

Концептуална Архитектура

Концептуалната архитектура е првиот чекор во одговорот на барањата на засегнатите страни и претставува иницијален дизајн на архитектурата на апликацијата која се развива. Нејзината главна цел е да ги дефинира основните компоненти и нивните одговорности на ниво на домен, обезбедувајќи едноставен, јасен и структуриран поглед на функционалностите на системот. Овој пристап помага во разјаснувањето на основните концепти, истовремено обезбедувајќи фокус на основните потреби на корисниците и цели на апликацијата. Концептуалната архитектура, исто така, служи како основа за понатамошниот развој и детали на системот, додека истовремено ги води дизајнерите и развивачите преку целиот процес на изградба.

1. СОФТВЕРСКИ БАРАЊА

1.1 Функциски барања

1.1.1 Преземање на Податоци

- 1.1.1.1 Системот треба да презема историски податоци за период од минимум 10 години
- 1.1.1.2 Системот треба да проверува за нови податоци и да ја ажурира базата
- 1.1.1.3 Системот треба автоматски да ја презема комплетната листа на засегнати страни (издавачи) од веб-страницата на Македонската берза на дневна база
- 1.1.1.4 Системот треба да ги филтрира сите податоци (обврзници) од листата на засегнати страни
- 1.1.1.5 Системот треба да ги презема податоците (отворената цена, затворената цена, највисоката цена, најниската цена, обемот на тргување, датумот на тргување) за секој трговски ден

1.1.2 Обработка на Податоци

- 1.1.2.1 Системот треба да ги трансформира податоците (датумите) во конзистентен формат системот треба да овозможи проверка на последниот датум на достапни податоци
- 1.1.2.2 Системот треба да ги конвертира сите податоци (датуми) во формат "YYYY.MM.DD"
- 1.1.2.3 Системот треба да ги форматира сите податоци (цени) со точка како децимален разделувач и записка како разделувач за илјади
- 1.1.2.4 Системот треба да ги валидира сите податоци (нумерички вредности) пред зачувување

1.1.3 Складирање на Податоци

- 1.1.3.1 Системот треба да ги зачувува податоците во структурирана база
- 1.1.3.2 Системот треба да спречи дупликати на податоци при внесување на нови податоци
- 1.1.3.3 Системот треба да овозможи спојување на нови со постоечки податоци

1.1.3.4 Системот треба да обезбеди интегритет на податоците при зачувување

1.2 Нефункционални барања

1.2.1 Перформанси

1.2.1.1 Системот треба да биде способен да обработи голема количина на историски податоци

1.2.1.2 Системот треба да комплетира целосно иницијално полнење на базата за помалку од 30 минути

1.2.1.3 Системот треба да процеси дневно ажурирање на податоците за помалку од 5 минути

1.2.1.4 Системот треба да го оптимизира времето на преземање и обработка на Податоците

1.2.1.5 Системот треба да работи ефикасно со минимално користење на ресурси

1.2.2 Достапност и сигурност

1.2.2.1 Системот треба да биде достапен 99.9% од времето на месечно ниво

1.2.2.2 Системот треба да има механизми за справување со мрежни прекини

1.2.3 Скалабилност

1.2.3.1 Системот треба да подржува додавање на нови Функции (филтри)

1.2.3.2 Системот треба да подржува зголемување на бројот на засегнати страни за минимум 200% од иницијалните

1.2.3.3 Системот треба да биде способен да се прошири за додатни извори на податоци

1.2.4 Одржливост

1.2.4.1 Системот треба да генерира известување при неуспешно преземање или обработка

1.2.5 Безбедност

1.2.5.1 Системот треба да има механизам за верификација на интегритетот на податоците

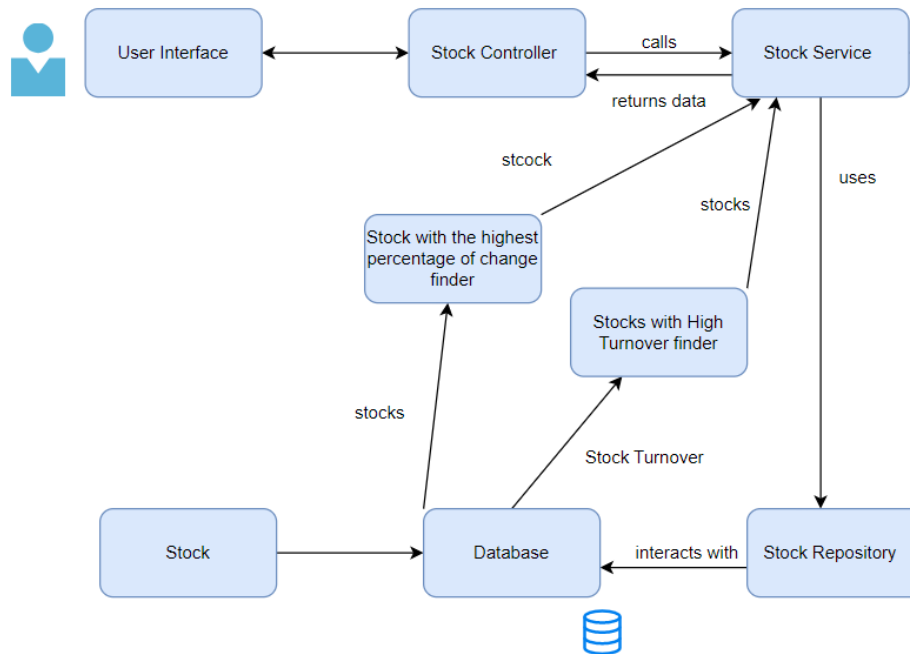
1.2.6 Усогласеност

1.2.6.1 Системот треба да работи во согласност со политиките за пристап до Податоци на Засегнатите страни (Македонската берза)

1.2.6.2 Системот треба да го ограничи бројот на барања до серверот при преоптовареност на берзата.

Data	Function	Stakeholder	System	Abstract Concept
историски податоци	презема	засегнати страни	Системот	интегритет
податоци	проверува	издавачи		ефикасно работење
обврзници	ажурира	Македонска берза		достапност
датуми	филтрира			сигурност
цени	трансформира			скалабилност
нумерички вредности	конвертира			одржливост
извори на податоци	форматира			безбедност
дупликати	валидира			усогласеност
количини на податоци	Огранчи(број на барања			политики за пристап
	зачувува			нови функции
	спречи (дупликати)			зголемување на бројот
	овозвозможи спојување			структурирана база
	обезбеди			
	обработи			
	комплетира			
	процесира			
	оптимизира			
	додавање			
	генерира известување			
	верификација			

2. Концептуална архитектура



3 Одговорности на компоненти

- **StockController**
 - Раководи со HTTP барања поврзани со акции. Комуницира со **StockService** за да ги преземе и манипулира податоците за акциите.
- **StockService**
 - Содржи бизнис логика за обработка на операции со акции. Презема податоци за акции од **StockRepository**. Извршува специфични операции како наоѓање акции со висок промет или највисока промена.
- **StockRepository**
 - Раководи со интеракцијата со базата на податоци за да преземе, зачува, ажурира или избрише записи за акции. Обезбедува методи како `findAllStocks`, `findStockById`, `saveStock`, и `deleteStockById`.
- **Stock**
 - Претставува ентитет за акција. Содржи својства како `id`, `turnoverBEST`, `percentAvg`, `date` и др.
- **Database**
 - Чува податоци за акции перзистентно.
- **Stocks with High Turnover finder**
 - Пронаоѓање на акции со висок обрт
- **Stock with the highest percentage of change**
 - Пронаоѓање на акција со највисок процент на промена