

Домашна работа 2

Архитектура за WineMapper

Нашиот тим одлучи дека ќе ги следи постапките по кои се прикажани низ презентациите, односно ќе се води по редоследот прикажан во нив. Па затоа ќе изработиме архитектури од трите дадени погледи во деловите. Во првиот дел ќе ја опфатиме концептуалната архитектура ќе започнеме со барањата кои ги наведовме во домашна број еден прикажувајќи ги клучните концепти и правејќи категоризација по што ќе започнеме со архитектурата од екстерните системи и стејкхолдерите. Во вториот дел ќе направиме архитектурен модел во извршен поглед. На крајот во третиот дел ќе ја прикажеме нашата финална (имплементациска) архитектура со детали како и на кој начин планираме да ја реализираме дадената архитектура.

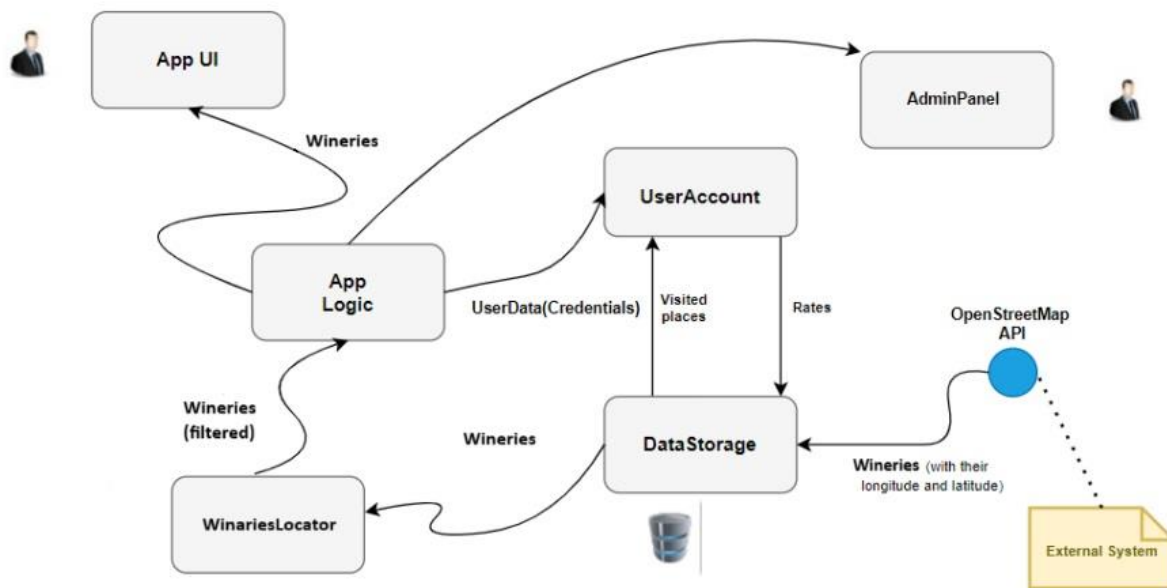
Прв Дел – Концептуална архитектура

1. Web апликацијата ќе овозможува најава преку емаил адреса на корисникот.
2. Web апликацијата ќе овозможува приказ на рута за пристигање до избраниот објект.
3. Web апликацијата ќе овозможува преглед на оценките кои се однесуваат за дадена винарија.
4. Web апликацијата ќе овозможува поставување на оценка на дадена винарија.
5. Web апликацијата ќе овозможува приказ на времето потребно да се стигне до дестинацијата.
6. Web апликацијата ќе престане да ја користи локацијата на корисникот во моментот кога ќе го исклучи пребарувачот.
7. Web апликацијата ќе нуди можност за пронаоѓање на места на територијата на Македонија.

System	Funtionality	Data	Stakeholder	Abstract Class
OpenStreetMap	најава	локација на винарија	корисник	оценка
	лоцирање	Територија на Македонија		време
	приказ на рута	Оставени оценки		дестинација
	оставање оценка			интерактивна мапа
	Следење на движењето			

Табела 1. Категоризација на клучни концепти

Со приказ на табелата можеме да забележиме дека имаме еден надворешен систем и еден stakeholder според барањата.



