

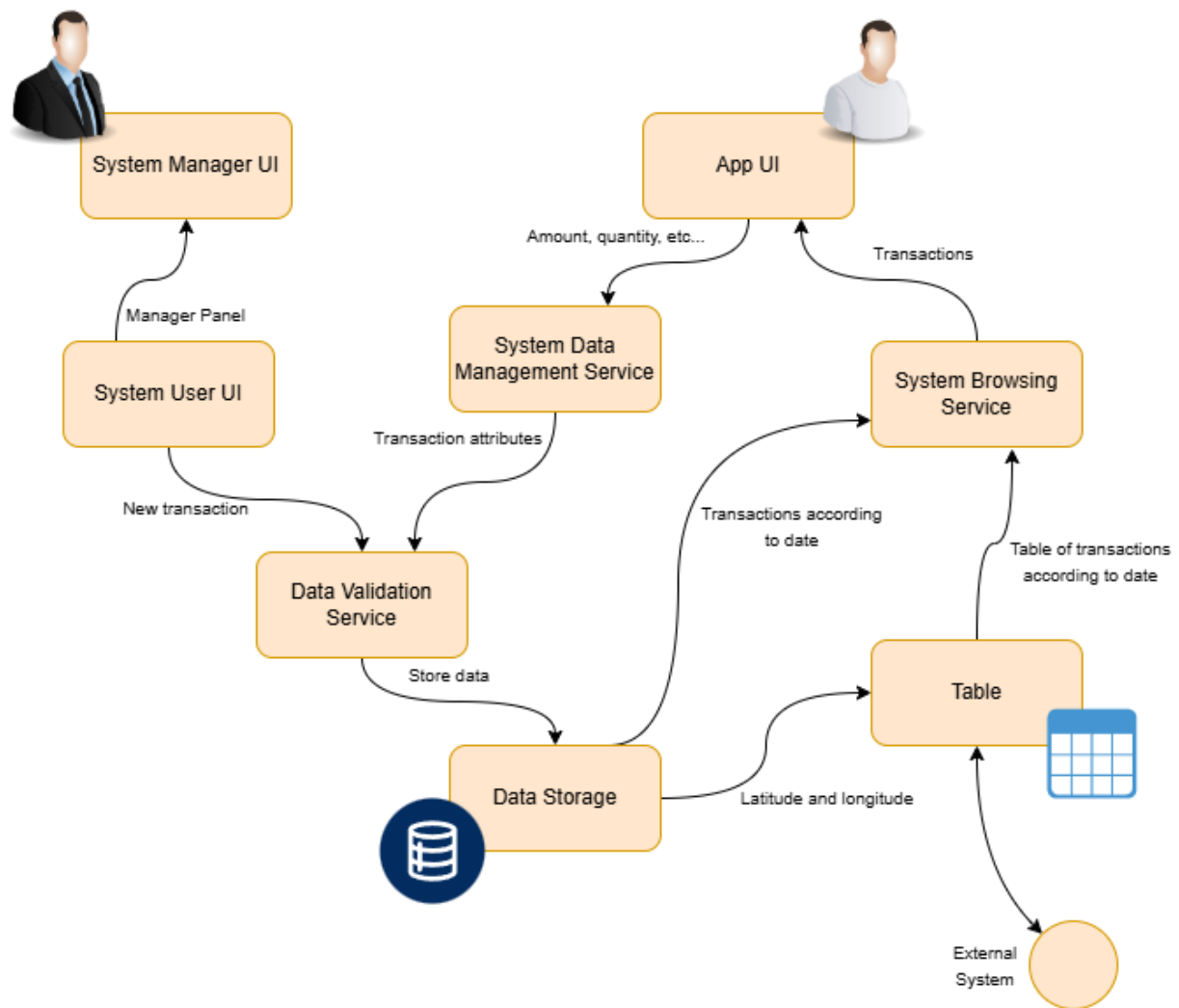
Архитектура

1. Концептуална архитектура

- Во првите чекори на концептуалната архитектура ги земаме функциските барања за системот каде се подвлечени клучните концепти.
- 1. Апликацијата мора да се поврзе на веб - страницата на Македонската берза и да добие листа на сите издавачи.
- 2. Мора да складира историски дневни податоци за акциите за секој издавач најмалку во последните 10 години.
- 3. Апликацијата мора да провери дали недостасуваат податоци во базата на податоци и да ги пополни до тековниот датум.
- 4. Податоците мора да се трансформираат и форматираат, вклучително и конзистентност на датумот и соодветно форматирање на броеви.
- 5. Апликацијата мора да ги складира обработените податоци на структуриран начин, ниво база на податоци или во структуриран формат на датотека (CSV, Excel, JSON).
- Клучните концепти ќе бидат категоризирани според следните категории: **Data, Function, Stakeholder, System, Abstract concept**.

