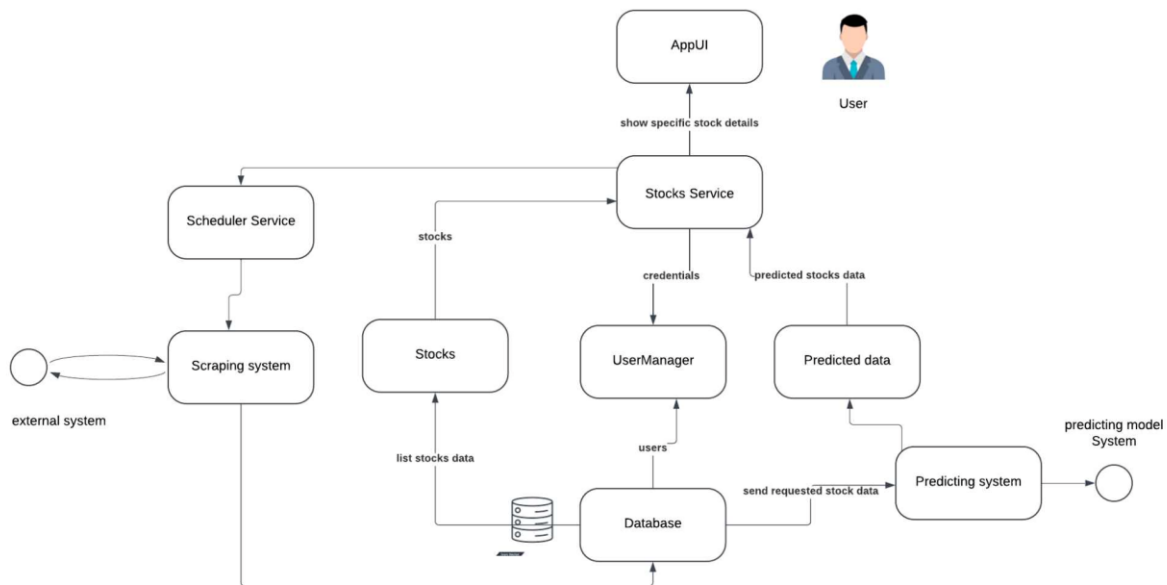


1. Концептуална архитектура

Оваа архитектура опишува систем за обработка и анализа на податоци од берза, каде што компонентите се организирани да обезбедат автоматизирано прибирање, обработка, и презентирање на податоци на крајниот корисник.

- **AppUI:** Претставува точката на интеракција со корисникот. Прикажува специфични детали за акциите добиени преку *Stocks Service*.
- **Stocks Service:** Управува со барањата од интерфејсот. Ги добива податоците за акциите од компонентата *Stocks* и предвидувањата од *Predicted Data*. Ги проследува деталите назад до интерфејсот.
- **Scheduler Service:** Автоматизирано иницира процеси за прибирање податоци. Испраќа барања до *Scraping System* за ажурирање на податоците.
- **Scraping System:** Презема податоци од веб-страницата на македонската берза и ги доставува до компонентата *Stocks*.
- **Stocks:** Ги чува и управува обработените податоци за акциите и ги доставува до *Stocks Service* и *Database* за понатамошна употреба.
- **Database:** Обезбедува информации за корисниците преку *UserManager* и податоци за акциите преку *Stocks* и *Predicting System*.
- **UserManager:** Управува со податоците за корисниците. Проследува информации за корисниците до *Stocks Service*.
- **Predicting System:** Генерира предвидувања базирани на податоците за акциите. Користи модел за машинско учење за предвидување за да ги анализира податоците од *Database*. На крај, ги испраќа обработените предвидувања до *Predicted Data*.
- **Predicted Data:** Ја складира информацијата за предвидувањата и ја доставува до *Stocks Service* за крајниот корисник.
- **Надворешен систем:** Извор на податоци од македонска берза.

