

Архитектурен дизајн

1. Концептуална архитектура

Концептуалната архитектура ја организира и дефинира распределбата на одговорностите на доменско ниво, поставувајќи ја основата за почетниот архитектонски дизајн. Таа претставува иницијален одговор на потребите и целите дефинирани од страна на стејкхолдерите. Создавањето на оваа архитектура е водено од темелна анализа на функционалните и нефункционалните барања, обезбедувајќи стратешка насока за понатамошен развој и имплементација на системот.

Концептуалната архитектура на нашата апликација за следење и анализа на берзански податоци е изработена така што ги издвоивме клучните концепти од иницијалните барања и сценарија за употреба, а потоа истите ги категоризиравме во следните категории: **Data**, **Function**, **Stakeholder**, **System**, **Hardware** и **Abstract concept**.

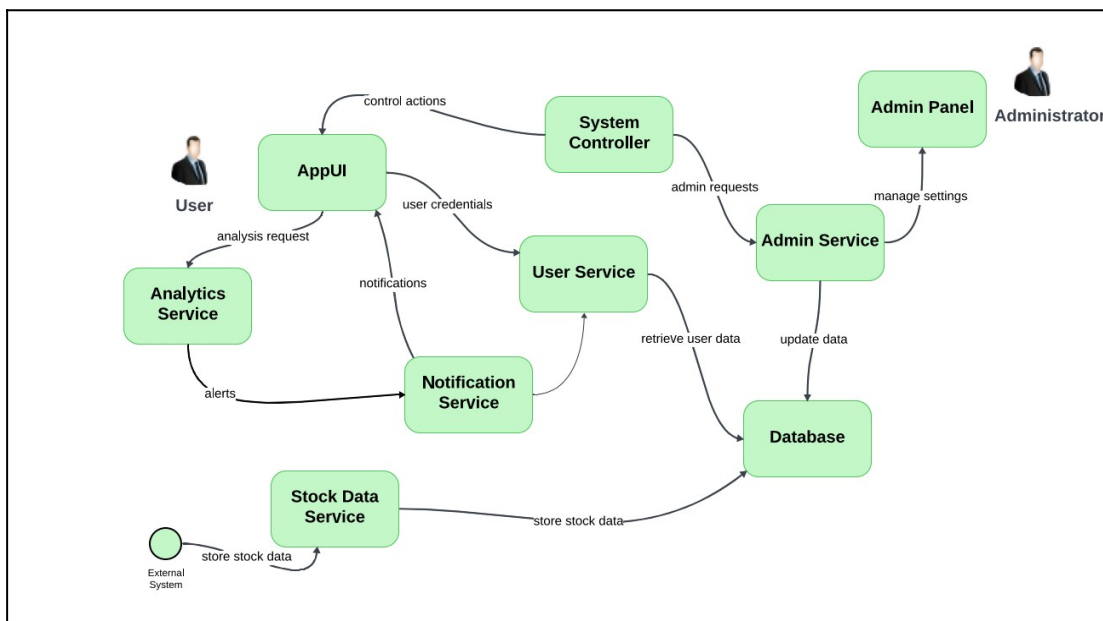
Data	Function	Stakeholder	System	Abstract concept
кориснички податоци	автентикација	корисник	User Service	безбедност на податоци
администраторски податоци	управување на систем	администратор	Admin Service	администраторски привилегии
известувања	испраќање аларми		Notification Service	важни информации
акции	анализирање податоци		Analytics Service	прогнозирање
историски податоци за акции	складирање податоци		Stock Data Service	интеграција

На сликата е прикажан концептуалниот поглед на нашата апликација. Корисникот комуницира со апликацијата преку **AppUI**, преку која се праќаат барања за анализа или преглед на податоци. **AppUI** ги проследува барањата до **System Controller**, кој има улога да ги оркестрира услугите на системот.

User Service ја обработува автентикацијата и ги повлекува корисничките податоци од базата.

Admin Service, од друга страна, овозможува администраторите да управуваат со поставките на системот, додека **Notification Service** испраќа известувања до корисниците за промени или аларми.

Податоците за акции и нивната историја се вчитуваат од **Stock Data Service**, кој директно ги презема од екстерниот систем. Тие податоци се складираат во базата на податоци и се обработуваат од **Analytics Service**, кој е одговорен за генерирање на анализи и нотификациите за корисниците.