Слика 1. Концептуална архитектура

App UI:

- Презентациски слој
- HTTPS конекција со AppLogic
- Менаџирање на output-от
- Рендерирање на мапа.

AppLogic:

- Пресметување на координати
- Конекција со база
- Автентикација на корисниците

DataStorage:

- Податоци за името на винаријата заедно со нејзините координати.
- Податоци за најава
- Оценки за кафетеријата

UserAccount:

- Дел од AppLogic.

-Менаџирање со податоците на корисникот.

AdminPanel:

-UI за интерфејс на корисниците со привилегии.

OpenStreetMaps:

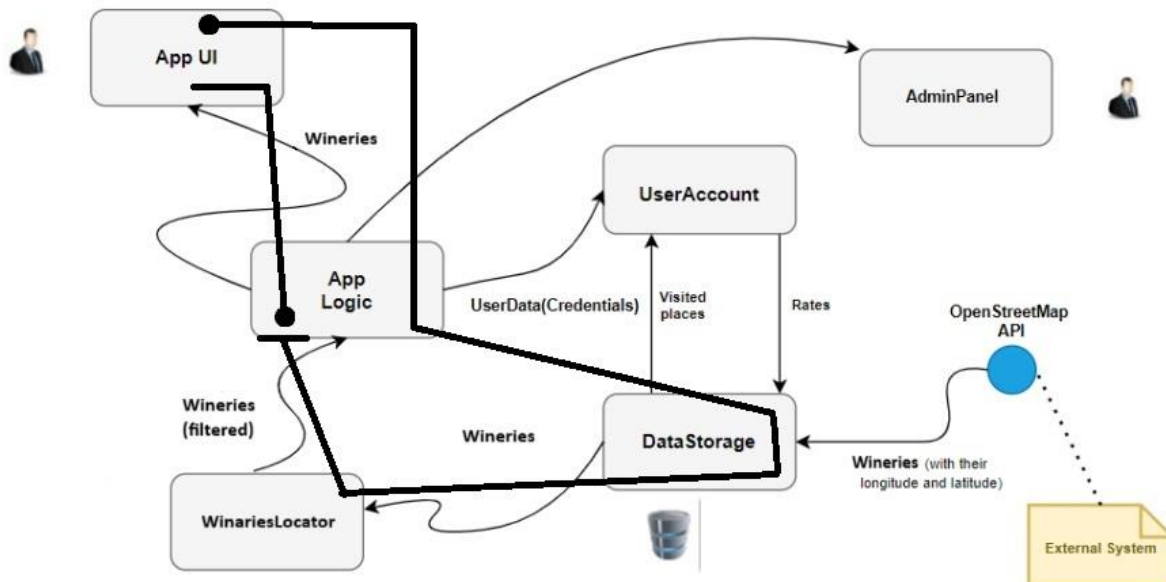
Надворешен систем од кој ги преземаваме податоците за објектите како и мапата на UI.

WineriesLocator:

