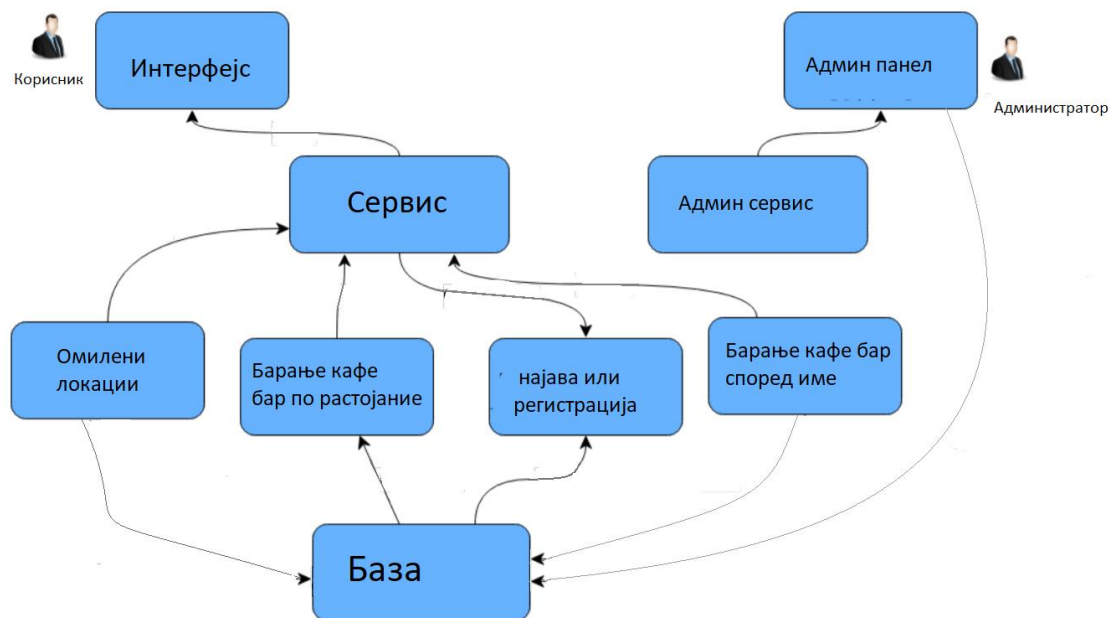


# Дефинирање на архитектурните погледи

## Архитектурни погледи:

### Концептуална архитектура

Концептот на оваа архитектура се фокусира на домесно ниво на одговорности, иницијален архитектурен дизајн и претставува одговор на барањата на стејкхолдерите. Грануларноста на компонентите во апликацијата е мала, бидејќи се вклучени повеќе компоненти кои извршуваат различни работи со висока кохезијата и цврсто врзување на компонентите.



Концептуален поглед 1.

Од коцептите во апликацијата застапени се пресметка на растојание- абстрактен концепт, места-податоци, дата база-податоци, пребарување функција, log-in/log-out функција, сортирање-функција, омилени локации-податоци.

Откако завршивме со анализата на софтверските барања и ги екстрахиравме функциските, нефункциските и контекстуалните барања, запонавме со концептуалната архитектура изградена од компоненти и конектори. Првично фокусот го насочивме на функциските барања, но бидејќи дизајнот е итеративен процес ги додадовме и нефункциските барања и контекстуалните.

Поконкретно според насоките, ги извлековме сите концепти поврзани со функционалноста од функциските барања и ги доделивме на соодветната категорија.

