

# Домашно број 2 – Архитектурен дизајн

## 1. Вовед

Овој документ претставува вовед во концептуалната, извршната и имплементациската архитектура на нашата веб апликација, со фокус на хибридна архитектура и вклучување на различни подархитектурни компоненти како *pipe* и *filter*, слоевата веб архитектура, дистрибуирани микросервиси и контејнеризација. Покрај дијаграмите на различните архитектурни погледи, опфатени се и чекорите при креација на истите, држејќи се до техниките наведени во материјалите за предавања. Дополнително, секој архитектурен поглед се држи до синтаксата дефинирана во материјалот од предавања.

Нашата цел е преку различни дизајни да се покаже и визуелизира системот заедно со неговите интеракции. Тоа ни помага во процесот на разбирање на системот и како би изгледала неговата имплементација.

## 1.2. Концептуална архитектура

Процесот на дизајн на концептуалниот архитектурен поглед започнува со подвлекување на клучните поими.

### 1.2.1. Клучни поими

r2.1.1. Системот треба да ги презема податоците за последните десет години на секој издавач од страната на Македонската берза.

r2.1.2. Системот треба да може автоматски да ја преземе листата на издавачи (компаниии или институции) од Македонската берза, исклучувајќи ги обврзниците или кодовите што содржат броеви.

r2.1.3. Системот треба да ја провери базата на податоци за секој издавач за последниот зачуван датум на податоци.

r2.1.4. Системот треба да ги преземе податоците за последните 10 години на дневно ниво за издавачи за кои нема претходно зачувани записи во базата на податоци.

r2.1.5. Системот треба да го идентификува последниот зачуван датум на издавачи за кои има претходно зачувани записи во базата на податоци.

r2.1.6. Системот треба автоматски да ги преземе сите податоци што недостасуваат до сегашниот датум за секој издавач.

r2.1.7. Базата на податоци на системот треба да ги чува податоците во стандардизиран македонски формат, конзистентен за сите записи.

r2.1.8. Системот треба да имплементира Веб апликација која ќе овозможи преглед на податоците за издавачите зачувани во базата на податоци.

r2.1.9. Веб апликацијата треба да овозможи визуелизација на податоците за секоја компанија преку графици кои се модел на временска серија.

r2.1.10. Веб апликацијата треба да овозможи споредба на графици на две компании.

r2.1.11. Веб апликацијата треба да овозможи табеларен приказ на податоците со можност за нивно сортирање и филтрирање.

