

Функцијски барања:

1. Системот управува со извори на податоци. Системот мора автоматски да се поврзе со избран извор на крипто податоци како на пример Coingecko, Binance, Coinbase, Kraken.
2. Системот ја презема листата со врвните 1000 криптовалути-символи по пазарна капитализација.
3. Системот ги идентификува невалидните или проблематичните валути. Тука спаѓаат валути кои се делистани, валути со премала ликвидност или нестабилна квота. Системот мора сите овие валути да ги филтрира без рачна интервенција.
4. За секоја криптовалута системот треба да провери кои податоци се веќе зачувани во базата.
5. За криптовалутите кои немаат зачувани податоци системот мора да превземе најмалку 10 години историски дневни “отвори високо ниско затвори волумен” (OHLCV) податоци или пак максимумот што API го поддржува.
6. Доколку податоците постојат потребно е системот да го земе последниот датум и да пресмета кои податоци недостасуваат, а потоа да ги додаде во базата/документот со филтрирани и финални податоци.
7. Системот мора да нуди механизам за преземање на сите податоци кои недостасуваат до тековниот ден.
8. Системот има Pipe-and-Filter трансформација.
9. Системот мора да имплементира филтер 1 (Extraction Filter) каде што презема листата од најдобрите-највредни 1000 валути, ги прочистува податоците и потоа прочистените податоци ги зачува.
10. Системот мора да имплементира филтер 2 (Date Check Filter) каде што ја проверува базата и потоа враќа симбол заедно со последниот датум, за валутите кои имаат податоци за нив, а оние кои немаат податоци ги додава на листа.
11. Системот мора да имплементира филтер 3 кој што ги додава OHLCV податоците за сите симболи за кои нема записи и тоа додава податоци од 10 години наназад или максимално достапно време па до тековниот датум.
12. Системот мора да имплементира филтер 4(Data Completion Filter) каде што ги додава OHLCV податоците за сите симболи до тековниот датум.
13. Системот мора да ги зачува податоците во база или датотечен формат како на пример (CSV/JSON).
14. Системот мора да избегнува дупликати при додавање на нови редови.
15. Системот мора да биде конзистентност во форматите. Да има исти колони, исти датумски формат...со цел да се избегне настанувањето грешки.
16. Системот мора да прикаже грешки доколку API не е достапен

17. Системот мора да прави “retry” при неуспешен повик, односно да се обиде уште неколку пати.
18. Системот мора да ги прати/следи сите трансформации и грешки што ќе настанат.
19. Системот треба да измери колку време му треба да ја исполнити празната база.
20. Системот треба да прави обработка на податоците, и да дава точни и прецизни информации и предвидувања.
21. Системот треба да нуди интуитивен и лесен за користење кориснички интерфејс преку кој графички ќе се претставуваат податоците.

Нефункцијски барања:

1. Перформанси
 - Системот треба да обработи 100 крипто валути во разумно време.
 - Системот треба да подржува 100 истовремени корисници.
2. Скалабилност
 - Системот треба да поддржи додавање на нови извори на податоци без промена на целата архитектура.
 - Системот треба да може да ги пренесе податоците од CSV во SQL без голема измена.
3. Конзистентност
 - Податоците мора да бидат во стандарден формат.
4. Безбедност
 - API клучевите (доколку се користат) мора да бидат во .env фajл.
5. Одржливост
 - Филтрите мора да бидат независни единици на код.
 - Промената на еден филтер не смее да влијае на другите.
 - Додавањето на нови функционалности не смее да се одрази на веќе постоечките.
6. Квалитетен код
 - Кодот треба да ги следи принципите кои обезбедуваат читливост и разбираливост
 - Кодот треба да го следи SOLID и DRY принципот
 - Кодот треба да биде лесно одржлив
 - Кодот треба да биде робустен и ефикасен.

7. Употребливост
 - Интерфејсот потребно е да биде едноставен и лесен за користење за анализа на податоци.
8. Доверливост
 - Системот мора да продолжи да работи и при делумен пад на API.
 - Retry механизам мора да постои за сите API повици.
9. Тестирање
 - Системот мора да се тестира со повеќе од 50 тест валути пред финалната верзија.

Персони

Персона 1: Крипто Инвеститор (Retail Trader)

Возраст: 24

Професија: студент по економија / делумен трејдер

Техничко ниво: средно

Локација: Скопје

Цели:

- Да добие централизирани, точни и ажурирани историски податоци за криптовалути.
- Да прави сопствени анализи без да пребарува податоци од повеќе веб-страни.
- Да идентификува трендови и шеми на дневно ниво.

Фрустрации / Болни точки:

- Различни сајтови имаат различни податоци, некои со грешки или празни денови.
- Тешко му е да најде податоци за криптовалути што се пониско рангирали.
- Понекогаш API серверите не враќаат комплетни OHLCV податоци.
- Не сака да губи време на мануелно преземање CSV датотеки.

Како му помага нашата апликација:

- Автоматски ги собира и форматира сите податоци на едно место.
- Ги чисти дупликатите.
- Обезбедува целосна историска база за сите топ 1000 криптовалути.
- Обработката е автоматизирана, без потреба од мануелна интервенција.

