THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

Kazakh National Research Technical University named after K. I. Satbayev

Institute of Information and Telecommunication Technologies

Department Of Cybersecurity, Data Storage and Processing

Mokhammad Yuz Alef Bissima

Development of Intellectual Chat Bot System on MVC WEB API JAVA spring/EE

DIPLOMA WORK

Major 5B070300 – «Information systems»

THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

Kazakh National Research Technical University named after K. I. Satpayev

Institute of Information and Telecommunication Technologies

Department Of Cybersecurity, Processing and Storage of Information

ADMITTED TO DEFENSE
Head of Department Department
Of Cybersecurity, Processing
and Storage of Information,
PhD, associate professor
___________N.A.Seilova
«_13_»___05_____ 2019.

DIPLOMA WORK

Theme of work: «Development of Intellectual ChatBot System»

Major 5B070300 - «Information system»

Performed: Mokhammad Yuz Alef Bissima

Reviewer,
PhD, professor of AlmaU
____________T.I.Bakibaev
«_13_» __May__ 2019.

Scientific advisor,
Tutor
_____________A. T. Azhenov
«_13_» __05______2019.

Almaty 2019

Almaty 2019

THEMINISTRYOFEDUCATIONANDSCIENCE OF THE REPUBLIC OF KAZAKHSTAN

Kazakh national research technical University named after K. I. Satpayev

Institute of Information and Telecommunication Technologies

Department of Cybersecurity, Processing and Storage of Information

5B070300 – Information system

AFFIRM
Head of Department of
Cybersecurity, Processing and
Storage of Information
PhD, associate professor
________________ N.A.Seilova
«_13_»_05_________2019.

TASK
to perform the thesis

Student Mokhammad Yuz Alef Bissima.
Theme of diploma work: «Development of Intellectual Chat Bot System on MVC WEB API JAVA spring/EE»
Approved by the order of the University№1162б from «_16_»_10_________2018.
Deadline for completion of work «_13_»_05_________2019.
Source data to work: Web application Chat Bot for Ecommerce projects, instant messengers integration project, chatbot development.
Summary of Diploma work:
a) the development of the website;
b) integration of chats from social channels;
c) the development of the chatbots structure.
Recommended main literature: 20 sources
The list of graphic material presenting: 15 slides

SCHEDULE
of preparation of the Diploma work

The name of the sections, a list of issues	Deadline for submission to supervisor and consultants	Notice
Overview and analysis of existing IS on the market	10.01.2019 - 08.03.2019.	
Writing the functional structure of IS	05.02.2019-10.03.2019.	
Writing program part	11.03.2019-28.04.2019.	

Signature
Consultants and normcontrol to complete a thesis indicating the related sections of the thesis

Section titles	Consultants, Full name (academic degree, rank)	Data of signing	Signature
Development of Program software	M.B Bauyrzhan, Master of Technical Sciences, tutor	8.05.19	
Normcontrol	O.V Kisseleva, senior-lector	8.05.19	

Scientific adviser ____________ Azhenov A. T.

Task was accepted for execution by the student ____________ Mokhammad Y. B.

Data «13» 05 2019 .

THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

SATBAYEV UNIVERSITY

REVIEW

on ________Diploma work________
(name of the type of work)

Mokhammad Yuz Alef Bissima
(full name student)

5B070300 – Information systems
(code and title of specialty)

Theme of work: «Development of Intelligent Chat Bot System»

Department Of Cybersecurity, Processing and Storage of Information

Done:
a) software part on 33-39 pages
b) the volume of work 39 pages

COMMENTS TO WORK

The work of Mokhammad Y.B. is devoted to the development of intellectual chatbot system for the E-commerce. The student made the choice of the system works of online consultants, studied the principles of the system, studied and applied in practice the methods of developing a web site based on the selected system. The Diploma work provides an overview and comparative analysis of currently used chatbots in the field of e-Commerce. In the students work were studied and applied programming languages Python and JavaScript, AJAX and MVC technology.The student had completed the integration of instant messengers and social networks and developed the chatbot with the training mode.The student demonstrated the use of a scientific approach to solving applied problems, good programming skills and the ability to justify the solutions used in the work.

Work evaluation

Generally, diploma work performed at a high level. Assessment of the work 87 points. M Yuz Alef Bissima 05070300 –"Information systems" with the qualification of bachelor, a degree, feel that they deserve.

Reviewer,
PhD, professor of AlmaU
__________ T.I.Bakibaev
«13» May 2019.

School of Engineering Management
Республика Казахстан г. Алматы Учреждение образования «Алматы Менеджмент Университет» * 43,2061596 76,8933833

THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

SATBAYEV UNIVERSITY

REVIEW

OF SCIENTIFIC ADVISOR

on ________Diploma work________
(name of the type of work)

________Mokhammad Yuz Alef Bissima________
(full name student)

________5B070300 – Information systems________
(code and title of specialty)

Theme of work:______«Development of Intelligent Chat Bot System»______

The work of Mokhammad Y. B. is devoted to the development of intellectual chatbot system for the E-commerce and study of the possibility of improvement online consultants through the integration of external channels and the connection of machine learning. The work provides an overview and comparative analysis of currently used chatbots in the field of e-Commerce , with an emphasis on those that allow you to use them in sales. The specificity of the task is that now basically all sales are conducted online this suggests that all requests, queries and questions about the products comes online to different channels at different times. To serve the customer efficiently and on time student use innovative approach to the task. The student had completed the integration of instant messengers and social networks which mostly questions in one and put the chatbot with the training mode.The author demonstrated the use of a scientific approach to solving applied problems, good programming skills and the ability to justify the solutions used in the work.The recommended rating is“excellent”.

Scientific advisor,
Tutor,
________A. T. Azhenov
«13» 05 2019.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Мохаммад Юз Алеф Бисима

Название: “Development of intellectual chatbot system"

Координатор: Алмат Аженов

Коэффициент подобия 1:1,2

Коэффициент подобия 2:0

Тревога:1

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

☑ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
□ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
□ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

....
....
....
....
....

....

Дата 13.05.19г

Подпись заведующего кафедрой /

начальника структурного подразделения

Окончательное решение в отношении допуска к защите, включая обоснование:

Допущен к защите!

Дата 13.05.19г

Подпись заведующего кафедрой /

начальника структурного подразделения

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Мохаммад Юз Алеф Бисима

Название: “Development of intellectual chatbot system"

Координатор:Алмат Аженов

Коэффициент подобия 1:1,2

Коэффициент подобия 2:0

Тревога:1

После анализа Отчета подобия констатирую следующее:

☑ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

………
………
………
………
………
………………………………………………………………

13.05.2019
…………………………… ……………………………………………

Дата *Подпись Научного руководителя*

Краткий отчет

ҚАЗ ҰТУ

Университет:	Satbayev University
Название:	"Development of intellectual chatbot system"
Автор:	Мохаммад Юз Алеф Бисима
Координатор:	Алмат Аженов
Дата отчета:	2019-05-06 05:38:21
Коэффициент подобия № 1:	**1,2%**
Коэффициент подобия № 2:	**0,0%**
Длина фразы для коэффициента подобия № 2:	**25**
Количество слов:	7 916
Число знаков:	47 589
Адреса пропущенные при проверке:	
Количество завершенных проверок:	41

!

К вашему сведению, некоторые слова в этом документе содержат буквы из других алфавитов. Возможно - это попытка скрыть позаимствованный текст. Документ был проверен путем замещения этих букв латинским эквивалентом. Пожалуйста, уделите особое внимание этим частям отчета. Они выделены соответственно. Количество выделенных слов 1

>> Самые длинные фрагменты, определеные, как подобные

№	Название, имя автора или адрес гиперссылки *(Название базы данных)*	Автор	Количество одинаковых слов
1	URL_ https://katasis.com/service/website-development/crm-development		11
2	URL_ https://katasis.com/service/website-development/crm-development		9
3	URL_ https://katasis.com/service/website-development/crm-development		9
4	URL_ https://katasis.com/service/website-development/crm-development		8
5	URL_ https://katasis.com/service/website-development/crm-development		7
6	URL_ https://katasis.com/service/website-development/crm-development		7
7	Spin Foam Models for Quantum Gravity and semi-classical limit (*RefBooks* - https://arxiv.org/) Wolters Kluwer	Maité Dupuis;	6
8	URL_ https://katasis.com/service/website-development/crm-development		5
9	URL_ https://katasis.com/service/website-development/crm-development		5
10	URL_ https://katasis.com/service/website-development/crm-development		5

>>

Документы, в которых найдено подобные фрагменты: из RefBooks i

Источник https://arxiv.org/

№	Название	Автор	Количество одинаковых слов (количество фрагментов)
1	Spin Foam Models for Quantum Gravity and semi-classical limit	(Maité Dupuis;)	11 (2)

>> Документы,содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>> Документы,содержащие подобные фрагменты: Из внешних

баз данных

Не обнаружено каких-либо заимствований

Документы,содержащие подобные фрагменты: Из интернета

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

№	Источник гиперссылки	Количество одинаковых слов (количество фрагментов)
1	URL_ https://katasis.com/service/website-development/crm-development	66 (9)
2	URL_ https://www.veracrypt.fr/en/FAQ.html	10 (2)
3	URL_ https://www.dol.gov/whd/overtime/fs17c_administrative.htm	5 (1)

АНДАТПА

Сұхбат бот зияткерлік жүйесін дамытуы
MVC WEB API JAVA spring/EE

Диссертация автоматтандырылған жұмыс орнында онлайн-кеңесші болып табылатын веб-қосымшаны әзірлеу болып табылады. Сондай-ақ, API технологиясымен жұмыс істеу зерттелді және сыртқы әлеуметтік арналармен интеграция қажет болды. Веб-қосымшаларды әзірлеу сындарлы үлгілерді пайдалану арқылы жүзеге асырылды. Мақалада веб-қосымшаларды әзірлеу және клиенттермен өзара әрекеттесуді автоматтандыру бойынша қолданыстағы шешімдер талданады. Ол пайдаланылған технологияларды, веб-қосымшаны жасау процесін, ботты дамыту модулін, деректер базасымен.

АННОТАЦИЯ

Разработка интеллектуальной системы чатбот на
MVC WEB API JAVA spring/EE

В дипломной работе проводится разработка веб-приложения, которое является автоматизированным рабочим местом онлайн-консультанта. Также исследовалась работа с API технологиями и было осуществлена интеграция с внешними социальными каналами. Разработка веб приложения было осуществлена с использованием конструкторных шаблонов. В работе проведён анализ существующих решений для разработки веб приложения и автоматизации взаимодействий с клиентами. Описаны использованные технологии, процесс построения веб-приложения, модуль разработки бота, взаимодействие с базой данных.

ABSTRACT

«Development of Intellectual Chat Bot System on MVC WEB API JAVA spring/EE»

The thesis is the development of a web application that is an automated workplace online consultant. The work with API technologies was also investigated and integration with external social channels was essential. The development of a web application was carried out using constructive patterns. The paper analyzed the existing solutions for developing a web application and automating customer interactions. It describes the technologies used, the process of building a web application, the bot development module, interaction with the database.