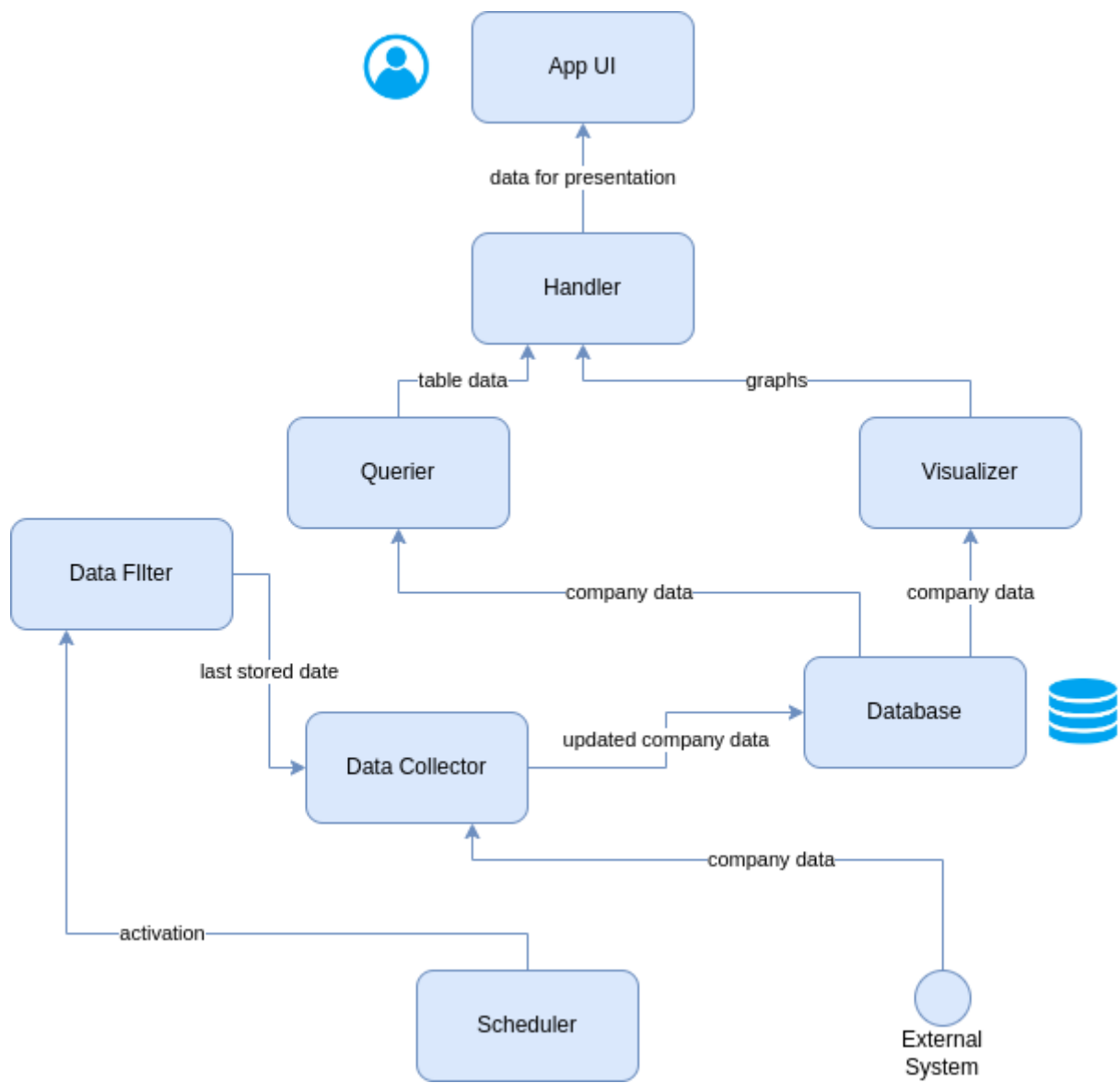
r2.1.12. Веб апликацијата треба да имплементира приказ на трендот и сезоналноста на временската серија.

r2.1.13. Веб апликацијата треба да имплементира движечка лента за процентуална промена на просечна цена на акција за издавачи со најголем профит.

Data	Function	Stakeholder	System	Abstract concept
издавач	визуелизација	корисник	страна на Македонска берза	последно зачуван датум
листа на издавачи	споредба			стандардизиран македонски формат
база на податоци	табеларен приказ			процентуална промена
график	движечка лента			најголем профит
временска серија	сортирање			
тренд	филтрирање			
сезоналност				

