

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, Data Storage and Processing

Mussabay Dariga

Development of the HR-management portal for office automation of the company

**DIPLOMA WORK**

specialty 5B070300 – «Information systems»

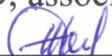
Almaty 2019

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, Data Storage and Processing

**ADMITTED TO DEFENSE**  
Head of Department Cybersecurity  
Data Storage and Processing,  
PhD, associate professor  
 N.A. Seilova  
«13» 05 2019.

**DIPLOMA WORK**

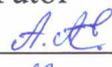
Theme: «Development of HR-management portal for office automation of the  
company»

specialty 5B070300 – Information systems

Performed

Mussabay Dariga

Reviewer,  
PhD, professor of AlmaU  
 T.I. Bakibayev  
«13» 05 2019.

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



Almaty 2019

THE MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF  
KAZAKHSTAN

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

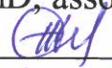
Institute of Information and Telecommunication Technologies

Department of Cybersecurity, Data Storage and Processing

5B070300 – Information system

**AFFIRM**

Head of Department Cybersecurity  
Data Storage and Processing,  
PhD, associate professor

 N.A. Seilova

« 13 » 05 2019.

**TASK**

**to perform the Diploma work**

Student Mussabay Dariga

Theme: Development of HR-management portal for office automation of the company

Approved by the order of the University № 1167 from « 16 » 10 2019

Deadline for completion of work « 13 » 05 2019

Source data to diploma work: HR-management portals, projects of office automation and company workflow, the results of pre-diploma practical work and the literature review, based on theoretical data.

The list of subject to the development of the thesis or a summary of its content:

- a) the development of the HR-management portal (website);
- b) the development of office automation;
- c) the creation of effective Electronic Document Management System.

The list of graphic material presented 15 slides presentation work

Recommended main literature: 20 sources

**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 normocontrol to complete a thesis indicating the related sections of the Diploma work

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

Scientific adviser

 Azhenov A. T.

Task was accepted for execution by the student

 Mussabay D.E.

Date

« 13 » 05 2019

**REVIEW**

**of Diploma work**  
*(name of the type work)*

**Mussabay Dariga**  
*(student full name)*

**Specialty 5B070300 – «Information systems»**

**Theme: Development of HR-management portal for office automation of the company**

Performed:

- a) software part on 14 pages  
b) the volume of work 30 pages

This diploma work is dedicated to the development of HR-management portal for office automation and document flow management of the company.

The submitted diploma work consists of introduction, main parts, and conclusion. The main parts are focused on general principles of HR-management and Electronic Document Management System.

When constructing the system, an object-oriented approach was used, which is a further development of the systems approach to software design. A detailed analysis of the advantages and disadvantages of various implementation options has been carried out and the relevance and necessity of creating this type of EDMS has been substantiated.

The advantages of the estimated diploma work include the excellent logic of constructing the text, the use of foreign subject literature, the practical significance of the findings. In the diploma work was not revealed gross mistakes. The work was performed in accordance with the requirements.

**Performance evaluation**

Diploma work is executed on "excellent" (90%), and its author deserves the academic degree "bachelor" on specialty 5B070300 – «Information systems».

**Reviewer**

PhD, professor of AlmaU

T.I. Bakibayev

« 17 » 2019.



**REVIEW**  
**of the scientific advisor**

**DIPLOMA WORK**  
*(name of the type work)*

**Mussabay Dariga**  
*(student full name)*

**5B070300 – Information systems**  
*(profession name and code)*

**Theme: Development of HR-management portal for office automation of the company**

This diploma work of student of Mussabay D. is dedicated to the development of HR-management portal for office automation and document flow management of the company.

In main part of diploma work presented are subject area, technologies which used and realization of program project. First chapter accurately described the sphere, overview office work and document flow in the company. Also provided basic requirements, purpose and classification of Electronic Document Management System. Second chapter presented describe methods of implementing Information System.

There are basic technologies used in diploma work for development system. Such as, Microsoft Visual Studio IDE , C# Programming language, ASP.NET MVC 5, PostgreSQL. In software part of diploma work presented programming realization of work, Document Management System, which can used for document flow in organizations or companies.

Diploma work is executed on "excellent" (90%), and its author deserves the academic degree "bachelor" on specialty 5B070300 – «Information systems».