Data	Function	Stakeholder	System	Abs. concept
Листа на издавачи	Добивање податоци	Администратори, корисници	Македонска берза	Преземање податоци
Историски податоци	Проверка на податоци	Корисници, систем	База на податоци	Конзистентност на податоци
Датуми на трансакции	Валидација	Систем	Систем	Форматирање на податоци
Финансиски податоци	Потполнување податоци	Систем	API за берзата	Ажурирање на податоци
Преработен и податоци	Транформирање и складирање	Корисници	CSV, JSON или база	Структурирано складирање

Табела 1. Категоризација на клучни концепти



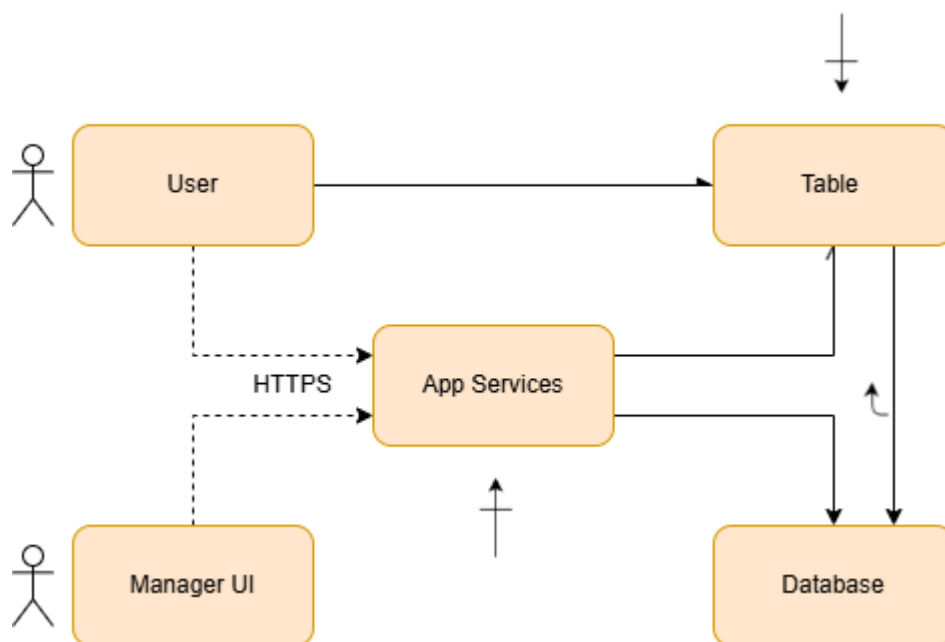
Слика 1. Концептуална архитектура

– Опис на компонентите:

1. **AppUI**: Главниот кориснички интерфејс кој им овозможува на корисниците директна комуникација со апликацијата, прикажувајќи ги сите неопходни функционалности и информации
2. **System User UI**: Крајниот интерфејс во форма на кориснички панел за внесување трансакции.
3. **System Manager UI**: Контролерот управува со профилот на администраторот и ги испраќа податоците за издавачот до сервисот за валидација.
4. **Table**: Табела каде ќе се прикажат изгенирираните податоци побарани од корисникот.
5. **Stock Browsing Service**: Сервис кој ги обработува филтрите и критериумите за пребарување внесени од корисникот и враќа податоци за акциите кои ги исполнуваат условите.

6. **Stock Data Management Service:** Секој внес на сума, количина и сл. треба да биде валидиран, дали е во правилна форма која одговара на базата на податоци. Овој сервис ги превзема податоците од корисникот и ги препраќа до сервисот за валидација во потребниот формат.
7. **Data Validation Service:** Сервис кој проверува дали пристигнатите податоци се во правилен формат за складирање во базата на податоци и, ако се валидни, ги запишува.
8. **Data Storage:** База на податоци која ги содржи следниве информации за трансакциите: датум, max, min, просечна цена, процент, количина, промет во БЕСТ во денари и вкупен промет.

