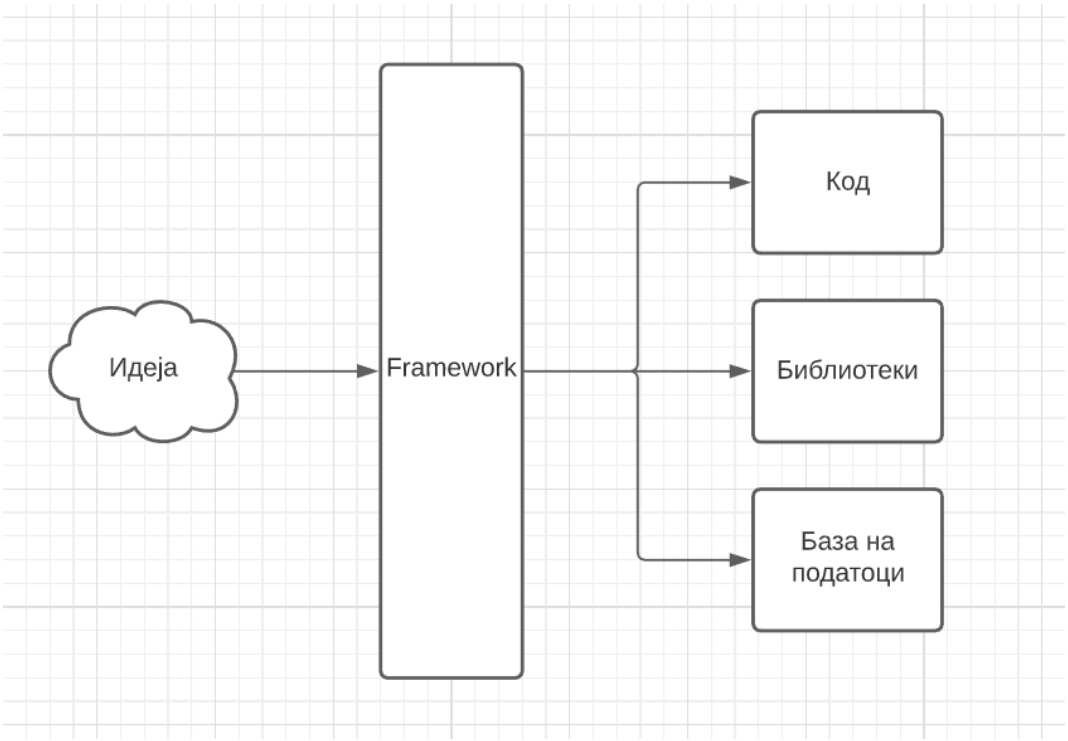
Имплементациска архитектура

Имплементациската архитектура е составена од компонентите и конекторите на системот, како различните елементи се поврзани меѓу себе, односно како е изграден системот.

Компонентите и конекторите овде ги дефинираат елементите на ниво на код.

Нашата цел е идејата којашто ја дефиниравме за овој проект да ја имплементираме во функционална веб апликација, односно преку framework да ги дефинираме класите, соодветните функции, библиотеки и пакети, како и базата којашто ги содржи податоците. Исто така, освен имплементација на идејата, целта ни е да се запазат претходно дефинираните функциски и нефункциски барања.



Слика 1:

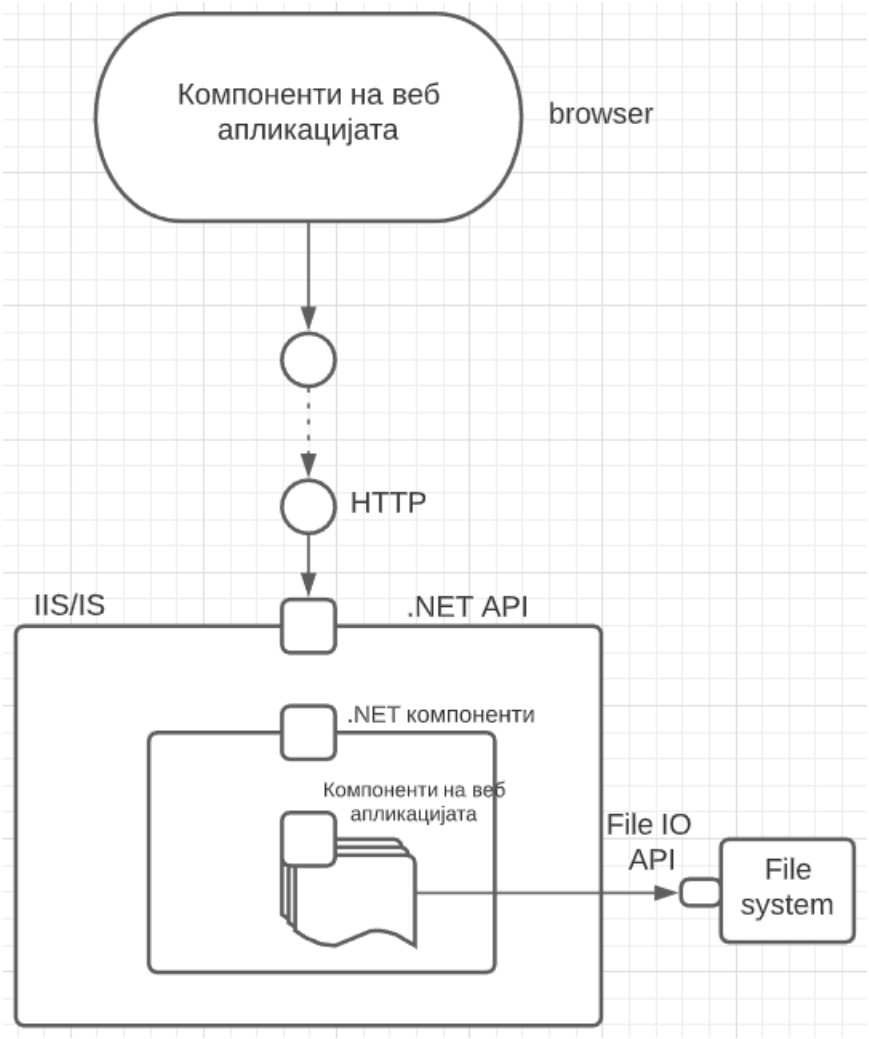
Нашата веб-апликација е заснована на клиент-сервер архитектурата и се користи HTTP протоколот. Клиентот ќе има пристап до апликацијата и ќе може да ја користи преку browser.

Двата типа на компоненти во имплементациската архитектура се апликациските и инфраструктурните компоненти.

Апликациските компоненти се одговорни за имплементирање на одговорностите на ниво на домен. Во нашата веб апликација source кодот претставува апликациска компонента.

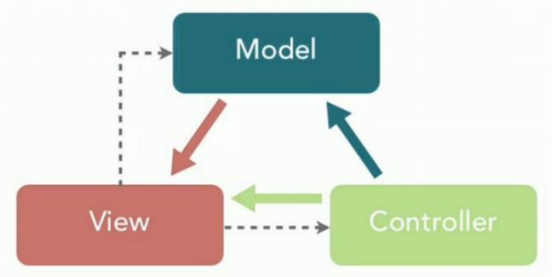
Нашата апликација има повеќе инфраструктурни компоненти. Бидејќи е веб апликација, следува дека HTTP Connection Handler-от е и browser-от се неопходни инфраструктурни компоненти. Преку browser-от се праќа барањето од корисникот и се рендерира одговорот.

Како инфраструктурна компонента ние го избравме ASP.NET framework-от, којшто е open-source и е наменет за изработка на динамички веб апликации.