1. Системот мора да овозможи приказ на кафулиња во ранг од 0 до 1 км.Ф-ја, податоци
2. Системот мора да овозможи корисникот да внесепомало растојание од дозволениот ранг. Ф-ја, абстрактен концепт
3. Системот мора да ги нудисите локации до дестинацијата обележани на мапата. Абстрактен концепт, data
4. Системот треба да има поле за пребарување по име.Ф-ја
5. Функцијата за пребарување треба да ги добие соодветните записи од базата на податоци на кафулињата. Ф-ја, податоци
6. Системот треба да прикаже „Не се пронајдени резултати“ ако ниту еден објект не одговара на критериумот за пребарување. Ф-ја,
7. Системот треба да регистрира нови корисници. Ф-ја
8. Системот треба да биде способен да ја ажурира листата на објекти според рангот внесен од корисникот. Ф-ја
9. Сите резултати мора да бидат јасни и конзистентни. Податоци
10. Системот мора да биде поддржан од актуелните пребарувачи: Google Chrome, Firefox, Opera, Safari и Internet Explorer.
11. Системот треба да овозможи користење на податоците кои предходно се земени со скрипта од Openstreetmap. Ф-ја
12. Серверот треба да испрати порака за добредојде, кога корисникот на апликацијата ќе креира сметка. Ф-ја
13. Системот мора да прави резервна копија на податоците на период од 30 дена. Податоци
14. Само администраторот на системот мора да има пристап до податоците во базата.Ф-ја

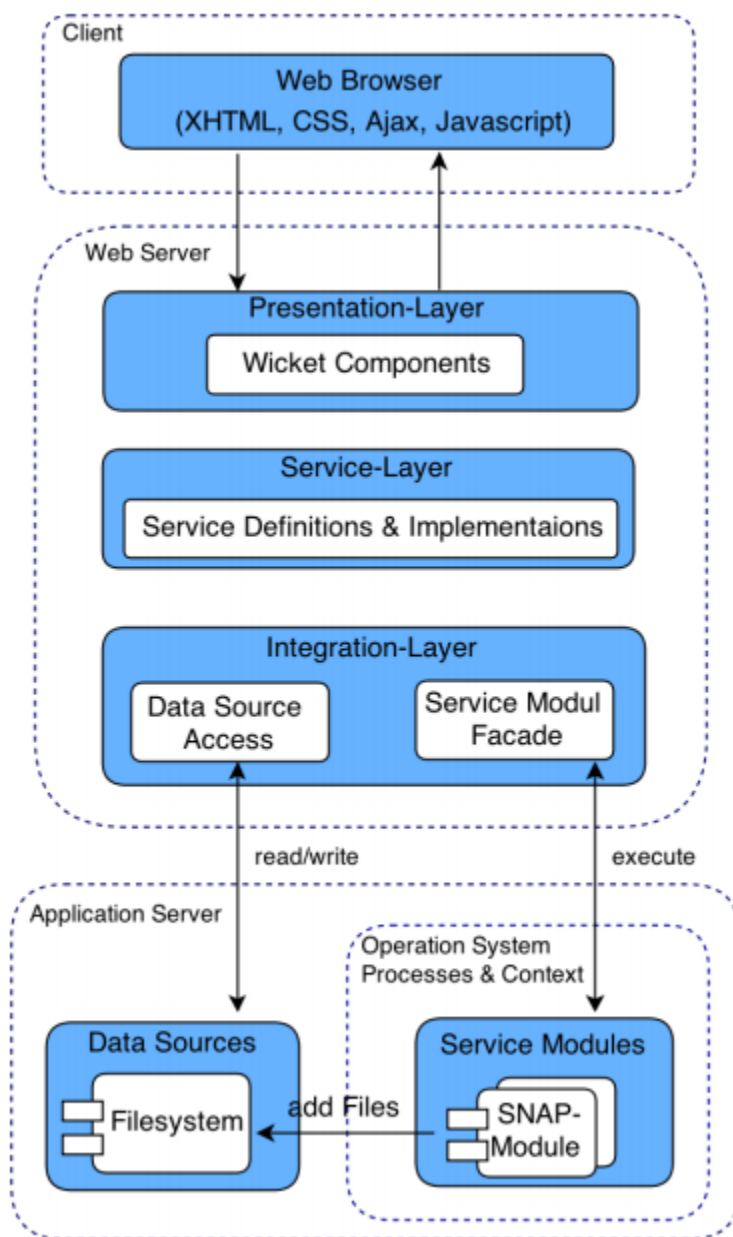
#### Поконкретно:

Идентификувавме актери во системот: клиентите кои имаат потреба од локација на кафуле, Создадовме use -cases, ги приоритизиравме функциските и нефункциските барања кои ги екстрахиравме од стејкохолдерите и го идентификувавме протокот на податоци.

