Лабараториски вежби по
Дизајн и архитектура на софтвер

Домашна работа 2 - Архитектурен дизајан

Изработиле:

- Матеа Иваноска, 201030
- Сара Глигорова, 201124
- Давор Ѓурчиноски, 201103
- Олег Столевски, 201226
- Христијан Ацоски, 196030

Скопје, декември 2022

Содржина

1. Концептуална архитектура

Концептуална архитектура е архитектура која се базира на доменско ниво на функционалности,основен архитектонски дизајн,таа е прв чекор во процесот на креирање на дизајнот на апликацијата, првиот одговор од засегнатите страни за тоа дали апликацијата се развива според замислата. Како и секој архитектонски дизајн и концептуалната архитектура се состои од компоненти и конектори.Компонентите претставуваат збир на доменско ниво на одговорности, кои произлегуваат од функционалните барања. Со цел да се поврзат компонентите на системот за да можат да разменуваат податоци (flow data) се користат конектори. На конекторите се додаваат лабели со цел подобро разбирање на системот и функциите. Концептуалните компоненти и конектори не се вклучуваат во финалната верзија на апликацијата тие ни служат како основа за понатамошен развој.Првиот чекор во креирање на основата на апликацијата е откако се добро дефинирани функционалните барања да се најдат key concepts, и на истите да им се доделат соодветни категории.
Динамичкиот аспект од концептуалната архитектура (Behaviour model) ,да се претстави протокот на податоците (flow data), најчесто се постигнува со визуелно претставување на структурата на системот преку use case мапи. Дополнително се додава и дел за одговорности на компонентите во кои се опишуваат функционалностите на секоја од компонентите заедно со конекторите со кои е поврзана.

Функционални барања

1. Апликацијата треба да биде достапна на македонски јазик.
2. Апликацијата во себе треба да содржи база на податоци во кои ќе се чуваат информации за секоја банка и банкомат вклучувајќи: име, id, координати.
3. Конекциите помеѓу локацијата на корисникот и банките, банкоматите се чуваат во база на податоци.
4. Апликацијата треба да овозможи приказ на сите финансиски институции на територија на град Скопје: банки.
5. Апликацијата треба да овозможи приказ на сите финансиски институции на територија на град Скопје: банкомати.
6. Апликацијата е навигациска алатка. Таа ќе овозможи да се пронајде најблиската банка од точката каде што се наоѓа корисникот.
7. Корисникот со улога на администратор на апликацијата ќе може да додава нови банки, банкомати, да брише, да врши измени.
8. Корисникот ќе може да ја избере банката, банкоматот од листата на понудени.
9. Избраната банка,банкомат од страна на корисникот ќе се прикаже на итерактивна мапа.

