

## ОПИС НА АРХИТЕКТУРАТА НА ВЕБ АПЛИКАЦИЈАТА

### *Под-архитектурни компоненти:*

#### **1. Pipes & Filters**

- a. За потребите на апликацијата, податоците кои ни се од интерес (сите аптеки и здравствени установи на територијата на градот Скопје) беа превземени од 'Google Maps'. Со цел да ги отстраниме сите непотребни информации, како и табеларно да ги структурираме истите за нивно понатамошно складирање во релациона база на податоци, го применивме концептот на 'Pipes & Filters' со употреба на 'bash' скрипти.

#### **2. Data-centered architecture**

- a. Еден од најпознатите примери на 'data-centered' архитектура е базата на податоци, која има централна улога во нашата веб апликација. Дополнително, станува збор за 'repository' под-архитектурниот стил.

#### **3. N-tier architecture**

- a. Веб апликацијата е пример имплементација за три-слојна веб архитектура. Корисничкиот интерфејс, бизнис логиката на апликацијата и складирањето на податоците постојат како три различни и јасно дефинирани логички целини. Иницијално, при градење на техничкиот прототип, овие три целини се соодветно имплементирани низ: Thymeleaf (за кориснички интерфејс), Java Spring Boot MVC (за бизнис логиката), PostgreSQL (за чување на податоците).

#### **4. Distributed architecture with microservices**

- a. За разлика од техничкиот прототип (кој во позадина го користи 'MVC' шаблонот), архитектурата на крајната верзија на апликацијата ќе биде дистрибуирана и имплементирана со помош на концептот на микросервиси. Нашата веб апликација ќе биде составена од вкупно два микросервиси: за работа со аптеки и здравствени установи и за менаџирање на форумот на апликацијата, т.е. за работа со коментари и здравствени заболувања. Делот од апликацијата кој се однесува на навигацијата и рутирањето на мапа, ќе биде имплементиран со помош на скрипти и 'Google API & Services'.

#### **5. Containerization**

- a. За имплементација на микросервисната архитектура, се јавува дополнителна потреба од користење на т.н. контејнери, кои ја овозможуваат потребната околина за исправна имплементација на микросервисите. За секој микросервис постои одделен контејнер, додека пак за секој контејнер соодветно постои негов "docker image". Таканаречената докер слика е непроменлива датотека што содржи изворен код, библиотеки, зависности, алатки и други датотеки потребни за извршување на микросервисот.