CONTENT

INTRODUCTION

Every day, the number of users of online resources is growing relative to the number of visitors to similar areas offline. We are talking about stores and their online versions, about music concerts and their online broadcasts, about food places and applications for food delivery, as well as many other spheres of life. And an important stage in the development of network resources is the emergence of not only technical support, but also a two-way communication channel between users and representatives of the organization, namely online consultation. This type of communication with customers has become not only a means of assistance, but also a tool for interaction. The ordering process is still quite complicated for most customers of e-Commerce companies. Some studies show that the share of cancellations reaches 68%. Even if the buyer has studied the products, selected the appropriate products and placed them in the basket, the chance that he will not make a purchase is very high.

However, thanks to modern technology, this stage of the purchase process can be significantly improved. Chatbot is one of such useful technologies that can help the client in the process of ordering. It is expected that chatbots will increase savings of billions of dollars per year in the field of e-Commerce

They help people solve their problems and reduce the costs of businessmen. As a result, the need for managers and employees of call-centers disappears. The main tasks of the smart chatbot are to automate interaction with the client at each stage of the sales funnel and to save the customer support service from solving typical issues, significantly reducing the cost of customer service.

The main reasons for the introduction of automated working of specialist:

- improving the quality of consultants ' work;
- significant increase in labor productivity consultant tov;
- increase user loyalty;
- an increase in profit resource;

At the moment, there are many automated systems of decision support, as well as software for online consultation, however, there are no tools that would combine these aspects of the work. The purpose of the thesis is to develop intellectual chatbot system with integration of messenger and social network.

Based on this goal, the following tasks were set:

- the development of the website;
- integration of chats from social channels;
- the development of the chatbots structure.

1 Analysis of available automation menegment systems on websites

1.1 The general principles of information system

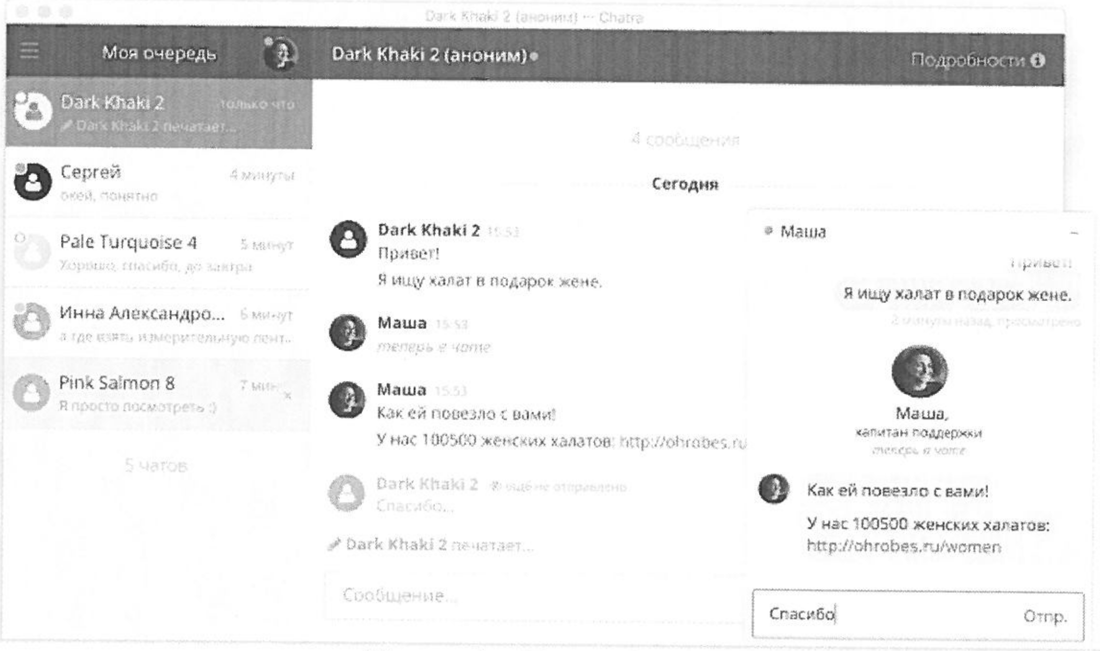
Chatbot is a virtual interlocutor that integrates into messengers and helps business to be closer to customers. It is an automated intelligent system of communication with users. Chatbots – also known as "online consultant" is a program that simulates human communication with the help of text or voice.

Today, the owners are trying all sorts of methods to optimize their sites so that visitors were more comfortable to interact with them. To improve you need every element — as a website design, and quality placed on it content. But it is important to take into account the fact that there are specialized tools with which the site can be made much more convenient for the audience. One of these tools is an online consultant for the site.

Online consultant is a special system designed for online communication with users. Managers have the opportunity 24 hours a day to advise clients on a number of issues, solve their problems when working with the site or describe the organization's activities in detail.

Many websites disagree that the online consultant for the site is one of its most important components. We are talking about online stores or sites for the services provided. Users always have a lot of questions, the answers to which they want to receive as soon as possible. Units write letters to tech support, since most users are sure that the answer will not come from there soon [1].

An online consultant for the site comes to the rescue: in the chat you can quickly answer questions and advise the client in real time like showed in figure 1. This is convenient in those situations when you need to get a link to a specific page or specific products on the site.



Dark Khaki 2 15:53
Привет!
Я ищу халат в подарок жене.

Маша 15:53
теперь в чате

Маша 15:53
Как ей повезло с вами!
У нас 100500 женских халатов: http://ohrobes.ru

Dark Khaki 2 ещё не отправлено
Спасибо...

Dark Khaki 2 печатает...

Я ищу халат в подарок жене.

Как ей повезло с вами!
У нас 100500 женских халатов: http://ohrobes.ru/women

Figure 1 - Example of chatbot

Now there are many online means of communication — it's Whatapp, Skype, Vk and Telegram. Owners of online stores usually provide contact information on their websites. Such programs — a great many, but not all of them have. For a huge number of users it is easier and faster to write through an online consultant for the site.

1.2 Areas of application of the online consultant

Installing an online consultant for the site solves the following tasks [2]:

- to perform lead generation (capturing leads). To contact the consultant, the userяmust leave contact information-e-mail, phone number. That is, there is a collection of data about potential buyers for further interaction with them, or, in other words, capture leads;

- promotional materials. The online chat window can be used not only to ask the consultant a question, but also for advertising, for example, a banner with information about the campaign, after clicking on which the client will go to the thematic page, etc. in addition, in the online chat, you can place a window with a mini-survey and collect user opinions about a par- to provide quick feedback to potential customers. If the company

provides quality service, it increases the number of customers and the size of the average check;

ticular product or service.

Of course, the online consultant for the site can be used in another way, but we still want to devote our article to the main options for using this program.

In practice, even capturing leads and displaying advertising messages for an online consultant are secondary features. Therefore, the main goal of the online consultant is a convenient and quick solution to customer problems through communication with them.

1.3 Advantages and disadvantages of online consultant

So, an online consultant is a software product that is installed on websites. Through it, a person can ask questions to the consultant or Manager. He does not need to spend money on a call and time waiting for his turn on the line. And this is only part of the advantages of using an online consultant.

Let's focus on the benefits of the program:

- improving the quality of service. The user does not understand something — he asked a question and immediately received an answer;
- promptly send a link to the desired product to the client who can not find it in the catalog or does not want to do so. But at the same time you find out whether the site is well-structured, if a person can not find the right product [3];
- you can see which page the client is on. Accordingly, it is possible to give him targeted advice on the products of interest;
- immediately receiving an answer;

- also shows where the person on what request was on the site. So you can adjust the optimization processes to meet demand;

- thanks to voice communication, which can be connected together with text chat, the client quickly receives the necessary information. For example, he was told the prices and throw off links to products;

- do not need to vkll special program for communication. Not all clients have Skype or ICQ. And no programs are needed for chat, you communicate with clients quickly and easily;

- adapted to the design of the site, forms an overall impression of comfort and ease of free communication. For a visitor who is not well versed in different "chips", the simpler the better;

- from a consultant may be the most recent information. It can tell users about the changes that for some reason have not yet been reflected on the site (not the most advantageous situation, but anything happens);

- if the client first time buys goods in the online store, the online consultant for the site comes in handy here (if a person "hangs" on the order page for half an hour, most likely, he had difficulties, and offer him help would be quite appropriate).

There are many advantages. If you use the right online consultant for the site, it brings a really good result. But there is always a significant BUT. This BUT occurs on many sites and can take the following forms:

- lot of unnecessary buttons in the chat. Unnecessary buttons and tabs knock the customer off. The purpose of creating an online chat is to facilitate communication. It should also be clear where the question is and where the answer is;

- if the chat is integrated with the overall design of the site, it is good, beautiful. But not all visitors are able to understand what's what. The chat should look like a chat, not a glossy magazine picture;

- grammar Problems. The consultant not only needs to know the promoted product "from" and "to", it is important and competent description. Illiteracy repels and always speaks of incompetence employee;

- pop-up is out of fashion. Don't be fooled into thinking that someone likes pop-UPS in the middle of the screen. On the contrary, the user will immediately leave the site, because the extra details will push him away.

1.4 Classification of existing software systems

Based on all of the above aspects of the work, it is possible to classify the software for each of them:

- modules for the site;
- instant messaging services;
- call center SOFTWARE;
- CRM system;
- decision support systems.

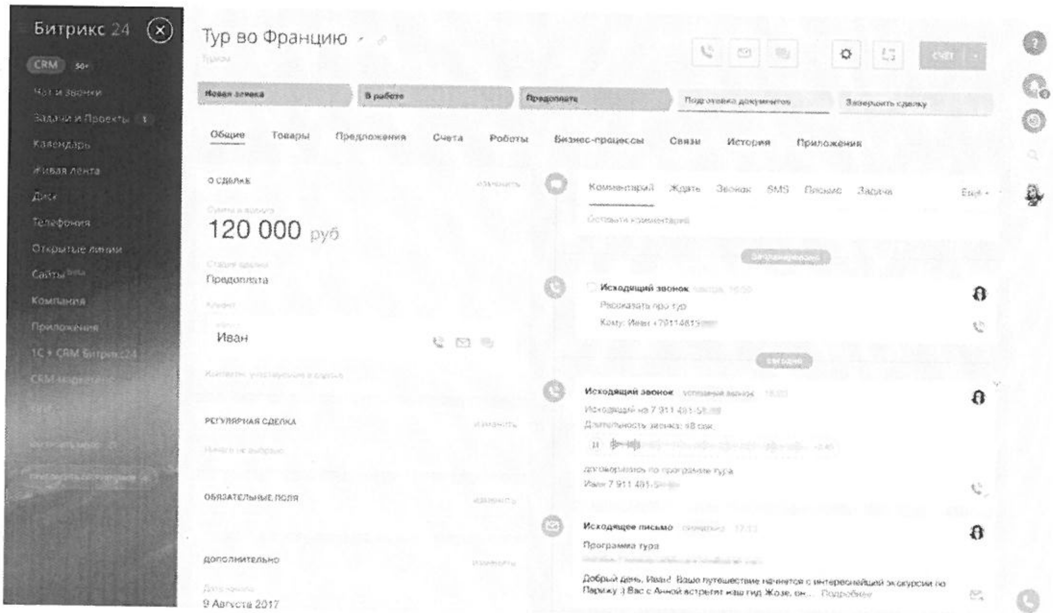
Based on the specifics of this work, it makes sense to consider systems that to some extent automate interaction with customers, as well as help in real-time decision-making.

1.4.1 CRM system

Customer Relationship Management is a special approach to business, in which the client is put at the forefront of the company's activities.

