

Cryptocurrency Market Tracker App

Апликација за следење и анализа на
крипто-пазарот

Изработено од:

Јасмин Абази 225039

Стефан Тасевски 233105

Кристијан Јовиќ 233140

Верзија: 1.3

Датум: 10/11/2025

Содржина

1. Историја на ревизии (Revision History)
2. Вовед (Introduction)
 - 2.1 Цел на документот (Document Purpose)
 - 2.2 Референци (References)
 - 2.3 Преглед на документот (Document Overview)
 - 2.4 Правен и регулаторен контекст
3. Преглед на производот (Product Overview)
 - 3.1 Перспектива на производот (Product Perspective)
 - 3.2 Функционалности на производот (Product Functions)
 - 3.3 Главни сценарија на користење (Main Use Cases)
 - 3.4 Доменски модел (Domain Model)
 - 3.4.1 Дијаграм на активности
4. Архитектура на системот (System Architecture)
 - 4.1 Архитектонски стилови
 - 4.2 Pipe-and-Filter архитектура
 - 4.3 Словита (Layered) архитектура
 - 4.4 Component-based архитектура
 - 4.5 Microservices архитектура
5. Функциски и нефункциски барања (Requirements)
6. Технологии и алатки (Technologies and Tools)
7. Тестирање и валидација (Testing and Validation)
8. Идни подобрувања (Future Work)
9. Заклучок (Conclusion)

1. Историја на ревизии

| Верзија | Датум | Опис на промени | Автор |
|---------|-----------------|--|-------------------|
| 1.0 | 11 Ноември 2025 | Почетна верзија на техничка документација | Тимот на проектот |
| 1.1 | 16 Ноември 2025 | Документот е ревидиран согласно насоките од професорот: фокусот е пренасочен од техничка имплементација кон опис на функционалностите и архитектурата на ниво 'што', а не 'како'." | Тимот на проектот |
| 1.2 | 21 Ноеври 2025 | Додадени архитектонски стилови; подобрен доменски модел; изработен Use Case дијаграм; формализирани барања; уредени секциите за тестирање, подобрувања и заклучок. | Тимот на проектот |
| 1.3 | 22 Ноември 2025 | Додадена подточка 1.4 со правен и регулаторен преглед на крипто-услуги и заштита на лични податоци. | Тимот на проектот |

2. Вовед

Овој документ претставува спецификација на софтверските барања (SRS) и високо-нивоен опис на проектот „Апликација за следење и анализа на крипто-пазарот“. Документот го дефинира опфатот на системот, главните функционалности, корисничките сценарија, доменскиот модел и основните архитектонски насоки кои ќе се применуваат во текот на развојот.

Целта на документот е јасно да ја опише потребната функционалност на системот, ограничувањата, очекуваните квалитети и општата структура на решението, без навлегување во конкретни имплементациски детали.

2.1 Цел на документот

Целта на овој документ е да ги дефинира функционалните и нефункционалните барања на системот, корисничките сценарија, доменските концепти и основниот опфат на проектот. Документот служи како референтен извор за сите членови на тимот и обезбедува заедничко разбирање за тоа што треба да биде изградено во текот на сите фази од развојниот процес.

2.2 Референци

- Материјали од предметот *Дизајн и архитектура на софтвер* (ФИНКИ)
 - Лекции: Conceptual, Execution, Implementation Architecture, Architectural Styles, Architectural Styles part 2.
- [Binance API Documentation](#)
- [MongoDB official Documentation](#)
- GitHub репозиториум: <https://github.com/TaseZmaj/cryptocurrency-market-tracker-app>

2.3 Преглед на документот

Овој документ содржи опис на производот, неговите главни функционалности, корисничките сценарија, доменскиот модел и спецификацијата на функционалните и нефункционалните барања. Дополнително, документот опфаќа преглед на архитектонските насоки, технолошките избори, процесот на тестирање и планираните идни подобрувања на системот.

2.4 Правен и регулаторен контекст

Во Република Северна Македонија криптовалутите (виртуелните средства) не се експлицитно регулирани како официјално платежно средство според Законот за Народна банка и Законот за платен промет. Тргувањето и поседувањето на криптовалута е дозволено, но се врши на сопствен ризик, при што Народната банка има издадено предупредувања за поврзаните ризици.

