Plankweb

Rozšírenie projektu Prankweb

16.12.2024

Slovník pojmov

- **Aminokyselina**: Organická molekula, ktorá je základnou jednotkou proteínov; pozostáva z amínovej skupiny, karboxylovej skupiny a charakteristickej bočnej skupiny (R-skupiny).
- **Proteín**: Makromolekula zložená z jednej alebo viacerých polypeptidových reťazcov, ktoré sa skladajú z aminokyselín.
- **Reziduum**: Jeden konkrétny aminokyselinový zvyšok v sekvencii proteínu, často sa používa na označenie aminokyselín v kontexte proteínovej štruktúry.
- **Binding (site)**: Väzobné miesto proteínu
- **Ligand**: Molekula alebo ión, ktorý sa viaže na proteín, často za účelom ovplyvnenia jeho funkcie alebo štruktúry.
- **Konformácia**: Priestorové usporiadanie atómov v molekule, ktoré môže ovplyvniť jej funkciu; v prípade proteínov ide o trojrozmernú štruktúru, ktorá vzniká ohýbaním a skladaním polypeptidového reťazca.
- **APO**: Apoproteín označuje proteín bez viazaného ligandu alebo koenzýmu.
- **HOLO**: Holoproteín označuje proteín, ktorý je kompletný s viazaným ligandom alebo koenzýmom.
- **PDB**: Skratka pre "Protein Data Bank," databáza, ktorá uchováva experimentálne zistené 3D štruktúry proteínov, nukleových kyselín a komplexov biomolekúl.
- **PDB ID**: Identifikačný kód priradený každému záznamu na jednoznačnú identifikáciu konkrétneho štrukturálneho záznamu.
- **UniProt**: Skratka pre "Universal Protein Resource," rozsiahla databáza, ktorá poskytuje rôzne anotácie pre proteínové sekvencie.
- **UniProt ID**: Identifikačný kód priradený každému proteínu na jednoznačnú identifikáciu konkrétneho proteínu a jeho súvisiacich informácií.
- **Pocket**: Špecifická oblasť na povrchu proteínu, ktorá môže viazať ligandy.
- **pLM**: Skratka pre “protein Language Model”, označuje jazykový model trénovaný na štruktúrach proteínov.
- **Query proteín**: Proteín zadaný užívateľom na vstupe.
- **Superpozícia proteínových štruktúr**: Technika, pri ktorej sa prekrývajú a porovnávajú 3D štruktúry proteínov, aby sa identifikovali podobnosti a rozdiely

medzi nimi. Ide o nájdenie najlepšieho možného prekrytia štruktúr, pričom sa minimalizujú rozdiely medzi atómovými pozíciami.

Úvod

Prankweb je webová aplikácia, ktorej hlavnou funkcionalitou je predikovanie a vizualizácia väzobných miest ligandov na základe štruktúr proteínov. Cieľom nášho projektu je rozšíriť túto webovú aplikáciu. Pôjde o rozšírenie v dvoch oblastiach - predikcie a analýzy.

Rozšírenie predikcie

V súčasnosti Prankweb dokáže predikovať väzobné miesta v proteínovej štruktúre pomocou metódy P2Rank, pričom vstupom je vždy 3D štruktúra proteínu. Cieľom rozšírenia je pridať možnosť predikcie na základe sekvencie a jej prevod do 3D štruktúry.

Na predikciu väzobných miest zo sekvencie sa využije existujúci Protein Language Model (pLM), ktorý vygeneruje tzv. embeddings – vektory opisujúce vlastnosti jednotlivých aminokyselín v závislosti na ich sekvenčnom kontexte. Každý embedding bude vstupom pre klasifikačnú neurónovú sieť, ktorá rozhodne, či daná aminokyselina je súčasťou nejakého väzobného miesta.

Po zadaní sekvencie proteínu bude na vizualizáciu štruktúry, a takisto na predikovanie väzobných miest pomocou P2Ranku, potrebný prevod zadanej sekvencie na 3D štruktúru. Na prevod bude využitá už existujúca predikčná metóda.

Rozšírenie analýzy

P2Rank a pLM sa dajú chápať ako zdroje dát, ktoré pre vstupný proteín predikujú väzobné miesta. V súčasnosti existujú rôzne ďalšie zdroje dát. Buď také, ktoré pre vstupný proteín vrátia experimentálne zistené väzobné miesta s konkrétnymi ligandmi, alebo také, ktoré dokážu nájsť proteíny, ktoré sú vstupnému proteínu podobné 3D štruktúrou a obsahujú informácie o väzobných miestach a ligandoch. Tieto získané informácie umožnia získať nové poznatky o vstupnom proteíne.

Rozšírenie analýzy sa skladá z 2 častí:

- Návrh **systému pre prácu s rôznymi dátovými zdrojmi**, ktorý bude jednoducho rozšíriteľný o nové dátové zdroje.
- Nové užívateľské rozhranie, **analytická stránka**, ktoré umožní vizualizáciu dát získaných z dátových zdrojov.

Systém pre prácu s dátovými zdrojmi

Systém zabezpečí celý priebeh od spracovania vstupu od užívateľa, získania dát z rôznych dátových zdrojov až po poskytnutie dát analytickej stránke. Súčasťou vytvorenia systému bude:

- Vytvorenie modulu, ktorý vytvorí identifikátor pre vstupný proteín a zabezpečí aby neprebehol výpočet pre opakované identické vstupy.
- Vytvorenie modulov s jednotným rozhraním pre prácu s dátovými zdrojmi.
- Návrh jednotného formátu pre dáta získané z dátových zdrojov a poskytnúť dáta v danom formáte užívateľskému rozhraniu – analytickej stránke.

Analytická stránka

Úlohou užívateľského rozhrania v kontexte rozšírenia analýzy, resp. analytickej stránky, je prezentácia dát poskytnutých užívateľom a dátovými zdrojmi. Pôjde o 2 druhy vizualizácii:

- Sekvenčný displej
- Štruktúrny displej

Sekvenčný displej má za úlohu vykresliť sekvenciu zadanú užívateľom (spolu s informáciami o nej) a taktiež aj dáta poskytnuté dátovými zdrojmi. Ak niektorý z dátových zdrojov bude poskytovať aj proteínovú štruktúru, tak rozhranie umožní užívateľovi zvoliť si daný proteín (resp. proteíny). Výber týchto proteínov bude mať vplyv na vizualizáciu v štruktúrnom displeji.