This strategy involves the creation in the company of such mechanisms of interaction with customers, in which their needs have the highest priority for the enterprise. Such customer focus affects not only the overall business strategy of the company, but also the corporate culture, structure, business processes and operations.

The main goal of the CRM strategy is to create a single ecosystem to attract new and develop existing customers. Manage relationships means to attract new customers, neutral buyers turn into loyal customers of loyal clients to build business partners. The main functions of the CRM shows in figure 2.



Добрый день, Иван! Ваше путешествие начнется с интереснейшей экскурсии по Парижу :) Вас с Анной встретит наш гид Жозе, он... Подробнее

Figure 2 - Bitrix CRM system

CRM systems are specialized software tools that automate business processes, procedures and operations that implement the company's CRM strategy.

Bitrix24 call themselves multichannel CRM (i.e. CRM with different channels through which the user can contact the company). This multichannel can be achieved thanks to "Open lines" — chat, callback, messengers and social networks. Bitrix has a free version [4] (which will interest CRM users, but will not be so interesting for those who are looking for channels to communicate with potential customers — there is only one channel to choose from). The number of open lines for the test is not limited (by the way, the test version is available for 30 days).

In Fact, it is still a CRM and task scheduling tool. To communicate with the audience and benefit from this, it is better not to use it. The cost, if you have a large staff and you plan to connect a lot of channels, is quite high.

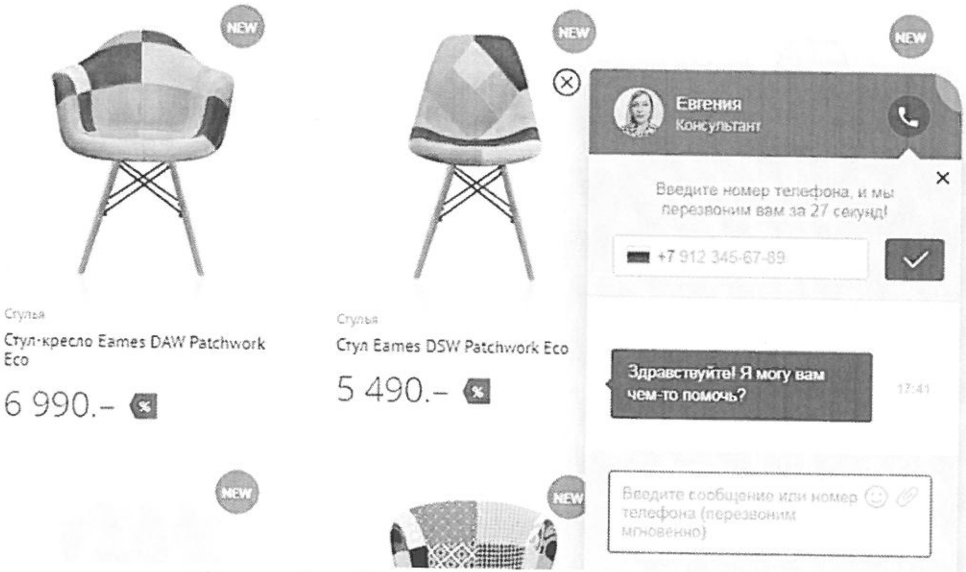
1.4.2 Instant messaging services

Jivosite is the most common online consultant for the site on the market. There is a free version, the possibilities of which are very limited, and therefore strongly rely on it is not necessary. In addition, there is a limit on the number of staff — up to five employees. Another point: a large company can not always afford to use a free program for reasons of prestige and credibility. And even if we talk about the paid version, this is one of the most budget solutions on the market.

Now about the advantages. These include the presence of a convenient personal account, the ability to simply and quickly connect all the necessary channels ant don't lose the customer. It shows in figure 3.

In Addition, the consultant's response history is kept in a special Annex, and a list of its operational responses is created, which makes it easier to respond to typical requests [5] (however, this function is incomplete, since the system itself determines what to save and, among other things, takes into account typos).

Reports on all dialogs are sent to e-mail. Statistics cannot be called functional. It is not very convenient to work with dialogues. In addition, the period of storing the history of communication with users is only 60 days. The last point is the most critical, as the sales cycle can last six months — you should not forget about working with loyal customers.



NEW
NEW
NEW
Стулья
Стул-кресло Eames DAW Patchwork Eco
6 990.-
%
Стулья
Стул Eames DSW Patchwork Eco
5 490.-
%
Евгения
Консультант
NEW
NEW

Введите номер телефона, и мы перезвоним вам за 27 секунд!

+7 912 345-67-89

Здравствуйте! Я могу вам чем-то помочь? 17:41

Введите сообщение или номер телефона (перезвоним мгновенно)

Figure 3 – Example of sales of chatbot

The program may well show offline, even if at this point you have several operators. When you configure a specific address for a particular scenario page, the system continues to show the client a default prompt. At the same time, there are a lot of script settings. There are many useful schemes by which you can get a buyer.

Jivosite settings also include section to connect all available channels, including social networks and messengers. If we understand omnichannel as a whole, then thanks to it the Manager sees the previous history of interaction between the user and the company (regardless of which channel the user came from).

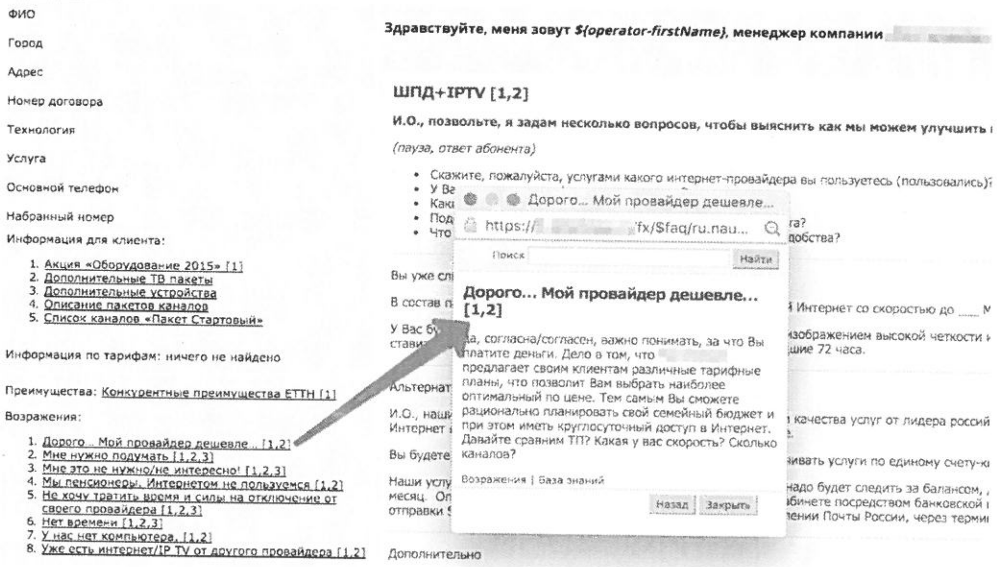
In Fact, the Jivosite operator does not see even the previous correspondence with the user who returned to the site on the same day after some time. In this case, if the user himself writes to the consultant, the story from time to time gets to the operator. But if the operator writes, the conversation always begins again. The same applies to fixing the history of the client in the personal account. The system will not replenish one client card (for example, even if he constantly leaves the same phone number), but will always create different applications and branches of the dialogue in the history [6].

Thus, it will never be clear that this is the same user (especially in practice, when a person comes to the site and through the window in the chat asks a question that you need to respond quickly). Of course, with social networks is still sadder. Through them, and the user does not always receive notifications

1.4.3 Call center software

Naumen Contact Center is a single solution for organizing corporate and outsourcing call centers, a full-featured call center built on the basis of VoIP-telephony technology.

The program developed by Naumen for the call center provides telephony, as well as reception and processing of requests through other channels (e-mail, Fax, SMS, web chat, call from the website, social media, messengers, mobile applications). The solution uses SIP-Protocol for data transmission, which allows connecting VoIP-gateways and subscriber terminals of different manufacturers, providing complete freedom of choice of server and telecommunication equipment. NAUMEN contact centers work imagine in 4 figure.



ФИО

Город

Адрес

Номер договора

Технология

Услуга

Основной телефон

Набранный номер

Информация для клиента:

1. Акция «Оборудование 2015» [1]
2. Дополнительные ТВ пакеты
3. Дополнительные устройства
4. Описание пакетов каналов
5. Список каналов «Пакет Стартовый»

Информация по тарифам: ничего не найдено

Преимущества: Конкурентные преимущества ETTH [1]

Возражения:

1. Дорого .. Мой провайдер дешевле .. [1,2]
2. Мне нужно подумать [1,2,3]
3. Мне это не нужно/не интересно! [1,2,3]
4. Мы пенсионеры. Интернетом не пользуемся [1,2]
5. Не хочу тратить время и силы на отключение от своего провайдера [1,2,3]
6. Нет времени [1,2,3]
7. У нас нет компьютера. [1,2]
8. Уже есть интернет/IP TV от другого провайдера [1,2]

Здравствуйте, меня зовут *${operator-firstName}*, менеджер компании

ШПД+IPTV [1,2]

И.О., позвольте, я задам несколько вопросов, чтобы выяснить как мы можем улучшить

(пауза, ответ абонента)

- Скажите, пожалуйста, услугами какого интернет-провайдера вы пользуетесь (пользовались)?
- У Ва
- Как
- Под … га?
- Что … добства?

Вы уже сл

В состав п … Интернет со скоростью до ____ М

У Вас бу … изображением высокой четкости
стави … шие 72 часа.

Альтернат

И.О., наш … качества услуг от лидера россий
Интернет

Вы будете … ивать услуги по единому счету-к

Наши услу … надо будет следить за балансом,
месяц. Оп … абинете посредством банковской
отправки … лении Почты России, через терми

Дополнительно

Дорого... Мой провайдер дешевле...

https:// fx/$faq/ru.nau...

Поиск Найти

Дорого... Мой провайдер дешевле... [1,2]

Да, согласна/согласен, важно понимать, за что Вы платите деньги. Дело в том, что предлагает своим клиентам различные тарифные планы, что позволит Вам выбрать наиболее оптимальный по цене. Тем самым Вы сможете рационально планировать свой семейный бюджет и при этом иметь круглосуточный доступ в Интернет. Давайте сравним ТП? Какая у вас скорость? Сколько каналов?

Возражения | База знаний

Назад Закрыть

Figure 4 - NAUMEN Contact Center Interface

The platform for the call center is based on freely distributed software: Linux OS, PostgreSQL or Oracle can be used as a DBMS[7]. Different operating systems are allowed in the workplace: Microsoft Windows, Linux, Mac OS.

1.5 **Problem statement**

The aim of this project is to develop an intelligent system of automation of the workplace online consultant, as well as the solution of a number of tasks.

Creating a system for processing customer requests from the Telegram vk messaging service:

- selection and implementation of website algorithm;
- creation of a modern and flexible interface for visualization of the dialogue graph and interaction with it;
- building an extensible web application architecture;
- implementation of interaction between the client application and the server application via WebSocket Protocol to work with the application in real time;
- integration with Web API of connection channels.

2 The choice of the methods and technologies for development of information system

The development of web portals, adapted for both desktop and mobile devices, is developing very actively. Many developers have realized that the use of REST-architecture has a number of advantages and can, in the future, avoid many problems, such as the translation of the application to another platform.

