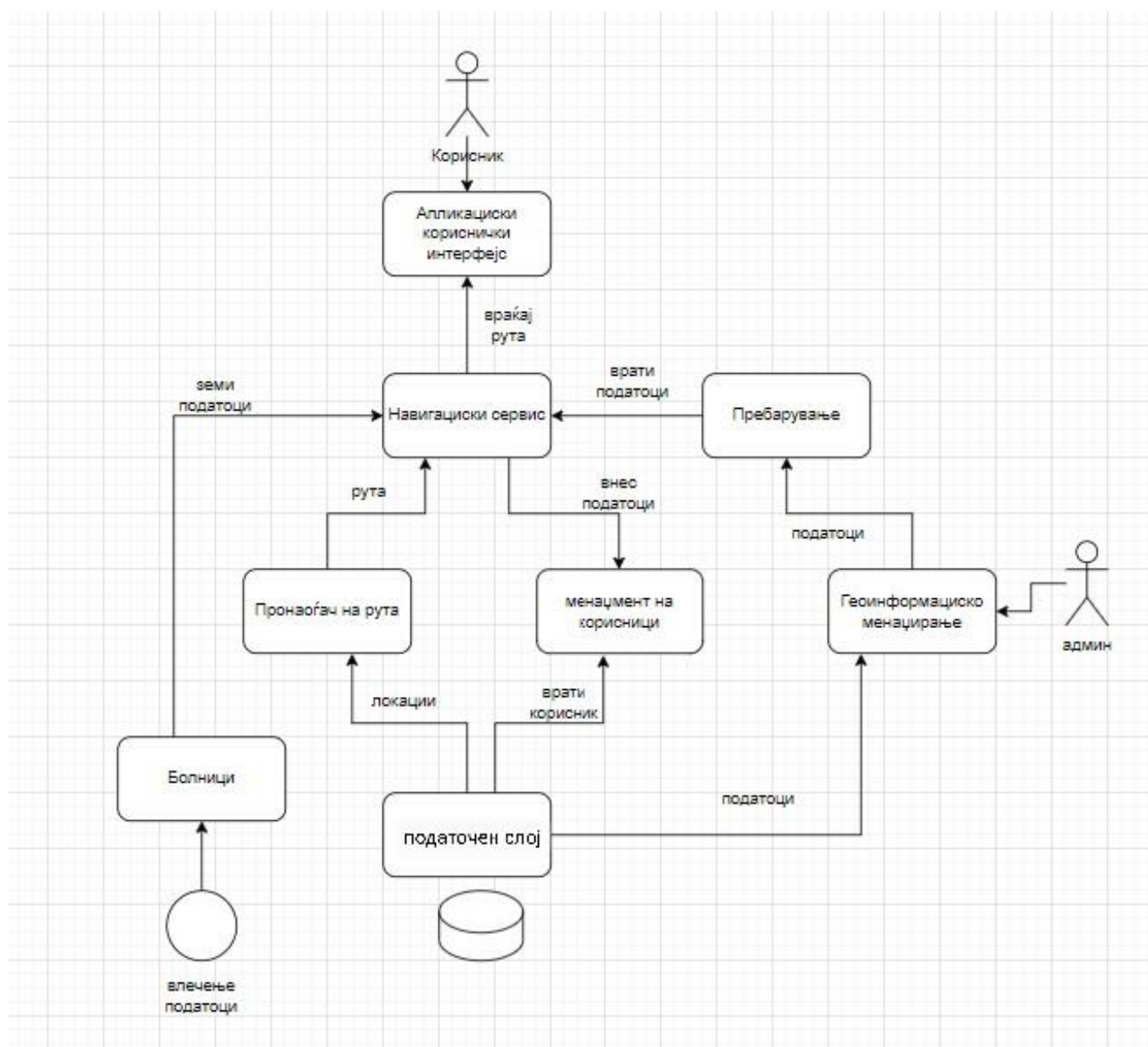


Архитектурни концепти на HospiMap

Во продолжение ќе ви ги претставиме трите слоеви на нашиот архитектурен дизајн преку кои на посликовит начин ќе ја доближиме функционалноста на нашата апликација, и начинот на кој таа ќе ги исполнува функциските, нефункциските и контекстуалните барања на клиентите. Во нашиот архитектурен дизајн се опфатени сите три погледи-концептуален, извршен и имплементациски. Овие погледи ги претставивме преку следните дијаграми. Нашата крајна апликација ќе биде составена од хибридна слоевита веб архитектура. Апликацијата е замислено да има добро дефиниран интерфејс, да биде едноставна за корисниците, а сепак технички да се разработени сите слоеви.