Štruktúrny displej bude slúžiť na vizualizáciu štruktúry vstupného proteínu, ako aj štruktúr proteínov vybraných zo sekvenčného displeja (poskytovaných dátovými zdrojmi). Pri vizualizácii viacerých štruktúr budú tieto štruktúry vzájomne superpozicované. Pre každý vizualizovaný proteín bude dostupná možnosť výberu a

zobrazenia konkrétnych ligandov (resp. viacerých typov ligandov). Okrem samotných štruktúr budú vizualizované aj ich väzobné miesta. Používateľ bude mať možnosť prepnutia do módu zvýraznenia väzobných miest, v ktorom budú tieto miesta farebne označené. Intenzita farebného zvýraznenia bude korešpondovať s počtom dátových zdrojov podporujúcich dané väzobné miesto, pričom vyššia saturácia farby bude indikovať väčšiu podporu.

Štruktúra projektu

Projekt bude rozdelený na 4 časti - frontend, backend, dátová a AI časť. Každý člen tímu bude mať na starosti jednu zo spomínaných častí.

Frontend

Hlavným cieľom frontendu bude vytvorenie Analytickej stránky, obsahujúcej sekvenčný a štruktúrny displej, ktorá bude užívateľom slúžiť ako nástroj na analýzu proteínov. Analytická stránka už bola spolu s displejmi opísaná v časti Analytická stránka v podkapitole Rozšírenie analýzy.
Okrem toho sa v rámci frontendu upravia existujúce vstupné metódy (“Experimental structure”, “Custom structure”, “AlphaFold structure”). Taktiež bude pridaná nová vstupná metóda - “Sequence”. Úprava pôvodných metód spočíva v tom, že užívateľ bude po potvrdení vstupu presmerovaný na analytickú stránku.

Backend

Hlavnou úlohou backendu je poskytnúť funkčné prostredie na chod systému pre prácu s dátovými zdrojmi. Architektúra bude pozostávať z viacerých navzájom komunikujúcich Docker kontajnerov.

Súčasťou tejto časti projektu je aj vytvorenie *application programming interface* (API). Toto rozhranie bude slúžiť frontendu na obdržanie dát získaných z jednotlivých Docker kontajnerov, ktoré pracujú s dátovými zdrojmi.

Dátová časť

Návrh a implementácia systému, ktorý spracuje vstup od užívateľa a zabezpečí získanie dát z dátových zdrojov. Súčasťou je analýza jednotlivých dátových zdrojov. Na základe tejto analýzy je potrebné navrhnúť rozhranie pre interakciu s týmito dátovými zdrojmi a definovať jednotný formát dát, v ktorom budú dáta poskytnuté analytickej stránke.

AI

Cieľom je rozšírenie predikcie, a to natrénovať klasifikačnú neurónovú sieť, ktorá z proteínových sekvencií predikuje väzobné aminokyseliny proteínu a využiť existujúcu metódu na predikciu 3D štruktúry zo vstupnej sekvencie. Do klasifikačnej neurónovej siete budú vstupovať embeddingy získané zakódovaním proteínových sekvencií pomocou existujúceho pLM.

Funkčné požiadavky

Táto kapitola opisuje funkčné požiadavky softvéru, predstavujúce špecifické činnosti a funkcionality, ktoré systém umožní vykonávať užívateľom. Funkčné požiadavky ďalej delíme na užívateľské a systémové požiadavky.

Užívateľské požiadavky

Užívateľské požiadavky (UP) špecifikujú funkcionalitu, ktorá bude dostupná koncovým užívateľom systému. Požiadavky sú rozdelené do dvoch skupín, ktoré sú zamerané na zadanie vstupu a na novú analytickú stránku.

UP1 - Zadanie vstupu

1. Užívateľ si bude môcť vybrať zo štyroch vstupných metód. Patria medzi ne pôvodné “Experimental structure”, “Custom structure”, “AlphaFold structure” a nová vstupná metóda “Sequence”.
2. V prípade novej vstupnej metódy “Sequence” bude môcť užívateľ zadať vstup ako text vo FASTA formáte a bude si môcť zvoliť, či bude použitá konzervovanosť.
3. Pôvodné vstupné metódy zostanú, až na správanie po potvrdení vstupu (viď nasledujúci bod), nezmenené.
4. Po zadaní a potvrdení vstupu je užívateľ presmerovaný na analytickú stránku, a to aj v prípade pôvodných vstupných metód, kde sa budú postupne načítavať vizualizácie.

UP2 - Analytická stránka

Stránka bude poskytovať dva rôzne druhy vizualizácií:

Sekvenčný displej:

1. Bude zobrazovať:
    - sekvenciu vstupného proteínu
    - experimentálne zistené väzobné miesta získané zo vstupného proteínu

    - dáta z dátových zdrojov (napr. predikované väzobné miesta pomocou pLM, mieru konzervácie aminokyselín atď.)
2. Všetky zobrazené dáta budú zarovnané na sekvenciu vstupného proteínu, t.j. predikované väzobné miesta a sekvencie podobných proteínov sa budú zobrazovať zarovnané na sekvenciu vstupného proteínu.
3. Ak bude dátový zdroj poskytovať proteínovú štruktúru, rozhranie umožní užívateľovi označiť/odznačiť daný proteín (resp. proteíny) pre zobrazenie na Štruktúrnom displeji.
4. Dáta zobrazené rôznymi dátovými zdrojmi budú farebne odlíšené.

Štruktúrny displej:

1. Bude vizualizovať štruktúru vstupného proteínu, prípadne štruktúry proteínov označených v sekvenčnom displeji.
2. Pri vizualizácií viacerých štruktúr, budú štruktúry superpozicované.
3. Pre každý proteín, ktorého štruktúra je vizualizovaná, bude existovať možnosť zapnúť/vypnúť vizualizáciu konkrétneho druhu ligandu (resp. viacerých druhov ligandov).
4. Zobrazené štruktúry budú farebne odlíšené.
5. Okrem samotných štruktúr budú vizualizované aj ich väzobné miesta.
6. Užívateľ sa bude môcť prepnúť do módu zvýraznenia väzobných miest, v ktorom budú farebne vyznačené väzobné miesta. Vyššia saturácia farby bude znamenať, že väzobné miesto je podporené viacerými zdrojmi, a teda jeho existencia je pravdepodobnejšia.

