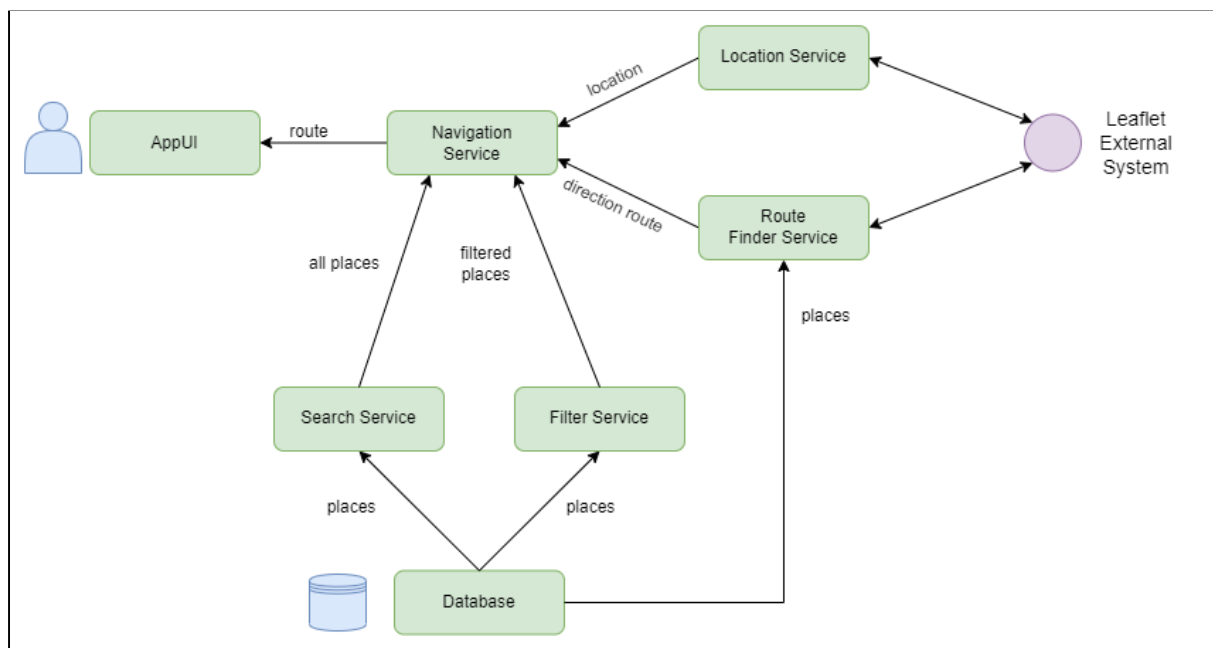


# Архитектурен дизајн

## Концепциски поглед

Концептуалната архитектура се залага за доменските одговорности, и се дизајнира при процесот на анализира на барања. Со концептуалната архитектура се објаснува кои се компонентите на системот и какви информации тие разменуваат помеѓу себе.



Слика 1. Концепциски поглед

Нашиот систем има за цел да му прикаже на корисникот каде се наоѓаат угостителските објекти во неговата област, како и дополнителни информации за нив. Системот започнува со работа при пребарување на одредена локација или избор на моментална локација на корисникот овозможено од компонентата Location Service. Дополнително корисникот може да избере одредени критериуми со кои ќе се прикажат само конкретни места или ќе подредат во листата за приказ на посебен начин. Овие функционалности системот ги постигнува со компонентите Search Service и Filter Service. На крај корисникот може да избере да му се прикаже навигација од местото на кое се наоѓа до избраното место од мапата или од прикажаната листа овозможено со компонентата Route Finder Service.

## Хибридна архитектура

Нашата апликација има хибридна архитектура која е составена од:

- Цевка и Филтер архитектура
- Словитата веб архитектура
- MVC архитектура
- Дистрибуирана архитектура со микросервиси