**Scientific advisor**

Tutor

  
signature

«P» may 2019 y.

**A.T. Azhenov**  
full name

**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:** Mussabay Dariga

**Название:** Developing of the HR-management portal for office automation of the company

**Координатор:** Алмат Аженов

**Коэффициент подобия 1:**1,6

**Коэффициент подобия 2:**0

**Тревога:**0

**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 13.05.19

Подпись заведующего кафедрой /  КБ Да ХУ

начальника структурного подразделения

Окончательное решение в отношении допуска к защите, включая обоснование:

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

Допущен к защите!

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

Дата 13.05.19

Подпись заведующего кафедрой / И.В.Клименко

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:** Mussabay Dariga

**Название:** Developing of the HR-management portal for office automation of the company

**Координатор:** Алмат Аженов

**Коэффициент подобия 1:**1,6

**Коэффициент подобия 2:**0

**Тревога:**0

**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:

Работа не обладает признаками  
интеллектуальной  
Признаки работы самостоятельной!

13.05.2019

A.K.

Дата

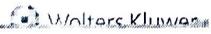
Подпись Научного руководителя

Краткий отчет



Университет:	Satbayev University
Название:	Developing of the HR-management portal for office automation of the company
Автор:	Mussabay Dariga
Координатор:	Алмат Аженов
Дата отчета:	2019-05-03 11:03:06
Коэффициент подобия № 1:	<b>1,6%</b>
Коэффициент подобия № 2:	<b>0,0%</b>
Длина фразы для коэффициента подобия № 2:	<b>25</b>
Количество слов:	11 353
Число знаков:	72 593
Адреса пропущенные при проверке:	
Количество завершенных проверок:	18

**Самые длинные фрагменты, определенные, как подобные**

№	Название, имя автора или адрес гиперссылки (Название базы данных)	Автор	Количество одинаковых слов
1	URL_ <a href="https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3">https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3</a>		17
2	URL_ <a href="https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3">https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3</a>		14
3	URL_ <a href="https://ru.bmstu.wiki/PostgreSQL">https://ru.bmstu.wiki/PostgreSQL</a>		14
4	URL_ <a href="https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/">https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/</a>		13
5	URL_ <a href="https://www.bookdepository.com/Visual-Studio-2017-IV-Robert-Wilson/9781547224050">https://www.bookdepository.com/Visual-Studio-2017-IV-Robert-Wilson/9781547224050</a>		13
6	LCG Monte-Carlo Data Base (RefBooks - <a href="https://arxiv.org/">https://arxiv.org/</a> ) 	A. Vologdin, L. Dudko, A. Sherstnev, I. Seluzhenkov, P. Bartalini, A. Kryukov,	12
7	URL_ <a href="https://www.pilotfishtechnology.com/supported-databases/">https://www.pilotfishtechnology.com/supported-databases/</a>		10
8	Mikrosocwisy <i>Polsko Japońska Akademia Technik Komputerowych (Warszawa.)</i>	Dominik Miszkiewicz nr. albumu s10727	10
9	CNUT/Олійченко_І_М_Впровадження_та_вдосконалення_електронного_документообігу_в_державному_управлінні_.pdfCNUT <i>Chernihiv National University of Technology (CNUT) (Deanery)</i>		9
10	URL_ <a href="https://ru.bmstu.wiki/PostgreSQL">https://ru.bmstu.wiki/PostgreSQL</a>		9

**Документы, в которых найдено подобные фрагменты: из RefBooks 1**

Источник <https://arxiv.org/>

№	Название	Автор	Количество одинаковых слов (количество фрагментов)
1	LCG Monte-Carlo Data Base	(A. Vologdin, L. Dudko, A. Sherstnev, I. Seluzhenkov, P. Bartalini, A. Kryukov.)	19 (2)

**Документы, содержащие подобные фрагменты: Из домашней базы данных**  
 Не обнаружено каких-либо заимствований

**Документы, содержащие подобные фрагменты: Из внешних баз данных**

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

№	Название (Название базы данных)	Автор	Количество одинаковых слов (количество фрагментов)
1	<b>CNUIT/Олійченко_І_М_Впровадження_та_вдосконалення_електронного_документообігу_в_державному_управлінні_.pdf</b> CNUIT <i>Служба National University of Technology (CNUIT) (Deanery)</i>		16 (2)
2	<b>Milosz Miszkiwscy</b> <i>Państwowa Akademia Techniki Komputerowych (Warszawa.)</i>	Dominik Miszkiewicz nr. albumu s10727	10 (1)