Слика 1 – Концептуална архитектура

2. Извршна архитектура

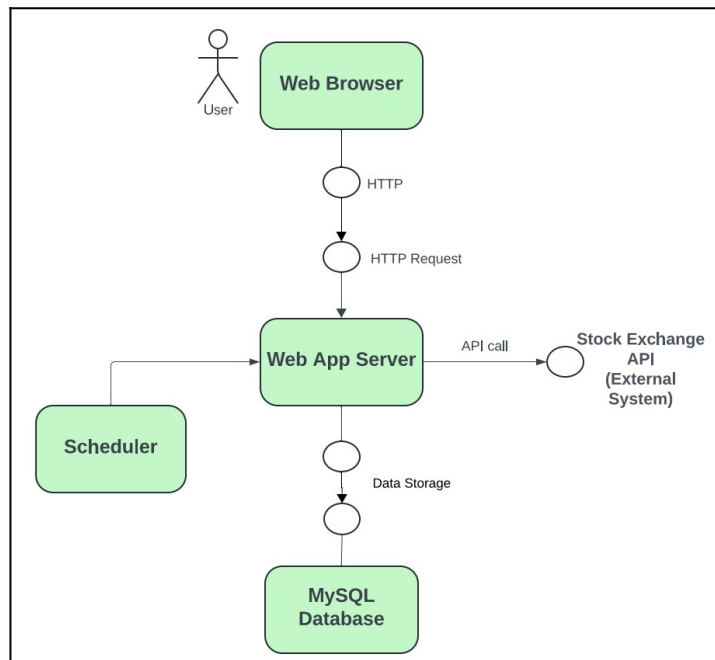
Извршната архитектура се насочува кон оперативната структура на системот за време на неговото извршување, вклучувајќи хардверски компоненти, потсистеми, процеси и нишки. Таа става акцент на нефункционалните карактеристики, како што се перформансите, безбедноста, употребливоста и скалабилноста, обезбедувајќи стабилна и ефикасна поддршка за работењето на системот.

Во прилог следува извршната архитектура на нашиот систем. Во оваа архитектура, клиентот комуницира со системот преку **веб прелистувач**, кој праќа HTTP барања до **апликацискиот сервер** каде е сместена бизнис логиката.

Апликацискиот сервер прави повици кон **Stock Exchange API** (екстерен систем) за да ги преземе податоците за берзата.

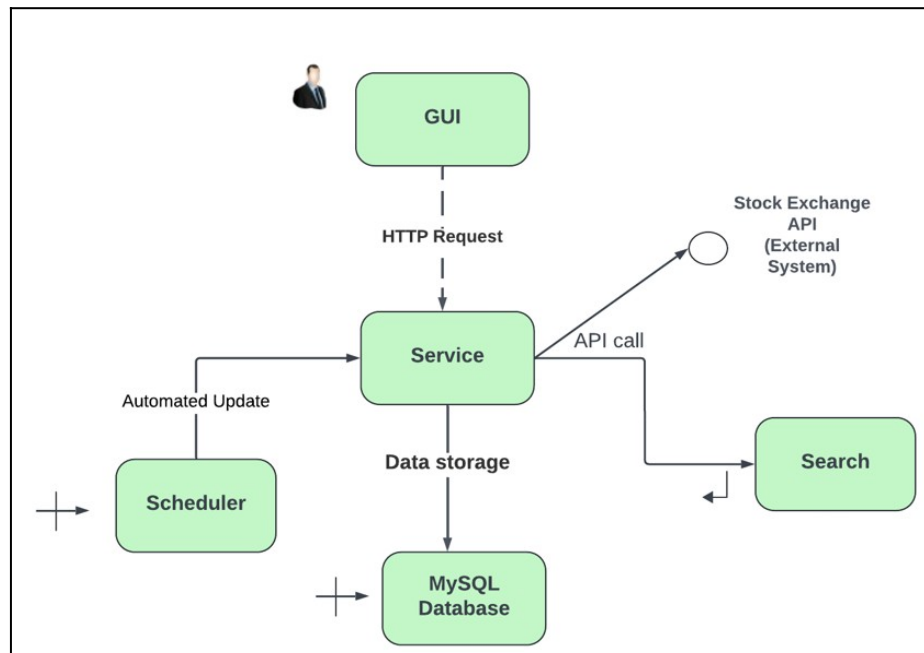
Дополнително, апликацискиот сервер комуницира со **MySQL базата на податоци**, каде се складираат и обработуваат податоците.

Scheduler-от, пак, работи независно за автоматизирање на задачи, како што се периодично вчитување на нови податоци од екстерниот систем или ажурирање на базата на податоци.



Слика 2.1 – Едноставна верзија на извршната архитектура

Во прилог е и подеталната верзија на извршната архитектура.