КОНЦЕПТУАЛНА АРХИТЕКТУРА



* објаснување на дијаграмот

Дијаграмот е направен според концептуална архитектура. Во него е разработена веб страната за наоѓање на најблиска рута до болници во зависност од локацијата на корисникот. 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 со клик на копчето пребарај, се прикажуваат најблиските болници на мапа. Со клик на одредена болница на мапата, информацијата оди до администраторот и веднаш му дава feedback на корисникот, односно му ја покажува рутата до болницата кој тој ја избрал. Сите податоци се чуваат во база на податоци, до која нормално пристап има само администраторот.

1. Системот треба да обезбеди автентикација на корисникот преку корисничко име и лозинка.
2. Системот треба да му овозможи на корисникот да ги види најблиските болници околу неговата моментална локација и насоките за одење до таа локација.
3. Со кликување на локацијата, системот на корисникот ќе му ја прикаже рутата до избраната локација.
4. Системот треба да обезбеди можност за постоење на повеќе слоевите мапа.
5. Системот треба да добие согласност од корисникот за користење на неговата моментална локација.
6. Податоците на системот за локации се сместени во база на податоци.
7. Системот треба да има функција за движење на мапата лево, десно, горе и долу.
8. Системот треба да има zoom in и zoom out функции.
9. Системот треба да има функција за ротирање на мапата лево и десно.
10. Доколку системот изгуби GPS сигнал, треба да се испрати известување до корисникот преку соодветен интерфејс.
11. Системот повторно ќе добие GPS сигнал доколку се изгуби во рок од 10 секунди по влегувањето во средина со мрежна покриеност.

Податоци	Функција	Засегнати страни	Систем	Апстрактен концепт
Болници	Пребарување болници	Администратор	Локации	Алатка за навигација
Локации за болниците	Регистрирање	Корисник		Оптимална рута
Информации корисниците	Log in/out			Најблиска рута
База со податоци	Zoom in Zoom out			Интерактивна мапа
Рута	Ротирање			
Корисници				