Systémové požiadavky

Systémové požiadavky (SP) špecifikujú funkcionalitu a obmedzenia systému. Požiadavky sú rozdelené do skupín, ktoré predstavujú hlavné funkcionality nového systému.

SP1 - Validácia vstupu metódy “Sequence”

1. Vstup bude validovaný na úrovni frontendu aj backendu, aby sa zabezpečilo, že je v platnom formáte FASTA a nepresahuje maximálnu veľkosť pre predikciu 3D štruktúry.

SP2 - Vytvorenie identifikátora

1. Systém zabezpečí vytvorenie unikátneho identifikátora pre zadaný vstup.
    a. Ak je vstupom PDB ID alebo UniProt ID, použije sa ako identifikátor dané ID.
    b. Ak bola použitá vstupná metóda “Custom structure” alebo “Sequence” systém vygeneruje identifikátor pre vstupný proteín.

SP3 - Prevod vstupu

1. Ak bola použitá vstupná metóda “Sequence”, systém zabezpečí predikciu 3D štruktúry zo vstupnej sekvencie. Využitá bude existujúca predikčná metóda, pričom uvažované sú ESMFold a AlphaFold.
2. Ak bola použitá vstupná metóda *iná* ako “Sequence”, systém zabezpečí získanie sekvencie vstupného proteínu.

SP4 - Dátové zdroje

1. Systém zabezpečí získanie väzobných miest a podobných proteínov z dostupných dátových zdrojov, ktoré sú podrobne popísané v kapitole Dátové zdroje.
2. Systém umožní využiť predpočítané dáta pre opakované identické vstupy.
3. Získané dáta z dátových zdrojov budú prevedené systémom do jednotného formátu.

SP5 - Prístup k dátam pomocou API

1. Systém zabezpečí prístup k dátam získaných z dátových zdrojov pre konkrétny proteín pomocou read-only API endpointu.
2. Endpoint bude verejný.
3. API endpoint bude pozostávať z:
    a. Názov dátového zdroja
    b. Identifikátor pre proteín (SP2)

Nefunkčné požiadavky

Táto časť opisuje nefunkčné požiadavky (NP) systému, ktoré špecifikujú technické vlastnosti a charakteristiky systému. Požiadavky definujú rozšíriteľnosť, dostupnosť a škálovateľnosť systému.

NP1 - Rozšíriteľnosť

1. Systém bude navrhnutý modulárne, aby umožňoval jednoduchú integráciu nových komponentov. Konkrétne:
    a. Docker kontajnery: Bude možné jednoducho pridávať nové kontajnery pre spracovanie dát z ďalších dátových zdrojov.
    b. Užívateľské rozhranie: Architektúra frontendu pomocou React komponentov bude umožňovať pridávanie nových prvkov do UI.

NP2 - Dostupnosť

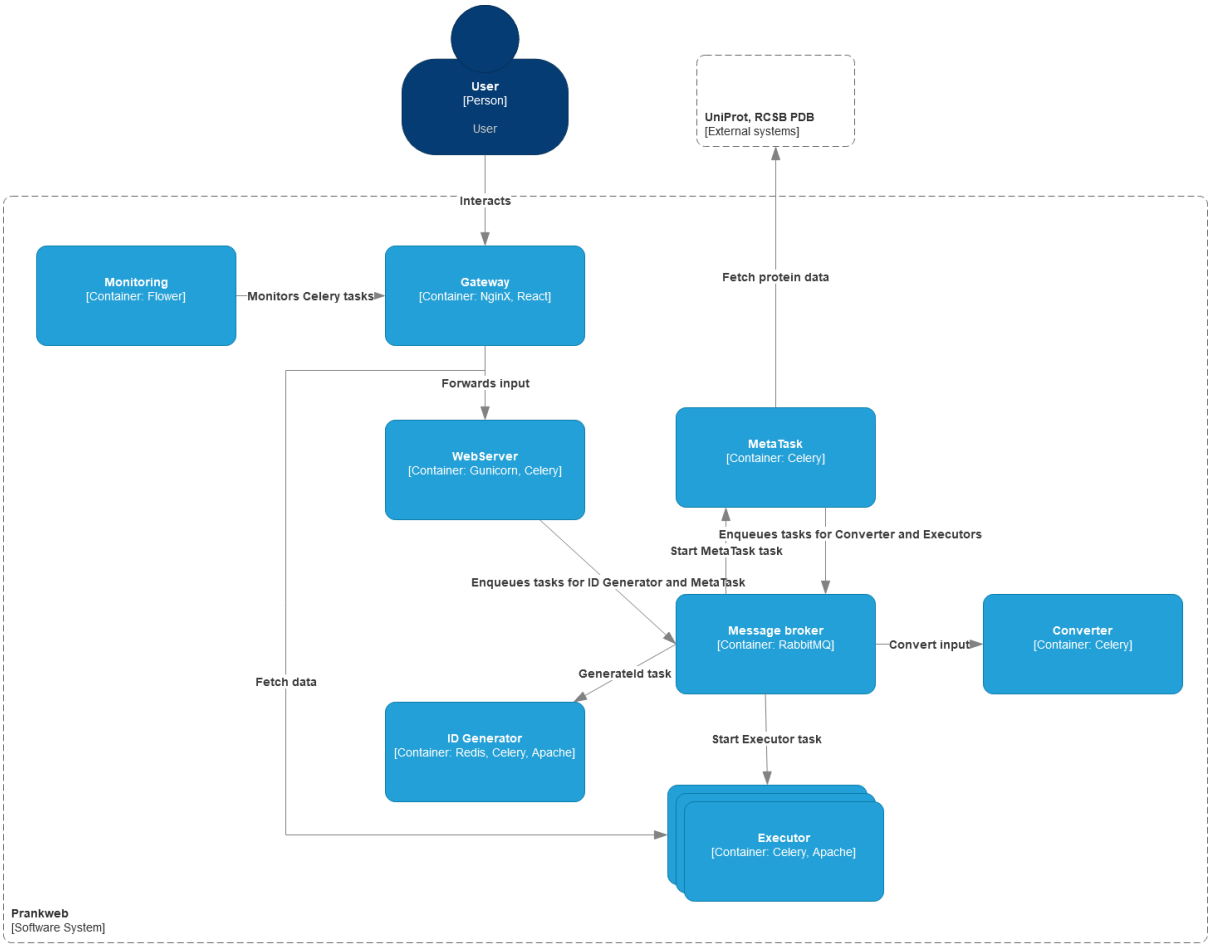
1. Neúspešné získanie dát z jedného z dátových zdrojov neovplyvní získavanie dát z ostatných dátových zdrojov.
2. Ak dôjde k chybe v systéme (napr. pri získavaní dát z dátových zdrojov), systém bude schopný diagnostikovať problém a informovať o tom užívateľa.
3. Bude vytvorený API endpoint pre prístup k dátam získaných z jednotlivých dátových zdrojov.