## Одговорности на компонентите:

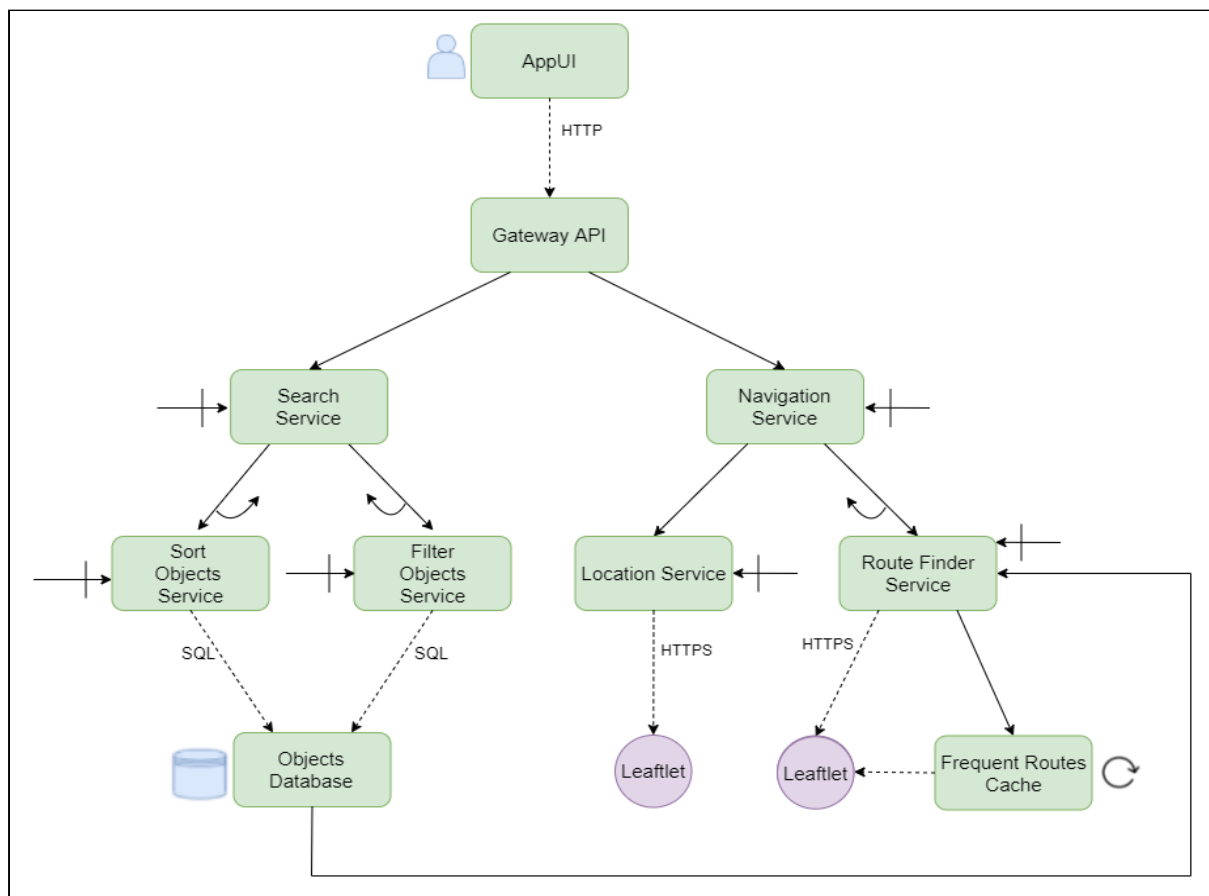
- AppUI
  - Прикажува мапа
  - Прикажува моментална локација
  - Прикажува сите најблиски угостителски објекти
  - Прикажува најблиски филтрирани угостителски објекти
  - Прикажува навигација од моментална локација до избрано место
  - Прикажува листа од сите угостителски објекти прикажани на мапата
  - Прикажува рецензија за избраниот угостителски објект
- Navigation Service
  - Земи локација за одредена адреса
  - Земи ја моменталната локација на корисникот
  - Земи ја навигацијата
  - Земи ги сите рецензии за одредено место
  - Земи листа од сите угостителски објекти во близина на моменталната позиција
  - Земи листа од филтрирани угостителски објекти во близина на моменталната позиција
- Location Service
  - Пресметај локација за одредена адреса
  - Пресметај моментална локација на корисникот
- Route Finder Service
  - Пресметај навигација помеѓу моментална и посакувана локација
- Filter Service
  - Пресметај и филтрирај ги сите угостителски објекти со одреден критериум во близина на моменталната позиција
- Search Service
  - Пресметај ги сите угостителски објекти сортирани по одреден критериум во близина на моменталната позиција

# Извршна архитектура

Извршната архитектура се фокусира на времето на извршување на системот.

Нејзините компоненти се конкурентните подсистеми, процеси, нитки и хардверски елементи со нивните граници. Извршната архитектура е само дел од целосната архитектура на системот, односно таа е потребна специфично за дистрибуираните и конкурентните подсистеми.

Пред да започнеме со дизајнот на архитектурата, јасно ги одредивме операциите кои ќе бидат временски најкритични. Со оглед на тоа што главната функционалност на нашиот систем ќе биде поврзана со мапирање и навигација на угостителски објекти, операциите за пребарување, филтрирање, сортирање и рутирање до објектите треба да бидат оптимални.