Табела 1: Категоризација на клучни поими

### 1.2.2. Концептуален дизајн

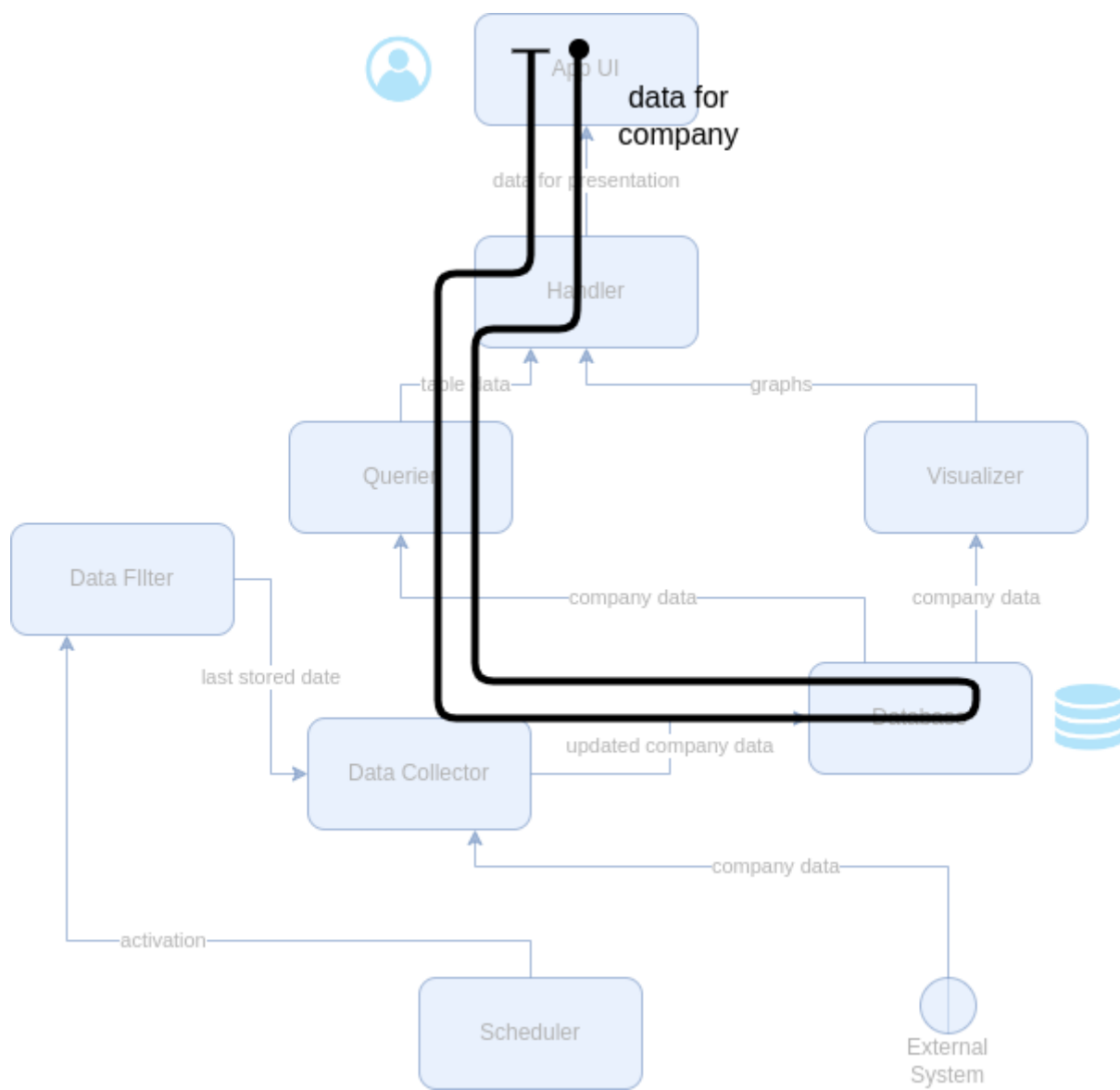


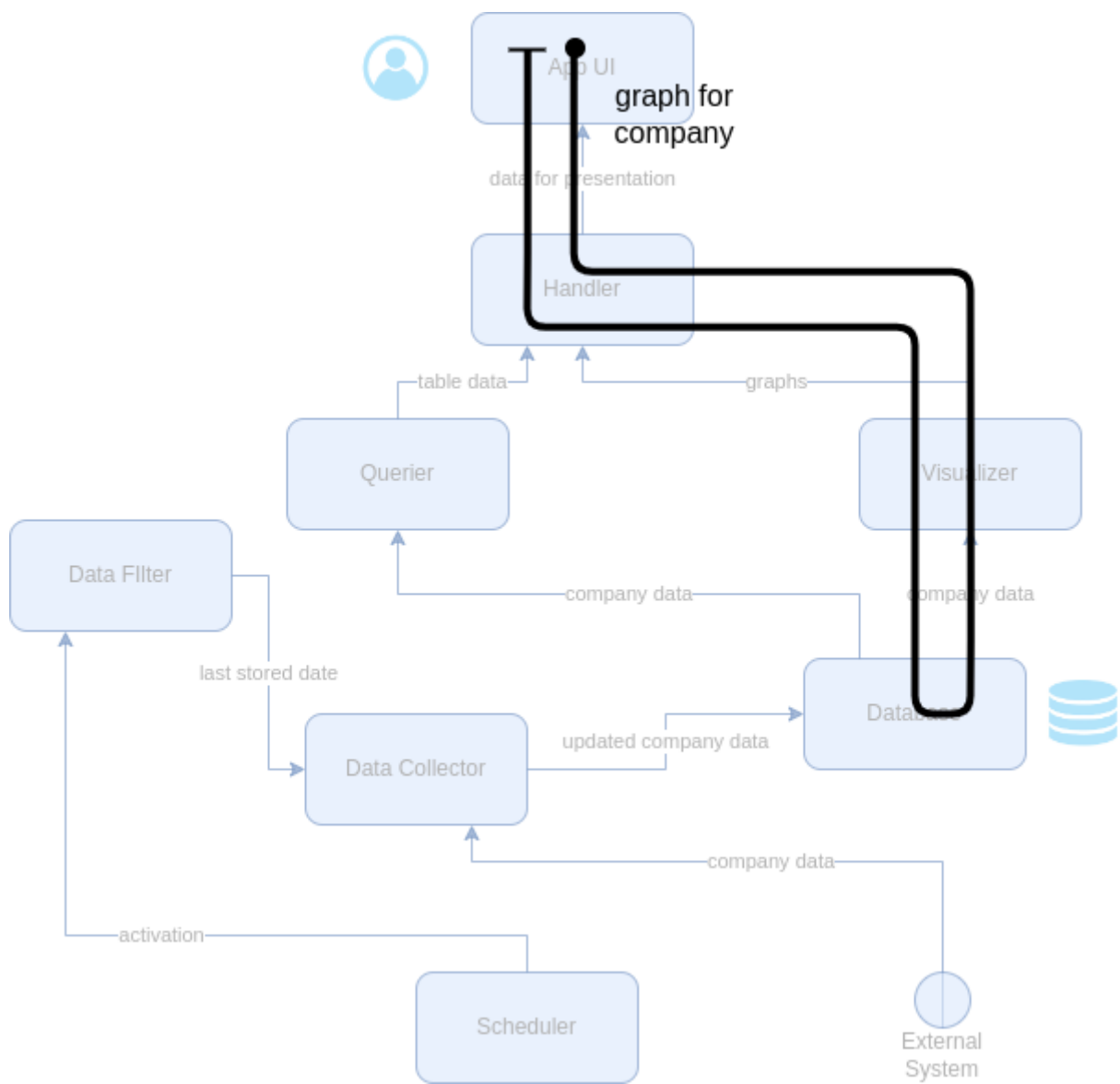
### 1.2.3. Одговорности на компонентите

- App UI
  - ➔ Презентациски слој
  - ➔ Приказ на движечка лента
  - ➔ Приказ на податоци и визуелизации на корисникот
  - ➔ Креирање на барања до веб апликацијата
- Handler
  - ➔ Обработка на барања од корисникот
- Querier
  - ➔ Генерирање табеларен приказ
  - ➔ Филтрирање
  - ➔ Испрашување на базата на податоци
- Visualizer
  - ➔ Креирање графици
  - ➔ Креирање временски серии
  - ➔ Испрашување на базата на податоци
- Database
  - ➔ Складирање на финансиски податоци за компании
- Data Collector
  - ➔ Преземање финансиски податоци од надворешен систем
  - ➔ Пополнување на базата на податоци со ажурирани податоци
- Data Filter
  - ➔ Проверка на последниот зачуван датум
- Scheduler
  - ➔ Автоматско активирање на преземање податоци

