-21 14:54			
Story	Vyriešiť, čo s článkami, ktoré budú v pôvodnom zdroji nekompletné			To Do	13-10-21 14:47			
Story	Zistiť, ako fungujú iné zdroje Google News, prípadne aký mechanizmus používa Google News.		Adam Šipka	To Do	13-10-21 14:31			
Story	Zistiť, čo spája kriminálne články			To Do	13-10-21 14:13			
Story	Analýza - multilingvistika			To Do	05-10-21 13:37			
Story	Násu API dát na portál verejných API v rámci propagácie			To Do	05-10-21 13:22			

Story	User databáza + zberanie aktivity používateľa		Táňa Poláková	IN PROGRESS		13-10-21 14:29			
Story	Lofi používateľského rozhrania		David Sliady	IN PROGRESS		05-10-21 13:20			
Story	Dokumentácia k riadeniu projektu		Táňa Poláková	Done		23-11-21 9:00	23-11-21 20:39	AMS Sprint 3	
Story	Dokumentácia k inžinierskemu dielu		Jakub Hlavačka	Done		23-11-21 9:00	23-11-21 19:32	AMS Sprint 3	
Story	Rozšírenie docker compose o MongoDB a Elasticsearch		Jakub Hlavačka	Done		15-11-21 13:16	21-11-21 19:16	AMS Sprint 3	
Story	Implementácia dummy Flask API		Dominik Horvath	Done		09-11-21 14:45	15-11-21 19:43	AMS Sprint 3	
Story	Návrh Flask API		Táňa Poláková	Done		09-11-21 14:44	15-11-21 12:50	AMS Sprint 3	
Story	Analýza možnosti implementácie testovania		Adam Šipka	Done		09-11-21 14:43	23-11-21 20:39	AMS Sprint 3	
Story	Zabezpečenie komunikácie medzi klientom a serverom		David Sliady	Done		09-11-21 14:38	23-11-21 14:36	AMS Sprint 3	
Story	Uprava dokumentácie zo stretnutí a šprintov tak, aby mohli ísť na stránku tímu		Táňa Poláková	Done		09-11-21 14:31	23-11-21 8:51	AMS Sprint 3	
Story	Integrácia MongoDB a Elasticsearch		Jakub Hlavačka	Done		09-11-21 14:22	16-11-21 12:54	AMS Sprint 3	
Story	Integrácia scraperu a vylepšenie parsera		Jakub Müller	Done		09-11-21 14:12	22-11-21 11:42	AMS Sprint 3	
Story	Rozšíriť úložisko pomocou nepriradeného disku		Dominik Horvath	Done		09-11-21 14:07	14-11-21 8:30	AMS Sprint 3	
Story	Zabrániť začleneným buildom		Dominik Horvath	Done		07-11-21 23:07	15-11-21 8:22	AMS Sprint 3	
Story	Vyhľadanie RSS dvojčíta k HTML článku - všeobecný postup		Dominik Horvath	Done		02-11-21 14:08	09-11-21 9:45	AMS Sprint 2	
Story	Naplnenie MongoDB priamo pri scrapovaní		Adam Šipka	Done		02-11-21 13:44	14-11-21 14:06	AMS Sprint 2	AMS Sprint 3
Story	Rozšírenie google news scraperu na lokáciu UK		Dominik Horvath	Done		02-11-21 13:42	02-11-21 16:53	AMS Sprint 2	
Story	Pamäťová zložitosť elastic search na vzorke dát		Jakub Hlavačka	Done		01-11-21 12:15	12-11-21 17:09	AMS Sprint 2	AMS Sprint 3
Story	Parsovanie tagov pomocou lxml, readability, traflatura		Táňa Poláková	Done		30-10-21 11:41	09-11-21 13:12	AMS Sprint 2	
Story	Zavesenie klienta		David Sliady	Done		27-10-21 14:45	08-11-21 0:51	AMS Sprint 2	
