

Архитектурен дизајн

1. Концептуална архитектура

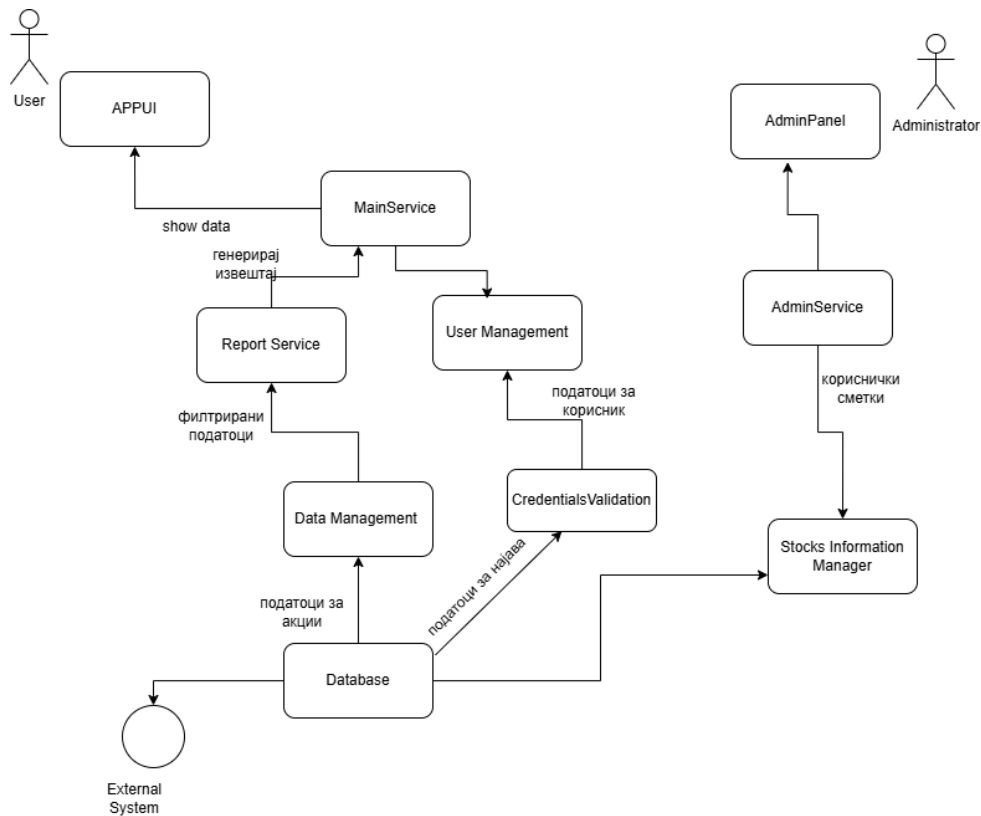
Концептуалната архитектура ја претставува големата слика на системот што го градиме. Таа се креира уште на почетокот, во раните периоди на развој на системот и се фокусира да ја долови главната цел на системот и неговите функционалности кои ќе ги понуди на крајните корисници. Оваа архитектура ги содржи главните компоненти кои би ги поседувал системот како и врските помеѓу овие компоненти кои ни покажуваат дека помеѓу два модули има комуникација односно размена на податоци. Ова е клучна фаза и прв одговор на потребите на засегнатите страни фокусирајќи се на функционалности на системот на доменско ниво.

Со анализа на функционалните барања ги одвоивме следните, најважни концепти:

Data	Function	Stakeholder	System	Abstract concept
Податоци за акции во последните 10 години	Достапни и ажурирани податоци во секое време	Корисник	Веб серверот на страната на Македонска Берза	Дозволи засновани на улоги
Податоци за кориснички сметки	Филтрирање на податоци	Администратор	Македонска берза	Управување со привилегии
Податоци за кориснички привилегии	Генерирање извештаи			Интуитивен интерфејс
	Пребарување акции, компании			Визуелизација со табели

Компоненти:

- APPUI
- AdminPanel
- AdminService
- MainService
- Report Service
- UserManagement
- DataManagement
- CredentialsValidation
- Database
- Stocks Information Manager