Персона 2: Финансиски аналитичар

Возраст: 31

Професија: аналитичар во финтек компанија

Техничко ниво: високо

Локација: Битола

Цели:

- Да има сигурен и автоматизиран data ingestion систем.
- Да користи историски податоци за моделирање, прогнозирање и бектестирање.

Фрустрации / Болни точки:

- Често се соочува со недостасувачки денови кај крипто податоци.
- Податоците доаѓаат во различни формати (JSON, CSV, различни timestamp формати).
- Сака конзистентен формат за да избегне грешки во анализа.
- Многу системи не поддржуваат долг историски опсег (10+ години).

Како му помага нашата апликација:

- Pipe-and-Filter архитектура овозможува чисти и верификувани податоци.
- Форматирањето е стандардизирано.
- Може да се комбинираат податоци од повеќе берзи со иста обработка.
- Сите податоци се со ист формат и директно подгответи за аналитички модели.

Персона 3: Blockchain Researcher

Возраст: 29

Професија: истражувач во институција за блокчејн технологии

Техничко ниво: напредно (Python, Solidity, Data Engineering)

Локација: Охрид

Цели:

- Да анализира податоци за долгорочни трендови на крипто пазари.
- Да ги споредува историските податоци со он-чејн метрики.
- Да користи стандарден формат на податоци за research papers.

Фрустрации / Болни точки:

- Повеќето бесплатни API даваат скратена историја или нецелосни записи.
- Податоците често се различни од берза до берза (различни timezone, timestamps, volumetric data).
- Многу datasets имаат “gaps”(дупки) поради непостоечки денови.
- Тешко се споредуваат податоци од centralized и decentralized извори.

Како му помага нашата апликација:

- Обезбедува комплетни OHLCV податоци, исправени и конзистентни.
- Pipe-and-Filter архитектура гарантира дека сите податоци се филтрирани и нормализирани.
- Може да увезе податоци од различни извори, но да ги стандардира во единствен формат.
- Подготвени податоци за научни анализи, понатамошна обработка и машинско учење.

Персона 4: Software Engineer

Возраст: 27

Професија: backend инженер во стартап

Техничко ниво: напредно

Локација: Велес

Цели:

- Да работи со модуларен и лесно проширлив систем.
- Да може да додава нови филтри, нови извори на податоци и нови типови трансформации
- Да користи архитектура која не бара големи промени при скалирање.

Фрустрации / Болни точки:

- Проблеми со различни API стапки (rate limits) и непредвидливо API однесување.
- Нестандардни формати на податоци го отежнуваат ingestion процесот.
- Недостаток на логирање и транспарентност при грешки.

Како му помага нашата апликација:

- Може да се додаваат нови филтри без промена на целата апликација.
- Централизирано логирање овозможува лесно debugging.
- Оптимизиран за скалабилност и паралелно преземање.

Наратив за кориснички сценарија

1. Крипто Инвеститор – Ажурирање на недостасувачки податоци

Александар, млад крипто инвеститор и студент, го отвора системот за да ја провери состојбата на криптовалутите што ги следи. Тој нема време да проверува повеќе веб-страни, ниту да презема CSV датотеки рачно.

Кога ја активира опцијата „Ажурирај ги податоците“, системот веднаш ја скенира базата преку Date Check Filter. Препознава дека за неколку симболи недостигаат податоци за последните неколку денови. Автоматски се активира Data Completion Filter, кој ги првзема податоците од изворот на API и ги додава во базата, избегнувајќи дупликати и користејќи стандарден формат.

Александар само гледа како графикот се освежува и добива целосна слика за пазарните трендови – без никаков напор.

2. Blockchain Researcher – Преземање 10+ години историски податоци

Марко, блокчејн истражувач, подготвува научна анализа за долгорочните трендови на пазарот на криптовалути. За потребите на трудот, му треба целосна историска база за повеќе стотици криптовалути.

Тој ја активира опцијата „Генерирај историска база“. Системот прво ја презема листата на топ 1000 криптовалути и ги филтрира сите невалидни или делистани симболи. Потоа Date Check Filter утврдува кои валути немаат никакви податоци во базата.

Историскиот филтер се активира и за секоја валута системот презема минимум 10 години OHLCV податоци – или максимумот што API го поддржува. Податоците се нормализираат, прочистуваат и се зачувуваат во консистентен формат.

Кога процесот е завршен, Марко има комплетен dataset подготвен за анализа и машинско учење – нешто што инаку би траело денови да го собере.

3. Финансиски аналитичар – Визуелна анализа на податоци

Анита, финансиски аналитичар во финтек компанија, секој ден ја користи платформата за визуелна анализа на крипто пазарот. Нејзината цел е да види трендови, волумен и движење на цените во различни периоди.

Таа отвора Dashboard, внесува симбол – на пример ETH – и системот веднаш ги повлекува сите OHLCV податоци. Платформата ги прикажува графиците за цена, волумен и индикатори како Moving Averages.

Ако недостигаат денови, системот автоматски инициира Data Completion Filter и ги дополнува. Анита не мора ни да знае дека имало „дупки“ во податоците – системот прави сè во позадина и секогаш прикажува комплетна анализа.