To successfully start the development need to take several important decisions. One of them will be choosing the right technology to create the application.

This Chapter describes the APIs and technologies that enable you to develop a web application with REST architecture, and provides an assessment to help you choose the appropriate API or technology. Design of REST you can see in figure 5.

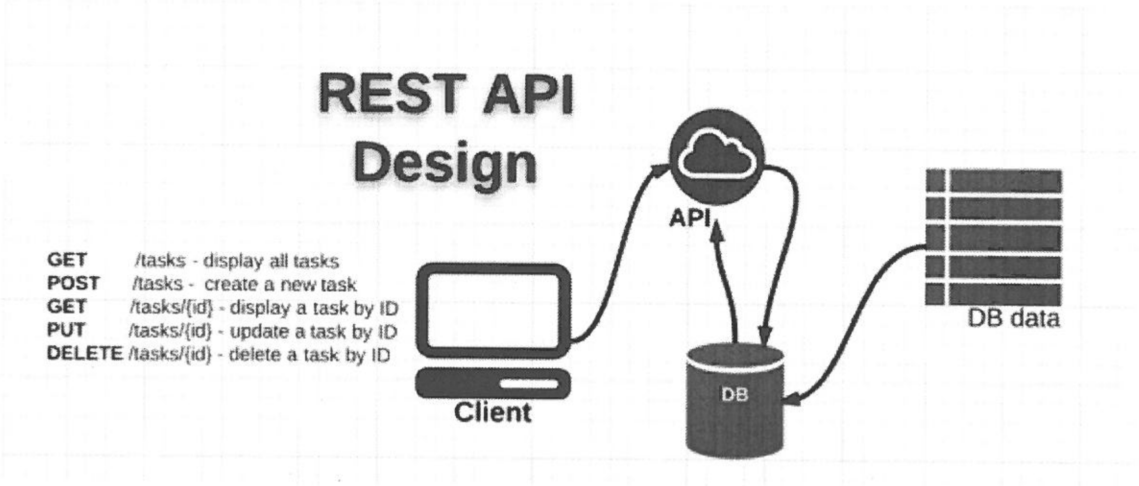

REST API
Design
GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID
Client
API
DB
DB data

Figure 5 – Design of Representational State Transfer

In the life of modern man firmly established mobile devices that use a variety of original and network applications for data aggregation. HTML5 and JavaScript have made a real revolution in the practice of Internet surfing. In such circumstances, the popularity of a new type of web services has increased significantly. These are web services that pass the view state, known as RESTful. With this in mind, many key players of the modern Network, such as Amazon, eBay, Google and Yahoo! abandoned previous web services in favor of RESTful resource - oriented services [8].

Representational State Transfer (REST) is a style of architecture based on the principles of the Network. Speaking about web services, can approved expect in this style of services are taken into account the nature of some features of the Network. In order to develop a web service with a presentation state, you must have a good understanding of the hypertext transfer Protocol (HTTP) and unique resource identifiers (URIs), and you must adhere to certain design principles. In essence, this means that each URI is a representation of a specific object. You can interact with this object using HTTP requests:

- GET - to get its contents;
- DELETE - deletes the contents of the;

- POST - to create content;
- PUT - to update content.

RESTful architectures quickly gained popularity, as they are based on a very reliable data transfer Protocol: HTTP. Web services with view state transfer reduce the rigidity of communication between the client and the server, which allows you to subsequently deploy an entire REST interface without disrupting the work of existing clients. Restful web services, like the Protocol on which they are based, do not retain States and can be affected by HTTP cache and proxy servers to make it easier to cope with high loads, and the system can easily scale [9]. Moreover, it is relatively easy to build such services, since no special tools are required (unlike, for example, WSDL).

2.1 Web application for front-end development

Single Page Application – abbreviated SPA, "single page application". In other words, SPA is a web application hosted on a single web page that loads all the necessary code along with the page itself to ensure operation. An application of this type appeared relatively recently, with the beginning of the HTML5 era. SPA is a typical representation of HTML5 applications [11].

If the application is quite complex and contains rich functionality, such as a distance learning portal, the number of files with scripts can reach several hundred or even thousands. To solve the problem of loading a large number of scripts in the SPA, an API called AMD is called. AMD implements the ability to load scripts on demand. That is, if three scripts are required for the start page of a single-page portal, they will be loaded immediately before the start of the program. And if the user has clicked on another page of the single-page portal, then AMD will load the script and markup only before going to that page.

The page of the site, which contains all links to all CSS, and links to scripts necessary for the SPA, is called "index.html." And the pages that the user switches within a single-page portal are called "modules"   lifecycle of SPA showed in 6 figure.

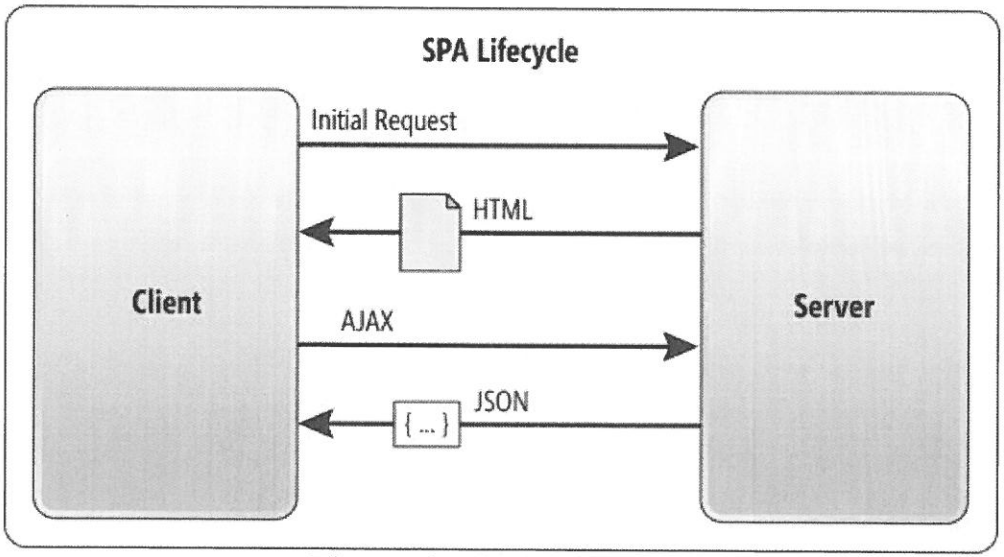


Figure 6 – SPA Lifecycle

2.2 ASP.NET Web Api platform for building REST service

With output ASP.NET Web API developers have the opportunity to quickly create REST-services in a convenient form, on the one hand fully implementing the principles of REST, and on the other using the full power of the platform ASP.NET.

The Web API tool is based on the addition to the application ASP.NET MVC Framework controller of a special kind. This type of controller, called an API controller, has two characteristics:

- action methods return model objects, not action result objects;
- action methods are selected based on the HTTP method used in the request.

Model objects returned by the API controller action method are encoded in JSON format and sent to the client.

API controllers are designed to deliver web data services, so they do not support views, layouts, or any other means that were used to generate HTML markup.

The inability of an API controller to generate HTML markup from the views is the reason why in single-page applications combine standard techniques ASP.NET MVC Framework with Web API. Infrastructure ASP.NET the MVC Framework performs the steps required to deliver HTML content to the user (including authentication, authorization, view selection, and visualization) [2]. Once the HTML content is delivered to the browser, Ajax requests generated by the JavaScript code inside will be processed by the Web API controller.

In conventional controllers, you can create action methods that return JSON data to support Ajax, but the API controller offers an alternative approach. This approach involves the separation of actions from running to data from the actions related with the performance of, and makes creating store apps Web API quick and simple [11].

The default route for the Web API has a static segment of the api, as well as the variable segments of the controller and id are optional. The key difference from the normal MVC route is that there is no action segment variable - this is where the behavior of the API controllers is formed.

When an application receives a request that corresponds to a Web API route, the action is determined based on the HTTP method used to issue the request

The ApiController class, which is the base class for Web API controllers, finds out the required controller based on the route and applies the HTTP method to find the appropriate action methods.

The API controller action method naming Convention provides that the action method name is prefixed with a supported HTTP method and includes a reference to the type of model with which the action method works. However, this is just a Convention because the Web API will ensure that any action method whose name contains the HTTP method used to execute the request is matched.

To decide which of these two action methods to choose, the controller looks at the arguments they accept and uses the routing variables to find the best match. In the case of a url like /api/courses, there are no routing variables except for the controller name, so a method that does not accept documents is selected.

When you request a URL of the form /api/ courses /3, a value is provided for the non - required ID segment variable, which will result in the selection of a method that takes the id argument. Other actions in the API controller is focused on other HTTP methods: POST, DELETE and PUT. This is the basis for the REST architecture which showed in figure 7 of the Web API tool, more commonly referred to as a service that supports REST when an operation is specified by combining a URL and the HTTP method used to request it.

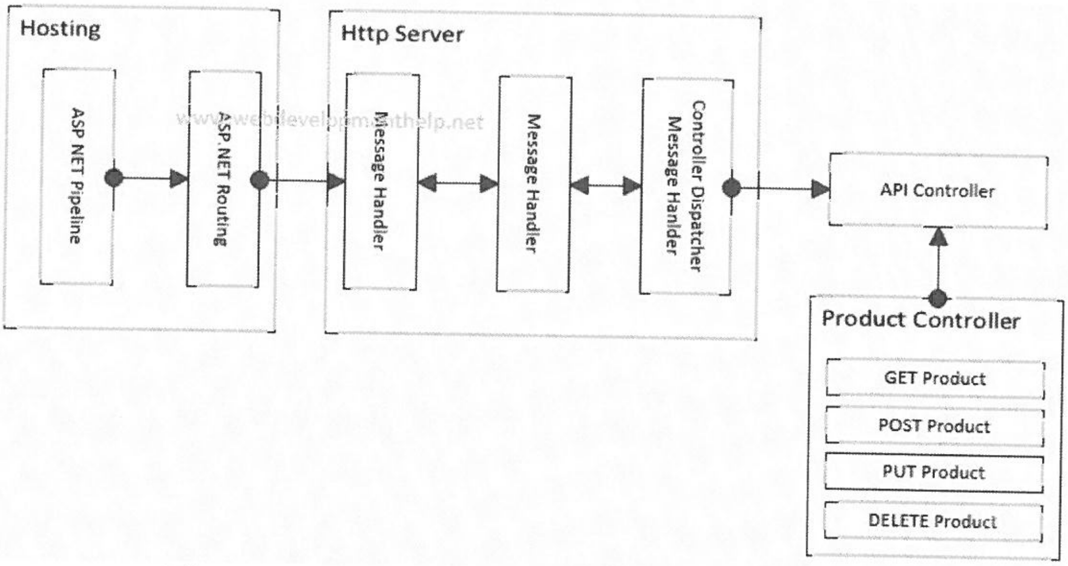


Figure 7 - ASP.NET WEB API processing architecture

Platform Web API ASP.NET allows you to easily create HTTP services for a range of clients, including browsers and mobile devices. The web API ASP.NET ideal for developing REST applications on the platform .NET Framework.

2.3 Framework for creating bootstrap web interfaces

Bootstrap - framework, a set of HTML+CSS tools and templates for layout and more efficient and fast creation of websites and web applications.

