

## ОПИС НА ДИЈАГРАМИТЕ

### 1. Концептуален модел

Според клучните концепции дефинирани во претходно конструираната SRS документација, можеме да добиеме 10 структурни компоненти во концептуалниот дизајн на архитектурата:

- GUI - почетна страница/кориснички интерфејс за сите корисници
- Админ Панел - преглед само за администратори на системот
- Админ Сервис - сервис специфичен за системските администратори
- Сервис за Локации - користен за управување со сите функционалности
- Пронаоѓач на Локации - користен за внесување на податоци од надворешен извор
- Пребарување - користено за пребарување на зачуваните податоци
- Управител на Корисници - користен за автентикација и авторизација на корисниците
- Пронаоѓач на Содржини - управување со дополнителни информации за институциите
- Менаџер на Геоинформации - користен за управување со локациите на мапата
- База на Податоци - користена за перзистентно зачувување на податоците

Податоци	Функција	Stakeholder	Систем	Апстрактен Концепт
Институции	Најава	Корисник	GUI	Институција
Корисник	Одјава	Администратор	Админ Панел	Мапа
Локација	Регистрирање		Админ Сервис	Коментар
Кориснички Креденцијали	Додавање		Сервис за Локации	Рејтинг
Работни часови	Уредување		Пронаоѓач на Локации	Најблиска институција
Адреса	Отстранување		Управител на Корисници	
Категорија	Зачувување		Пронаоѓач на Содржини	
Наслов	Приказ		Менаџер на Геоинформации	
Омилено место	Пребарување		База на Податоци	

## 2. Извршен модел

- Компонента за кориснички интерфејс на апликацијата (AppUI) - компонента која се активира со кориснички влез и извршува акција. Испраќа HTTP барање до Веб Серверот за да ги добие бараните податоци.
- Web Server компонента - сервисна компонента. Чека барања од други компоненти (AppUI и Пребарување) и генерира одговори за такви барања.
- Database компонента - сервисна компонента. Чека барања од други компоненти (Веб Сервер) и генерира одговори за такви барања.
- Компонента за пребарување по име или категорија - синхронска комуникација. Повикувачката компонента (Компонента за пребарување по име или категорија) чека одговор од повиканата компонента (Веб Сервер).
- Генерирање на мапа со локации од базата на податоци - користење на повик за поврат. Повикувачката компонента (Веб Сервер) добива одговор асинхроно со поставување средство преку кое повиканата компонента (компонента за генерирање на мапа) може да испрати одговор подоцна.
- Прикажување на детали за инфраструктурата компонента - синхронска комуникација. Повикувачката компонента (База на податоци) чека одговор од повиканата компонента (прикажување на детали за инфраструктурата компонента).

### Поврзување на извршниот и концептуалниот модел

Основниот цел на овој дијаграм е да се определи точната локација на концептуалните компоненти во извршниот модел. Најдобрата одлука во нашиот случај беше да се избере слабо поврзан дизајн на архитектурата кој ќе ни обезбеди сколабилност, конфигурирање, одржување и тестирање.

### Поведение при извршување

Основниот цел на овој дијаграм е да се верификува дека извршната архитектура поддржува желеното поведење и да се најдат грешки во архитектурата користејќи use-case мапи за моделирање на поведението.

Моделиравме поведението при извршување за use case "Пребарување на удобности".

### 3. Модел на изведба

Откако веќе разговаравме за предностите на слабо поврзаниот објектно-ориентиран дизајн и потенцијалните добивки што можеме да ги оствариме со користење на истото за дизајнирање на нашата апликација, одлучивме да користиме програмскиот јазик Java и неговиот Spring Framework. Тимот одлучи да користи Spring MVC бидејќи фрејмворкот е еден од најдобро-документираните за овој дизајн паттерн.

Моделот на изведба се опишува преку 3-слоен дизајн во кој Клиентскиот Ентитет, Веб Серверот и Серверот на База на Податоци се вклучени. Повеќе од тоа, Веб Серверот исто така комуницира со надворешниот ентитет, API-то на OpenStreetMap.

- **Клиентски Ентитет** е задолжен за изгледот на апликацијата користејќи HTML, CSS и JavaScript. Клиентите ќе можат да пристапат до апликацијата со било кој веб-прелистувач.
- **Веб Серверот** комуницира и со клиентскиот веб-прелистувач и со Серверот на База на Податоци. Тој е составен од 3 слоја: Претставителен Слој, Сервисен Слој и Интеграциски Слој:
  - Претставителен Слој користи Spring Framework за имплементација на интерактивната природа на апликацијата.
  - Сервисен Слој содржи Студент Апликациски Сервис кој го имплементира нашиот бизнис логик и комуницира со API-то на OpenStreetMap за добивање на функционалности на мапата.
  - Интеграциски Слој е одговорен за комуникацијата со базата на податоци за перзистентно чување на податоци.
- **Серверот на База на Податоци** е имплементација на релационата база на податоци PostgreSQL.