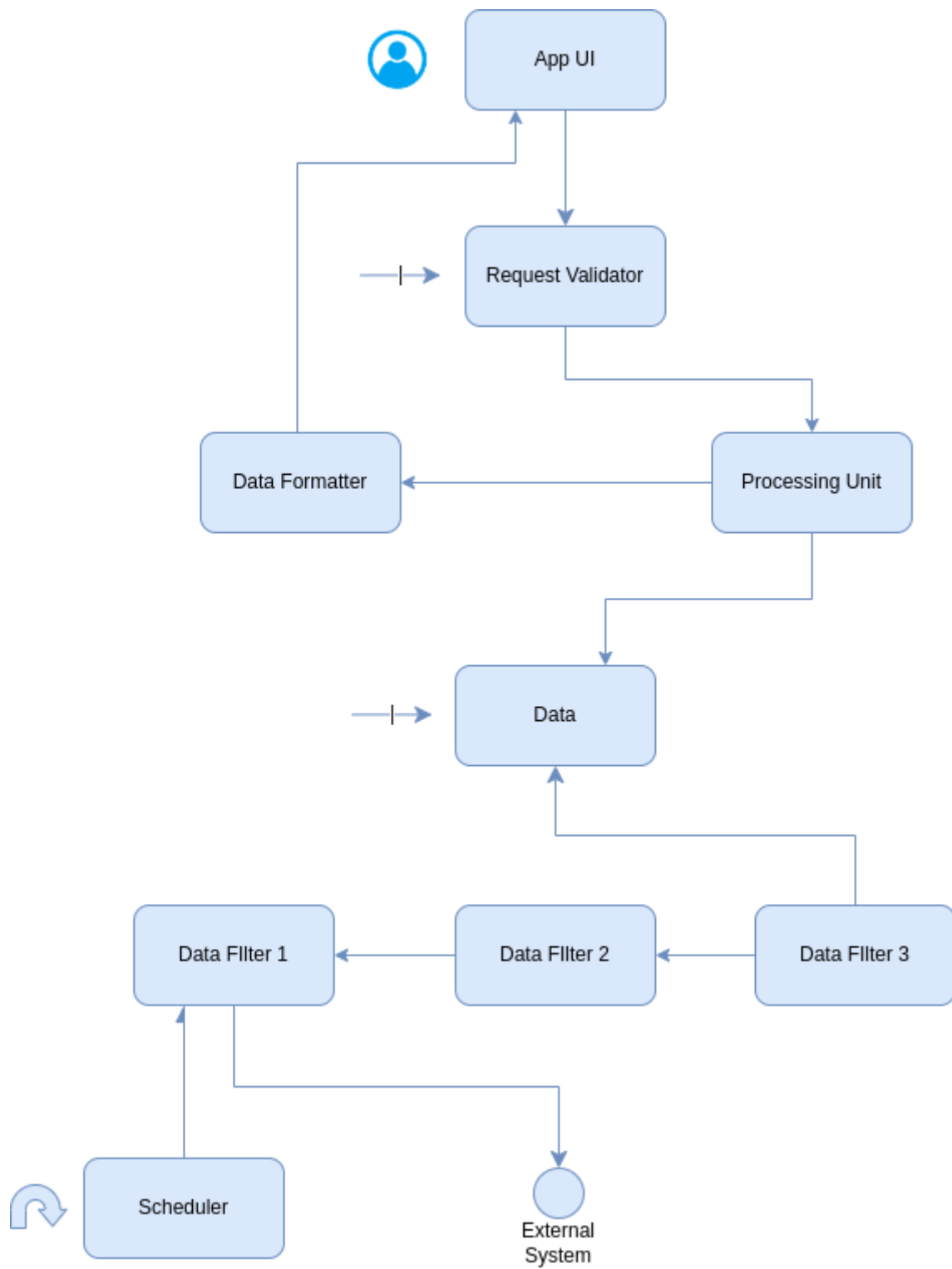
#### 1.2.4. Динамички аспект на концептуалниот дизајн