NP3 - Škálovateľnosť

1. Bude možné spustiť viacero inštancií backendu naraz (napr. pomocou workerov v Gunicorn).

Architektúra

Projekt bude navrhnutý ako modulárny systém zložený z viacerých komponent s využitím kontajnerizácie pomocou Dockeru.
Nasledujúci diagram zachytáva hlavné runtime časti systému.



Obr. Architektúra softvéru

Komponenty

Projekt možno rozdeliť na nasledovné komponenty: Gateway, WebServer, Message broker, Flower a viacero taskov, konkrétne: IdGenerator, MetaTask, Converter, Conservation a niekoľko Executorov. Task definujeme ako komponent, ktorý po prijatí správy od Message brokera zabezpečí vykonanie svojej funkcionality.

Gateway

S využitím HTTP serveru Nginx bude tento komponent plniť dve hlavné funkcie:

1. zabezpečenie obsluhy Frontendu
2. reverzný proxy server, sprostredkujúci požiadavky prijaté Frontendom WebServeru

WebServer

Úlohou WebServera bude:

1. validovať prichádzajúci vstup
2. získať jedinečný identifikátor vstupu pomocou IdGeneratora
3. spustiť MetaTask

Pomocou aplikačného servera *Gunicorn*, bude možné spustiť viacero inštancií WebServera naraz.

IdGenerator

IdGenerator bude zodpovedný za generovanie jedinečného identifikátora pre dvojicu: vstupný proteín a vstupná metóda. Tieto identifikátory budú uložené v *Redis* databáze. Jeho úlohou bude detegovať, či sa jedná o opakovaný vstup, pre ktorý už prebehol výpočet.

Message broker

Tasky komunikujú prostredníctvom Task queue, ktorá na svoju prevádzku vyžaduje komponent Message broker. V tomto projekte bude využitá kombinácia *Celery* ako Task queue a *RabbitMQ* ako Message broker. Projekt bude ako Message broker využívať existujúci Docker image *RabbitMQ*.

Flower

Komponent Flower slúži na monitorovanie *Celery* taskov v reálnom čase.

MetaTask

Na vstupe dostane proteín, vstupnú metódu, jedinečný identifikátor a príznak identifikátora. Jeho úlohou bude:

1. získať pomocou Convertera sekvenciu alebo 3D štruktúru proteínu výslednú dvojicu (sekvencia a 3D štruktúra) a uložiť výsledok do svojho súborového systému
2. spustiť jednotlivé Executory, ktorým sa predá vstupný proteín vo vhodnom formáte

Converter

Úlohou Convertera bude získať sekvenciu, resp. predikovať 3D štruktúru proteínu.

- Ak bola využitá vstupná metóda “Sequence”, modul dostane na vstupe sekvenciu a pomocou predikčnej metódy (AlphaFold, ESMFold) vráti vypredikovanú 3D štruktúru.
- Ak nebola využitá vstupná metóda “Sequence”, modul dostane na vstupe štruktúru proteínu v podobe súboru, z ktorej získa jej sekvenciu.

Executor

Bude zodpovedný za získanie a spracovanie dát z dátového zdroja pre vstupný proteín. Pre prístup k dátam získaných z dátových zdrojov bude vytvorené read-only API, na ktoré sa bude môcť dotazovať frontend.

- Ak má Executor výsledok uložený vo svojom súborovom systéme, vráti ho frontendu
- Ak výsledok nemá, vráti frontendu adekvátnu správu o neexistujúcom výsledku

Funkcionalitu budú zabezpečovať Docker kontajnery, na ktorých bude nasadený:

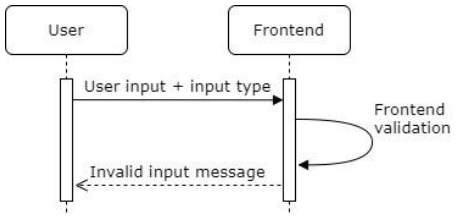
1. Celery Worker
2. HTTP server

Kontajner, na ktorom bude nasadený HTTP server, bude spoločný pre všetkých Executorov využívajúcich rovnaký súborový systém. Kontajner so Celery Workerom je pre každého Executora unikátny.

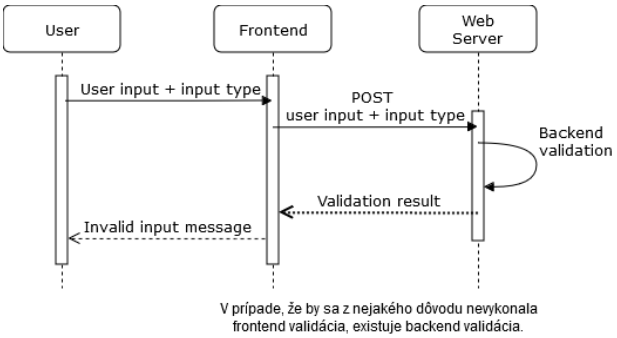
Priebeh výpočtu

1. Užívateľ zadá vstupný proteín.
2. Frontend vstup validuje, výsledok môže byť:
    a. Validný - vstup sa predá serveru aj s informáciou o vstupnej metóde.
    b. Nevalidný - frontend zobrazí užívateľovi správu o chybnom vstupe, končí výpočet.
3. Webserver znova validuje vstup, výsledok môže byť:
    a. Validný - požiada IdGenerator o jedinečný identifikátor pre vstup. Spolu s identifikátorom sa webserveru vráti aj príznak o tom, či už daný identifikátor existoval.
    b. Nevalidný - vráti frontendu status kód o nevalidnom vstupe, končí výpočet.
4. Webserver spustí MetaTask prostredníctvom RabbitMQ, ktorému predá vstup, vstupnú metódu, identifikátor a príznak identifikátora.
5. MetaTask na základe príznaku vykoná:
    a. Ak je príznak *true*, MetaTask overí, či výsledky Executorov existujú a ak nie, tak dané Executory spustí a predá im vstupný proteín vo vhodnom formáte (sekvencia alebo 3D štruktúra).
    b. V prípade príznaku *false*, MetaTask spustí Converter, na získanie potrebnej sekvencie alebo 3D štruktúry. Výsledok si uloží a spustí Executorov, ktorým predá vstupný proteín vo vhodnom formáte. Príznak *false* totiž znamená, že príslušný identifikátor je použitý prvýkrát.
6. Executory získajú a spracujú dáta z dátových zdrojov a uložia si ich do svojho súborového systému.
7. Frontend sa pomocou polling prístupu dotazuje Executorov o dáta.
8. Frontend zobrazí získané dáta užívateľovi.