-Трансформација на податоци ако е потребно

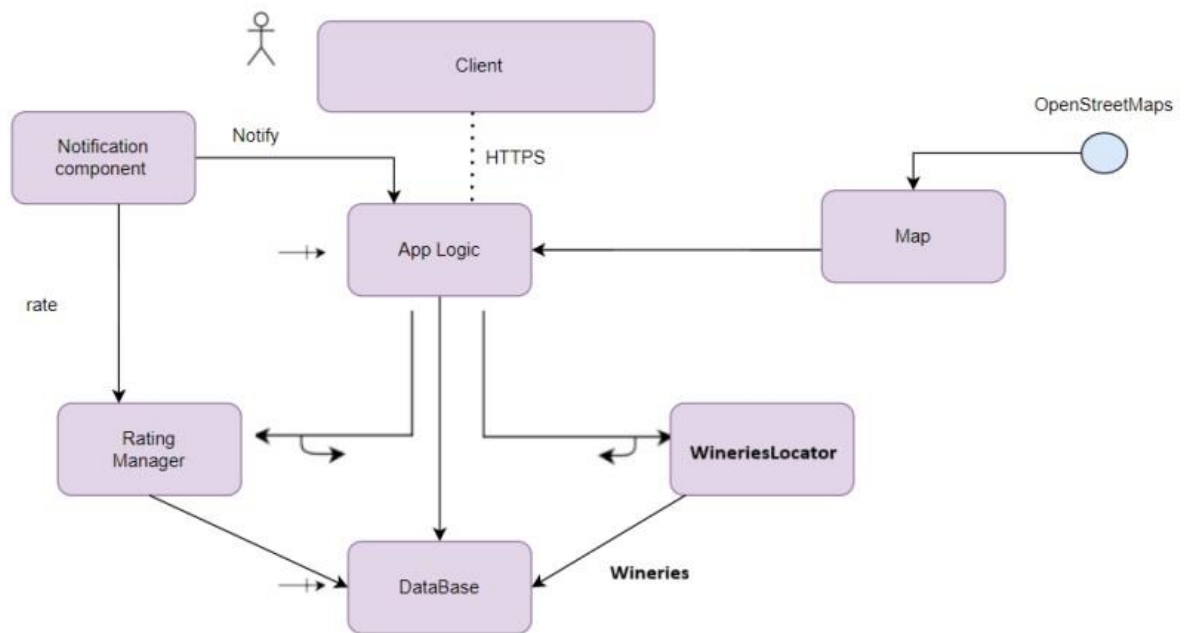
Со цел порано утврдување на грешките, сметаме дека соодветно е да користиме Use Case мапа, со која преку персона ќе ја прикажеме соодветната употреба. Во овој метод ќе измислиме сценарио во кое може да се најде нашиот измислен корисник. Нашиот корисник е ентузијаст за вино кој е лоциран во Скопје. Едно неделно утро тој сака заедно со неговото семејство да патува до некоја винарија која се наоѓа на територија на Македонија.

Persona: Корисник, 54 години, ентузијаст за вино



Слика 2. Use Case мапа на концептуалната архитектура

Втор Дел – Извршна Архитектура



Слика 3. Извршна архитектура на WineMapper

Client:

- Апликациски интерфејс кој се презентира на клиентот
- Client sender и handler за requests.

Notification component:

- Ќе го извести системот кога ќе има внес на нова оценка за објектот.

RatingManager:

- Оваа компонента е поврзана со AppLogic преку Callback врска.
- Внес на оценка
- CRUD функционалности.

AppLogic:

- Скоро сите функционалности
- Автентикација
- Трансформирање на податоци

- Синхрона врска со базата
- Функција како MVC контролер

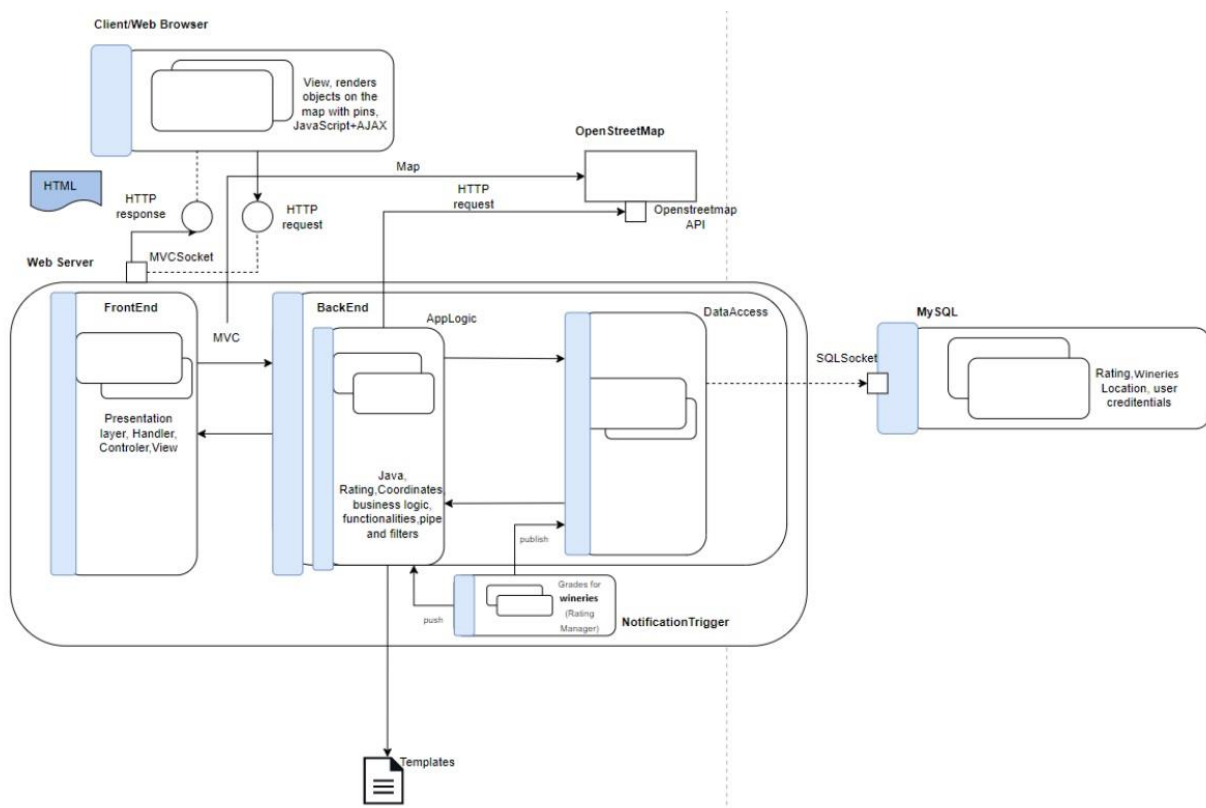
Map:

- Интерактивна мапа превземена од OpenStreetMap.
- Сервис на апликацијата.

DataBase:

- Податоци за името на винаријата заедно со нејзините координати.
- Податоци за најава
- Оценки за кафетеријата
- CRUD функционалности

Трет Дел – Имплементациска Архитектура



Слика 4. Имплементациска архитектура

Client/Web Browser:

- Пребарувач кој користи униформен апликациски протокол HTTPS
- Рендерира поглед на корисникот
- На клиентска страна како pushed функционалност ќе биде исцртувањето на мапата
- Праќа HTTPS барање и чека одговор (синхрона комуникација)

WebServer:

- Frontend технологија која ја користиме HTML/CSS со JavaScript и тука се наоѓа првиот слој MVC.
- Backend технологија Java+Spring, тука се наоѓа вториот слој а тоа е AppLogic.

MVC:

- Првиот слој во архитектурата кој е од делот Frontend
- Задолжен за комуникација со клиентите и известувања.
- Задолжен за препраќање на барањата од клиентите до серверот (и обратно).
- Одговорен за креирање на погледите кои ги нуди интерфејсот.
- Имплементација на интерактивна мапа преземена од OpenStreetMap

AppLogic:

- Вториот слој во архитектурата
- Може да земе и модифицира податоци од базата
- Ги подготвува податоците за првиот слој да ги испрати на корисникот.

NotificationTrigger:

- Одговорен за пренос (push) на настанот (внесена оценка) до AppLogic.
- Користи нотификациска компонента API за креирање нотификација во системот.
- Самата компонента ќе биде иницираната од секој нов настан односно внес.
- Има комуникација и со самата база на податоци каде ги праќа внесените оценки.

DataAccess:

- Користи CRUD операции за податоците од базата.

DataBase:

- PostgreSQL која ќе ја користиме за чување на конзистентните податоци потребни за имплементирање на функционалностите кои се опишани во барањата.