Корисникот придонесува со влез за кафуле, а системот како излез го дава кафулето за кое пребарувал корисникот или листа на сите кафулиња во радиус од 1км од Плоштад Македонија.

## Имплементациска архитектура

Се фокусира на тоа како е изграден системот. Кои технолошки елементи се потребни за спроведување на системот. Софтверски пакети, библиотеки, рамки, класи... Се осврнува на барањата што не се извршуваат и на атрибутите за квалитет: конфигурабилност, проверливост, повторна употреба... Составен од компоненти и приклучоци (конектори). Компонентите и приклучоците ги рефлектираат софтверските ентитети и нивните односи на ниво на изворен и бинарен код. Типично голем број модели на имплементација. Секој модел се фокусира на еден од истовремените подсистеми или процеси од погледот на извршување.



Имплементациски поглед 1.[1]

Ги констатиравме двата типа на компоненти за имплементација на нашата веб апликација: апликациски и инфраструктурни компоненти. Апликациските компоненти се одговорни за имплементирање на одговорностите на ниво на домен. Во нашата веб апликација HTTP управувачот со конекции претставува типична апликациска компонента.

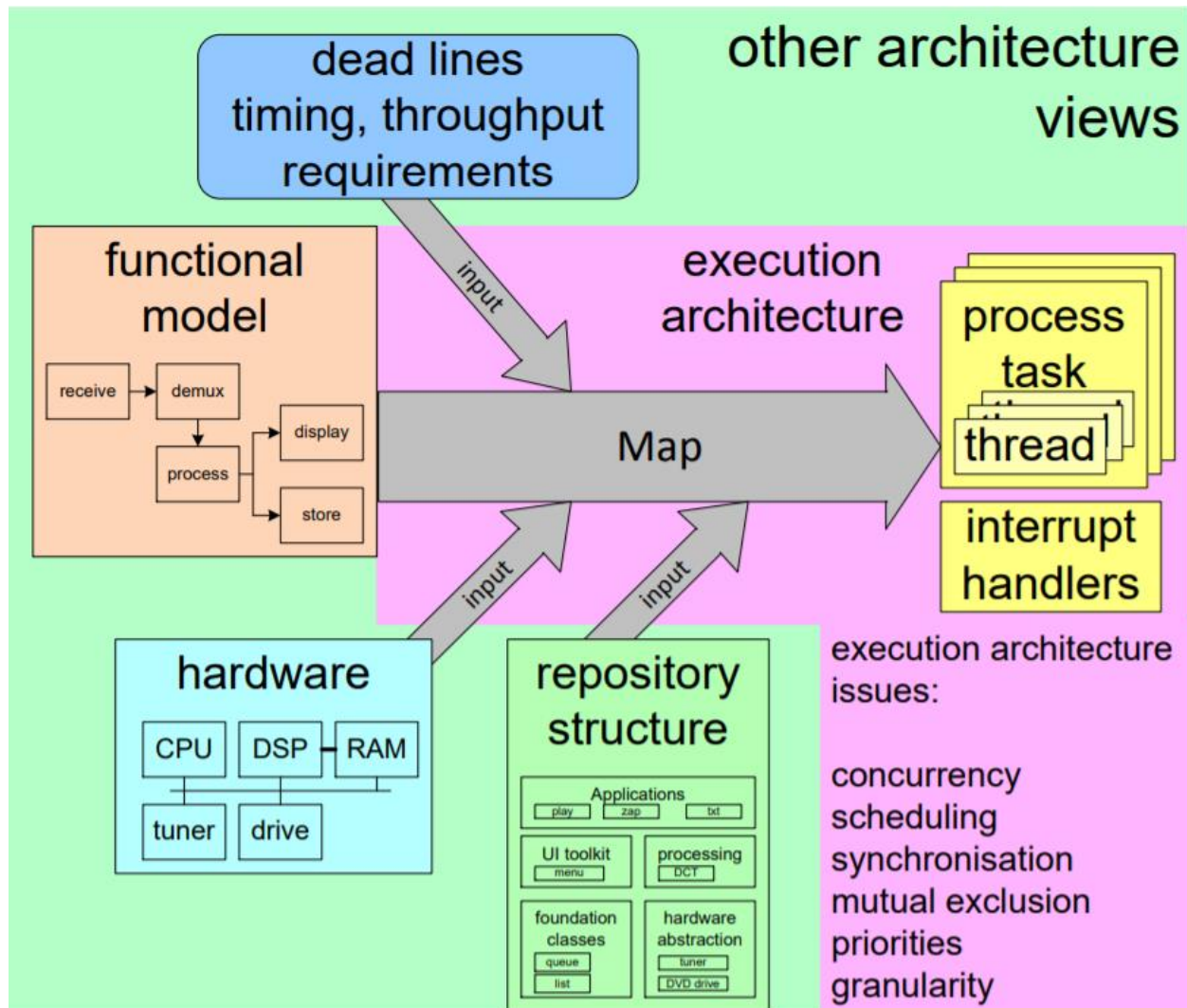
Инфраструктурни компоненти се потребни за да се активира системот, но не се поврзани со функционалност на апликацијата. Овде базата со податоци за кафеулиња е типична инфраструктурна компонента. Исто така и првичниот документ како излез од архитектурата за проток на податоци е инфраструктурен претставник.