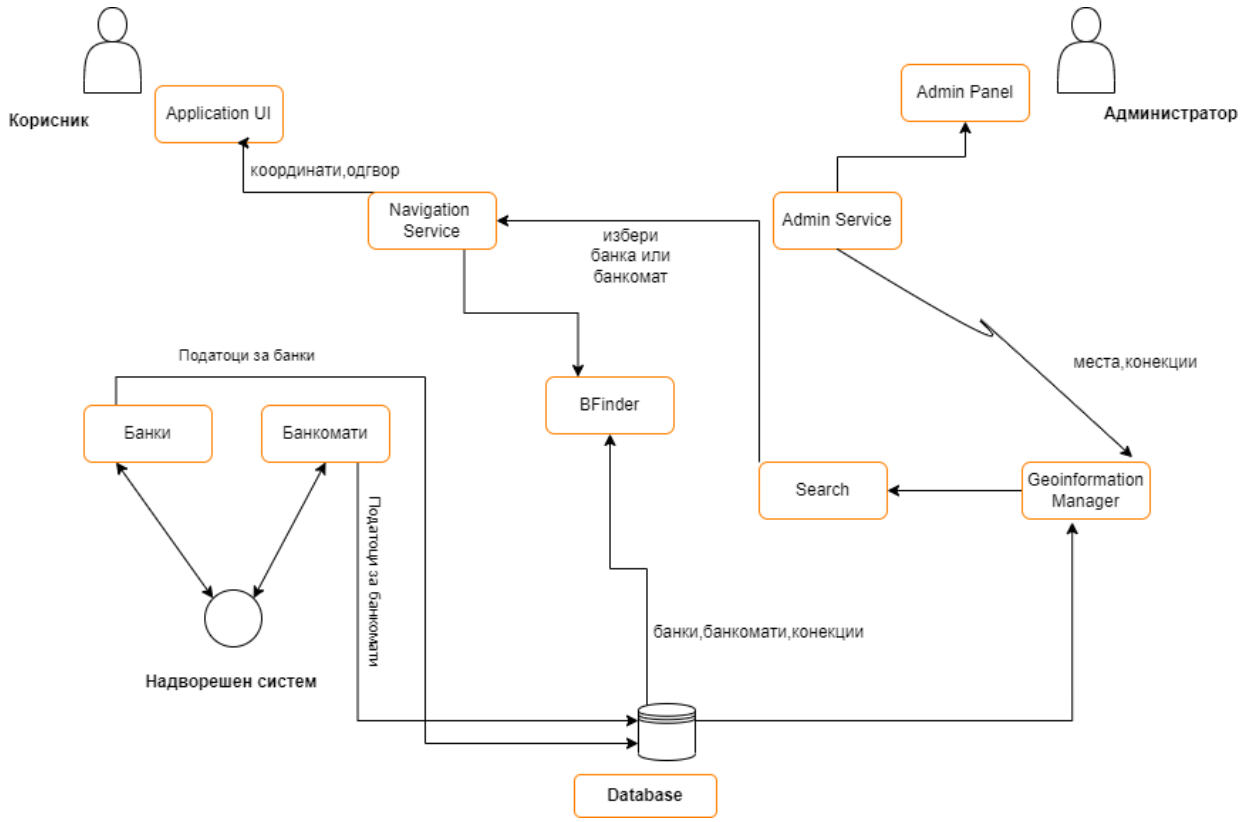
10. На корисникот ќе треба да му е вклучена локација на неговиот уред за да се лоцираат точните координати.

11. Лоцирање на локацијата на корисникот е овозможена преку опцијата во секој пребарувач за превземање на локацијата на корисникот.
12. Корисникот ќе треба да има овозможено и пребарувачот преку кој ќе пристапи до апликацијата да биде вклучена опцијата за дозвола за земање на локацијата на корисникот.
13. Апликацијата да е во интеракција со надворешни системи.

Клучни концепти (Key concepts)

Податоци	Функции	Засегнати страни (Чинители)	Систем	Абстрактен концепт
Банка	чуваат	Корисник	Надворешни системи	Итерактивна мапа
Банкомат	приказ	Администартор	Open Street Map	опција
База на податоци	пронајде		API	Најблиска банка
id	додава(банки, банкомати)		Веб преарувач (browser)	Најблизок баномат
Име	брише(банки, банкомати)			Навигациска алатка
Координати	врши измени			
Локација	избере(листа банки)			
Уред	превземање			
Финансиски институции	пристап			
Информации	вклучена			
Конекции				

Динамичкиот аспект од концептуалната архитектура (Behaviour model)



Одговорности на компонентите

- Admin Panel
  -листа од банки
  -листа од банкомати

- Admin Service
  -додавање на нова банка,банкомат
  -бришење на постоечка банка,банкомат
  -измена на постоечка банка,банкомат

- Geoinformation Manager
  -ги листа сите банки и банкомати внесени во системот на територија на град Скопје

- Search
  -избери од дадената листа на банки и банкомати

- Application UI
  -приказ на сите банки и банкомати
  -избор на банка или банкомат
  -приказ на мапа за избраната финанциска институција