**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	<b>URI</b> <a href="https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3">https://www.facebook.com/ISSSTUDENTS/photos/a.245115978931440.49196.125857620857277/1301339059975788/?type=3</a>	39 (3)
2	<b>URI</b> <a href="https://ru.bmstu.wiki/PostgreSQL">https://ru.bmstu.wiki/PostgreSQL</a>	29 (3)
3	<b>URI</b> <a href="https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/">https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/</a>	26 (3)
4	<b>URI</b> <a href="http://www.bookdepository.com/Visual-Studio-2017-IV-Robert-Wilson/9781547224050">http://www.bookdepository.com/Visual-Studio-2017-IV-Robert-Wilson/9781547224050</a>	20 (2)
5	<b>URI</b> <a href="https://www.webdss.com/wordpress-website/">https://www.webdss.com/wordpress-website/</a>	15 (2)
6	<b>URI</b> <a href="http://www.pilotfishtechnology.com/supported-databases/">http://www.pilotfishtechnology.com/supported-databases/</a>	10 (1)

Copyright © Plagiat.pl 2002-2019

## ANNOTATION

This diploma work is related to the development of a Document Management System for the HR-portal for office automation in companies.

For research subject area were taken existing workflow, Electronic Document Management Systems. Analytical work was done on the advantages and disadvantages of Document Management System (DMS). During the development of the system, the following tools were used: C# programming language, Microsoft Visual Studio 2017 Integrated Development Environment, ASP.NET MVC and Entity frameworks and PostgreSQL Data Base Management System.

The system processes and verifies information about employees, workers, managers. It registers users and gives the opportunity to exchange electronic documents and view their statuses in their workplace.

## АННОТАЦИЯ

Данная дипломная работа связана с разработкой системы документооборота HR-портала для автоматизации делопроизводства в компаниях.

Для исследования предметной области и в качестве исходных материалов были взяты существующие системы документооборота. Была проведена аналитическая работа над преимуществами и недостатками систем документооборота. В процессе разработки системы были использованы такие инструментв как: язык программирования C#, Интегрированная среда разработки Microsoft Visual Studio 2017 , фреймворки ASP.NET MVC, Entity и СУБД PostgreSQL.

Система обрабатывает и проверяет информацию о сотрудниках, рабочих, руководителей и менеджеров. Регистрирует пользователей и дает возможность обмениваться электронными документами и смотреть их статусы на своем рабочем месте.

## АНДАТПА

Бұл дипломдық жұмыс компанияларға арналған электронды құжат айналымының HR-порталындағы кеңсе жұмысын автоматтандырушы жүйені құрумен байланысты.

Дипломдық жұмысқа негіз ретінде қолданыстағы электронды құжат айналымының үлгілері алынды. Ақпараттық жүйені құру барысында келесі құралдар пайдаланылды: C # программалау тілі, Microsoft Visual Studio 2017 интеграцияланған құрылыс ортасы, ASP.NET MVC және Entity фреймворктары және PostgreSQL МББЖ.

Ақпараттық жүйе тіркелген жұмысшылардың жіберген ақпараттарын тексеріп, өңдеп, оларды мәліметтер базасына еңгізеді. Сонымен қоса, ақпараттық жүйеге қолданушыларды тіркеп, жұмыс аймағында электронды құжаттармен алмасуға мүмкіндік беріп, оларды қадағалап отырады.

## CONTENT

	Introduction	7
1	Overview office work and document flow in company	8
1.1	Office work in Human Resource Management	9
1.2	General structure of the organization document flow	10
1.3	Basic principles and types Electronic Document Management	11
1.4	Requirements and purpose Electronic Document Management System	12
1.5	System setting tasks	15
2	Methods of implementing Information System	16
2.1	Microsoft Visual Studio - Integrated development environment	16
2.2	C# Programming language	18
2.3	Object-oriented programming	20
2.4	ASP.NET MVC 5 framework	21
2.5	PostgreSQL Data Base Management System	22
2.6	Object-Relation Mapper and Entity Framework	24
2.7	Bootstrap framework in ASP.NET MVC 5	25
3	Software for Information system	27
3.1	Structure of software	27
3.2	Description of the developed application software	28
3.3	Work algorithm with Programming tools	29
3.4	Realization of Information system	35
	Conclusion	38
	References	39
	Appendix A	40
	Appendix B	41
	Appendix C	43

## INTRODUCTION

Work with documents has been and remains one of the most important processes in any enterprise. The organization of such work is an important part of management processes that significantly affect the efficiency and quality of decisions made.