Клиентот до апликацијата пристапува преку пребарувач - генерична инфраструктурна компонента. (кога ќе одредеме кој фремјворк ќе го користеме за веб апликацијата и тоа ќе го пишеме тука). Апликативното решение ќе биде хостирано во “облак” што е исто така претставник на инфраструктурната архитектура (контејнер) за апликативните компоненти. На сите инфраструктури им зададовме соодветна важност и удел во крајното решение и имплементациониот алгоритамот сами го дизајниравме. Слакд 32??

Бидејќи имплементационата архитектура ни нуди поглед на build-time структурата на системот овде ги разработивме атрибутите во нереално време. Бидејќи апликацијата ќе биде хостирана во “облак” не треба да се грижаме за одржувањето на податоците и апликацијата. Другиот бенефит е што немавме потреба од инвестиции во хардвер туку имаме облак архитектурата нуди неограничена големина на хардвер, и доверливоста овозможувајќи секогаш да бидеме известени за промените што се случуваат.

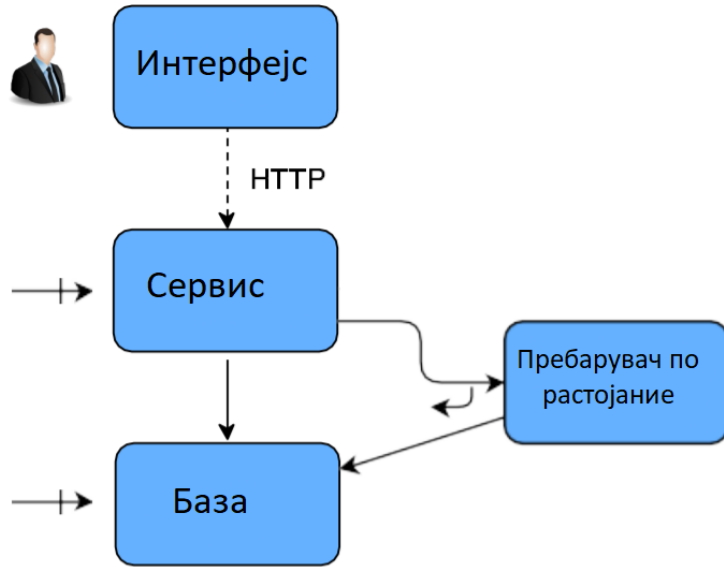
## Извршна архитектура

Поле на интерес на овој дел од архитектурата е системот во извршувачко време, хардверските елементи, подсистемите, процеси и нишки.