- Navigation Service
  -превземање на координати од корисникот преку прелистувачот

- BFinder
  -превзмање на податоци од база за сите банки,банкомати
  -превземање на координати од корисникот
  -пресметка на најблиската финанциска институција до корисникот

- Банки
  -пркажување на име на банка
  -прикажување на id на банка
  -прикажување на координати на банка

- Банкомати
  -пркажување на име на банкомат
  -прикажување на id на банкомат
  -прикажување на координати на банкомат

Нефункционални барања

1. Апликацијата треба да биде достапна за користење преку следните на типови уреди: персонален компјутер, лаптоп, мобилен телефон, таблет.
2. Апликацијата треба да биде разбирлива и лесна за употреба за секој корисник.
3. Апликацијата треба да биде скалабилна.
4. Апликацијата треба да содржи сортирање и филтрирање кој ќе овозможи лесна употреба на корисниците.
5. Апликацијата треба да има кориснички интерфејс кој ќе им овозможи лесна употреба и навигација на своите корисници.
6. Апликацијата ќе биде достапна во форма на веб апликација, достапна за повеќњто веб прелистувачи (Firefox, Chrome, Edge, Safari).
7. Време на одговор на апликацијата не треба да биде повеќе од 5 секунди.
8. Превземање на податоци од база не треба да е повеќе од 1 секунда.
9. Апликацијата ќе биде развиена backend со Java Spring.
10. Апкикацијата ќе биде развиена frontend со Angular.

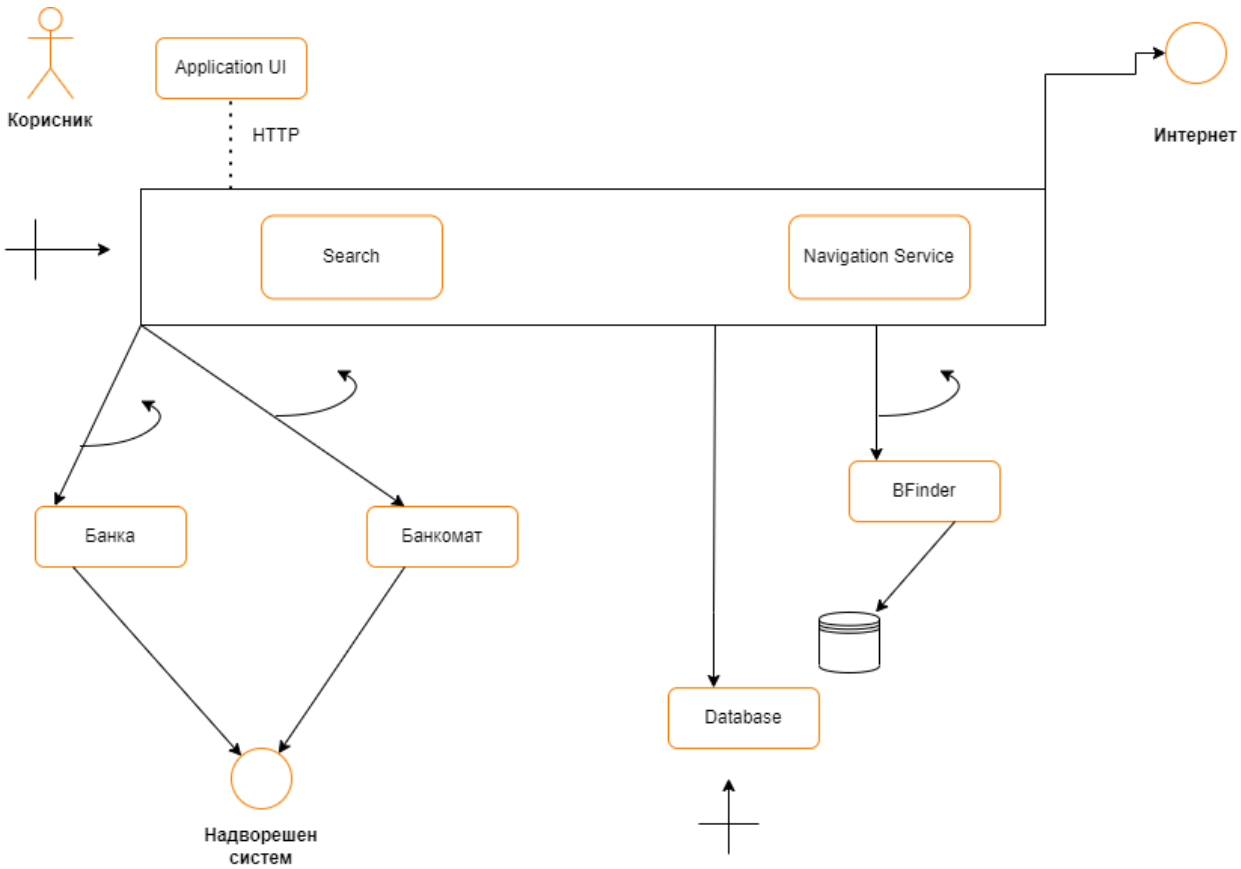
Клучни концепти (Key concepts)