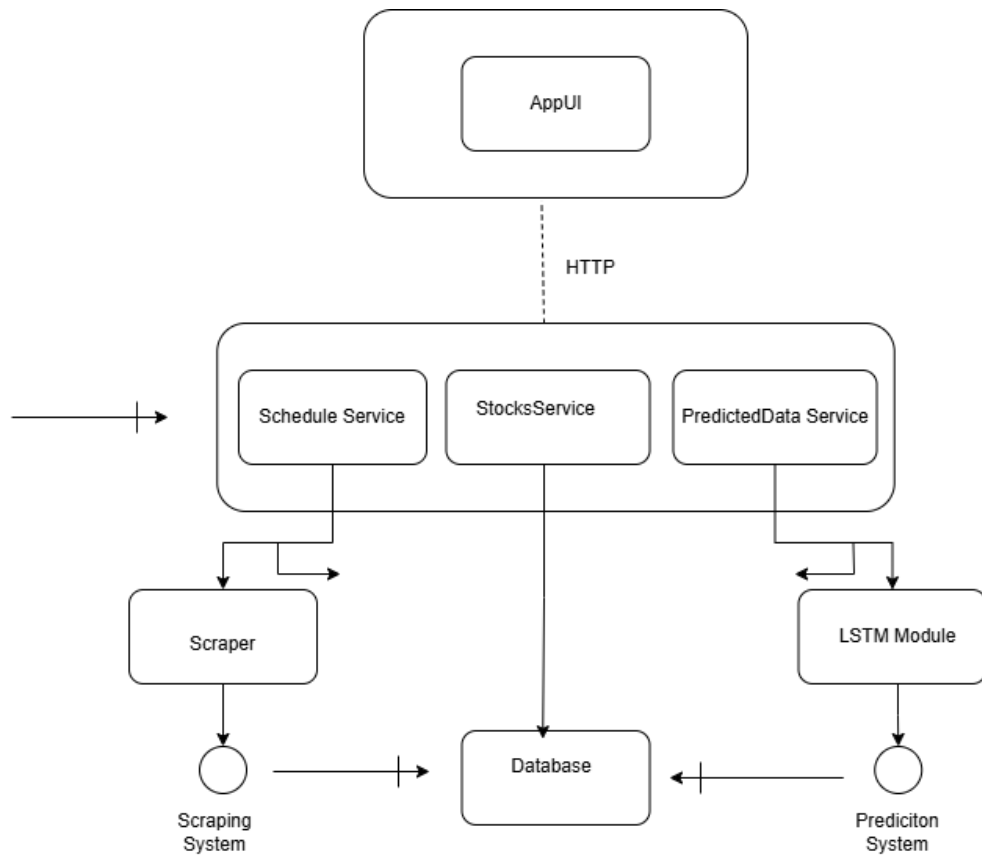
*Корисникот испраќа барање преку AppUI, кое се обработува од *Stocks Service*. Ако податоците се достапни, тие се добиваат од *Stocks* или *Predicted Data*. Ако податоците се застарени или недостасуваат, *Scheduler Service* активира *Scraping System* за да ги освежи податоците. Обработените податоци се чуваат во базата за понатамошна употреба.



2. Извршна архитектура

Ги илустрира интеракциите помеѓу различните компоненти, како што се:

- **AppUI:** Им овозможува на корисниците пребарување, прикажување податоци и предвидувања за соодветните компании.
- **Scheduler Service:** Периодично иницира задачи за собирање податоци преку **Scraping** системот.
- **Stocks Service u PredictedData Service:** Клучни за обработка и анализа на податоци за берзата.
- **LSTM модул:** Модул за машинско учење кој предвидува идни вредности базирани на историски податоци.



3. Имплементациска архитектура

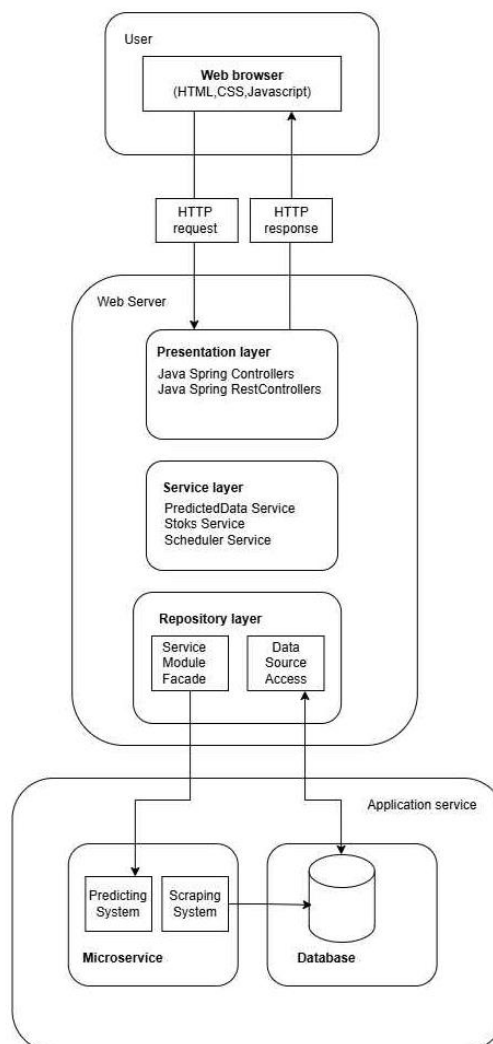
Го прикажува деталниот дизајн на имплементацијата на компонентите.

- **Прелистувач:** Овозможува на корисниците да комуницираат со системот преку интерфејс.
- **Слоевита веб архитектура:**
 - Имплементирани се три главни слоеви:
 - **Presentation Layer:** Содржи Spring MVC контролери и REST контролери за пречекување на HTTP барања.
 - **Service Layer:** Сервисите како PredictedDataService, StocksService, и SchedulerService ја имплементираат логиката за трансформација на податоците.

- **Repository Layer:** Се занимава со податоците преку модулите Service Module Facade и Data Source Access, кои обезбедуваат пристап до базата на податоци и микросервисите.
- Слоевитоста овозможува јасно одвојување на одговорностите, што ја прави имплементацијата лесно одржлива и скалабилна.

- **Микросервисна архитектура:**

- Модулите Predicting System и Scraping System се независни микросервиси што комуницираат со базата на податоци и другите делови на системот.
- Ова ја зголемува флексибилноста и можноста за независно ажурирање на специфични функционалности.



4. Примена на архитектурните стилови

1. **Слоевита архитектура:** Користена за фронтенд и бекенд комуникација, како и за јасно дефинирање на слоеви со различни одговорности.
2. **Микросервиси:** Клучни за независно развивање и скалирање на Scraping System, Predicting System, и сервисите.
3. **Цевки и филтри:** Податоците од македонската берза се обработуваат преку низа чекори, од преземање, трансформација, до предвидување.
4. **Контејнеризација:** Микросервисите се хостираат во контејнери за лесно управување и распоредување.