Објаснување на компонентите:

1. APPUI

- Приказ на акции и податоци за нив(цена,име на компанија)
- Поле за пребарување според име на компанија и датум
- Форма за најава и регистрација
- Визуелен приказ на податоци со помош на табели,графици

2.AdminPanel

- листање на сите компании и историски податоци
- листање на сите кориснички сметки
- листање и преглед на сите акции

3.AdminService

- управување со кориснички сметки и нивни привилегии
- управување со процесот на превземање податоци
- води грижа за точност и конзистентност на податоците

4.MainService

- Започни пребарување
- Генерирај извештај
- прикажи кориснички сметки

5.Report Service

- генерирање на извештај според избрани компании и временски период
- приказ на податоци табеларно,графички

6.UserManagement

- Регистрирање на нови корисници
- Најава на корисници

7.DataManagement

- филтрирање и трансформација на податоци кои што се добиени со scrapping од страната на Македонска берза(филтрирање на непотребни или невалидни информации)

8.CredentialsValidation

- проверува дали податоците на корисникот се валидни при најава

9.Database

- чување на историски, дневни податоци за сите достапни издавачи (име на издавач,највисока цена,најниска цена,волумен,отворена цена,затворена цела и сл.)
- чување на аналитички податоци и предвидувања

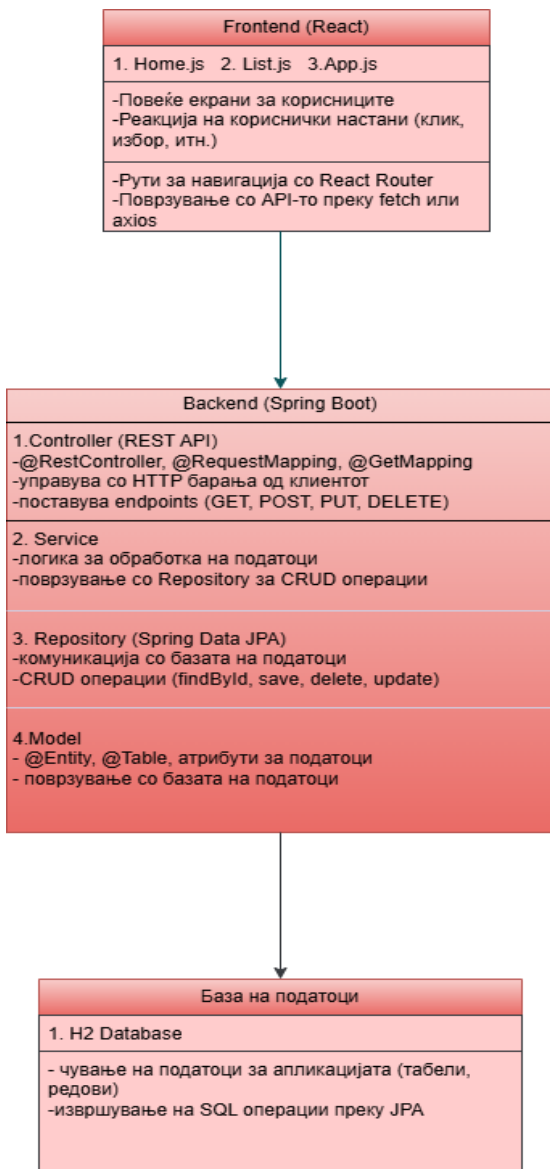
10.Stocks Information Manager

- Додавање нов корисник
- Бришење на корисник
- Листање акции
- Пребарување акции

2.Извршна архитектура

Извршаната архитектура ја опишува конкретната имплементација на различни технолошки компоненти во системот и како тие ќе функционираат во продукциско опкружување.

Извршната архитектура на оваа апликација користи **Java (Spring Boot)** и **React** како главни технологии. Spring Boot обезбедува серверска страна и логика, додека React се грижи за корисничкиот интерфејс. Базата на податоци е интегрирана преку Spring Data JPA, и комуникацијата помеѓу frontend и backend се врши преку REST API. Системот е лесно скалабилен и може да биде оптимизиран во зависност од барањата на апликацијата и продукциските услови.