In any organization, in the conduct classical paperwork, as a rule, after a while, problems begin to arise: the loss of documents received, duplicating documents; difficulties in finding the necessary documents; the inability to control the status of work on a specific document; an increase in the time required for documents by authorities, etc. The consequence of the inefficient workflow of the document management system is the inability to answer the key questions related to the company's activities: What decisions were made on this project? What documents are in work? How responsibility is divided between specific units and individual employees?

Paper documents have long been used in storing information. However, paper can be costly and, if used excessively, wasteful. Document management software is not simply a tool but it lets a user manage access, track and edit information stored. Document management software is an electronic cabinet that can be used to organize all paper and digital files. The software helps the businesses to combine paper to digital files and store it into a single hub after it is scanned and digital formats gets imported. Web based document management software are becoming the staple of the industry.

The most effective way to solve these problems of organizing workflow is office automation - that is, managing the movement of documents using electronic systems.

Actuality of the work lies in the fact is that today the improvement of management, improving the level of organization and efficiency of management work largely depends on how the office work is delivered at the enterprises.

The purpose of the work is to research and development workflow of the document management system for office automation.

For realization this purpose, the following tasks were formulated: disclosure concept of the terms “HR-management”, “office work automation” and “document flow”; describe office automation subsystems; describe the functions of the document workflow system; to characterize electronic document circulation as a form of modern office work; analyze the office automation of the company.

## **1 Overview office work and document flow in company**

In modern society, the relationship between government and enterprises, on the one hand, and individuals - on the other, as well as enterprises among themselves are carried out through the exchange of documentation, which captures all the most important management decisions. Documented data is the basis of management. The document created a legal or natural person, prepared and authenticated in the prescribed manner, is called a official document. Electronic Document (ED) - a document created by means of computer data processing, which can be signed by an electronic signature (ES) and stored on machine medium as a file appropriate format.

Management documents can be divided into two groups:

- documents on the general management of the company (organization) and its production activities (on general and administrative issues);
- documents on management functions (financial, accounting, personnel, etc.).

The number of cases and types of documents is directly related to the specifics of the company. Often the document is divided into blocks, which are experts in areas of work. But there is a list characteristic of any company, be it commercial or state:

Incoming - coming into the organization from the outside - everything that came by mail, with couriers, personally delivered by representatives of other organizations.

Outgoing - sent from the organization - everything, addressed to other legal entities, as well as to individuals (answers, letters, and decisions).

Internal - not going beyond the organization, regulating the work of the enterprise (orders, orders, regulations, regulations, etc.).

The documents contain information that is a valuable resource, as well as an important element of business activities. A systematic approach to document management allows an enterprise to protect or preserve them as evidence of actions.

Office work is the term used in office practice for a set of rules for working with documents. There are document flow systems that are customized to the rules of office work.

Document flow is a movement (turnover) of documents, both within the organization, and between organizations that cooperate among themselves on any issues or in any area. The concept of documentation is quite broad, and the tasks are very many-sided, since the documentation system in any organization includes entire departments and services (financial, administrative, administrative, etc. documents are mandatory in the activities of any enterprise). The document flow tree also displays the document type and number, the ID of the user who created the document, the creation date, and the document status.

Efficient workflow and document management, it is necessary to develop the least time-consuming paths of movement of documents both from executors “up” in the development of a draft document, and “down”, to the execution of an already prepared one, as well as control over their execution. It is necessary to avoid re-registration of the same document, because it means a waste of time and unproductive burden on the performers [1].

## **1.1 Office work in Human Resource Management**

Office work is constantly expanding, including more and more different directions. Personnel office work, archival office work, processing of incoming / outgoing documentation, all this is only a small part of the concept of office work. Office, secretary, personnel service, archivists, clerks, personnel managers, timekeepers - all employees who work in the field of office work.