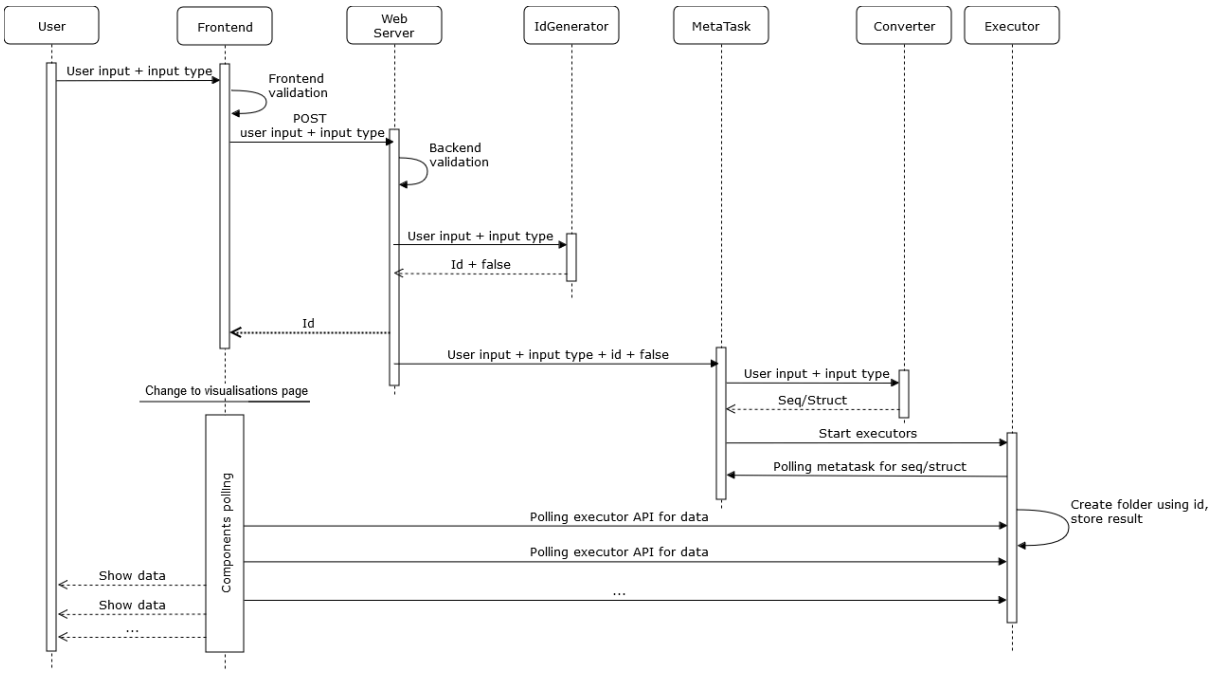
Možné scenáre výpočtu sú zobrazené pomocou sequence diagramov.



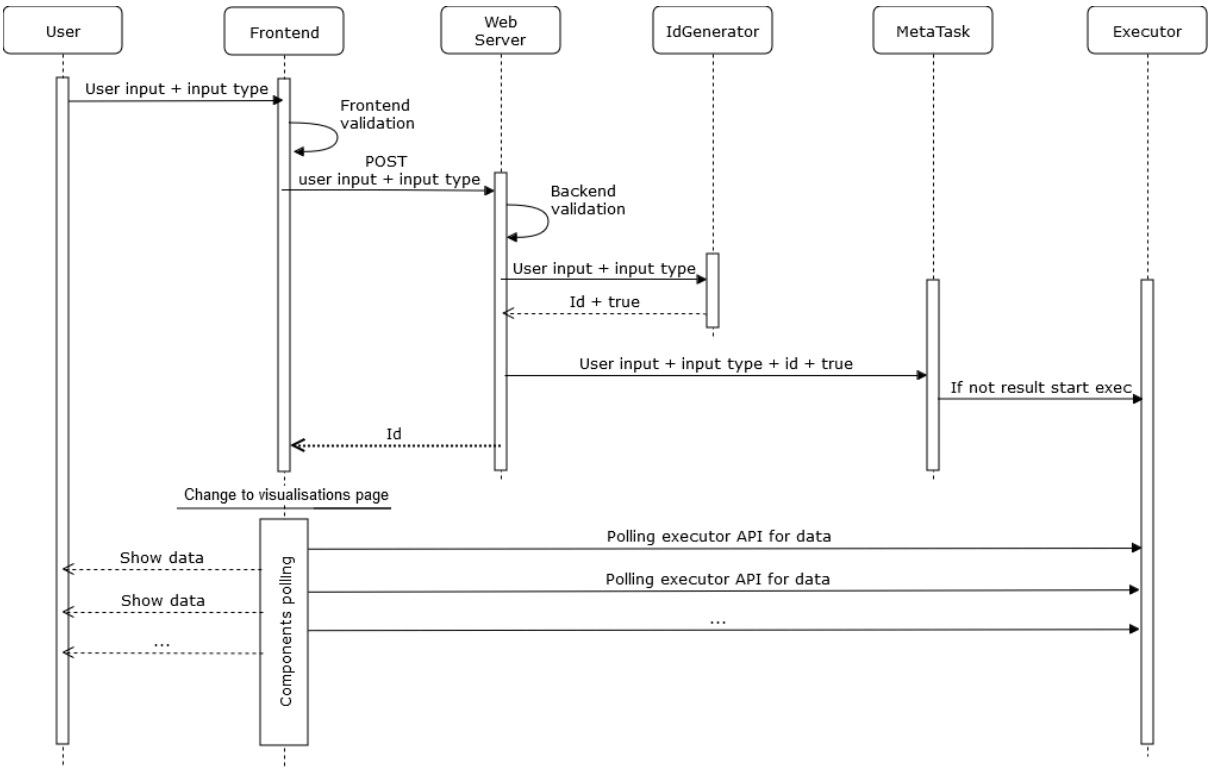
Obr. Invalidný vstup, validácia na Frontende