Објаснување на компоненти од горенаведениот дијаграм на компоненти:

1. Frontend (React)

-Home.js: Почетната страница на апликацијата која има линкови или копчиња за навигација.

-List.js: Прикажува листа на записи од серверот (податоци).

-App.js: Главната компонента која ги имплементира сите рутирања и користи други компоненти (Home, List).

2. Backend (Spring Boot)

-Controller: Сервисот кој управува со HTTP барања од клиентот. Овде се дефинираат ендпоинтите за обработка на податоците.

-Service: Логика за обработка на податоците, поврзување со репозиторијата и извршување на операции (добавување, менување, бришење податоци).

-Repository: Интерфејс за комуникација со базата на податоци преку Spring Data JPA. Овозможува CRUD операции.

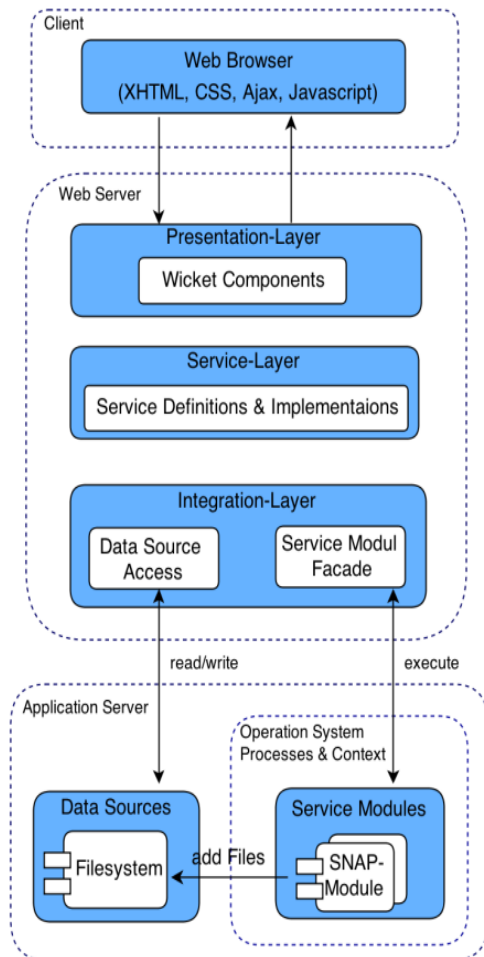
-Model: Класите кои се мапираат на базата на податоци. Секој модел ќе има соодветни анотации за да биде препознат како ентитет.

3. База на податоци

-H2 Database: Вградена база на податоци која се користи за развој. Ако се преминува во продукциско опкружување, може да се користи **PostgreSQL** или **MySQL**.

-JPA (Java Persistence API): Користи се за мапирање на објектите во базата на податоци, што овозможува CRUD операции.

3..Имплементациска архитектура



Имплементациската архитектура го претставува структурниот дизајн на системот што дефинира како ќе се имплементираат и интегрираат различните компоненти на софтверската апликација. Имплементациската архитектура е деталниот план за тоа како различните делови на системот ќе се реализираат и како ќе комуницираат меѓусебно за се се постигне функционалноста што е дефинирана од проектот.

Имплементациската архитектура го опишува техничкиот аспект на апликацијата.

Главни компоненти на архитектурата:

Нашата апликација ќе се базира на 3-слојна архитектура која ги содржи следниве основни компоненти:

1.1.Фронтенд: React

*Главна цел - користење за иградба на кориснички интерфејс (UI) и комуникација со бекенд преку REST API.

*Технологии-React, React Router (за рутирање помеѓу различни екрани)

*Функционалности:

- прикажување на податоци;
- праќање на податоци до бекендот преку HTTP барања;
- навигација помеѓу различни страници/екрани.

*Корисничкиот интерфејс комуницира со Spring Boot апликацијата преку REST API.