Однесувањето на системот е доловено со помош на две use-case мапи кои ги опфаќаат главните функционалности на системот, табеларен приказ и визуелизации.





### 1.3. Извршна архитектура

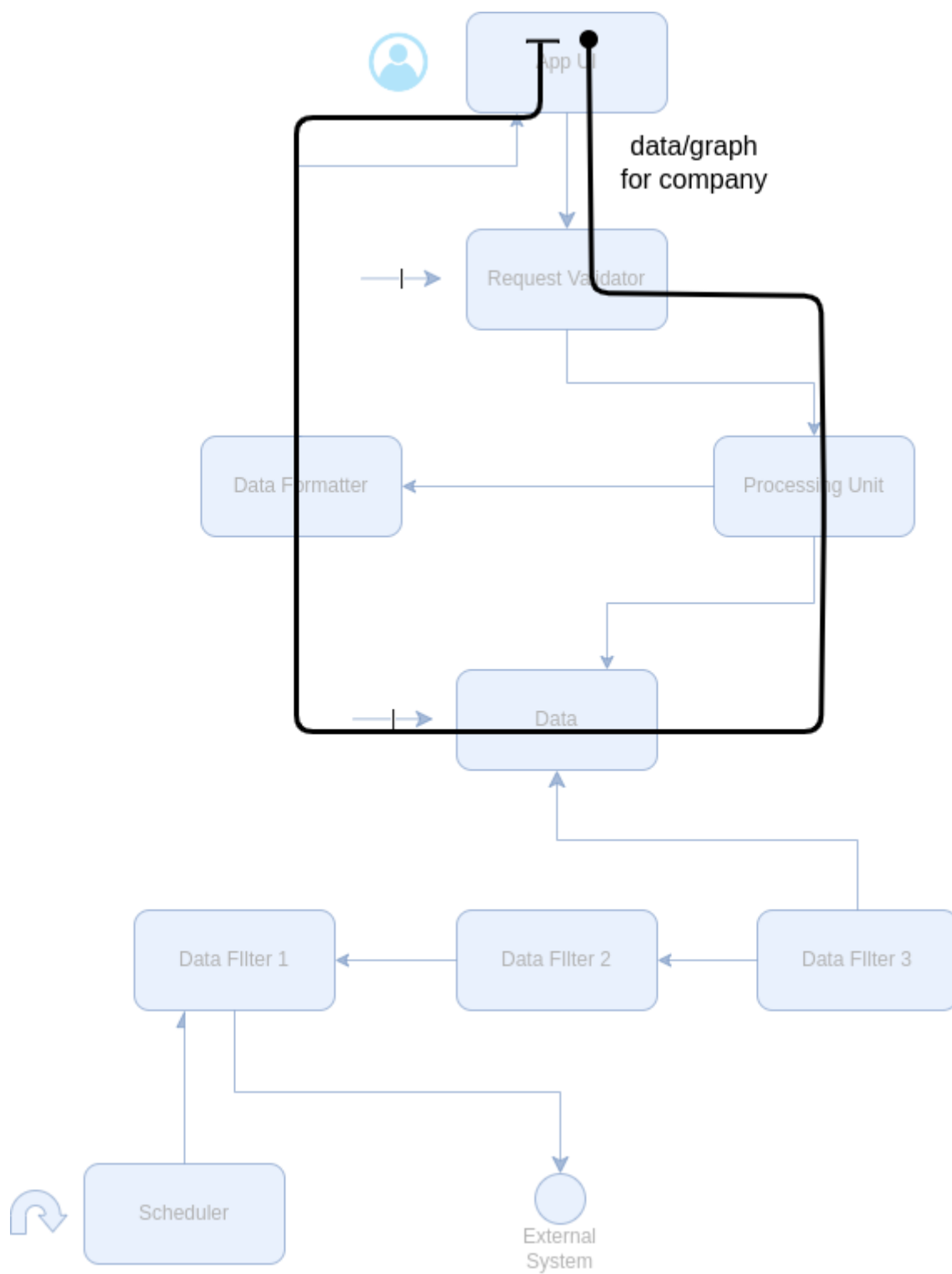


Опис на компонентите:

- App UI
  - ➔ Креирање на барања до веб апликацијата
- Request Validator
  - ➔ Обработка на барања за прикажување
- Processing unit
  - ➔ Генерирање на одговор за клиентот
  - ➔ Испрашување на базата на податоци
- Data Formatter
  - ➔ Пакување на одговор на клиентот
- Data
  - ➔ Складирање на финансиски податоци за компании
- Data Filter 1
  - ➔ Преземање листа на издавачи
- Data Filter 2
  - ➔ Проверка на последниот зачуван датум
- Data Filter 3
  - ➔ Пополнување на база со податоците кои недостасуваат
- Scheduler
  - ➔ Автоматско активирање на преземање податоци



### 1.3.1. Динамички аспект на извршната архитектура



## 1.4. Имплементациска архитектура

### 1.4.1. Метод на тежинско оценување

За изборот на frontend и backend технологии употребени се тежинските табели во прилог.

Критериум	Тежина	React.js	HTML	Django	Spring	.net
r2.1.1	10	9	5	10	8	8
r2.1.2	9	8	8	10	7	7
r2.1.3	8	9	6	10	8	8
r2.1.4	9	8	5	10	7	7
r2.1.5	9	8	7	10	7	7
r2.1.6	10	8	7	10	8	6
r2.1.7	7	6	6	9	7	7
r2.1.8	9	10	7	8	7	8
r2.1.9	8	9	7	9	7	7