Obr. Invalidný vstup, validácia na WebServeri



*Obr. Validný **nový** vstupný proteín*



*Obr. Validný, no **opakovaný** vstupný proteín*

Dátové zdroje

P2Rank a pLM v spojení s klasifikačnou neurónovou sieťou môžeme považovať za dátové zdroje poskytujúce informácie o predikovaných väzobných miestach pre daný vstupný proteín. V súčasnosti sú k dispozícii aj ďalšie dátové zdroje. Niektoré z nich poskytujú už vopred vypočítané väzobné miesta pre konkrétne ligandy. Iné dokážu nájsť proteíny, ktoré majú podobnú 3D štruktúru a obsahujú údaje o experimentálne získaných väzobných miestach.
Takéto dáta je možné spracovať z lokálnych aj externých zdrojov. Ako lokálne zdroje chápeme výsledky modulov, ktoré bežia v rámci prostredia aplikácie. Externé zdroje sú nástroje a databázy, ktoré nie sú priamou súčasťou Prankwebu.

Lokálne dátové zdroje

Medzi lokálne zdroje patrí v systéme už existujúci P2Rank, novo vyvíjaný modul Plank využívajúci pLM a nástroj Foldseek.

P2Rank

Poskytuje predikcie väzobných miest pre vstupný proteín na základe jeho 3D štruktúry.

Plank

Predikuje pre každú aminokyselinu vstupnej sekvencie proteínu či je súčasťou väzobného miesta.
Predikcia väzobných miest prebieha nasledovne:

1. Sekvencia je predaná pLM, z ktorého získame reprezentáciu sekvencie vo forme embeddingov.
2. Embeddingy sú následne predané klasifikačnej neurónovej sieti, ktorá pre každú aminokyselinu daného proteínu rozhodne či je súčasťou väzobného miesta.

V súčasnosti existuje viacero pLM, ktoré kódujú sekvenciu proteínov do embeddingov. Uvažovaný je ESM-2 model.

Foldseek & PDB100

Foldseek je nástroj na vyhľadávanie homologických proteínov založený na podobnosti ich 3D štruktúr, čo je často efektívnejšie ako porovnávanie sekvencií aminokyselín. Zohľadňuje priestorové usporiadanie proteínov, ktoré je kľúčové pre ich biologickú funkciu. Dáta, s ktorými by Foldseek pracoval by boli získané z databázy PDB100, ktorá pre štruktúry poskytuje informácie o experimentálne získaných väzobných miestach.

Potenciálny externý dátový zdroj

Táto oblasť je v neustálom vývoji a dátové zdroje sa časom menia. Jeden z uvažovaných externých zdrojov, z ktorého sa budú získavať dáta, je databáza AHoJ-DB.

AHoJ-DB

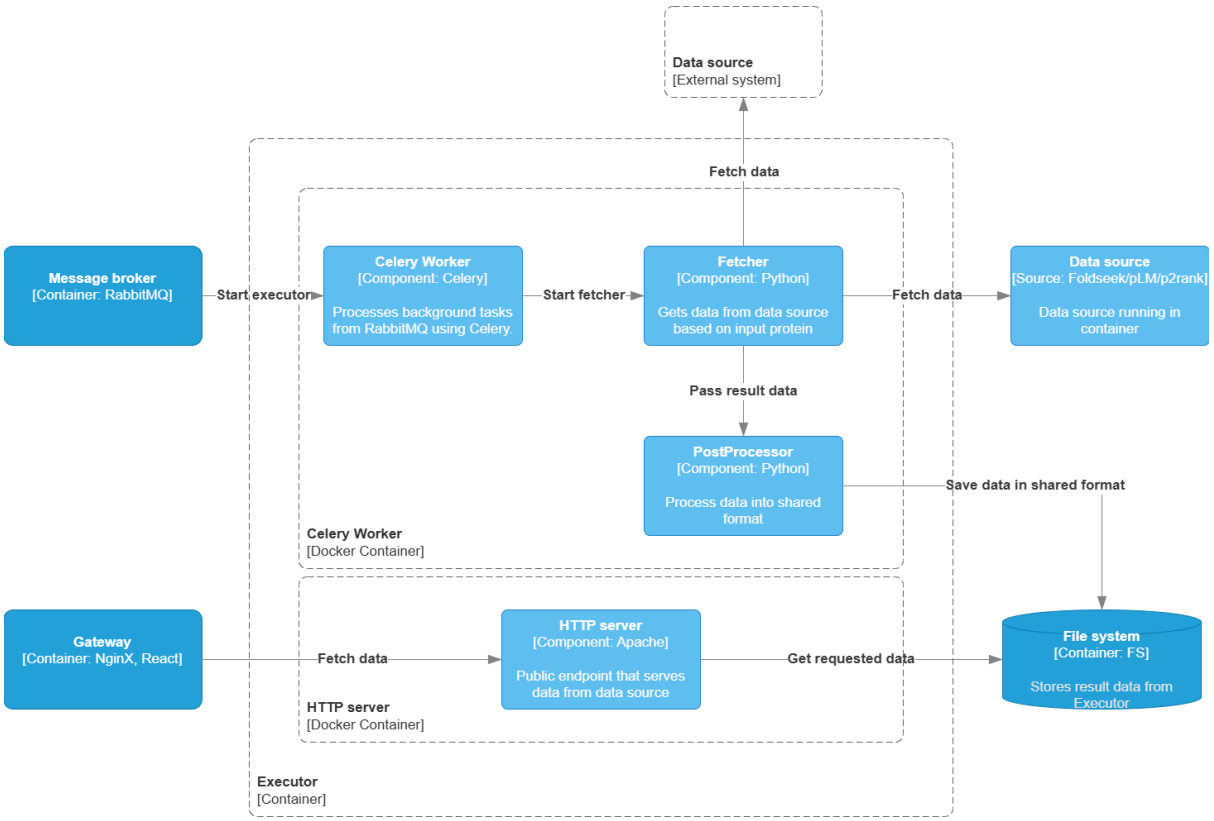
Databáza, ktorá obsahuje experimentálne získané väzobné miesta ligandov. Pre každé väzobné miesto poskytuje AHoJ-DB dva typy štruktúr:

- APO (ligand free) - miesto s nenaviazaným ligandom
- HOLO (ligand bound) - miesto s naviazaným ligandom

AHoJ-DB API je momentálne vo vývoji – preto nie je isté, či bude využitá ako dátový zdroj v tomto projekte.

