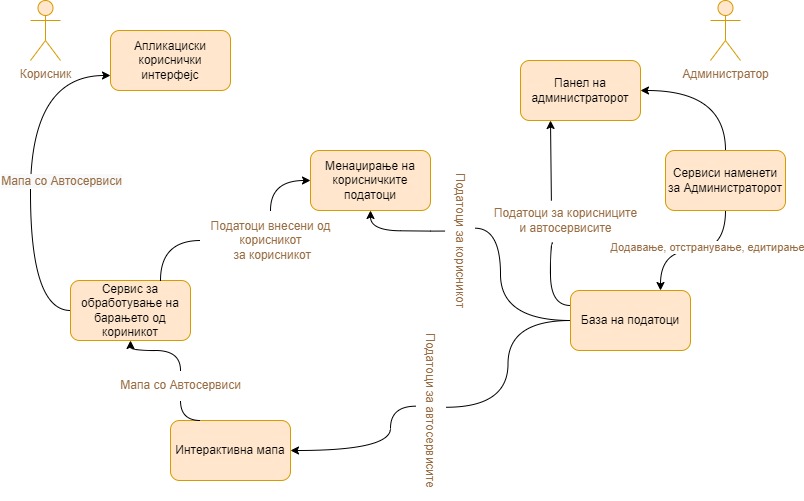
Архитектурен дизајн

1. Концептуална архитектура

Чекор 1: Класификација

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функции** | **Апстрактни концепти** | **Податоци** | **Засегнати страни** | **Системи** |
| Регистрација | Форма за регистрација | Податоци за корисник | Корисник | База на податоци |
| Најава | Форма за најава | Податоци за автосервиси  (локација, телефонски број) | Администратор |  |
| Одјава | Интерактивна мапа |  |  |  |
| Пребарување |  |  |  |  |
| Додавање и отстранување на автосервиси |  |  |  |  |
| Додавање и отстранување на корисници |  |  |  |  |

Чекор2: Дизајн



* Грануларност

**Апликациски кориснички интерфејс:**

Тригер копче за поднесување на податоците од регистрацијата,  тригер копче за најава, тригер копче одјава, поле за пребарување

**Сервис за обработување на барањата на корисникот:**

Регистрација, најава, одјава, прикажување на мапа

**Менаџирање на корисничките податоци:**

* Регистрирација на нови корисници

Дополнителни функционалности:  Пребарување на постоечките корисници според корисничко име. Не дозволувај регистрација доколку веќе постои корисничкото име, проверка на совпаѓањето на лозинката со лозинката за потврда, сите полиња во формата задолжително мора да се пополнат

* Најава на веќе регистрирани корисници ( Најавата е дозволена доколку  за корисничкото име се соодветствува со лозинката).

**Инратерактивна мапа:**

Мапа на која се прикажани автосервисите

**Панел на администратор:**

Пристап до изворниот код за навигациската апликација

**Сервис наменет за администратор:**

Додавање, бришење и едитирање на податоците за корисникот. Можност за додавање нови функционалности и ажурирање на постоечките.

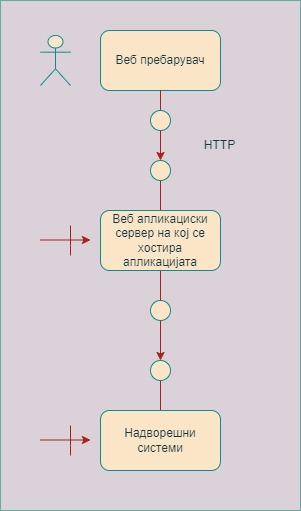
**База на податоци:**

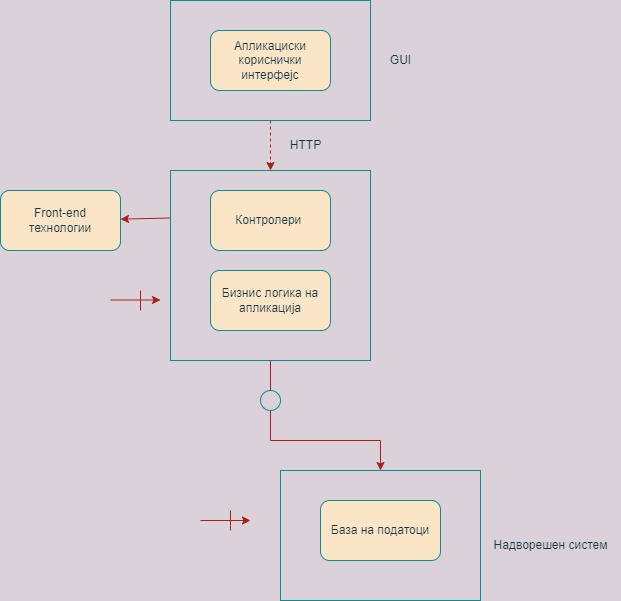
Креирање табели во кои ќе се чуваат податоците за корисниците. Во овие табели ќе се може да се додадат нови корисници или да се избришат.

* Кохезија

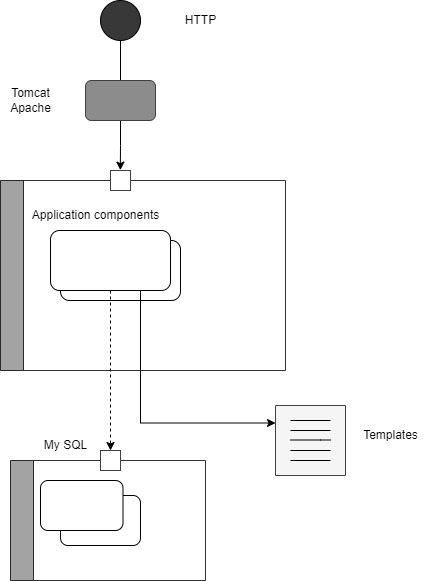
N-слојната архитектура овозможува висока кохезија меѓу компонентите. Одговорностите се поделени на секој слој соодветно на тој начин се овозможува и реискористливост на кодот како и измени во слоевите притоа да не се нарушуваат функционалностите на другите апликации.

1. Извршна архитектура

За апликацијата да работи потребно е корисникот преку пребарувачот да иницира барање, кое користејки го HTTP протоколот ќе пристигне до веб серверот на кој е хостирана апликацијата и чека барање за да го обработи. Доколку за опслужување на барањето е потребен некој надворешен систем (во нашиот случај база на податоци), серверот воспоставува комуникација со истиот, на тој начин доаѓа до потребните податоци за успешно да го реализира барањето од клиентот и да го врати како HTTP одговор.



1. Имплементациска архитектура



Инфраструктурни компоненти

* Веб пребарувач
* Веб апликациски сервер
* Apache Tomcat (веб контролер кој инстанцира requests, кој го носат брањето до контролерот го обработува и враќа одговор во инстанца response која исто така е инстанцирана од веб контенерот, инстанцира нишки за обработка на повеќе барања истовремено )
* MVC frameworks

Интерфејси

* Кориснички интерфејс HTML/HTTP

1. Дополнителни спецификации за архитектурата

* Слоевита архитектура -> N - слојна архитектура
* Data-flow architectures -> Цевки и филтри за прочистување на податоците кои треба да се постават во базата на податоци, филтер за дозволен пристап до главната страна
* Data-centered Architectures -> База на податоци и целата веб архитектура