Надзор и обврски при понуда на услуги:

Со усвојувањето на новиот **Закон за спречување перење пари и финансирање на тероризам** („Службен весник на РСМ“ бр. 120/18 и измени), се воспоставува надзор и над субјектите што даваат услуги поврзани со виртуелни средства. Доколку апликацијата нуди функции како:

- размена меѓу криптовалута и фиат-валута,
- чување на криптовалута (дигитален паричник),
- посредување преку платформи за тргување,

тогаш ќе се смета за „давач на услуги поврзани со виртуелни средства“ и подлежи на регистрација кај **Управата за финансиско разузнавање**, согласно член 4 и член 86 од наведениот закон. Потребно е и усогласување со процедурите за идентификација на клиенти, известување за сомнителни трансакции и водство на евиденции.

Заштита на лични податоци:

Секој информациски систем што обработува лични податоци мора да ги почитува обврските од **Законот за заштита на личните податоци** („Службен весник на РСМ“ бр. 42/20), кој е усогласен со **Општата регулатива за заштита на податоци (GDPR)** на Европската Унија. Апликацијата мора да ги почитува принципите на транспарентност, минималност, законитост и безбедност при обработка на лични податоци на корисниците.

Заклучок:

Бидејќи оваа апликација во својата почетна фаза **не врши размена, чување, ниту посредување со криптовалута**, туку служи само за **информативен приказ и анализа на податоци**, таа **не подлежи на обврска за регистрација како крипто-сервис**. Единствена обврска е усогласување со прописите за приватност и заштита на лични податоци. Доколку во иднина се додадат функционалности за тргување, размена или дигитални паричници, ќе биде потребно дополнително усогласување со надлежната регулатива.

3. Преглед на производот (Product Overview)

3.1 Перспектива на производот (Product Perspective)

Апликацијата претставува веб-базиран систем кој им овозможува на корисниците следење, анализа и визуелизација на податоци за највредните активни криптовалути. Системот функционира како посредник меѓу корисниците и надворешните крипто-берзи, од кои автоматски се преземаат историски и тековни податоци за пазарните движења.

Системот се состои од повеќе логички модули:

- **Модул за автоматско прибирање на податоци (ingestion)**
- **Модул за обработка, филтрирање и агрегирање податоци**
- **Модул за долгорочно складирање**
- **Модул за анализа и визуелизација**
- **Веб кориснички интерфејс**

Апликацијата е дизајнирана да биде модуларна и лесно проширлива, со можност за додавање нови функционалности и интеграции со дополнителни берзи и аналитички компоненти во идни фази од развојот.

3.2 Функционалности на производот

- Системот автоматски ги презема податоците за највредните активни криптовалути од надворешни крипто-берзи.
- Системот ги обработува и филтрира податоците за да обезбеди валидни, актуелни и конзистентни информации.
- Системот ги складира обработените податоци во долгорочен формат погоден за пребарување и анализа.
- Системот овозможува пребарување и филтрирање на историски и агрегирани податоци за секоја криптовалута.
- Системот нуди визуелизација на податоците преку графички прикази, статистики и аналитички компоненти.
- Системот ги ажурира податоците за да обезбеди тековни и точни информации.
- Системот овозможува мерење на перформансите на обработката и прикажување на метрики како вкупно просечно време за ingest, трансформација и складирање.