Rozhranie

Pre každý dátový zdroj bude vytvorený samostatný Docker container, tzv. Executor. Pre zachovanie konzistentnosti vstupu a výstupu z rôznych Executorov je navrhnuté rozhranie popisujúce základné funkcionality, ktoré by mal každý Executor spĺňať a taktiež bude definovaný jednotný formát výstupu, ktorý bude poskytnutý užívateľskému rozhraniu.



Celery endpoint pre RabbitMQ

Každý Executor bude obsahovať komponent s endpointom, pomocou ktorého bude komunikovať s RabbitMQ na získanie úloh. Po jej obdržaní bude zodpovedný za spustenie komponentu Fetcher, ktorému poskytne vstupný proteín.

Fetcher

Komponent bude zodpovedný za komunikáciu s dátovým zdrojom. Zabezpečí spracovanie dát potrebných pre dátový zdroj a bude obsahovať logiku pre pripojenie k dátovému zdroju (ktorý môže byť súčasťou aplikácie “lokálny” alebo samostatný systém “externý”), získanie dát a ich následné poskytnutie komponentu PostProcessor.

PostProcessor

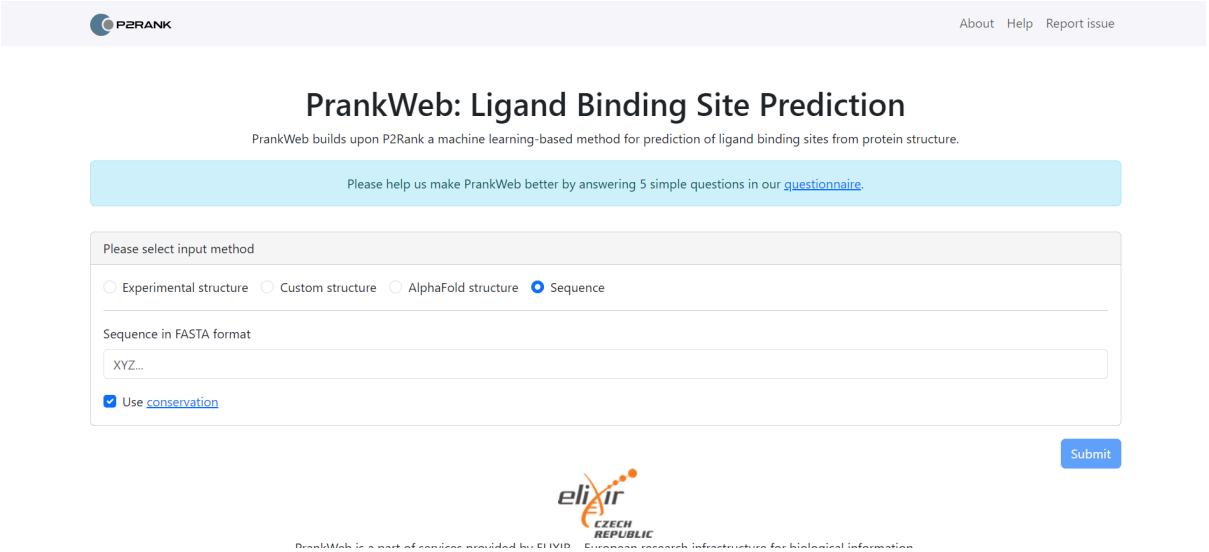
Zabezpečí prevod dát získaných z dátového zdroja do jednotného formátu pre užívateľské rozhranie. Dáta uloží do priečinku určeného pre konkrétny výpočet v súborovom systéme.

Užívateľské rozhranie

V tejto kapitole si predstavíme návrhy vzhľadu užívateľského rozhrania. Konkrétne sa pozrieme na rozšírenie úvodnej stránky so vstupom a predstavíme si návrh vzhľadu analytickej stránky.

Úvodná stránka

Úvodná stránka umožní užívateľovi výber vstupnej metódy a zadanie vstupu.



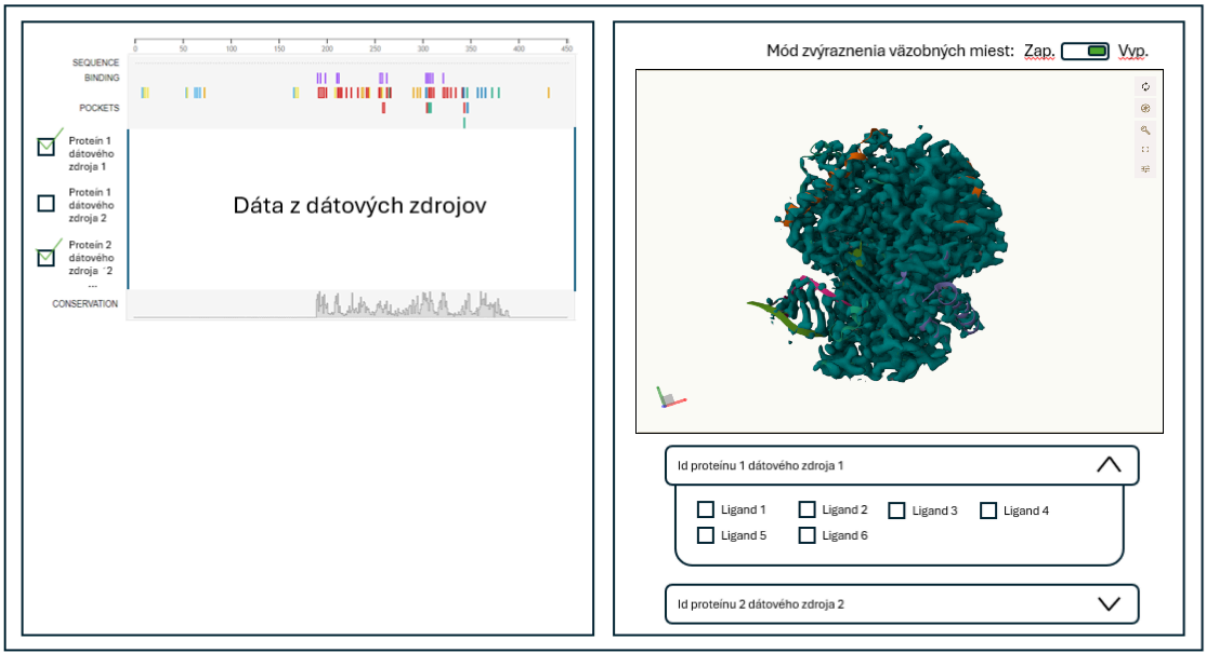
Obr.: Návrh úvodnej obrazovky s novou vstupnou metódou