Слика 2. Извршен поглед

Со дијаграмот е претставена целосната извршна архитектура на системот. Апликацискиот кориснички интерфејс и веб серверот ќе бидат најважните конкурентни подсистеми, при што интерфејсот ќе биде поврзан со веб серверот преку HTTP конекција.

Веб серверот е директно поврзан со сервисите за навигација, пребарување, лоцирање, рутирање, филтрирање, и сортирање според параметрите пренесени преку барањето иницирано од корисникот.

Сервисите ќе бидат организирани како слоевата архитектура при што најпрво ќе се повикуваат сервисите за пребарување и навигација, кои пак според потребите ќе ги повикуваат и комуницираат со останатите сервиси.

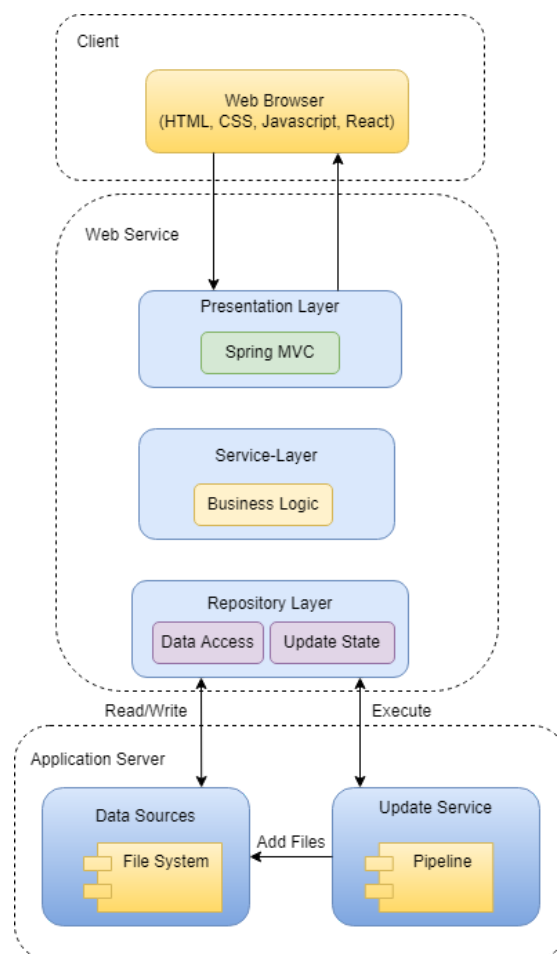
Дополнително ќе имаме кеш во кој ќе се чуваат најчесто бараните рути, бидејќи се очекува најголемиот број од барањата за рутирање да бидат само до одреден дел од објектите.

## Имплементациска архитектура

Имплементациската архитектура се заснова на тоа како е граден системот и кои технолошки елементи се потребни за истиот да се имплементира. Софтверски пакети, библиотеки, рамки, класи се само дел од технолошките елементи кои се користат за имплементација на системот.

Клучни елементи на имплементациската архитектура се компонентите и конекторите кои рефлектираат софтверски ентитети и нивните врски на ниво на изворен код. Често инфраструктурната компонента има улога како „контејнер“ за апликациските компоненти. За имплементација на истоимената архитектура потребни ни се модели кои се фокусираат на конкретен подсистем или процес од извршниот поглед притоа запазувајќи ги сите претходно дефинирани функционални и нефункционални барања од документот со софтверски барања.

Во првата фаза од проектот користевме OpenStreetMapAPI и OSM Filter за да ги повлечеме податоците и да овозможиме нивна манипулација според однапред зададени критериуми.



*Слика 3. Имплементациски поглед*

Во фазите кои следуваат ќе го користиме програмскиот јазик Јава како основен јазик врз кој ќе ја градиме нашата апликација. Од инфраструктурни компоненти, како рамка за Веб апликацијата ќе ја користиме најпопуларната и најкористената базирана на Java код -- Spring Boot. При што архитектурата која ќе биде развивана со помош на Spring Boot ќе да биде генерално трослојна со примеси од други архитектури со цел да се запазат барањата на проектот. За презентациониот слој ќе користиме jQuery со Ajax повици до Rest API за преземање на потребните податоци. Во следните фази овие Rest контролери може да ги користиме во ReactJS.

Манипулацијата на податоците, нивното складирање и останати слични операции ќе бидат изведени со PostgreSQL (индексирање по име во базите на податоци). Како рамка за развој на front-end планираме да ја искористиме ReactJS или евентуално би работеле со чист JavaScript код.