r2.1.10	7	9	7	9	6	6
r2.1.11	9	9	8	9	8	9
r2.1.12	6	8	6	9	6	6
r2.1.13	6	9	7	8	9	9
r2.2.1	10	7	6	10	7	7
r2.2.2	8	7	7	9	8	7
r2.2.3	9	7	6	9	7	6
r2.2.4	7	8	8	8	8	8
r2.2.5	8	8	8	8	8	8
r2.2.6	9	8	6	9	9	8
r2.2.7	9	7	6	9	6	7
r2.2.8	7	7	7	9	7	6
r2.2.9	8	7	6	9	9	9
r2.2.10	8	8	7	9	9	9
r2.2.11	8	8	6	9	9	9
r2.2.12	9	10	10	9	9	10
<b>Тежински резултат:</b>		41814	34983	47196	39537	39123

Табела 2: Избор на frontend и backend

Согласно добиените резултати, за frontend технологија се избира React.js, а за backend технологија Django.

### 1.4.2. Имплементациски дизајн

Во прилог е прикажана имплементациската архитектура, каде се користи React на клиентска страна и Django на серверска страна. React комуницира со Django така што му ги препраќа барањата од корисникот, а Django потоа ги обработува и му враќа одговор назад. Од серверска страна се пристапува преку HTTP до екстерниот систем, односно страната на македонска берза од каде се повлекуваат финансиски информации за компаниите. Исто така, од серверска страна се пристапува до базата која ги складира финансиските податоци за компаниите преку ORM (Object-Relational Mapping). Тоа е техника на програмирање или шаблон за дизајн кој се користи во развојот на софтвер за да се поврзат релационите бази на податоци и објектно-ориентираното програмирање.