Úvodná stránka zostane rovnaká ako v pôvodnom Prankwebe, s výnimkou pridania novej vstupnej metódy “Sequence” do výberového poľa. Používateľ bude môcť pri tejto metóde zadať proteínovú sekvenciu vo formáte FASTA do textového poľa. Okrem toho bude mať možnosť zaškrtnúť pomocou začiarkavacieho políčka (checkboxu), či má byť pri spracovaní sekvencie zohľadnená aj konzervovanosť.

Analytická stránka

Po zadaní a potvrdení vstupu bude používateľovi zobrazená analytická stránka s dvoma vizualizáciami:

- Sekvenčný displej
- Štruktúrny displej



Obr. Návrh analytickej stránky

Po príchode na stránku môže užívateľ vidieť, že sa jednotlivé časti načítavajú (ak ešte nedobehli výpočty). Po úplnom načítaní stránky bude užívateľ vidieť vľavo sekvenčný displej implementovaný pomocou knižnice rcsb-saguaro a vpravo štruktúrny displej implementovaný s pomocou knižnice Mol* (viď obr.).

Sekvenčný displej

Sekvenčný displej bude mať začiarkavacie políčka (checkboxy) pri riadku vstupnej sekvencie a riadkoch proteínov získaných dátovými zdrojmi, ktoré poskytnú aj ich štruktúry. Pomocou týchto políčok si môže užívateľ zapínať/vypínať vizualizáciu štruktúr odpovedajúcich proteínov v štruktúrnom displeji.

Štruktúrny displej

Štruktúrny displej bude vizualizovať 3D štruktúru vstupného proteínu a 3D štruktúry proteínov zvolených v sekvenčnom displeji. Pod vizualizáciou štruktúr sa bude nachádzať pre každú vizualizovanú proteínovú štruktúru panel s id odpovedajúceho proteínu a šípkou dole. Po kliknutí na šípku dole sa šípka zmení na šípku hore a zároveň sa otvorí (“rozbalí”) okno s ligandami (resp. druhmi ligandov) odpovedajúceho proteínu.

Pri každom ligande sa bude nachádzať začiarkavacie políčko (checkbox). Zaškrtnutím políčka sa zobrazia na štruktúre odpovedajúceho proteínu ligandy vybraného druhu, naopak odškrtnutím políčka ligandy zmiznú. Kliknutím na šípku hore sa okno zavrie (“zabalí”).

Vpravo hore (nad štruktúrnym displejom) sa bude nachádzať prepínač, pomocou ktorého si bude môcť užívateľ zapnúť/vypnúť mód zvýraznenia väzobných miest.

Realizácia projektu

Táto kapitola opisuje dôležité časti realizácie projektu, ako je rozdelenie častí projektu medzi jednotlivých členov tímu, plán realizácie a využité nástroje.

Role

Tím sa skladá zo štyroch členov, pričom každý člen má nasledovné úlohy počas realizácie projektu:

- **Katarína Bucková:**
  - AI - integrácia pLM, vývoj a trénovanie klasifikačnej neurónovej siete.
- **Richard Fedák:**
  - Data/BIO - analýza a využitie dátových zdrojov.
- **Samuel Karaš:**
  - Backend - návrh architektúry a dockerizácia projektu.
- **Milan Truchan (Team leader):**
  - Frontend - návrh a implementácia UI.

Míľniky

Táto sekcia obsahuje kľúčové míľniky projektu. Niektoré z nich už boli splnené v rámci vypracovania tejto špecifikácie, zatiaľ čo iné budú realizované počas samotného vývoja.

Analýza požiadaviek

Prvým krokom bolo oboznámenie sa s doménou a súčasne existujúcim projektom, ktorý rozširujeme. Táto fáza prebiehala formou niekoľkých stretnutí s konzultantmi, počas ktorých sme si vytvorili predstavu o doméne a základných požiadavkách na rozšírenie. Neskôr nasledovala definícia funkčných požiadaviek a tvorba špecifikácie projektu.

Vytvorenie mockupov

Po definovaní funkčných požiadaviek sme sa sústredili na návrh vzhľadu aplikácie.

Konkrétne na potrebné zmeny na existujúcej úvodnej stránke a návrh analytickej stránky s vizualizáciami.

Návrh architektúry

V ďalšej fáze sme definovali nefunkčné požiadavky a vytvorili návrh architektúry aplikácie. Zamerali sme sa na definovanie hlavných komponentov a spôsob komunikácie medzi nimi, pričom sme vychádzali z komponentov a architektúry pôvodného projektu.

Implementácia

Ďalším míľnikom, ktorý bude nasledovať je samotná implementácia rozšírenia. Cieľom je implementovať všetky spomenuté časti projektu, t.j. AI časť, dátová časť, frontend a backend.
Počas tejto fázy očakávame, že sa budú konať stretnutia s konzultantmi projektu kvôli možným pripomienkam ohľadom implementovaných požiadaviek. Tento proces budeme opakovať iteratívne, kým nebudú splnené všetky spomenuté požiadavky. Súčasťou bude aj tvorba dokumentácie a testovanie aplikácie. Odhadovaný čas venovaný tejto časti je šesť mesiacov.

Použité nástroje

Pre tvorbu špecifikácie aj následnej implementácie využívame nástroje popísané v tejto sekcii.

Verzovanie

Na zdieľanie zdrojových kódov, dokumentácie a ich verzovanie, sledovanie issues atď. sme si vybrali GitHub.

Komunikačná platforma

Primárnym komunikačným kanálom medzi členmi tímu a vedúcim/konzultantmi bol zvolený fakultný Mattermost. Výhodou je možnosť oddelených kanálov pre konzultácie rôznych častí projektu a dohadovanie stretnutí.

Editor pre písanie dokumentov

Zámer bol napísaný v Markdowne a špecifikácia v Google Docs.