1.2.Бекенд:Spring Boot

*Главна цел-управување со логиката на серверот, обработка на барањата и поврзување со базата на податоци.

*Технологии-Java, Spring Boot, Spring Data JPA, H2.

*Функционалности:

- управување со HTTP барањата (REST API) од фронтендот;
- обработување и валидација на податоците;
- комуникација со базата на податоци преку Spring Data JPA.

*Подсистеми:

-Controller - управува со HTTP барањата (GET, POST, PUT, DELETE) и враќа резултати на фронтендот.

-Service - логика за управување со податоците, како што се CRUD операции.

-Repository - комуникација со базата на податоци, користејќи Spring Data JPA.

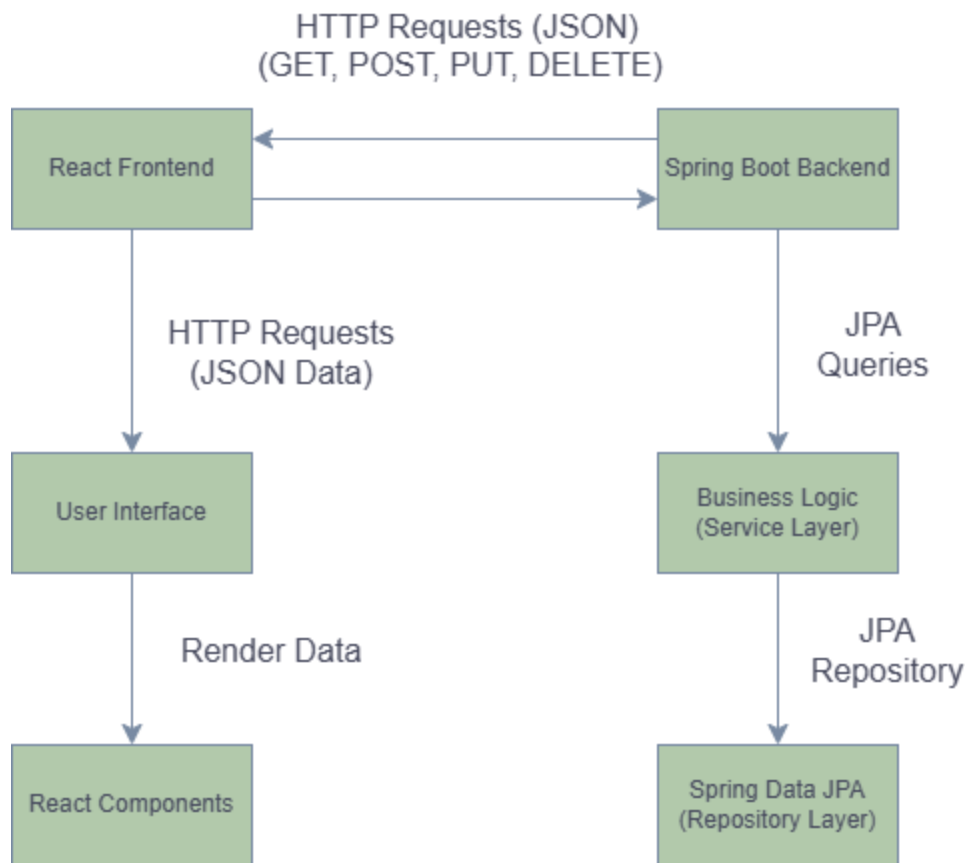
1.3.База на податоци : H2

*Главна цел - чување на податоците

*Потребни технологии - H2

*Функционалности:

- чување на податоци во табели
- преземање на податоци кога ќе се побараат од Spring Boot апликацијата



Опис на дијаграмот:

1. React Frontend : корисникот комуницира преку фронтенд апликацијата која испраќа HTTP барања кон Spring Boot серверот.

2. Spring Boot Backend : Бекенд апликацијата обработува барања и ги враќа одговорите, користејќи сервисни класи и репозиториуми.

3. База на податоци (H2) : Податоците се чуваат во базата на податоци, која е поврзана со Spring Boot преку JPA.