Bootstrap is a modern assistant, interface developers, designers and webmasters available for use under an open license. The framework is very dynamic and regularly updated, so not all its functions can be correctly supported by old browsers [12].

Bootstrap has the following templates:

- fonts;
- buttons;
- forms;
- labels;
- navigation;

- grid;
- JavaScript extensions.

The modern world is very dynamic, and web programmers often became - nod with a choice - to make quick prototypes and run them or slowly and long to redo the layout, bringing everything to an ideal state. Bootstrap perfectly implements the first approach.

Advantages of Bootstrap framework:

- time saving - achieved through the use of ready-made classes and design. This allows you to direct the saved energy and money to develop additional functionality;

- adaptability (high speed and optimization, standardization of interfaces) - dynamic layouts are qualitatively displayed on a variety of devices without the need to make changes to the layout;

- simplicity and openness — using Bootstrap is so easy that even novice web developers can cope with it, and open source code allows you to participate in the development, modify it for once or just use a good free solution.

At the same time, the HTML, JavaScript and CSS code in Bootstrap is thought out by hundreds of developers from all over the world — all in order for ordinary webmasters and webmasters to easily configure the site grid or embed the necessary elements into the interface.

Bootstrap also uses a dynamic style language LESS, which extends the capabilities of CSS: developers can control colors, create nested columns and variables [13].

The disadvantages of Bootstrap include:

- monotony - Bootstrap is often criticized for the fact that the sites look the same. All sites using Bootstrap are similar to each other, i.e. there is no uniqueness. Such twin sites are simply not remembered;

- inflexibility - if you want to create something different from the Bootstrap interface, you have to struggle with styles from the very first steps. In practice, it turns out a double work.

2.4 Choosing Web framework

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise JavaBeans (EJB) model. The Spring Framework is open source.

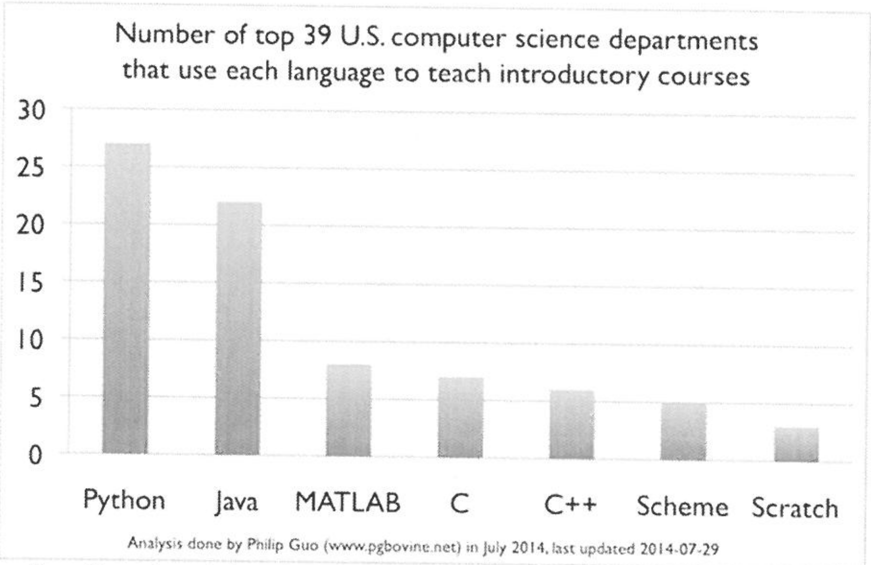
Objects created by the container are also called managed objects or beans. The container can be configured by loading XML (Extensible Markup Language) files or detecting specific Java annotations on configuration classes. These data sources contain the bean definitions that provide the information required to create the beans.

Objects can be obtained by means of either dependency lookup or dependency injection.[12] Dependency lookup is a pattern where a caller asks the container object for an object with a specific name or of a specific type. Dependency injection is a pattern where the container passes objects by name to other objects, via either constructors, properties, or factory methods.

In many cases one need not use the container when using other parts of the Spring Framework, although using it will likely make an application easier to configure and customize. The Spring container provides a consistent mechanism to configure applications and integrates with almost all Java environments, from small-scale applications to large enterprise applications.

For the development of the thesis, it was decided to use Python, namely the Django framework. Since it is written and intended for Python, this framework tries to follow the same philosophy. The developers sought to make it as simple as possible, straightforward, obvious, easily portable. Consider the main advantages of using Django. Python is probably the easiest programming language to learn. Thanks to the use of natural language constructs (paragraphs and indents) and simple syntax, understanding the structure of the program is significantly simplified compared to other popular programming languages. As proof of this, a large proportion of different programming courses in institutes and colleges use Python [14]. The figure below shows a graph showing how many of the top 39 schools in the United States use a particular programming language. Python is also universal. It is the basis of many server SOFTWARE as well as user programs on PC and Mac.

A comparison of the popularity of different programming languages in educational institutions is shown in figure 8.



Analysis done by Philip Guo (www.pgbovine.net) in July 2014, last updated 2014-07-29

Figure 8 - Graph of the popularity of YAP in educational institutions

The package includes many of the things that are often used in web development, the Django community has already taken care of, so starting to make a project you don't have to write everything from scratch, it will often be enough to set it up. Examples of such functionality are: application for administration, authentication, work with database, work with special pages (like user agreement), geolocation, session and cookie processing, etc.

I prefer dynamic languages for writing web applications and my dynamic language preference is Python. Python is both practical and beautiful programming language with huge ecosystem so you can find a library for anything.

There are good Python frameworks and two popular choices are Django and Flask. I chosed Django over Flask because of batteries included philosophy and some great Django libraries namely Django Rest Framework, Wagtail(Currently I am integrating wagtail to jeviz.com) and Django Channels. Plus:Django has LTS releases!

Here some comparison of Python/Django stack over Java/Spring stack:

- Python/Django feels quicker and lighter than Java/Spring stack which result in productivity boost;
- Django ORM has the least ceremony and more productive;
- Java technologies servlet, beans, XML configurations etc have more mental overhead which is hard for development;
- Django deployment far more easier;
- For MVP project I recommend Django;
- For enterprise (next banking, ERP so on...) , multi programmer web apps use Java/Spring;

Because of all above reasons and mainly for productivity boost I used Django for intellectual chatbot system and I am happy with my decision.

2.5 Selection the technology for website structure

The concept of MVC (Model-View-Controller: model-view-controller) is often mentioned in web programming in recent years.

MVC is not a project template, it is a design template that describes how to build an application structure, the scope of responsibility, and the interaction of each part in a given structure.

The concept of MVC was first described in 1979, but for a different concept, because there was no concept of a web application. The template used today is an adaptation of the original template to web development.

The popularity of this structure in web applications has developed thanks to its inclusion in two development environments, which have become very popular: Struts and Ruby on Rails. These two frameworks have outlined a development path for hundreds of working environments that are created later.

The idea behind the MVC design pattern is very simple: you need to clearly share responsibility for the different functions in your applications:

- model - data processing and application logic;
- view - presenting data to the user in any supported format;
- controller - processing of user requests and calls appropriate resources.

The application is divided into three main components, each of which is responsible for different tasks.

Controller manages user requests (received in the form of HTTP GET or POST requests when the user clicks on interface elements to perform various actions). Its main function is to call and coordinate the action of the necessary resources and objects needed to perform the actions specified by the user. Typically, the controller calls the appropriate model for the task and selects the appropriate view.

A model is data and rules that are used to work with data that represents the concept of application management. In any application, the entire structure is modeled as data that is processed in a certain way.

The model contains the most important part of the application logic that solves the problem that the user is dealing with (forum, store, Bank, and the like). The controller contains mostly organizational logic for the application itself.

The view provides a variety of ways to represent the data that is derived from the model. It can be a template that is populated with data. There can be several different types, and the controller chooses which one is best suited for the current situation.

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of DispatcherServlet. Here, DispatcherServlet is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.The block diagram of the MVC is shown in figure 9.

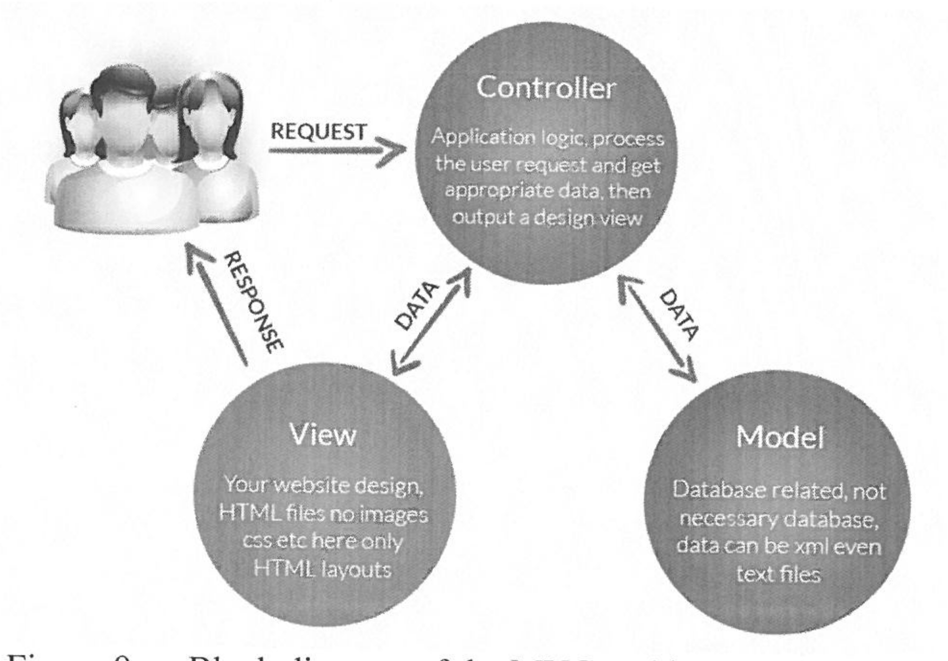

REQUEST
Controller
Application logic, process the user request and get appropriate data, then output a design view
RESPONSE
DATA
DATA
View
Your website design, HTML files no images css etc here only HTML layouts
Model
Database related, not necessary database, data can be xml even text files

Figure 9 — Block diagram of the MVC architectural template

In addition to isolating views from application logic, the MVC concept significantly reduces the complexity of large applications. Code is much more structured, and thus easier to support, test, and reuse solutions.

Spring Web MVC is the original web framework built on the Servlet API and has been included in the Spring Framework from the very beginning. The formal name, "Spring Web MVC," comes from the name of its source module (spring-webmvc), but it is more commonly known as "Spring MVC". Parallel to Spring Web MVC, Spring Framework 5.0 introduced a reactive-stack web framework whose name, "Spring WebFlux," is also based on its source module (spring-webflux). This section covers Spring Web MVC. Spring MVC, as many other web frameworks, is designed around the front controller pattern where a central Servlet, the DispatcherServlet, provides a shared algorithm for request processing, while actual work is performed by configurable delegate components. This model is flexible and supports diverse workflows.

The DispatcherServlet, as any Servlet, needs to be declared and mapped according to the Servlet specification by using Java configuration or in web.xml. In turn, the DispatcherServlet uses Spring configuration to discover the delegate components it needs for request mapping, view resolution, exception handling, and more.

The following example of the Java configuration registers and initializes the DispatcherServlet

2.6 Selection of Database management systems

Database management systems or DBMS is a general term that refers to all possible solutions (i.e. computer programs or integrated libraries), and often works very differently. These applications manage or assist with the management of information packages. As the information received can vary in size and shape, many databases have been developed.

