Дизајн и архитектура на софтвер на Винарии.мк

Нашата веб апликација е подржана со хибриден стил.Имплентирана е со шаблонот MVC,pipe and filter со чија помош извлековме и филтриравме податоци од xml до json,од базата на OpenStreetMap.Добиените податоци од ова филтрирање ги искористивме во нашата MongoDb база.

Во нашата веб апликација има микросервиска архитектура составена од:

-Презентациониот слој се занимава со корисничкиот интерфејс и како податоците се презентираат на корисниците.

-Сервисниот слој (делумно претставен од контролорот) управува со логиката на апликацијата и делува како посредник помеѓу Погледот и Моделот.

-Слојот на податоци (претставен од моделот) се фокусира на управување со структурите на податоци, валидација и интеракции со базата на податоци.

-Слој на база на податоци е вистинското физичко или виртуелно складирање каде што податоците од винаријата постојано се складираат и се преземаат со помош на MongoDB.

Концептуална архитектура

Целта на оваа архитектура е да ни ги прикаже одговорностите на доменско ниво,добиено од потребите и функциските барања на засегнатите страни.Оваа архитекура ни го прикажува и основниот архитектонски дизајн.Подоле се прикажани најважни концепти кои се среќаваат во нашата веб апликација.

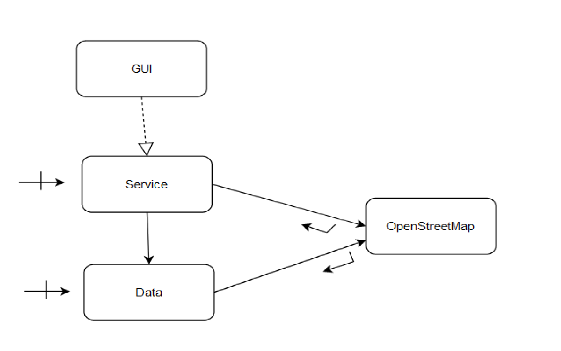
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Function** | **Stakeholder** | **System** | **Abs. concept** |
| винарии | Покажи листа | корисник | NodeJS | Локација |
| град | Прикажи на мапа |  | OpenStreetMap | Интерактивна мапа |
| локација | Прикажи име и град |  |  | Едноставност |
| оцена | Пребарај низ локации |  |  | Безбедна |
| Веб прелистувач | Остави оценка |  |  | Јавно |
|  | Достапна |  |  | Редовно одржување |
|  | Ажурирај |  |  |  |

Сликовито претставување на концептуалната архитектура



Извршна архитектура

Се однесува за самиот изглед на системот во однос на извржувачката фаза.Овдека се претсавени два погледа во однос на извршувањето на системот.





Имплементациска архитектура

Се заминава и фокусира за тоа како системот е изграден и сите техники што се потребни за негово функционирање.

Во нашиот проект за Винарии.мк, имаме различни views (home.pug, aboutUs.pug, mapa.pug, winery.pug) кои претставуваат различни делови од корисничкиот интерфејс.

Views:

home.pug, aboutUs.pug, mapa.pug, winery.pug: овие датотеки претставуваат различни страници или прикази на нашата апликација за винарија. Тие содржат код сличен на HTML (со користење на синтакса на Pug) кој ја дефинира структурата и содржината на секоја страница прикажана на корисниците.

Model:

vinariiModel: Ова го претставува слојот Model во нашиот проект, кој се справува со манипулација со податоци и интеракција со базата на податоци.

vinariiSchema: Ја дефинира структурата и шемата на податоците што ќе се складираат во базата на податоци MongoDB. Вклучува различни полиња како што се име, оценки, телефон, опис, слики, детали за локација итн.

Mongoose: Се користи за интеракција со MongoDB преку дефинирање на шемата, ракување со валидација на податоците и создавање на модел за колекцијата Vinarii.

Controller:

ViewController: Ова служи како контролер во нашата апликација. Содржи логика за справување со барањата на корисниците и интеракција со Моделот и Прегледите.

Рутер (viewRoutes): Овој рутер ги дефинира рутите (URL крајните точки) за различни views во нашата апликација. Користи методи како GET, POST, итн., за да се справи со дојдовните барања и да ги мапира на одредени функции на контролорот.

Поврзување и проток:

Во нашиот случај Web Browser-от кој ќе ја користи апликацијата ќе испраќа HTTP барање до серверот

Рутер (viewRoutes): Прима барања од клиентот/прелистувачот за различни URL-адреси (на пример, дома, за нас, винарија итн.).