2. Извршна архитектура

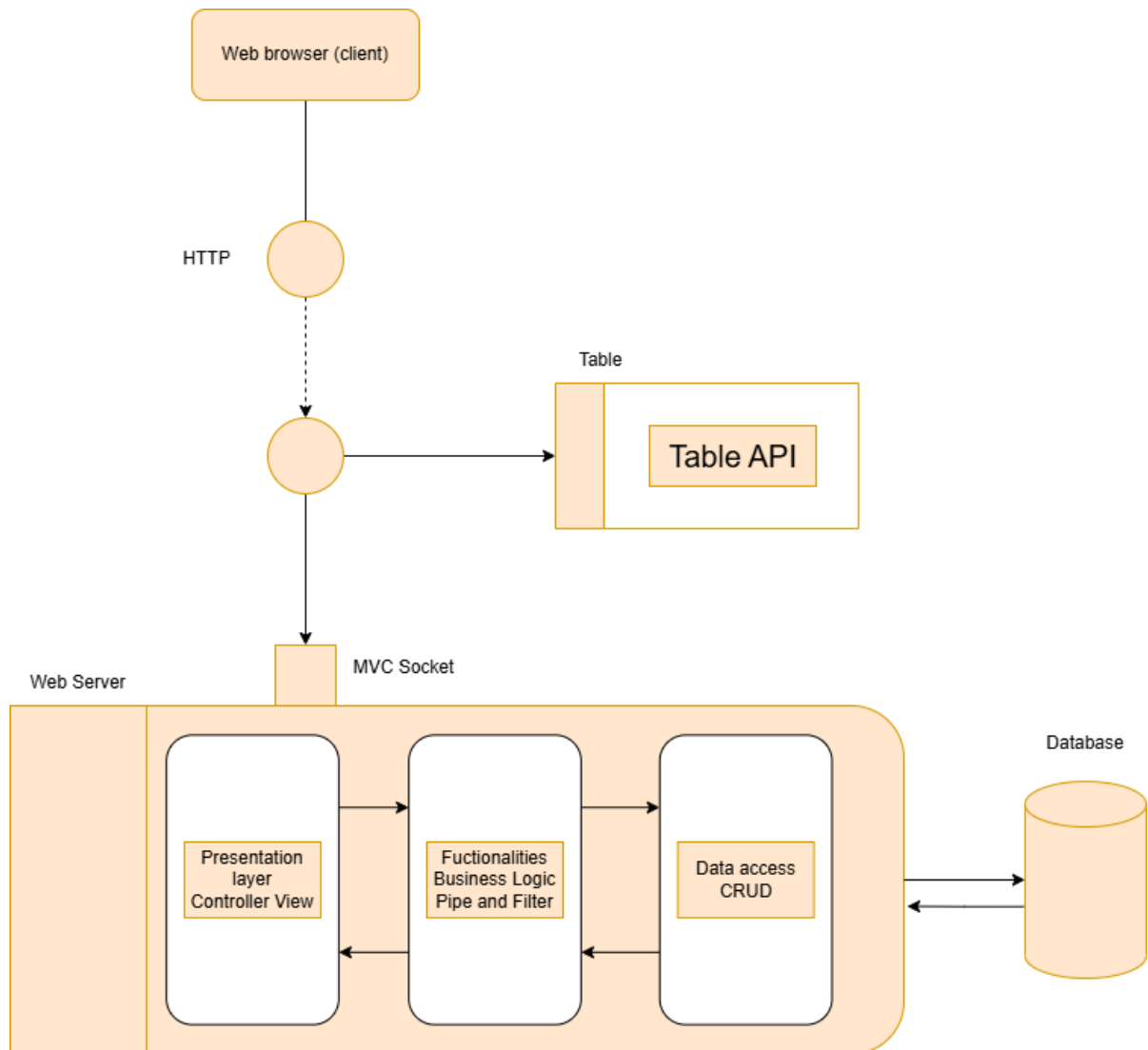


Слика 2. Извршна архитектура

- Опис на компонентите:
 1. **UserUI:** Графички интерфејс на обичниот корисник на системот.
 2. **ManagerUI:** Графички интерфејс на менаџерот на системот.
 3. **App Services:** *CRUD* податоци, валидација на податоци, обработка на добиени податоци, синхрона врска со databазата.
 4. **Table:** Приказ на трансакциите.

5. **Database:** Податоци за трансакциите, *CRUD* функционалности.

3. Имплементациска архитектура



Слика 3. Имплементациска архитектура

– Опис на компонентите:

1. **Web Browser:** Рендерира поглед (*view*) на корисникот. Праќа *HTTP* барање и чека одговор од *Web Server*.
2. **Web Server:** Користи *MVC* архитектонски шаблон со три различни слоеви:
 - i. **Презентациски слој (Frontend):** Задолжен за обработка на барањата од клиентите и испраќање на одговорите назад до

клиентите. Обезбедува визуелна претстава за корисникот преку употреба на технологии како *HTML*, *CSS* и *JavaScript*.

- ii. **Апликациски слој:** Содржи деловна логика која управува со барањата од презентацискиот слој. Обработува текстуални пребарувања, применува филтри и враќа соодветни резултати врз основа на поставените услови.
- iii. **Слој за пристап до податоци (Data Access):** Одговорен за директна комуникација со базата на податоци.