The relational model was introduced into the 1970s system. It provides a mathematical way the very structure of data storage and use, as well as the expansion of old concepts such as a flat model, a network of models through communication service providers offer the advantages of Automatic Data Storage limited collections grupero, by which the table that contains information in a structured way (e.g. name and address) connect all input data, Shiva values of different attributes [15].

Although the relational model is flexible, very strong, there are several problems that cannot be solved.

More recently, a number of solutions called nosql have gained popularity. This was done because of the promise to solve problems where relational databases failed, as well as because of some additional functions. By eliminating the data structure to store method descriptions in the re-population model databases, this work offers a much freer way to work with the data, thus providing ease and flexibility. By using this unstructured (or on-the-fly) approach, NoSQL solutions seeks to eliminate the restrictions imposed by strong relationships and offers many different ways to store and use data for specific use cases. Although there are many different databases, there are only a few of the solutions to each period that quickly become popular and remain so for a long time. The last decades of unconditional popularity

The DBMS takes its relational name from the model it implements, which is relational, as previously mentioned. Today, for the foreseeable future, it is the most common option for reliable, efficient and secure data storage.

These databases require a well-defined set of data acceptance plans. These schemes are user-specific and would constitute the way data is stored and used.

To develop the project was chosen SQLite solid integrated database was selected. It's a self-sufficient library, connecting the emo to the application it's going to use. Provides sqlite with a good set of the tools for work with any type of information with simplicity and fewer restrictions of the Server databases. When an application asks about integration processes with competence and direct to the file that contains the data instead of communicating through different interfaces (ports, sockets). This makes SQLite very fast and efficient.

2.7 Interaction between server and client via Websocket

It is difficult to imagine modern web applications without a web site, when being on the same Internet resource, you can spend hours not to update the web page, however, the information on it will be constantly updated. Since the main purpose of the development was to create an application to work in real time, it is impossible not to resort to this technology.

Communication between the client and the server takes place through the transmission of links via websockets. There are the following types of messages:

- "Connect response". Sent from the server to the client when the connection is fixed. Contains the string "Connected";
- "Update json". The packet with this event sends a message that the client has asked a question. The package passes the question itself, as well as the expected answers from the server. Transmitted from the server to the operator-consultant;
- "Receive answer". The packet with this event sends a message that the operator has chosen the appropriate response. The selected response is transmitted in the packet;
- "Finish dialog". A packet with this event sends a message that the client has finished the conversation;
- "Disconnect." Comes from the client to the server at the time of disconnection of the first. Contains no data.

Packets are exchanged through a dedicated namespace "/socket" in order to avoid noise in the common namespace.

2.8 API integration with connection channels

2.8.1 VK Web API

To work with all API methods, You need to pass a special access_token in the request. It is a string of Latin letters and numbers and can correspond to an individual user, community, or your application itself.

VK API is an interface that allows you to get information from the database vk.com using http requests to a special server. You do not need to know in detail how the database is arranged, from which tables and fields of what types it consists — it is enough that the API request "knows"about it. The syntax of the queries and the type of data they return are strictly defined on the side of the service itself.

Consider separately all its components:

- connection Protocol;
- API service address;
- VK API method.

It is a string of Latin letters and numbers and can correspond to an individual user, community, or your application itself.

2.8.2 Telagram Web API

The Bot API is an HTTP interface for working with bots in Telegram. Each bot is a special account created for automatic processing and sending of messages. There are two logically opposite ways of receiving updates from the bot:

- long pulling the application automatically polls the serve Telegram for any updates to the bot. Default 100ms;
- webhook – server inform themselves of Telegram app on the server as soon as there are any updates.

Incoming updates will be stored on the server until they are processed, but not longer than 24 hours. Regardless of the method of obtaining in response, an Update object serialized in JSON is sent [16].

In order to get a token, you need to write a special bot @BotFather. Examples of methods available for the API are described below:

- getUpdates – this method is used to get software updates long polling technologies;
- setWebhook – the method binds to the bot the url of the domain where it contains running the bot;
- sendMessage – the method sends a text message to the client Telegram;
- sendLocation method sends a message with coordinates in telegram client;
- getFile – the method returns the downloaded file by its name, etc.

Post and GET requests are allowed. To pass parameters to Bot API there are 4 ways:

- query in the URL;
- application/x-www-form-urlencoded;
- application/json ;
- multipart/form-data.

3 Development of website with intellectual chatbot

3.1 Django MVC website development

For Django projects, set a directory on hard disk. For example, whether it's a directory C: \ django In the console, use the cd command to access this directory.

When installing Django, the script is installed in the django-admin.py virtual environment folder! And under Windows, the executable file django-admin.exe. They are in the folder of the virtual environment where Django was installed: on Windows - in the Scripts subdirectory.

Django-admin provides a number of commands to manage Django project. In particular, the startproject command is used to create a project. The project name is passed as an argument to this command.

Therefore, first enable the previously created virtual environment (for example, the Hello environment created in the previous topic, if it was not already enabled), and then run the following command:

- django-admin startproject hello;
- django-admin startproject.

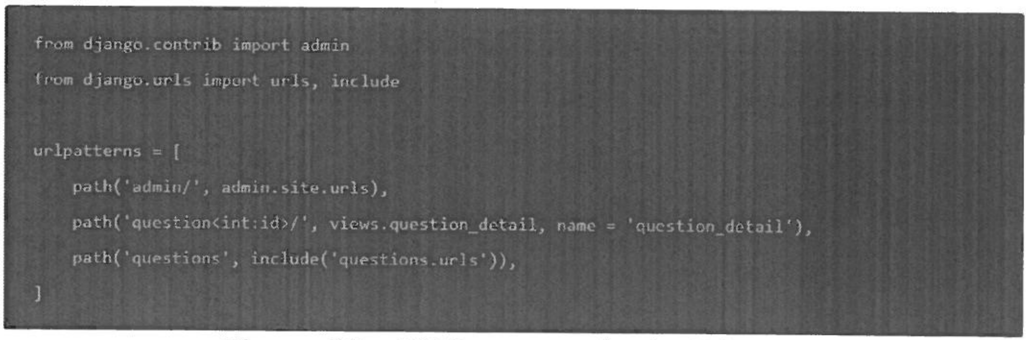
Now create a hello project. The project name does not have same as name of virtual environment.After executing this command, the hello subdirectory will be created in the current folder.

The project will include the following elements:

manage.py: runs various project teams, for example, launches the application:

- init__.py: This file indicates that the folder in which it is will be treated as a module. This is a standard file for Python;
- settings.py: contains project configuration settings;
- urls.py: contains the URL templates, essentially defines the project's routing ;
- wsgi.py: Contains WSGI (Web Server Gateway Inerface) configuration properties it is used when deploy a project.

A URL mapper in the Django project is urls.py. In the below example, the mapper called as url patterns defines different types of url patterns and corresponding view function. If an HTTP request is received that has a URL matching in the url patterns invokes the corresponding view function and passes that request. It show in figure 10.



```
from django.contrib import admin
from django.urls import urls, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('question<int:id>/', views.question_detail, name = 'question_detail'),
    path('questions', include('questions.urls')),
]
```

Figure 10 - URL mapper in the Django

The urlpatterns object is a Python list with items path().The first argument that the path() takes is the pattern(Route) that will be matched.

path() methods use angle brackets (<>) to capture the data from the URL and passes to the view function as a named argument(s). For example, if have10 question and the url looks like this 'questions/1', then the <> takes the id, i.e., 1 from the URL to display corresponding content in the pages. Will see this in detail in coming Voting App Tutorial With Django.

The second argument is another function that invokes corresponding functions when the URL is matched. The notation views.question_detail indicates that the function is called question_detail(). This can be found in the views module (views.py).

A third argument name is an option which can be used to shorten the URL.

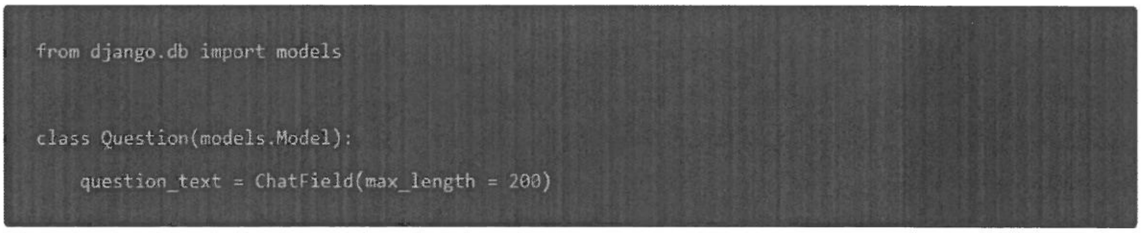
Handling the requests (views.py) views are the heart of the Django applications. They are responsible for the HTTP requests and HTTP responses. In between they perform some activities like database operations, rendering HTML templates, etc.,

The example below shows a minimal view function index(). It receives the HttpRequest as a parameter request and returns the HttpResponse object. In this example we don't do anything with a request, we simply return a string.

Views are usually stored in a file called views.py. Creating data models {models.py}.The code below shows how to create a simple model in Django. Querying Data {Views.py}

Django provides a simple Data query API for the database.

The code is shown below in figure 11, how to select all the questions from the Question model .



```
from django.db import models


class Question(models.Model):
    question_text = ChatField(max_length = 200)
```

Figure 11 – Data query of question model

The function render() is the shortcut method to create a HttpResponse that is sent back to the browser. It creates an HTML file by combining the corresponding template and data we inserted, i.e., the variable 'all_questions'. It inserts the data that we provided in the HTML file.

Templates allow you to specify the structure of the output document. Templates are often used to create HTML documents. Also create other types of documents also if needed like CSS, JS, XML, JSON, etc.., Django has its own template system. Also use the popular Python library called Jinja2 for template system.

The code shown below is the HTML template file. It is called by the function render().

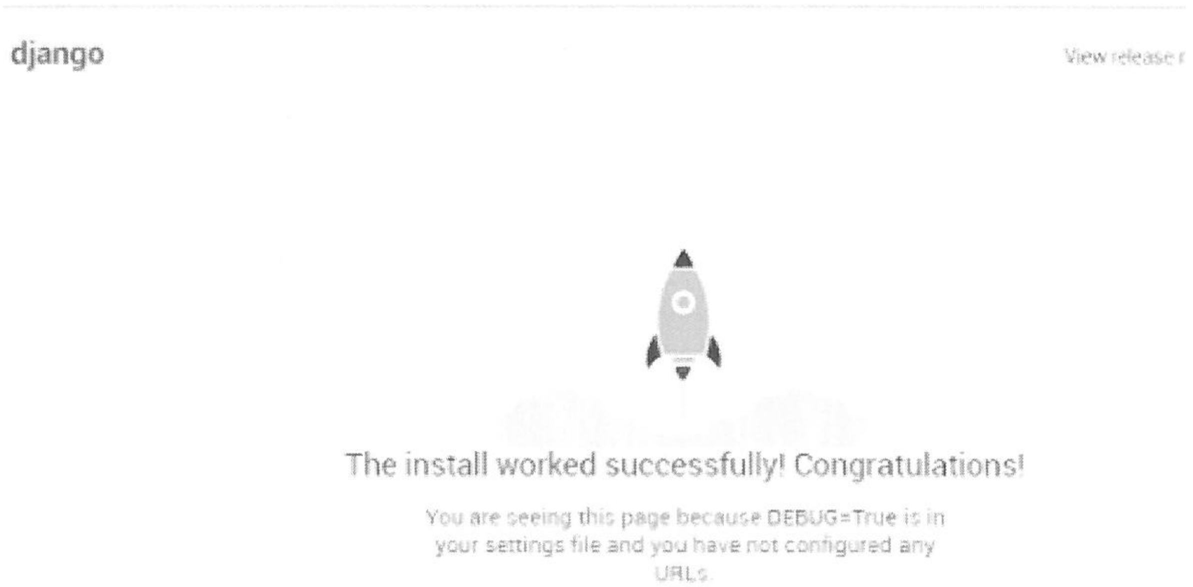
Write the python code in between {% %} pounds, so that Django recognizes that it is a Python code. And to display the variable in HTML template. Write a variable that it has to display in between 2 curly braces.