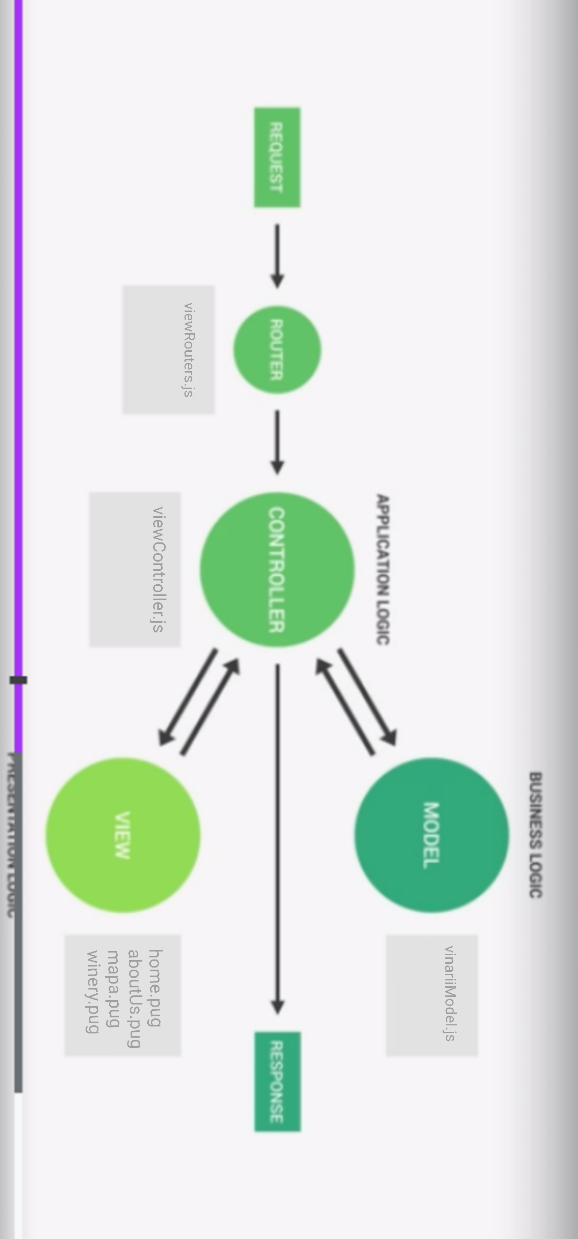
Контролер (viewController): Врз основа на добиените барања, се повикуваат функциите на контролорот што одговараат на овие правци. На пример, барањето за прегледување на почетната страница би активирало одредена функција во контролорот.

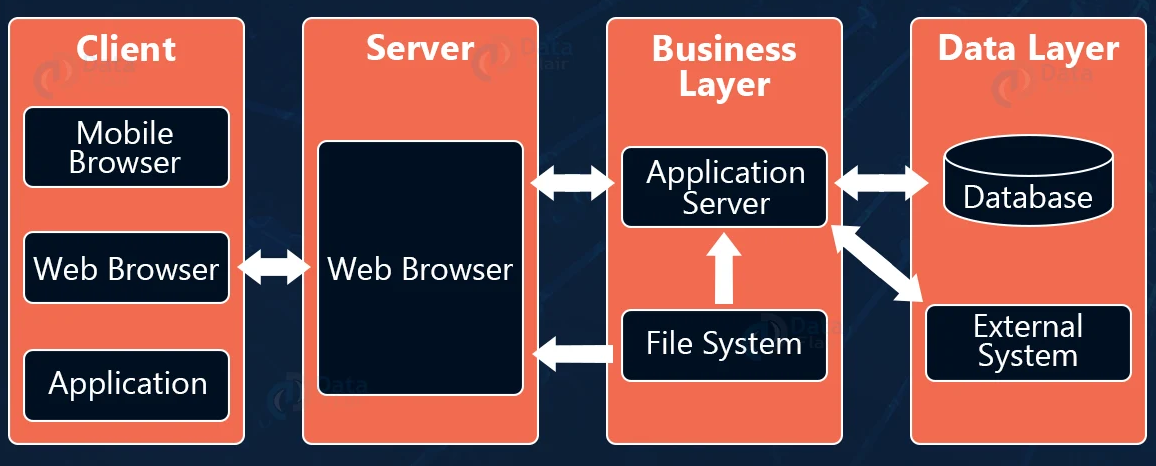
Интеракција контролер-модел: Внатре во функциите на контролорот, податоците може да се преземат или манипулираат со помош на vinariiModel (модел Mongoose). На пример, при рендерирање на страницата за винарија, контролорот може да преземе конкретни податоци за винаријата од базата на податоци користејќи го моделот Vinarii.

Прегледи за рендерирање:

Откако ќе се преземат потребните податоци, контролорот ги пренесува овие податоци до соодветниот приказ (датотека Pug) за рендерирање. Погледот потоа ги користи овие податоци за динамички да генерира HTML содржина, која се испраќа назад како одговор на прелистувачот на корисникот.

Накратко, кога корисникот пристапува до различни URL-адреси или страници во нашата апликација за винарија, рутерот го насочува барањето до соодветната функција на контролорот. Контролорот е во интеракција со моделот за да преземе или манипулира со податоци, а потоа ги пренесува овие податоци до соодветниот view за прикажување, што резултира со тоа што корисникот ја гледа саканата содржина врз основа на бараната страница.





Пример за секвенцијален дијаграм за како корисник пристапува до листата на винарии.Преку web browserot се испраќа HTTP request до веб серверот.Од серверот се праќа повик до бизнис делот на апликацијата за да искомуницира до сите останати делови и од базата да се превземе соодветната листа на винарии,која се препраќа до сите делови на архитектурата и до самиот корисник.