Office work can be:

- general - relating to the main activity of the company;
- personnel - for personnel documentation, conducted by the personnel department;
- special - found in some government agencies and has standards of conduct;
- electronic - when part of the information is transmitted in digital format.

In state-owned companies, the algorithm for working with documents is regulated by law. Private firms have the right to decide on their own how to manage office work.

There are three main areas of office work:

- documentation - preparation, execution, coordination;
- organization of work with documents - movement, control of execution, storage and use;
- systematization- the transfer of documents for storage.

1.1.1 Personnel office work Human Resource Management is an integral part of quality management systems of the company. It includes the actions of all persons who make decisions aimed at the planning process, evaluation and implementation of the plan. HR management activities - a targeted impact on the human component of the organization, focused on matching the capabilities of employees and goals, strategies, conditions for the development of the organization. To manage an organization (company, enterprise) it is always necessary to create numerous types of management documents, without which it is impossible to solve the tasks of planning, financing, accounting and reporting, staffing, etc. In order to develop a concrete management decision, it is first necessary to collect and process data.

Each employee of the company, without even noticing it, generates a variety of different documents necessary for processing and recording its activities. A person must be selected for the position, recruited and formalized this fact, documenting the departure for a business trip and return from it. Require documentation of the facts of illness of employees and exit to work after it, vacation, dismissal and other events.

Personnel office work is the documentation of labor relations and the organization of work with personnel documents.

Personnel office work is conducted in any organization. It is a mistake to assume that the documentation of labor relations is only a design of recruitment and dismissal. Personnel office work includes issues of developing and maintaining documentation related not only to personnel management, personnel movements and

personnel records, but also with appraisal, working time, sending on business trips, drawing up incentives and disciplinary penalties, drawing up staffing schedules, etc.

The main function of personnel documents is to regulate the relationship between the employee and the employer, and the main purpose of personnel documentation is to give labor relations transparency.

1.1.2 The automation of office work is to implement the functions of the integrated automation of the tasks of developing, coordinating, distributing, searching and archiving documents.

The need for office automation occurs when the problems of paper work begin to appear significantly in an organization. Some of them become critical. For example, when there are several answers to a client's request, and containing directly opposite or mutually exclusive information. This arises because of the inconsistency of the executors of documents in the preparation of responses and the lack of control over the formation and sending responses to appeals. Loss of documents, multiple duplication, time-consuming and inefficient control of execution, a long search - all these problems together become the impetus for office automation.

Office automation is primarily aimed at ensuring efficiency and reducing errors in working with documents, as well as reducing the time for routine operations such as registration, accounting, searching, etc.

For complex office automation in the system for most documents, it is possible to automate the full document life cycle. To do this, in the program each document type has its own route, which has a graphical block diagram determining the composition and sequence of document processing steps, employees participating in the process (Process Roles), and various automated procedures and checks that are performed during the process execution.

## **1.2 General structure of the organization document flow**

A document flow is the movement of documents from the moment they are created or received until completion of the execution, sent to the addressee or transferred to the archive. There is a general structure for organizing document flow and workflow in an enterprise, stages and tasks that will be present even in the smallest company.

Organization of incoming document flows is the manage documentation that is created in other organizations and institutions and comes to the company in paper or electronic form: letters, claims, commercial offers and invoices for payment. Organization of outgoing document flows is organizational structure of the enterprise. After creation, they pass the path of coordination and endorsement in several structural divisions and are sent to external bodies: trading partners, suppliers and customers, to regulatory bodies, and so on. Organization of internal document flows is the most extensive section of office work and workflow in large enterprises. Moreover, it is often their movement that is the most disordered. Internal documents can combine the features of incoming and outgoing [2].

## 1.3 Basic principles and types Electronic Document Management

1.3.1 Electronic Document Management is a way of organizing work with documents in which documents are used electronically and stored centrally. This is a system of automated processing of electronic documents, which implements the concept of “paperless office work”.

The first electronic document management systems were developed directly in enterprises and were completely individualized. It was almost impossible to change the structure of such a system, and the cost of the program was very high.

Today, more and more organizations are seeking to implement an electronic document management system in order to increase the efficiency of working time and minimize the costs of manual processing of documents.