It will display all the question(s) that are present in questions list in HTML template.Till now you have learned some keywords and methods in the Django framework. Now, let's learn by creating a simple Voting app in Django.

Run the Django server to check whether it's working or not. To run the server go to the outer first_site directory and run the following command. It shows the following output in the command line.

Run 'python manage.py migrate' to apply them. Started the Django server. Django server was written in Python. Let's check it by visiting the url http://127.0.0.1:8000/ or http://localhost:8000/ in browser.

It shows the following page in your browser in figure 12.



The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

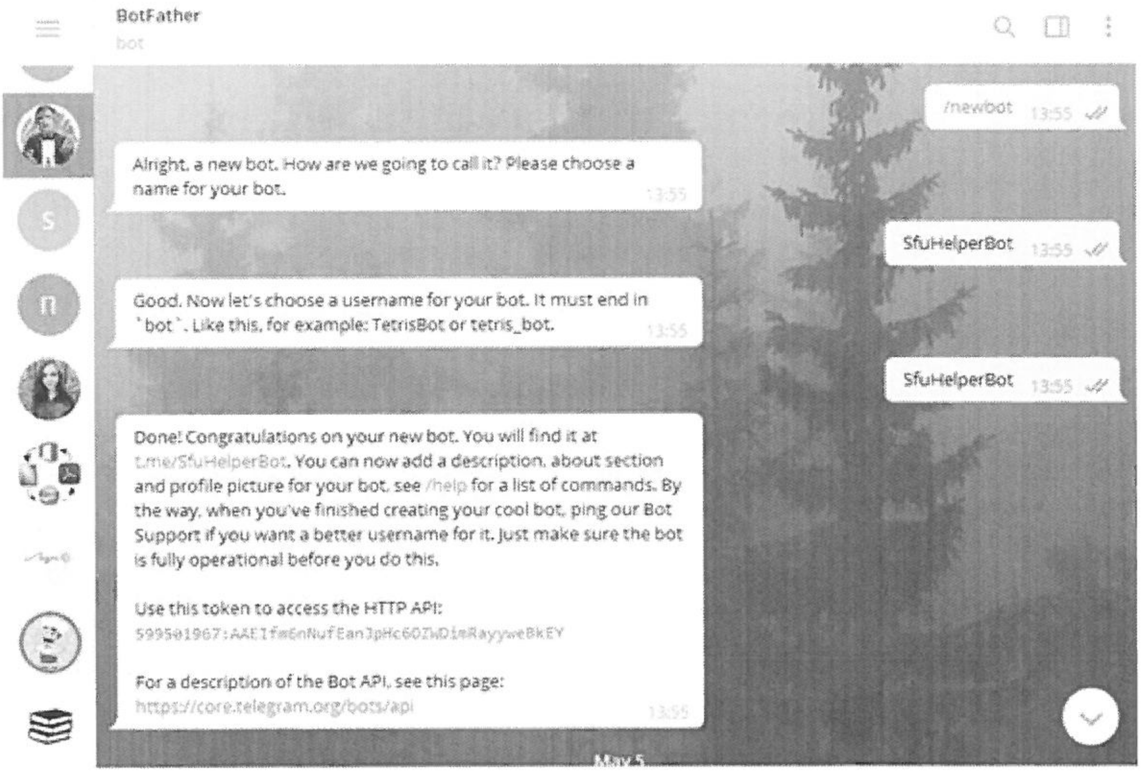
Figure 12 – First Django server page

3.2 Registration chat bot Bot for Telegram API

The first step in application development is to register to the special chat bot "BotFather". Registration starts with the command "/newbot", after which prompted to enter the name of the chatbot with a prerequisite: at the end of the name should be specified "Bot" or "_bot". If all conditions have been satisfied, the "BotFather" issues a token (a special set of symbols to access the HTTP API Telegram Bot) and the URL to access the chat bot. An example of chatbot registration is shown in figure 13. Create a list of available commands:

- /setname Changes an existing name;

- /setdescription Assigns the text to be displayed when you first open a chat;
- /setabouttext Assigns the text in the "About the chat bot»;
- /setuserpic Assigns the selected picture;
- /setcommands Allows;
- /deletebot Removes the selected chat bot.



/newbot 13:55

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 13:55

SfuHelperBot 13:55

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 13:55

SfuHelperBot 13:55

Done! Congratulations on your new bot. You will find it at t.me/SfuHelperBot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
599501967:AAEIfm6nNufEan3pHc6OZwDimRayyweBkEY

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 13:55

Figure 13 – Example of registering a chat bot

In addition to commands to change the basic parameters of the chatbot, there are a number of commands that allow you to display immutable parameters.

Command Descriptions:

- /token returns the previously received token fromselected bot;
- /setinlinegeo enables or disables the ability to transfer the bot's location from another chat's;
- /setinlinefeedback allows you to get information about the number of selected users teams;
- processes and sends back information separately for each user in the chat.

After the settings on the Telegram side and receiving the token, you can start developing the software part of the chatbot.

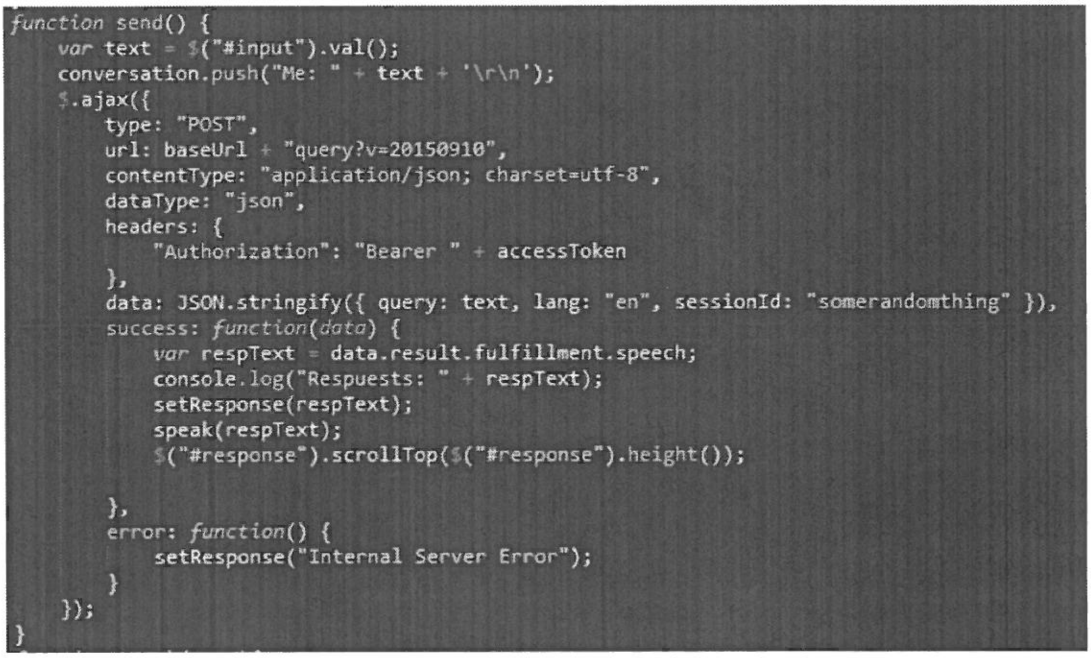
Telegram users can interact with the chat-bots in 2 ways: command ("/start", "/new", "/location" and other) settings, or the built-in keyboard (inline keyboards). For the convenience of users, it was decided to make an interface with a built-in keyboard.

Five different menus are required to achieve these goals:

- language selection menu;
- main menu;
- selection menu by date;
- sport selection menu;
- menu for selecting locations of sports facilities.

In Telegram messenger, each keyboard is implemented as an object, and its buttons as a JSON string. The Telegram Bot API documentation says that each keyboard must have one mandatory parameter – the name of the button (text), and six optional – link (url), return data (callback_data), the ability of the built-in query (switch_inline_query), the ability to output the keyboard from another chat (switch_inline_query_current_chat), call the description of the running game (callback_game) and the button with the ability to buy (pay).

An example of the keyboard implementation of one of the chatbot menus is shown in figure 14.



```
function send() {
    var text = $("#input").val();
    conversation.push("Me: " + text + '\r\n');
    $.ajax({
        type: "POST",
        url: baseUrl + "query?v=20150910",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            "Authorization": "Bearer " + accessToken
        },
        data: JSON.stringify({ query: text, lang: "en", sessionId: "somerandomthing" }),
        success: function(data) {
            var respText = data.result.fulfillment.speech;
            console.log("Respuests: " + respText);
            setResponse(respText);
            speak(respText);
            $("#response").scrollTop($("#response").height());

        },
        error: function() {
            setResponse("Internal Server Error");
        }
    });
}
```

Figure 14 -Implementation of one of the chatbot menus

3.3 Used libraries for developing chatbot

For developing intellectual chatbot system we used python libraries for different purposes:

- tensorFlow;
- scrapy;
- pattern;
- bokeh.

TensorFlow is an open source library for fast numerical computing.

It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API.

Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems, not least RankBrain in Google search and the fun DeepDream project.

It can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines.

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017.[11] While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as *tensors*. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google[16].

Scrapy is a fast, open-source web crawling framework written in Python, used to extract the data from the web page with the help of selectors based on XPath. In other words, the Scrapy framework provides a set of Python scripts that contain most of the code required to use Python for web scraping. We need only to add the last bit of code required to tell Python what pages to visit, what information to extract from those pages, and what to do with it. Scrapy also comes with a set of scripts to setup a new project and to control thhe scrapers that we will create.

It also means that Scrapy doesn't work on its own. It requires a working Python installation (Python 2.7 and higher or 3.4 and higher - it should work in both Python 2 and 3), and a series of libraries to work. If install Scrapy as suggested there, it should take care to install all required libraries as well.

Pattern is a web mining module for the Python programming language. It has tools for data mining (Google, Twitter and Wikipedia API, a web crawler, a HTML DOM parser), natural language processing (part-of-speech taggers, n-gram search, sentiment analysis, WordNet), machine learning (vector space model, clustering, SVM), network analysis and <canvas> visualization.

Bokeh is a Python library for interactive visualization that targets web browsers for representation. This is the core difference between Bokeh and other visualization libraries. Look at the snapshot below, which explains the process flow of how Bokeh helps to present data to a web browser. These bindings produce a JSON file, which works as an input for BokehJS (a Javascript library), which in turn presents data to the modern web browsers.