Story	Rozšíriť docker compose		Dominik Horvath	Done		27-10-21 14:43	31-10-21 20:22	AMS Sprint 2	
Story	Pridať info o projekte a opísať členov tímu		Jakub Müller	Done		27-10-21 14:42	01-11-21 11:37	AMS Sprint 2	
Story	Request URL s odpoveďou inou ako 200 uložiť do súboru		Adam Šipka	Done		27-10-21 14:40	01-11-21 11:35	AMS Sprint 2	
Story	Lokálna MongoDB so získanými dátami		Adam Šipka	Done		27-10-21 14:35	01-11-21 11:35	AMS Sprint 2	
Story	Analýza bezstratovej kompresie textu		Jakub Hlavačka	Done		27-10-21 14:25	01-11-21 12:25	AMS Sprint 2	
Story	Parsovanie tagov pomocou regex		Táňa Poláková	Done		27-10-21 14:19	02-11-21 13:49	AMS Sprint 2	
Story	Parsovanie tagov pomocou CSS Selector		Jakub Müller	Done		27-10-21 14:17	08-11-21 19:48	AMS Sprint 2	
Story	Stránka tímu		Jakub Müller	Done		18-10-21 15:03	01-11-21 11:37	AMS Sprint 1	
Story	Otvorenie portov na serveri		David Sliady	Done		18-10-21 15:01	18-10-21 17:05	AMS Sprint 1	

Story	Vyriešiť, aby fiktívkar nemohol urobiť "sudo su" na virtuálnom stroji (aby nemal šancu sa zmeniť na roota)		Táňa Poláková	Done	13-10-21 14:34	19-10-21 18:04	AMS Sprint 1	
Story	Prieskum databáz		Jakub Müller	Done	13-10-21 14:26	18-10-21 14:28	AMS Sprint 1	
Story	Vytvorí github projekt		Jakub Müller	Done	13-10-21 14:22	18-10-21 12:09	AMS Sprint 1	
Story	Príháška na TP CUP			Done	13-10-21 14:13	02-11-21 12:50	AMS Sprint 1	
Story	Set up elastic search v docker container		Jakub Hlavačka	Done	13-10-21 14:12	16-10-21 16:45	AMS Sprint 1	
Story	Získanie dát pre prototyp		Dominik Horvath	Done	13-10-21 14:09	26-10-21 13:29	AMS Sprint 1	
Story	Vytvorí Slack server		Táňa Poláková	Done	05-10-21 13:49	11-10-21 12:08		
Story	Analýza parametrov Google News		Adam Šipka	Done	05-10-21 13:43	13-10-21 14:26		
Story	Reálny scrapping - vytvorenie vzorky		Dominik Horvath	Done	05-10-21 13:27	10-10-21 18:14		
Story	Zistiť, kore miestnosti su na karticky		David Sliady	Done	05-10-21 13:24	18-10-21 16:12		
Story	Revízia požiadaviek		Táňa Poláková	Done	05-10-21 13:19	13-10-21 14:05		
Story	Spísať denník z prvého stretnutia s vedúcim		Táňa Poláková	Done	03-10-21 15:50	05-10-21 13:30		
Story	Případy použita		Dominik Horvath	Done	01-10-21 14:23	05-10-21 12:08		
Story	Prieskum získavania dát, práca s knižnicou		Adam Šipka	Done	01-10-21 14:22	05-10-21 13:24		
Story	Specifikácia požiadaviek		Jakub Hlavačka	Done	01-10-21 14:22	03-10-21 15:17		
Story	Vysoka architektura		David Sliady	Done	01-10-21 14:21	03-10-21 14:56		
Story	Zaoberať stroj v škole		Táňa Poláková	Done	01-10-21 14:18	13-10-21 14:23		
Story	Zoznam trestných činov, ktoré budu použite ako queries na google news		Jakub Müller	Done	01-10-21 14:18	04-10-21 21:20		