Basic principles of electronic document management (EDM):

- single registration of the document, which allows to uniquely identify the document;
- the possibility of parallel execution of operations, allowing to reduce the time of movement of documents and increase the efficiency of their execution;
- the continuity of the movement of the document, allowing to identify the person responsible for the execution of the document (task) at each time point in the life of the document (process);
- single (or coordinated distributed) database of document information, which allows to exclude the possibility of duplication of documents;
- effectively organized document search system that allows you to find a document with minimal information about it;
- developed reporting system for various statuses and attributes of documents, allowing to control the movement of documents on the processes of workflow and make management decisions based on data from the reports;

Types and functions of electronic document management:

Depending on the specifics of the organization's activities, there are several types of electronic document management:

- production EDM;
- management EDM;
- archive business;
- personnel EDM;
- accounting EDM;
- warehouse EDM;
- technological EDM;
- private and confidential EDM.

EDMS refers more specifically to a software system that handles digital documents, rather than paper documents, although in some instances, these systems may also handle digital scanned versions of original paper documents. EDMS provides a way to centrally store a large volume of digital documents [3].

## **1.4 Requirements and purpose Electronic Document Management System**

1.4.1 Requirements for EDMS. Electronic Document Management Systems (EDMS) is a corporate information system designed to work with corporate documents presented in electronic form throughout their life cycle: creating, modifying, coordinating, signing, registering, storing, searching, classification and so on.

In practice, the EDMS also automates other processes, such as archival storage of documents, electronic archives, in compliance with all standards and documents storage regulations.

Electronic Document Management System (EDMS) is a special system that provides participants with the exchange of electronic documents of legal significance. All electronic document management systems can be classified according to several criteria:

- EDMS with advanced information storage and retrieval systems. Their second name is electronic archives;
- EDMS with advanced routing systems that ensure the movement of documents along specified routes;
- EDMS with a system for supporting management of organization and accumulation of knowledge. Usually these systems combine the properties of the two previous ones. At the same time in such a system it is possible to use both hard and free routing. Similar EDS are used in large companies and government agencies;
- EDMS with employee collaboration support. Such systems are aimed at organizing the collective work of employees, even if they are divided geographically. Provide the ability to search for information, discussions and appointment meetings, including real and virtual, as well as services for storing and publishing documents;
- EDMS with additional services: project management, e-mail, billing, CRM service.

The most popular functions of the ERMS are:

- document storage and retrieval;
- office support;
- routing and control of the execution of documents: drawing up routes of documents, supporting actions during routes, notifying employees of new documents, automatic control of deadlines;
- preparation of analytical reports, such as a report on current employment, on the performance of work on documents and on overdue orders.
- ensuring information security, including user authentication, support for electronic digital signatures (EDS), encryption of documents and letters, audit of work in the system.

Some experts point out that the electronic document management system has a lot in common with a content management system (CMS). This systems involve handling a variety of Web content from a central site, while a document management system is often primarily used for archiving [4].

1.4.2 Purpose and classification of EDMS. There is no official definition of the electronic document management system (EDMS) approved by the standard. In essence, EDMS is a system that allows you to automate the main business procedures of a company. It covers the processes of creation, processing, replication, transfer, storage of documents, control over their execution and is intended for the effective management of the enterprise. When introducing an ERMS in an enterprise, the following purposes are usually set:

- reduction or total failure of paper workflow;
- creation of a single information base of the company;
- reducing the risk of losing a document;
- structuring of all documentation for the approved item;
- increased discipline among employees due to the possibility of tracking the activities of the performer of a particular document control over the execution of documents in accordance with the resolutions of the head;
- improving the efficiency of the company.

Classification of EDMS: universal "boxed" EDMS, individually developed EDMS, combined EDMS. Each system can be characterized by one or several signs, each of which will be useful in selecting a EDMS for a specific organization.

Universal "boxed" EDMS has standard feature set, the inability to fully meet the needs of a particular organization, low time costs for the purchase and installation, relatively low cost, the need to purchase a license for each implemented workplace.

Individually developed EDMS is the most personalized system, large time costs, high development cost and associated costs: the cost of training employees, the purchase of new equipment and software.

Combined EDMS has the basic platform to which the necessary additional modules are being developed, full compliance with the needs of the enterprise, small time costs for development and implementation, the cost includes the price of the basic platform and the cost of individual customization, depending on the complexity of the order, transfer of product rights to the customer, ease of development and use, full localization, convenient interface, interaction with existing office applications.