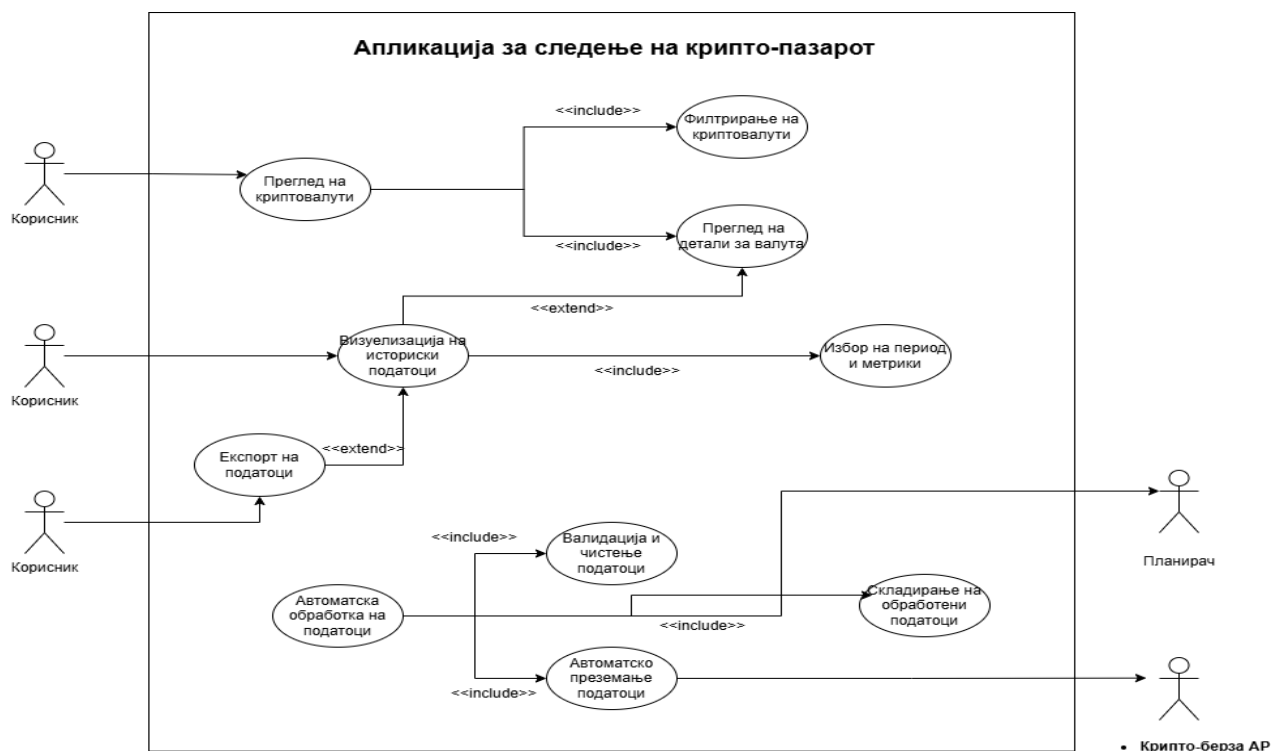
3.3 Главни сценарија на користење (Main use cases)

| ID | Назив на сценарио | Краток опис |
|-----|-------------------------------------|---|
| UC1 | Преглед на криптовалути | Корисникот преку веб интерфејс добива листа на највредните активни криптовалути. Податоците се доставуваат до системот кој ги складира и ажурира. |
| UC2 | Визуелизација на историски податоци | Корисникот избира криптовалута и добива графички приказ на дневни цени, волумен и други релевантни показатели. |
| UC3 | Автоматска обработка на податоци | Системот автоматски ги презема, проверува и надополнува податоците, осигурувајќи конзистентна историска база. |
| UC4 | Експорт на податоци (опционално) | Корисникот може да преземе агрегирани и историски податоци во стандарден формат. |

3.4 Доменски модел (Domain Model)

Доменскиот модел ги опишува главните концепти во системот и нивната меѓусебна поврзаност.

3.4.1 Дијаграм на активности



Главни ентитети:

➤ **Currency (Валута)**

Претставува криптовалута што се следи во системот. За секоја валута се чуваат основни информации, како симбол, име, пазарни показатели и тековна состојба.

➤ **HistoricalData (Историски податок / OHLCV)**

Претставува дневен запис за цените и обемот на тргување за одредена криптовалута. Секоја валута има повеќе историски записи поврзани преку нејзиниот симбол.

➤ **Exchange (Извор / Берза)**

Претставуваат надворешни системи (крипто-берзи) од кои апликацијата ги презема пазарните податоци. Може да постојат повеќе берзи како извори на податоци.

Врски и поврзаност меѓу ентитетите:

➤ **Exchange → Currency**

Една берза може да обезбедува податоци за повеќе валути.

➤ **Currency → HistoricalData**

Една валута има повеќе историски записи (дневни OHLCV податоци).

