

**Технически университет – София**

**Дипломна работа**

*„Дизайн и имплементация на система за предлагане на благотворителни събития посредством изкуствен интелект и неблокиращи операции“*

Изпълнена от:

Студент: Симеон Димитров Корчев: ФКСТ

Специалност: КСИ

Факултетен №: 121215125

Група: 38

Дата: Научен ръководител:

Град: София /маг. инж. Д. Андреев/

**Съдържание**

[Въведение 3](#_Toc516941266)

1. [Постановка на дипломната работа, цели и задачи 5](#_Toc516941267)

[1.1 Същност на проблема и съществуващи решения. 5](#_Toc516941268)

[1.2 Софтуерни системи за мониторинг 11](#_Toc516941269)

[1.3 Сравнителен анализ 13](#_Toc516941270)

[Втора Глава 14](#_Toc516941271)

[2.1. Функционално описание на приложението. 14](#_Toc516941272)

[2.2 Използвани технологии. 16](#_Toc516941273)

[2.3 Обобщение 20](#_Toc516941274)

[Трета глава 21](#_Toc516941275)

[3.1. Структура на софтуера 21](#_Toc516941276)

[3.2 Вътрешна реализация 28](#_Toc516941277)

[3.3 Цялостен процес на работа 32](#_Toc516941278)

[Четвърта глава 35](#_Toc516941279)

[4.1 Инструкции за конфигуриране и използване на системата. 35](#_Toc516941280)

[4.2 Представяне работата на приложението 36](#_Toc516941281)

[Пета глава 40](#_Toc516941282)

[5.1 Описание на използваните елементи 40](#_Toc516941283)

[5.3 Тестване на приложението 42](#_Toc516941284)

[Заключение 46](#_Toc516941285)

Източници…………………...……………………………………………………………….48

Приложение……………...…………………………………………………………………..51

2.

Увод

Живеем във време на "информационна експлозия", в което количеството публикувана информация е огромно. Освен това напредъка в сферата на информационните технологии, както и лесният достъп до интернет, и голямата популярност на услугите, които световната мрежа доведоха до информационен бум. Ако в миналото проблемът на човек е бил липсата на информация, то днес проблемът е може би противоположен – наличието на прекалено много информация.

В процеса на ползване на интернет и при посещаването на различни уеб сайтове, съвременният потребител бива заливан от множество данни под различни форми и от различни източници. Една стандартна новинарска уеб страница би съдържала десетки, дори стотици новини за текущия ден, сайта на една книжарница би съдържала стотици, а дори и хиляди книги, една система за събития би съдържала стотици събития за всеки ден. Поставен в хипотеза на множество нефилтрирани данни генерирани от дадена система, потребителя би бил неориентиран в избора си на адекватен артикул, стока или услуга. За него става много трудно да направи правилен и лесен избор. Системите от този род не дават персонализирана информация, нито отчитат предпочитанията на потребителите, както и не взимат предвид техните интереси.

От друга страна изискванията към приложенията са се променили драстично през последните години. Само преди няколко години едно голямо приложение имаше десетки сървъри, секунди време за реакция, часове поддръжка офлайн и гигабайти данни. Днес нещата са коренно различни. Приложенията са разположени на всичко - от мобилни устройства до клъстери, базирани на облаци, работещи с хиляди многоядрени процесори. Потребителите очакват милисекундни време на реакция, както и това, че всяко приложение трябва да е достъпно всеки ден, по което и да е време на денонощието - винаги. Количеството генерирани данни от приложенията също е изменено. Днес е нормално за данните да говорим в петабайти. За съжаление обаче, днешните изисквания към приложенията просто няма как да бъдат удовлетворени с повечето досегашни софтуерни архитектури. Именно изискванията приложенията да бъдат responsive, resilient, message driven и elastic провокираха създаването на нов вид системи – реактивни системи. За постигането на тези характеристики, приложенията използват похватите на асинхронните, неблокиращи операции. Те са написани по такъв начин, че при който нишките, които приложенията използват, никога не блокират. Всеки път, когато нишката трябва да блокира (например при четене, писане към файл или сокет), тя просто бива известявана, когато данните са налични или прочетени, докато самата нишка в същият момент прави нещо друго. Подобен механизам обикновено бива имплементиран и реализиран чрез предаване на съобщения (или още наричани събития). Вече споменахме както асинхронност, така и не блокиране, като за тези две понятия следва да поясним, че в повечето думи това са синоними, тъй като процеса (потока) на работа в едно такова приложение е такъв, че ние изпращаме заявки за различни ресурси, не чакаме за отговор, а продължаваме изпълнението на полезна бизнес логика, а в даден момент от времето получаваме съобщение (или биваме уведомени за събитие), че отговора на нашата заявка е вече наличен.