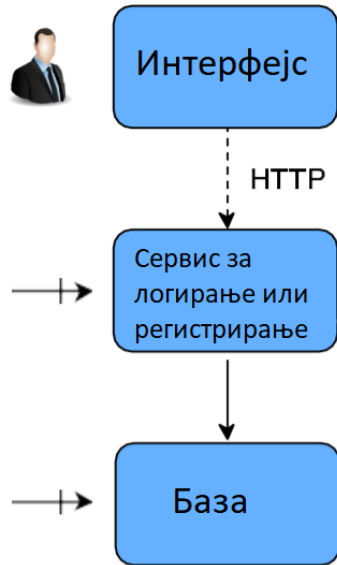


Сликовит приказ на извршна архитектура[2]

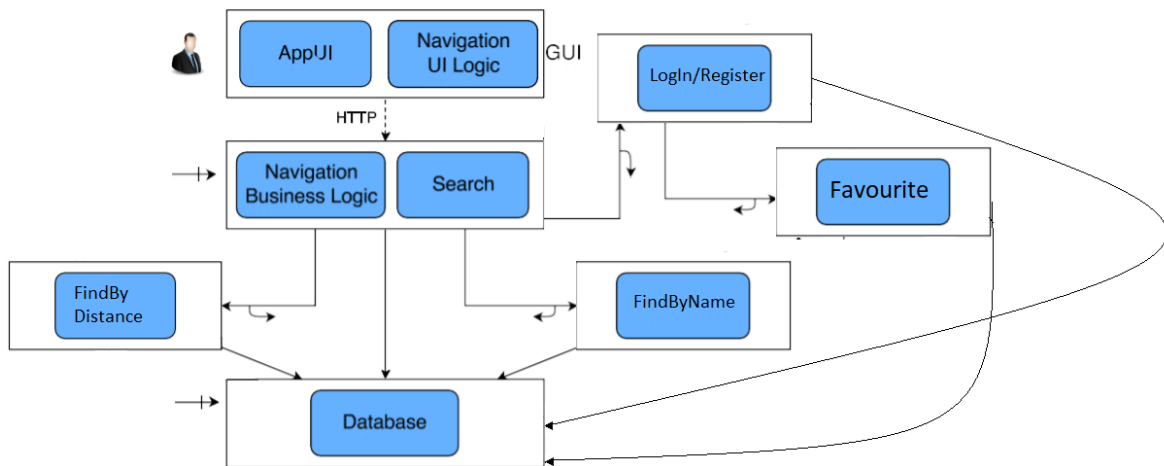
Апликацијата користи синхрона комуникација, што значи дека повикувачката компонента чека одговор од повиканата компонента пред да продолжи со понатамошно извршување. Конкурентни подсистеми се повеќе процеси кои се третираат како еден како на пример индексирањето во апликацијата. Конкурентни системи се и Веб-прелистувачот и Веб-серверот кои ја делат логиката за апликацијата.



Извршен поглед 1.  
Пребарување на кафе бар според растојание



Извршен поглед 2.  
Логирање/регистрација на корисник



Извршен поглед 3.  
Функционирање на апликацијата

## **Користена литература:**

[1]-Предавање број 4.1 Имплментациска архитектура слајд 42

[2]-<https://www.gaudisite.nl/ExecutionArchitectureConceptsPaper.pdf>