4. Архитектура на системот (System Architecture)

4.1 Архитектонски стилови

Архитектурата на системот е дизајнирана да обезбеди јасна организација, модуларност и можност за понатамошно проширување. Во продолжение се опишани архитектонските стилови и нивната улога во системот.

- **Pipe-and-Filter архитектура** за обработка и трансформирање на податоците добиени од крипто-берзите.
- **Слоевита (Layered) архитектура** за организирање на логичката структура на системот (презентациски слој, апликативен слој и слој за податоци).
- **Component-based архитектура** за имплементација на корисничкиот интерфејс.

Комбинацијата на овие стилови овозможува подобра повторна употребливост, скалабилност и едноставно одржување на системот.

4.2 Pipe-and-Filter архитектура

Модулот за обработка на податоци во системот е организиран според Pipe-and-Filter архитектонскиот стил. Обработката е поделена на повеќе последователни чекори (филтри), при што секој филтер има јасно дефинирана и независна функција. Излезот од еден филтер претставува влез во следниот, што овозможува јасен и модуларен тек на податоци.

Главни филтри во обработката:

- **Филтер 1 – Избор и филтрирање на симболи**

Ги прибира највредните активни криптовалути од избраната берза и ги отстранува невалидните или делистирани записи.

- **Филтер 2 – Проверка на постојни податоци**

За секоја валута се утврдува до кој датум постојат зачувани податоци, со цел да се идентификуваат недостасувачки временски периоди. За оние податоци за кои не постои датум (не се складирали во базата на податоци) – се превземаат потребните податоци од соодветните API endpoints.

- **Филтер 3 – Преземање на недостасувачки податоци**

Се преземаат сите податоци што недостасуваат за секоја валута од датумот кој постојат до денешниот датум и се подготвуваат за складирање.

Pipe-and-Filter пристапот овозможува јасна поделба на одговорностите, полесно тестирање на секој филтер одделно, како и можност за паралелна обработка и проширување со дополнителни филтри во иднината.

4.3 Словита (Layered) архитектура

Backend делот од системот е логички организиран во повеќе слоеви, при што секој слој има јасно дефинирана улога и одговорност. Овој пристап овозможува подобра организација на кодот, полесно одржување и можност за независен развој на поединечни делови од системот.

Главни слоеви:

- **REST API/Веб слој**

Ги претставува крајните RESTful endpoints со кои се поврзува клиентот преку корисничкиот интерфејс. REST архитектурата во системот служи како механизам за комуникација помеѓу клиентската (frontend) апликација и функционалностите за обработка и пристап до податоци.

- **Апликативен (логички) слој**

Го содржи бизнис-логичкиот дел на системот. Тука се дефинираат правилата за обработка, валидација и трансформација на податоците, како и управување со главната функционалност.

- **Слој за податоци**

Ги опфаќа механизмите за пристап, организирање и складирање на податоците. Овој слој обезбедува сигурно и конзистентно чување на историските и тековните пазарни информации.

Слоевитата архитектура овозможува секој слој да се развива независно, а истовремено обезбедува структурираност и транспарентна комуникација помеѓу деловите на системот.

4.4 Component-based архитектура

Component-based архитектура ја организира frontend апликацијата во независни компоненти наменети за повеќекратна употреба, така што секој е одговорен за одреден дел од корисничкиот интерфејс и неговата логика. Овие компоненти се комбинираат како градежни блокови за да го формираат целиот кориснички интерфејс.

4.5 Microservices архитектура

Во подоцнежните фази од развојот на системот се предвидува примена на микросервисна архитектура, со цел да се постигне поголема скалабилност, независен развој на модулите и подобра изолираност на функционалностите. Во рамки на овој архитектонски пристап, системот се дели на повеќе самостојни сервиси, при што секој сервис има јасно дефинирана одговорност и може да се развива, тестира и поставува независно од останатите компоненти.

Главни принципи на микросервисната поделба:

- **Делба според доменска логика**

Функционалностите за обработка на податоци, историски податоци, аналитика и управување со корисници се изолираат во посебни сервисни единици.