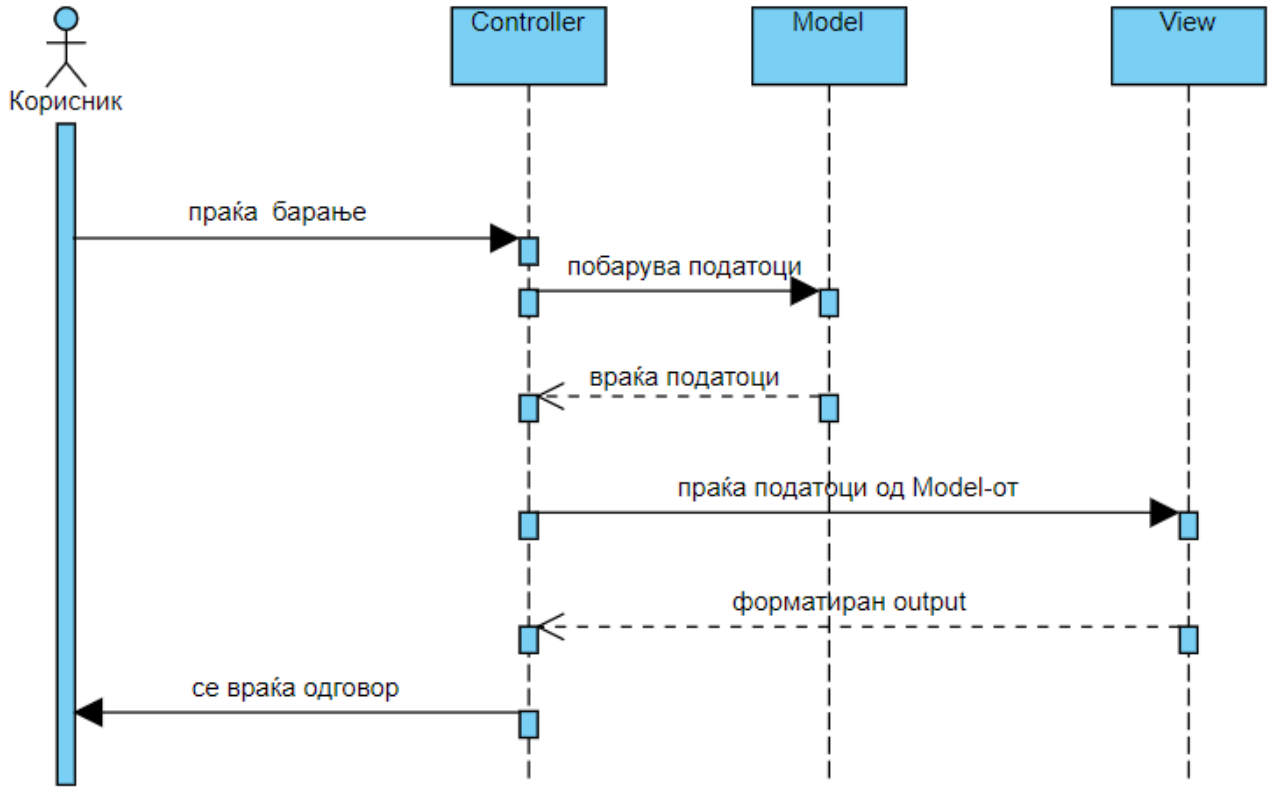


Слика 2: Имплементациска архитектура на веб апликацијата

Веб апликацијата се заснова на MVC шаблонот. MVC шаблонот е составен од три компоненти секоја со засебна улога. Компонентите се наречени Model, View и Controller.



Слика 3: MVC шаблон



Слика 4: Секвенцен дијаграм за имплементациската архитектура

Model - тука се дефинирани податоците и однесувањето на апликацијата, како и бизнис правилата за менување и управување со податоци(се однесува на доменот на проблемот). Моделот е независен од корисничкиот интерфејс

View – е всушност приказот на апликацијата, односно генериран изглед кој му се прикажува на корисникот. Не може да се посочи browser-от кон view и истото да се изрендерира. Controller-от треба да овозможи некои информации на view-то, па предава

податочен објект наречен модел. View-то го трансформира моделот во формат кој ќе се презентира на корисникот.

Controller – е задолжен за комуникација со корисникот и текот на апликацијата, односно се справува со барањата. Тие се одговорни за:

- Давање одговор на кориснички влез
- Правење промени во моделот како одговор на даден влез
- Тек на апликацијата
- Работат/обработуваат со податоците кои ќе ги добијат
- Праќаат податоци кон соодветен поглед (view)

Тек на апликацијата - Корисникот преку browser-от праќа барање, во зависност од барањето се избира Controller-от којшто ќе се справи со него. Барањето е насочено кон соодветниот Controller. Controller-от го процесира барањето и формира податочен Model. Model-от се пренесува до View-то. View-то го трансформира Model-от во соодветен излезен формат и одговорот се праќа до корисникот и browser-от го рендерира. Исто така во MVC шаблонот е присутен и таканаречениот Router кој го избира соодветниот Controller за да се справи со барањето од корисникот.

Уште една инфраструктурна компонента е базата на податоци, во нашата веб апликација csv фајловите коишто ги добивме со филтрирање претставуваат база на податоци, односно тука ни се сместени податоците потребни за апликацијата.