Bokeh can produce elegant and interactive visualization like D3.js with high-performance interactivity over very large or streaming datasets. Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

Flow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017.[11] While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

CONCLUSION

As a result of the work, various technologies for workplace automation of specialists working with clients, in particular, online consultants, were investigated. According to the results of the study we can say that even with the rapid modern development of intelligent systems, their role in software for consulting is reduced to zero. This is partly due to the fact that consulting needs to be done in real time, and most of the algorithms for machine learning require a large amount of time for preprocessing and training itself. Thus, the aim of the work was to create a web application for online consultants, the analyzing part of which could suggest possible ways of developing dialogue in real time. Moreover, it was necessary to provide for the possibility of additional training in the course of work to improve the quality of the proposed answers, and also to handle a large number of special situations.

It is a high standard and provides reliable and fast responses to users compared to those of traditional methods. The average time spent interacting with a chatbot is very short, as it provides an efficient way for users to make quick purchases and learn information. The low interaction time reflects the high rates of comprehension and speech recognition offered by adopting conversational user interfaces, allowing users to freely interact with the chatbot to meet the needs of modern life. Chatbot has proven to meet the demand of users wanting instant access and availability of information and services.

As a result of the development, the goal was fully achieved. And, as a result, the final product was a full-fledged web application for online consultants, which has a high response rate and visual visualization of the dialogue process.

REFERENCES

1 Mitchell S., Walter S., seven D. ASP.NET: tips, recommendations, examples. [Text], M.: Williams, 2002 – 863 p.

2 Search engine optimization on ASP.NET for professionals. The developer's guide to SEO / Christian Dari , Jamie Sirovich. - M.: Dialectics, 2008. - 400 p. (date accessed: 17.04.2019).

3 Messina, C. 2016 will be the year of conversational commerce [Electronic resource] / C. Messina // Medium. – 2016. – Access mode:https://medium.com/chris-messina/2016-will-be-the-year-of-conversationalcommerce- (date accessed: 19.03.2019).

4 Telegram Bot API [Electronic resource] : Telegram Documents. – Mode of access: https://core.telegram.org/bots/api

5 Agaltsov, B. Database: In 2 books. Book 2: Distributed and remote databases : Textbook / V. Agaltsov – Moscow : Forum, 2018. – 271 p. (date accessed: 12.04.2019).

6 Archakov, D. NodeJS: Make the buttons in the Telegram API [Electronic resource] / D. Arkhakov // Blog about programming. – 2016. – Access mode: https://archakov.im/post/nodejs-make-buttons-on-telegramapi html (date accessed: 12.04.2019).

7 Ivanov A. D., the Chat bot in the Telegram and VK as the new channel distribution of news. [Electronic resource] / A. D. Ivanov // Volzhsky (date accessed: 17.04.2019).

8 University named after V. N. Tatishcheva. – 2016. – №3 – p. 126-132. – Access mode: https://elibrary.ru/download/elibrary_26673675_34058358.pdf (date accessed: 29.04.2019).

9 Kozlov A. A., Telegram bot as a simple and convenient way obtaining information [Electronic resource] / A. A. Kozlov, A. V. Batishchev // The territory of science. – 2017. – №5. – p. 55-64. – Access mode: https://cyberleninka.ru/article/v/telegram-bot(date accessed: 27.03.2019).

10 Matveeva N. Yu., Technologies of creation and application of chatbots [Electronic resource] / N. Yu Matveeva, A. Zolotaryuk . // Scientific notes (date accessed: 21.04.2019).

11 Why Redis beats Memcached for caching [Electronic resource], Infoworld. - URL: http://www-в InfoWorld. com / article / 3063161/application development/why-direct-beats-like Memcached for caching.html (date accessed: 10.04.2019).

12 Machine learning algorithm cheat sheet for Microsoft Azure Machine Learning Studio [Electronic resource], Microsoft. - - URL: https://docs.microsoft.com/en-us/azure/machine-learning / machine learning-algorithms-cheat sheet (accessed: 15.04.2019).

13 10 Interesting Telegram stats [Electronic resource], expanded ramblings. — URL: expandedramblings.com/index ahhh! php/telegram-stats/ (accessed 25.04.2017).

14 VK Slack stats [Electronic resource], expandedramblings. — URL: http://expandedramblings.com/ index.php/block-statistics/ (accessed 27.04.2017).

15 Search engine optimization on ASP.NET for professionals. The developer's guide to SEO / Christian Dari , Jamie Sirovich. - M.: Dialectics, 2008. - 400 p. (date accessed: 17.04.2019).

16 Programming using Microsoft ASP.NET 4. Web application development using ASP.NET and AJAX / Dino Esposito. - M.: Peter, 2013. - 919 p. (date accessed: 14.04.2019).

17 Roy Fielding's dissertation "transfer of representative state (REST)". [2017 2017] Chapter 5. Updated: 07.04.2017. URL: http://pc-information-guide.ru/periferiya/vidy-i-ustrojstvo-kompyuternyx-myshej.html (accessed 3.05.2019).

18 REST in Practice: Hypermedia and Systems Architecture/ Jim Webber, Savas Parastatidis, Ian Robinson - "O'Reilly Media, Inc.", 2010

19 Korobov Mikhail. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts / Ed. by Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko et al. — Springer International Publishing, 2015. — Vol. 542 of Communications in Computer and Information Science. — P. 320–332. — URL: http://dx.doi.org/10.1007/ 978-3-319-26123-2_31. (accessed 13.04.2019).

20 Semantic Web Technologies: Trends and Research in Ontology-based Systems /John Davies, Rudi Studer - Peter publish, 2015. - 418 p. (date accessed: 24.04.2019).

Application

Listing of run.py

```
from flask import Flask , render_ template , json
from flask_ socketio import SocketIO , emit
from main import preprocessing from telebot import types , TeleBot from telebot. util
import async import _ thread
import redis
from functools import reduce import operator
import config
# Создание объекта приложения Flask
app = Flask (__ name__ )
# Создание объекта SocketIO
socketio = SocketIO ( app )
# Создание объекта TeleBot
bot = TeleBot( config . token )
# Массив для хранения сообщений диалога
answer_ array = []
# Словарь для хранения json представления графа диалога
data = {}
def     getFrom Dict ( dataDict , map List):
"""Функция получения ветви в json по массиву значений в каждом у зле
Keyword arguments :
dataDict -- json дерево mapList -- массив значений"""
return reduce( operator. getitem , mapList , dataDict)
def
setIn Dict( dataDict , mapList , value):
"""Функция задания значения ветви в json по массиву значений в к аждом узле
Keyword arguments :
dataDict -- json дерево mapList -- массив значений value -- новое значение
""""
def     getFrom Dict ( dataDict , map List [: -1])[ map List [ -1]] = value
verbose_ to_ compact ( verbose):
"""Функция для преобразования подробного представления json в ко мпактное
Keyword arguments :

verbose         подробный json
""""
return { item [' title ']: verbose_ to_ compact ( item [' properties '])
for item in verbose }
def compact_ to_ verbose ( compact):
"""Функция для преобразования компактного представления json в подробное
Keyword arguments :
compact         компактный json

""""
return [{ ' title ': key , ' properties ':
compact_ to_ verbose ( value)}
for key , value in compact. items ()]
def update_ json ( answers , client_ id ):
"""Функция для обновления объекта графа и id клиента
Keyword arguments :
```

Continue of application

```
answers          массив диалога
client_ id       id клиента
"""
global current_ chat_ id
current_ chat_ id = client_ id
result  =  preprocessing ( answers)
branch  =  {' title ':  answers [-1],
' properties ':
[{ ' title ': key , ' properties ': []} for key in
result]

}

i f  len ( answer_ array )  ==  1: get_ view . data  =  branch
else :
new  =  verbose_ to_ compact ([ get_ view . data ]) new_ branch  =  verbose_ to_ compact ([
branch ]) setIn Dict( new ,  answer_ array [:-1],  new_ branch ) get_ view . data  =  compact_ to_
verbose ( new )[0]
socketio . emit(' update  json ',  json . dumps( get_ view . data ,  indent =2 , separators =(',  ',  ':
')),  namespace ='/ socket ')

# ВЫВОД ЛОГОВ
print ( bot. get_ me ())

def  log ( message ,  answer):
"""Функция−лог гер
Keyword arguments :
message −− объект полученного сообщения
emit(' Connect  response ',  {' data ':  ' Connected '})
@ socketio . on(' disconnect ',  namespace ='/ socket ')
def handler_ disconnect ():
"""Функция обработчика отключения клиента"""
print (' Client  disconnected ')
@ socketio . on(' receive  answer ',  namespace ='/ socket ')
def get_ javascript_ data ( text):
"""Функция обработчика полученного ответа с веб−клиента
Keyword arguments :
text −− текст сообщения
"""
answer_ array . append ( text)
r_ server  =  redis. StrictRedis (' localhost ',  charset=" utf -8 ", decode_ responses = True)
qty = r_ server. dbsize ()
for  ans  in  answer_ array :
r_ server. lpush (' qa_ ' + str ( qty + 1), ans) tg_ send ( text)
@ async ()
def tg_ send ( text):
"""Функция асинхронной отправки сообщения
```

Continue of application

```
Keyword arguments :
text -- текст сообщения
bot. send_ message ( current_ chat_ id , text)
@app . route('/ viewer/')
def  get_ view ():
"""Функция рендеринга начального экрана
"""
get_ view . data  =  {
" title": " Ожидание диалога"
}
return  render_ template (' index . html ',
context= json . dumps( get_ view . data ,  indent
=2 ,  separators =(',  ',  ':  ')))
@bot. message_ handler ( commands =[' start '])
def  handle_ start ( message):
"""Функция обработчика запуска бота
Keyword arguments :
message -- объект сообщения
"""
answer  =  " Начало  диалога" log ( message ,  answer)
keyboard  =  types. Reply Keyboard Markup ( True ,  False) keyboard . add (*[ types. Keyboard
Button ('Начать  диалог')]) bot. send_ message ( message. chat.id ,
"""Здравствуйте , вас приветствует система консу льтации.""",
reply_ markup = keyboard , parse_ mode =" Markdown ")
@bot. message_ handler ( func=lambda  message:  message. text  ==  'Начать  ди алог',
content_ types =[' text '])
def  handle_ begin ( message):
"""Функция обработчика сообщения "Начать диалог"

Keyword arguments :
message -- объект сообщения
"""
keyboard  =  types. Reply Keyboard Remove () answer  =  """Что  вас  интере сует?"""  bot.
send_ message ( message. chat.id ,
answer ,          reply_ markup = keyboard , parse_ mode =" Markdown ")

log ( message ,  answer)

@bot. message_ handler ( func=lambda  message:  message. text  !=  '', content_ types =[' text '])
def  handle_ text ( message):
"""Функция обработчика сообщенияKeyword arguments :
message -- объект сообщения"""
answer_ array . append ( message. text) answer  =  """Пожалуйста ,  подождите ."""
bot. send_ message ( message. chat.id ,  answer)log ( message ,  answer)update_ json ( answer_
```

Continue of application

```
array ,   message. chat. id )
bot_ thread ():
"""Функция для запуска треда бота"""
bot. polling ( none_ stop = True)
i f   __ name__   ==   "__ main__ ":
_ thread . start_ new_ thread ( bot_thread , ())
socketio . run ( app )
```