AppUI

- ShowHospitals
- DisplayMap
- DisplayRoute

Admin

- ListHospitals

NavigationService

- RouteComputed

RouteFinder

- ComputeRoute

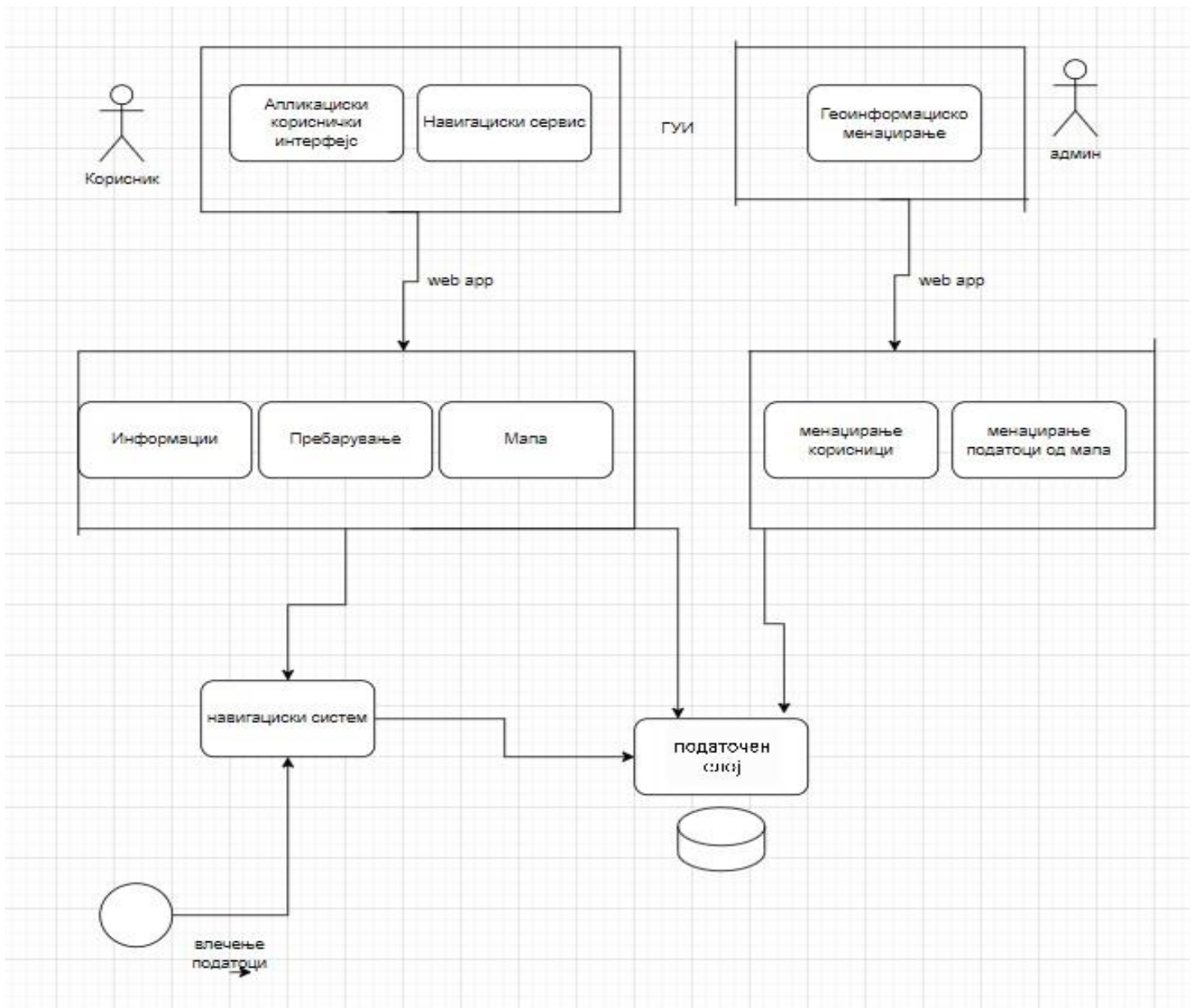
UserManager

- RegisterUser
- LoginUser

GeoinformationManager

- SearchForHospital

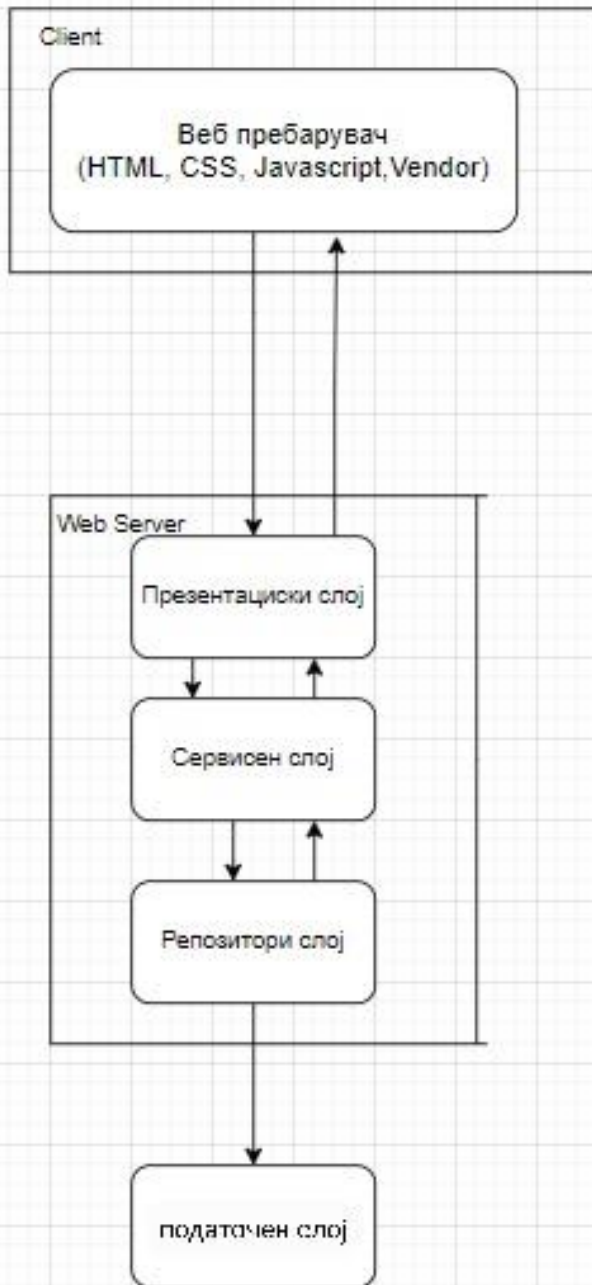
ИЗВРШНА АРХИТЕКТУРА



*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 администратор. Апликацијата асинхронно 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 информации назад до корисникот по негово барање. Сета оваа 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. Веб пребарувачот праќа повик до веб серверот до Presentation-Layer, тој до Service-Layer, па до Persistence-Layer. На крај комуницираат со податочниот слој кој кај нас се наоѓа во самата апликација.

