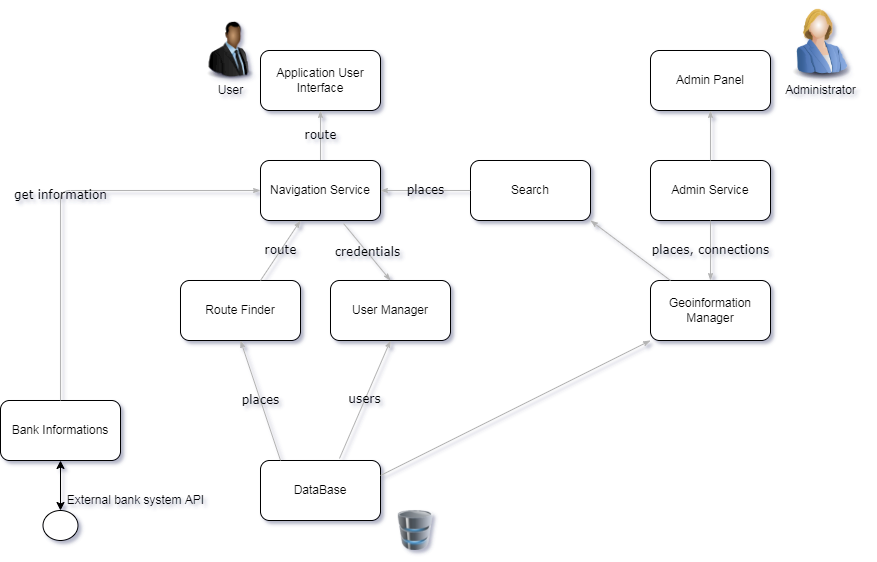
За нашата апликација, односно во нашиот архитектурен дизајн се потрудивме да ги опфатиме сите три погледи-концептуален, извршен и имплементациски.Овие погледи ги претставивме преку следните дијаграми.Од нив зависи нашата крајна апликација која ќе биде составена од хибридна слоевита веб архитектура и сметаме дека преку нив полесно ќе дојдеме до завршната фаза. Значи апликацијата треба да има добро дефиниран интерфејс,да биде едноставна за корисниците, а сепак технички да се разработени сите слоеви.

**КОНЦЕПТУАЛНА АРХИТЕКТУРА**



\*кратко објаснување на дијаграмот

Дијаграмот е направен според концептуална архитектура.Во него е разработена веб страната за наоѓање на банки во градот Охрид.Како што можеме да видиме од самата слика имаме само две засегнати страни корисникот и администраторот(ние).Системот е едноставен- при вклучување на самата апликација, системот ја има локацијата на корисникот(секако доколку тоа е дозволено) и во споредба со неа му ги прикажува блиските банки на корисникот.Исто така има и копче Search каде што корисникот ја пребарува посакуваната банка.Оваа информација оди до администраторот и веднаш му дава feedback на корисникот, односно му ја покажува рутата до банката кој тој ја избрал.Дополнително имаме и информации за секоја банка, ова е доста поволно за туристите.Сите овие информации се чуваат во база на податоци, во која нормално пристап има само администраторот.

1. Системот треба да му ги прикаже на корисникот сите филијали на банки во градот Охрид.
2. Системот треба да ја прикажува рутата до најблиската филијала на банка на

корисникот.

1. Системот треба да биде Веб – базирана апликација.
2. Системот да го прикажува работното време на банките.
3. Системот ќе покажува дополнителни информации за банките.
4. Системот треба да му овозможи на корисникот да дојде до информации за посакуваната банка со помалку од 4 кликови.
5. Системот ќе овозможи користење на веб сајтот преку мобилен уред.
6. Системот ќе овозможи користење на веб страницата преку сите оперативни системи.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Податоци** | **Функција** | **Засегнати страни** | **Систем** | **Апстрактен концепт** |
| Банки во Охрид | Пребарување на одредена банка | Администратор | Локации | Алатка за навигација |
| Локации за банките | Регистрирање | Корисник |  | Оптимална рута |
| Информации за банките | Log in/out |  |  | Најблиска рута |
| База со податоци |  |  |  | Интерактивна мапа |
| Рута |  |  |  |  |
| Корисници |  |  |  |  |

**AppUI**

* ShowPlaces
* DisplayMap
* DrawRoute

**AdminPanel**

* ListPlaces

**NavigationService**

* RouteComputed

**RouteFinder**

* ComputeRoute

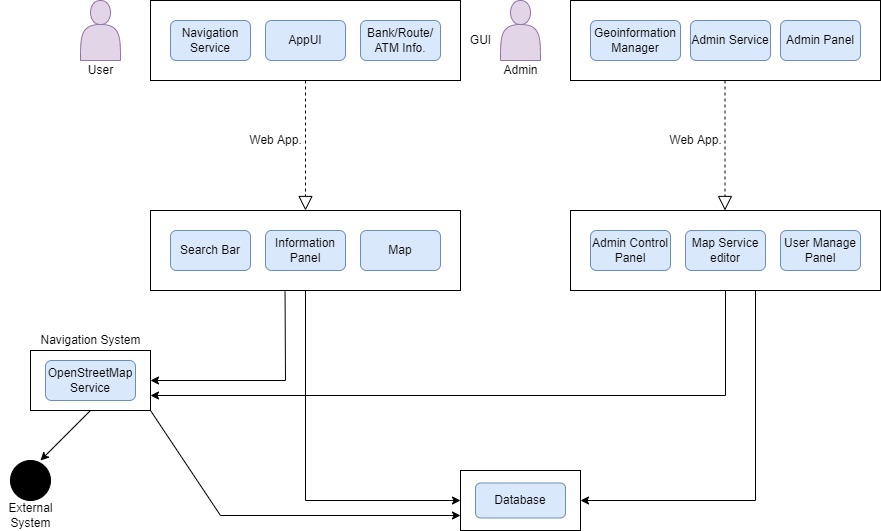
**UserManager**

* RegisterUser
* LoginUser

**GeoinformationManager**

* SearchPlace

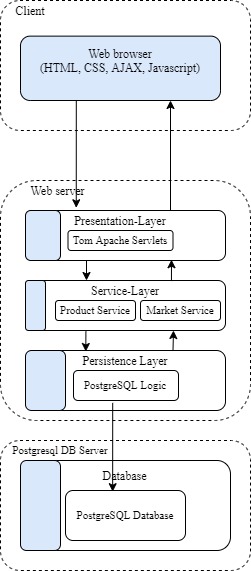
**ИЗВРШНА АРХИТЕКТУРА**

****

\*кратко објаснување на дијаграмот

Во овој извршен поглед влегуваат функционалните барања, а во некои итерации влегуваат и нефункционалните барања. Според нив апликацијата треба да изгледа како слоевит архитектурен стил.Повторно имаме корисник и админисратор. Апликацијата асинхроно комуницира со слоевите преку HTTP конекција.Секоја информација која корисникот ја бара и ја добива назад како одговор се наоѓа во базата на податоци и само администраторот може да праќа инфомации назад до корисникот по негово барање.Сета ова комуникација се случува во интерфејсот на WebApp.

**ИМПЛЕМЕНТАЦИСКА АРХИТЕКТУРА**

****

\*кратко објаснување на дијаграмот

Имаме три инфраструктури:клиент, веб сервер и база на податоци.Секој од нив помеѓу себе комуницира со API Call.Веб пребарувачот праќа повик до веб серверот до Presentation-Layer, тој до Service-Layer, па до Persistence-Layer.Компонентата PostgreSQL исто така испраќа API Call за читање/запишување до Postrgresql базата на податоци.