Податоци	Функции	Засегнати страни (Чинители)	Систем	Хардвер	Абстрактни концепти
Кориснички интерфејс	достапна	корисник	Веб апликација	Персонален компјутер	разбирлива
	употреба		Веб пребарувач (browser)	Лаптоп	лесна
	сортирање		Firefox	Мобилен телефон	скалабилна
	навигација		Chrome	Таблет	време на одговор
	превземање податоци од база		Edge		
			Safari		

2. Извршна архитектура

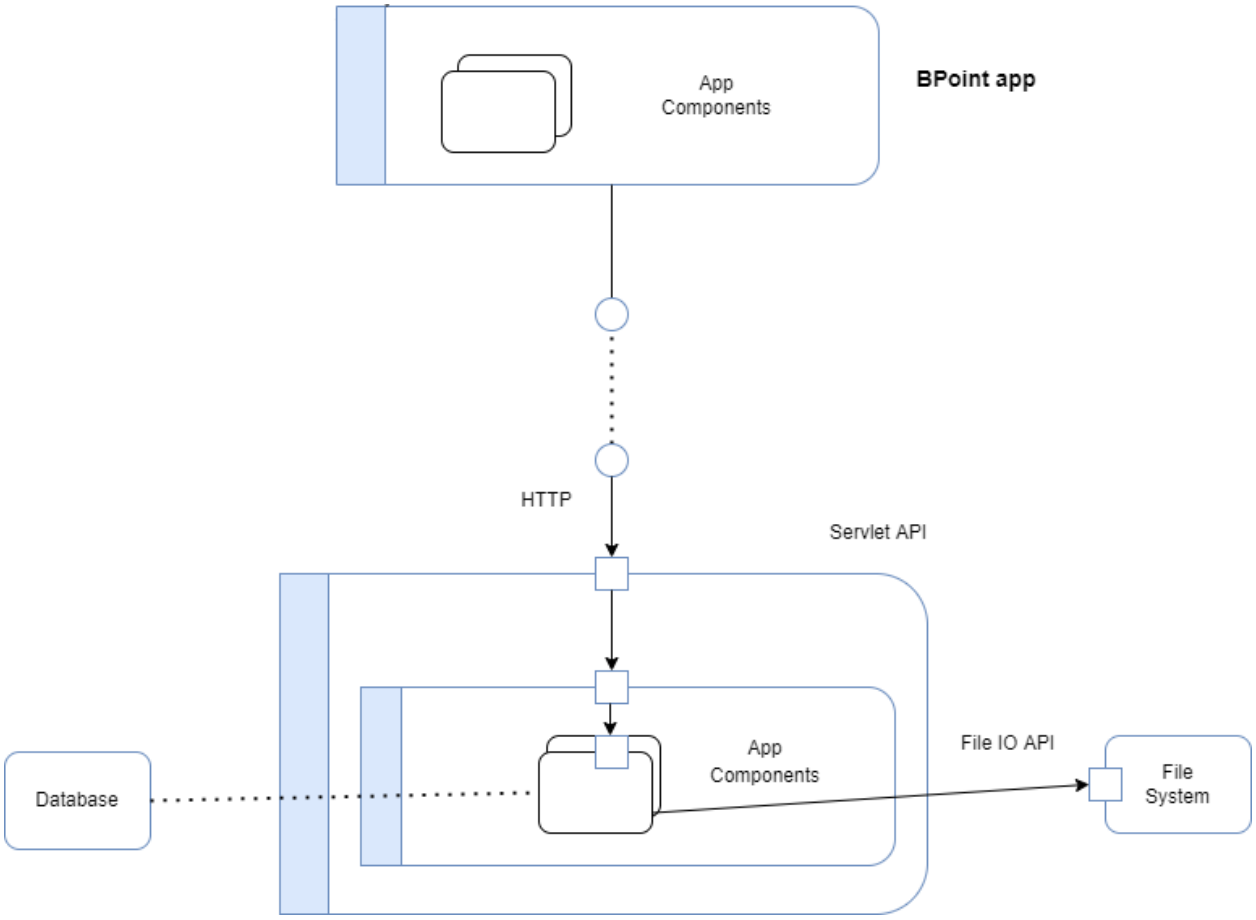
Извршната архитектура се фокусира на run-time структурата на системот на хардверските елементи,подисистеми,процеси и нишки. Оваа архитектура е најпогодна за испитување на атрибутите за квалитет на системот, перформансите. Се состои од компоненти и конектори кои ја даваат сликата за оваа архитектура. Компоненти се поврзани со конекторите на тој начин што е овозможен повик помеѓу конекторите за да се изврши дадена функција. Стереотипи на компонентите означуваат некаква конкурентна активност. Постојат три типа на стериотипи. User-initiated, во нашиот дијаграм Корисник, е компонента која извршува некоја активност како резултат на кориснички внес (input). Стереотипот Active, во нашиот дијаграм Банка,Банкомат, BFinder, се компоненти кои генерираат некоја активност внатрешно во нив, ови се компоненти кои постојано се извршуваат при повик на системот. Service компонентата, како што се Search, Database, чекаат некакво барање од другите компоненти и генрираат одговор за секое од поставеното барање.Погледот на извршната архитектура е повеќекратен. Динамичкиот аспект (Behaviour model) од извршната архитектура се претставува со помош на use case дијаграми.

Динамичкиот аспект од извршната архитектура (Behaviour model)



3. Имплементациска архитектура

Имплементациска архитектура е архитектура која опишува како е изграден системот во целост. Опишува кои технологии се потребни за да се имплементира апликацијата, софтверски пакети,библиотеки,класи.Постојат два типа на компоненти во оваа архитектура. Апликациски компоненти во нашиот дијаграм тоа се ,File System , кои се одговорни за имплементација на доменско ниво на одговорности од концептуалната архитектура. Инфраструктурни компоненти се потребни за системот да се извршува но не се поврзани со функционалностите на апликацијата како што се ,HTTP . Container component овозможува извршувачка околина за компонените кои ги содржи и го менаџира нивниот животен циклус. Во нашиот дијаграм тоа се BPoint app и Servlet API. Динамичкиот аспект од имплементациската архитектура се претставува со помош на секвенцијален UML дијаграм.



Динамичкиот аспект од имплементациска архитектура (Behaviour model)- Секвенцијален дијаграм