

## 1. Концептуална архитектура

Оваа архитектура обезбедува преглед на високо ниво на одговорностите, компонентите и нивните интеракции на апликацијата на ниво на домен. Обезбедува усогласување со функционалните и нефункционалните барања на системот.

### Компоненти:

- **GUI(Кориснички интерфејс):** Им овозможува на корисниците да иницираат преземање податоци, да прегледуваат обработени податоци за акции и да комуницираат со системот.
- **Платформата за обработка на податоци (филтри):**
  - **Филтер 1:** Презема шифри на издавачи од веб-страницата на Македонската берза.
  - **Филтер 2:** Проверува кои податоци се последно зачувани во базата.
  - **Филтер 3:** Ги презема податоците кои недостасуваат.
  - **Филтер 4:** Ги форматира и зачувува податоците во базата.
- **База на податоци:** Чување на податоците за издавачите и историските податоци за акции.
- **Надворешен систем (Веб-страница на берзата):** Извлекува податоци за акциите од веб-страницата на Македонската берза.

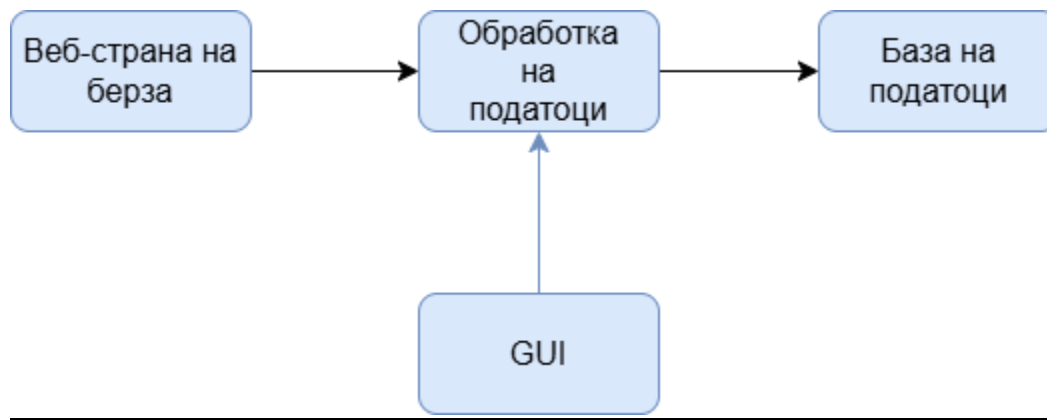
### Конектори:

- **Проток на податоци:**
  - Секвенцијален проток низ филтрите на платформата.
  - Двонасочна комуникација помеѓу GUI и бекендот (платформа и база на податоци).

### Наратив за сценарио

- **Корисникот го иницира процесот преку GUI:**
  - На пример, внесува име за кое што сака да види податоци.
- **GUI испраќа барање до платформата:**
  - Преку REST API, барање се праќа до Филтер 1.
- **Платформата го обработува барањето низ филтрите:**
  - Презема податоци, проверува постоечки записи, презема недостасувачки податоци и зачувува.
- **Базата на податоци враќа одговор:**
  - Резултатите се враќаат до GUI за прикажување.

**Дијаграм:**



## **2. Извршна архитектура**

Извршната архитектура се фокусира на тоа како системот ќе функционира во текот на неговото извршување. Ова вклучува процеси, нишки и како тие ќе комуницираат.

**Процеси:**

**1. Процес на веб-сервер:**

- Комуницира со корисничкиот интерфејс и ја координира обработката на податоците.

**2. Процес на платформа:**

- Користи филтри за обработка на податоците.

**3. Процес на база на податоци:**

- Базата на податоци ќе извршува SQL прашања за да чува, враќа или менува податоци на серверот.

**Конкурентност:**

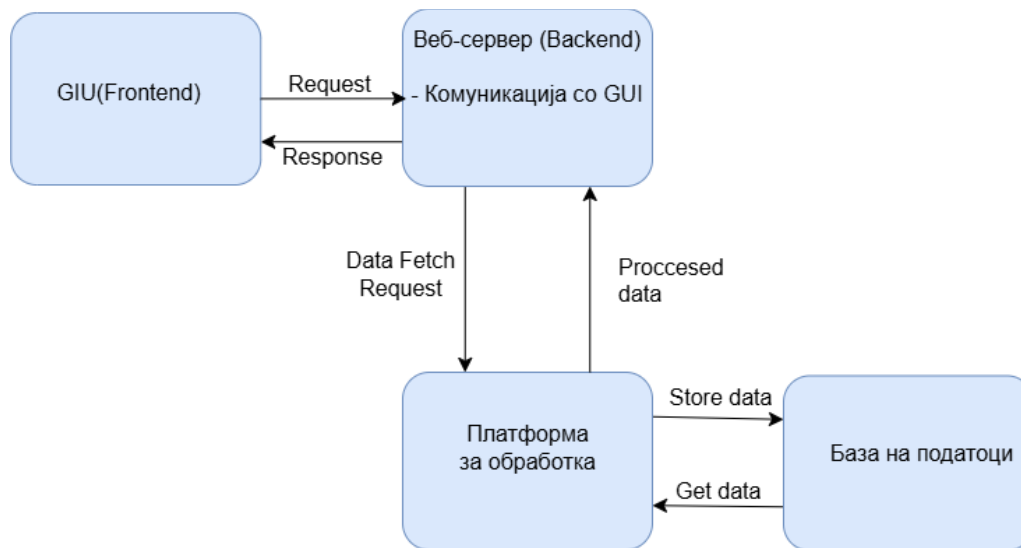
**1. Мултитрединг:**

- Секој филтер се извршува во своја нишка за да се подобри перформансот.

**2. Асинхрона комуникација:**

- GUI и Backend ќе комуницираат асинхронно за да се овозможи одзивност на апликацијата.

### Дијаграм:



### **3. Имплементациска архитектура**

Имплементациската архитектура го опишува изборот на технологии и како тие се распределени во системот. Таа прикажува како компонентите од концептуалната и извршната архитектура ќе бидат имплементирани со конкретни технологии.

#### Технолошки компоненти:

##### **1. Frontend(GUI):**

- HTML/CSS и JavaScript ќе се користат за креирање на структуриран и стилски кориснички интерфејс.
- React.js ќе се користи за динамично ажурирање на податоците на страната.

##### **2. Backend(Server):**

- Python (Flask/Django) за управување со API и координација на процесите.

##### **3. Платформа за обработка на податоците:**

- Python функции за обработка на податоци (BeautifulSoup за парсирање HTML и Playwright за понапредни можности за оптимизација на преземањето податоци).

##### **4. База на податоци:**

- SQLite за прототипирање.

##### **5. Контејнеризација:**

- Docker за пакување на апликацијата и нејзината инфраструктура во контејнери.