- **Самостојност на сервисите**

Секој сервис работи како независна логичка целина и може да биде надграден или заменет без значително влијание врз останатите модули.

- **Јасни комуникациски интерфејси**

Сервисите комуницираат преку добро дефинирани интерфејси, што овозможува стабилна размена на податоци и лесно проширување.

- **Подоброена скалабилност**

Сервисите што обработуваат големи количини податоци или извршуваат сложени операции можат да се скалираат независно, со што се оптимизира искористеноста на ресурсите.

Микросервисната архитектура обезбедува подготвеност на системот за идни подобрувања, интеграции со надворешни услуги и работа во распределено или cloud опкружување.

5. Функциски и нефункциски барања

Функциски барања

| ID | Барање | Тип |
|-----|--|-----------|
| FR1 | Системот автоматски ги презема податоците за највредните активни криптовалути од надворешен извор (крипто-берза). | Функциско |
| FR2 | Системот ги обработува и филтрира податоците преку јасно дефиниран процес на трансформација (Pipe-and-Filter). | Функциско |
| FR3 | Системот ги складира обработените податоци во конзистентен, долгорочен и пребарлив формат. | Функциско |
| FR4 | Системот овозможува следење на перформансите при обработка на податоците преку мерење на временски и ресурсни метрики. | Функциско |
| FR5 | Системот треба да овозможи визуелизација и пристап до историски и агрегирани податоци. | Функциско |
| FR6 | Системот треба да обезбеди експорт на обработени или агрегирани податоци во стандарден формат. | Функциско |

Нефункциски барања

| ID | Барање | Тип |
|------|--|-------------|
| NFR1 | Обработката на голем обем податоци треба да биде изведена во разумно време (на пр. дневната обработка да не надминува неколку минути). | Нефункциско |
| NFR2 | Системот мора да обезбеди идемпотентна обработка, односно да не создава дупликати и да не нарушува постојни податоци. | Нефункциско |
| NFR3 | Компонентите за обработка треба да бидат модуларни, изолирани и повторно употребливи. | Нефункциско |
| NFR4 | Системот треба да биде стабилен, отпорен на грешки и способен да продолжи со работа при делумни неуспеси или прекини. | Нефункциско |
| NFR5 | Системот треба да биде лесен за проширување, одржување и интеграција со нови извори на податоци или сервиси. | Нефункциско |
| NFR6 | Системот треба да биде скалабилен, со можност за обработка на поголем број валути и податоци без значително намалување на перформансите. | Нефункциско |

6. Технологии и алатки

- ✓ **Backend:** Java Spring Boot
- ✓ **Frontend:** React (JavaScript)
- ✓ **Database:** MongoDB
- ✓ **Податочен извор:** Binance API, Coingecko API
- ✓ **Контрола на верзии:** GitLab (официјален репозиториум на ФИНКИ), Github
- ✓ **Алатки:** Postman, IntelliJ IDEA, Visual Studio Code, MongoDB Atlas

7. Тестирање и валидација

- ✓ Тестирање на точноста и конзистентноста на обработените податоци.
- ✓ Интеграционо тестирање на процесот за автоматска обработка.
- ✓ Основни тестови за проверка на стабилноста и модуларноста на филтрите во процесот на трансформација.
- ✓ Тестирања на перформанси со цел да го измериме времето потребно за обработка на поголем обем податоци.

8. Идни подобрувања

- ✓ Додавање визуелен кориснички интерфејс со интерактивни графици и статистики.
- ✓ Интеграција на аналитички модули за напредни погледи и споредби помеѓу криптовалути.
- ✓ Додатоци за известување, алармирање или препораки.
- ✓ Поддршка за cloud архитектура во подоцнежни фази.

9. Заклучок

Апликацијата е дизајнирана да обезбеди сигурен и структуриран начин за прибирање, обработка и анализа на податоци од криптовалути пазари. Во оваа фаза е поставена основата за обработка на податоците преку јасно дефиниран процес, што овозможува стабилност, модуларност и можност за идно проширување на системот.

Во наредните фази се предвидува развој на кориснички интерфејс, дополнително надградување на архитектурата и проширување со нови функционалности со цел да се постигне целосно функционална и скалабилна веб апликација.