

# **АРХИТЕКТУРЕН ДИЗАЈН**

# Концептуална архитектура

Концептуалната архитектура се однесува на дизајн на софтверски систем, на високо ниво, кој ја вклучува неговата општа организација, функционалност и важни карактеристики.

Типично создадена во почетокот на циклусот на развој на софтвер, концептуалната архитектура нуди целосен преглед на способностите и функционалностите на системот, како и неговите главни елементи и меѓусебни врски.

Компоненти:

- AppUI
- Admin Panel
- Admin Service
- Stocks Service
- Stocks Finder
- Stocks Information Manager
- Database

Објаснување на компонентите:

- 1 AppUI
  - Приказ на графици со дневни податоци за цените на акциите
  - Приказ на текстуални информации како опис на апликацијата, најдобри акции за купување/продавање и слично.
  - Поле за пребарување на компании или симболи на акции
  - Приказ на детални информации за избрана компанија
  - Визуелен приказ на предвидувањата за идните трендови врз основа на AI модели
- 2 Admin Panel
  - Листање на сите компании и нивните историски податоци
  - Преглед и управување со податоци за акциите
  - Управување со корисници и нивните привилегии
- 3 Admin Service
  - Додавање и ажурирање на информации за компании и акции
  - Валидација на внесени податоци за акции и компании
  - Управување со пристапот и привилегиите на корисниците на апликацијата
- 4 Stocks Service
  - Обработка на податоци од базата и нивното испраќање до другите модули
  - Пресметка на важни индикатори
  - Поддршка за API барања од корисничкиот интерфејс и Admin Panel
- 5 Stocks Finder
  - Scraping за добивање историски дневни податоци за акции
  - Редовно ажурирање на базата со нови податоци
  - Проверка за интегритет на податоците
- 6 Stocks Information Manager
  - Додавање и отстранување на податоци за компании и акции
  - Организација на податоците по временски периоди и компании
  - Пребарување податоци за специфични акции или компании
  - Поддршка за подготовка на податоците за анализа
- 7 Database
  - Чување на податоци за компаниите (име, симбол)
  - Чување на аналитички податоци и предвидувања
  - Чување на историски дневни податоци за сите достапни издавачи (пр. отворена цена, затворена цена, највисока цена, најниска цена, волумен)

# Извршна архитектура

Овој дијаграм ја прикажува извршната архитектура на апликацијата за анализа на акции базирана на Македонската берза. Архитектурата е дизајнирана како трислоен систем кој вклучува кориснички интерфејс (React Frontend), сервисен слој (Spring Boot Backend) и база на податоци (PostgreSQL). Дополнително, се вклучени два специфични модули за обработка на барања - модул за пребарување компании и модул за генерирање извештаи и визуелизација на графици.

1. Кориснички интерфејс (GUI - React Frontend):
  - React Frontend е компонентата која ја користат крајните корисници за интеракција со апликацијата. Овде корисниците можат да пребаруваат компании, да гледаат графици или да генерираат извештаи.
    - Реализира HTTP барања до Backend сервисот за обработка на податоците.
2. Сервисен слој (Service - Spring Boot Backend):
  - Backend компонентата развиена во Spring Boot е централен дел од архитектурата.
  - Прима барања од React Frontend преку HTTP протокол.
  - Комуницира со PostgreSQL базата на податоци за читање, ажурирање и зачувување податоци. Ги повикува модулите за специјализирани задачи:
    - Генерирање извештаи и прикажување графици.
    - Пребарување на податоци за компании.
3. База на податоци (Data - PostgreSQL):
  - PostgreSQL е базата на податоци каде што се складираат сите информации за компании, акции и историски податоци.
  - Backend сервисот пристапува до овие податоци и ги користи за одговарање на корисничките барања.
4. Модул за генерирање извештаи и прикажување графици (Report Generator and Chart Display):
  - Оваа компонента е одговорна за креирање извештаи базирани на податоците од базата.
  - Истовремено, врши обработка и визуелизација на графици што се прикажуваат во Frontend-от.
5. Модул за пребарување компании (Company Search):
  - Ги обработува барањата за пребарување информации за компании. Користи податоци од базата и ги враќа резултатите преку Backend до корисничкиот интерфејс.
6. Процес на интеракција
  - Корисникот внесува барање преку React Frontend, на пример, пребарување за одредена компанија.
  - Барањето преку HTTP се праќа до Spring Boot Backend.
  - Backend сервисот:
    - Ако е потребно пребарување, го повикува модулот за пребарување на компании.
    - Ако е потребна визуелизација или извештај, го иницира модулот за генерирање извештаи и графици.
    - Ги користи податоците од PostgreSQL базата за секое барање.
  - Резултатите (на пример, графици или детали за компанија) се враќаат назад преку Backend до Frontend и се прикажуваат на корисникот.

# Имплементациска архитектура

Архитектурата на имплементацијата е дизајнирана се со цел да обезбеди ефикасна комуникација и обработка на податоци преку различни слоеви на системот. Овој систем вклучува спецификација на технички компоненти, технологии и алатки кои се користат за изградба и управување со системот, како и начините на кои ќе бидат интегрирани за да се постигне оптимално извршување на задачите и функциите.

Презентациониот слој делува како посредник помеѓу корисникот и системот, комуницирајќи со сервисниот слој, кој пак соработува со интегративниот слој за пристап до податоци и извршување на операции. Интегративниот слој овозможува читање и запишување на податоци во FileSystem и извршување на команди преку специфицирани сервисни модули.

**1. Клиентски интерфејс:** За динамично и интерактивно корисничко искуство, клиентскиот интерфејс ќе се развива користејќи веб-прелистувач со поддршка за HTML, CSS, JavaScript и Ajax. Овие технологии ќе овозможат брзо и ефикасно поврзување на корисникот со системот преку визуелно атрактивни и флексибилни елементи на интерфејсот.

**2. Веб сервер:** Презентациониот слој ќе се реализира со помош на React и TypeScript, што ќе овозможи изградба на високофункционален и модуларен кориснички интерфејс. Сервисниот слој ќе биде одговорен за дефинирање и имплементација на логиката за обработка на кориснички услуги, додека интегративниот слој ќе го обезбеди неопходниот пристап до податоци и комуникација со сервисните модули.

**3. Апликациски сервер:** FileSystem ќе служи како примарен извор за складирање и управување со податоци. За специфицираните операции со сервисните модули, ќе се користи SNAP-MODULE, кој ќе комуницира директно со оперативниот систем, овозможувајќи извршување на процеси и контекстуални команди.

Оваа архитектура нуди цврст и добро организиран систем, со јасно дефинирани улоги на секој слој, што резултира во ефикасно управување со податоци, лесна интеграција и високо оптимизирано корисничко искуство.