Слика 2.2 Детална верзија на извршната архитектура

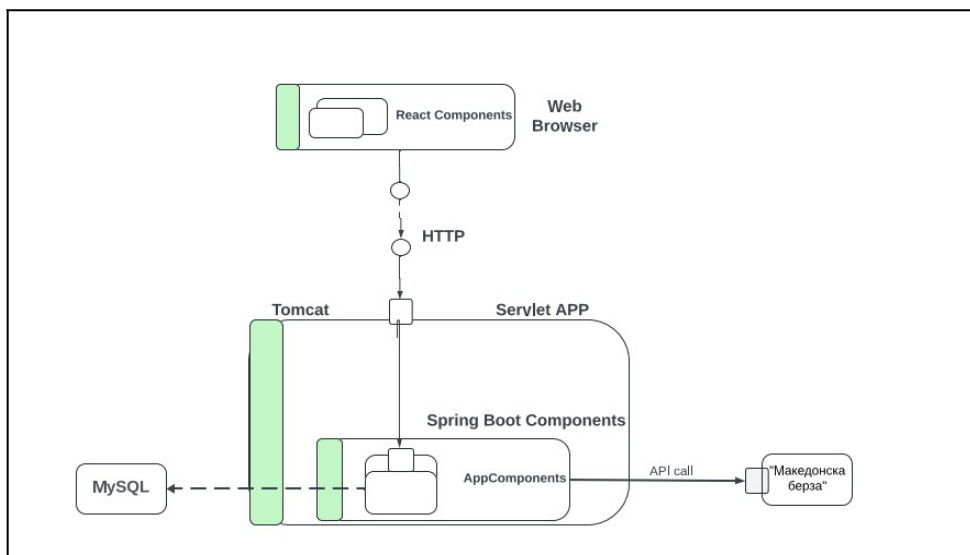
3. Имплементациска архитектура

Имплементациската архитектура се концентрира на начинот на кој системот е конструиран, опфаќајќи ги сите технички компоненти што треба да бидат имплементирани. Таа вклучува софтверски пакети, библиотеки, рамки (frameworks), класи и слично. Главниот фокус е насочен кон нефункционалните барања кои не се поврзани со времето на извршување (non-runtime) и кон обезбедувањето на квалитетни атрибути, како што се одржливоста, модуларноста и леснотијата на интеграција.

Веб прелистувачот, кој комуницира со апликацијата, испраќа HTTP барања до апликацискиот сервер, во нашиов случај Tomcat. Tomcat го пренасочува барањето до соодветниот сервлет во Spring Boot компонентите. Овие сервлети, со помош на сервисите и репозиториумите, го обработуваат барањето.

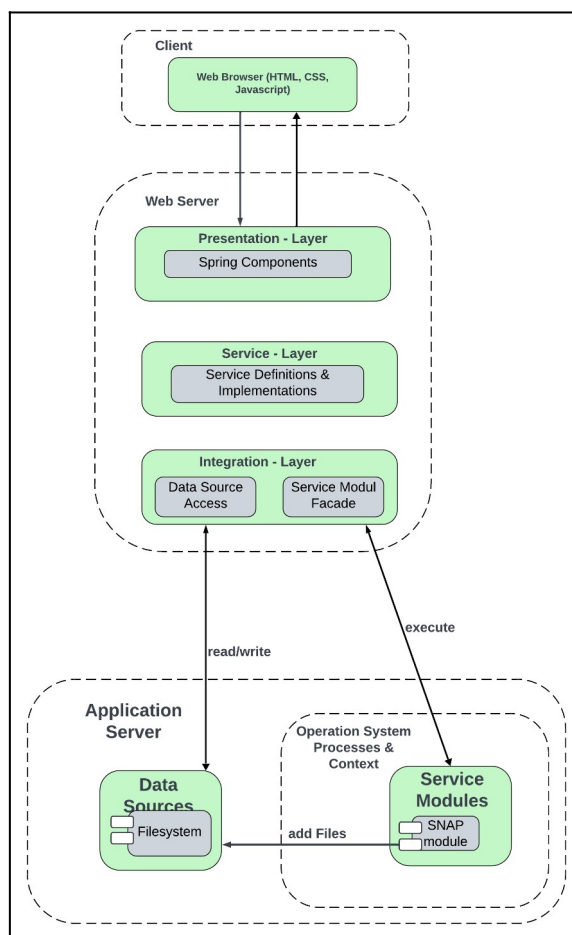
Во процесот на обработка, се прават API повици до базата на податоци – MySQL, како и надворешен API повик до "Македонска берза" за дополнителни податоци.

Одговорот од сервлетот се испраќа назад до веб прелистувачот. React апликацијата, со своите компоненти, го прикажува резултатот на корисникот, заедно со дополнителни информации и функционалности за пребарување или филтрирање.



Слика 3.1 – Едноставна верзија на имплементациска архитектура

Подеталната верзија на имплементациската архитектура е прикажана на следната слика.



Слика 3.2 – Детална верзија на имплементациска архитектура