The main problem in classifying information systems, including EDMSs, is seen in the fact that common bases (signs) that are sufficient to describe their properties have not been worked out. There is even an opinion among experts that it is almost impossible to develop a common classification that would satisfy both theoreticians and practitioners, since its author will inevitably have disagreements with the companies–developers of software about the “incorrect classification” of their systems. Each system can be characterized by one or several signs, each of which will be useful in selecting a EDMS for a specific organization. We have studied the approaches to the classification of EDMSs. Analysis of the revealed publications permits us to make the conclusion that the available classifications of EDMSs usually consider any one criterion or do not define the criteria at all based on which the division into classes takes place; there are terminological discrepancies [4].

### 1.4.3 Differences between EDMS and ERMS.

Electronic Records Management System (ERMS) - management of official documents of the organization from the moment of their creation until the final destruction. This practice includes the formation of documents in office work, their storage, protection and destruction (or, in some cases - archival storage).

The term "record" is used in the context of building a single electronic archive of the company. To avoid confusion, first we need to decide on the difference between the concepts of "document" and "record". We will assume that a "record" is a special kind of document containing actual versions of document files at certain stages of its life cycle with a subset of contextual metadata that cannot be changed.

For example, this type of document may include official company documents. If this type of data is stored in the company's archive, then when building an archive solution, it is necessary to take into account that records management largely determines the architecture of the archive and the technical requirements for the software.

What is the difference between electronic information management systems (document) and official documents (record)? It seems like the same functions are largely needed to work on both types of documents. Below table 1.1 presented the key differences between EDMS and ERMS.

Table 1.1 – The key differences between EDMS and ERMS:

№	EDMS	ERMS
1.	Allows you to modify documents.	Protects documents from change.
2.	Allows for the existence of multiple versions of the same document.	Allows for the existence of documents in one version - the final.
3.	Allows deletion of documents by their owners;	Prohibits the removal of documents, except for some strictly controlled situations.
4.	Include some long-term storage management functions.	Include strict storage rules.
5.	The document repository structure, if present, can be controlled by the users themselves.	Must contain documents in a strictly ordered manner (in accordance with the classification scheme) determined by the system administrator.
6.	Designed primarily to support daily work with documents;	Can support daily work, but the main purpose is to provide a reliable repository (storage) for documents relevant to the business.

So that these differences do not look mysterious, it is enough to take into account the accents of each of these two classes: for EDMS, user efficiency is important, for ERMS - the safety of documents [5].

### 1.4.3 Storage in Electronic Document Management System.

It is no secret that each of us has quite a lot of files that are located on local disks, and not in EDMS - sketches, thoughts, notes, pictures, prototypes, archives, letters, photos, e-books. Even having access to a corporate archive of documents organized with the help of an electronic document management system, we are all free to decide which documents should be placed in the system and which ones should not.

And more often the decision is made not in favor of the EDMS. Moreover, there is only one criterion - unwillingness to spend time: in order to place a document in the DMS, you need to fill out its card and assign rights, and this is always additional work. Nevertheless, I would like to speculate in which cases the document should still be stored in the system, even if "I do not want to".

First, documents important for the company's business should always be posted to the DMS. Correspondence with customers, partners or suppliers can be very useful in the subsequent analysis of conflict situations. Saved drafts of projects or contracts will allow your colleagues to pick up your work if you are sick.

Secondly, documents containing useful knowledge and experience for the employees of the company. Electronic books, useful articles and just a list of links to Internet resources need to be placed in the EDMS: the time saved by your colleagues in searching for them will fully pay for two minutes of creating the card by you.

Thirdly, private information about various assets, for example, passwords to Internet resources or license keys of the software used in the company, should fall into the if you are responsible for it. Storing this information in the EDMS with restricted access is much safer than on local disks, and if necessary, if the responsible person changes, for example, this information can be transferred to the new employee without any problems [5].

## 1.5 System setting tasks

For realization work purpose, the following tasks were formulated: disclosure concept of the terms "HR-management", "office work automation" and "document flow"; describe office automation subsystems; describe the functions of the document workflow system; to characterize electronic document circulation as a form of modern office work; analyze the office automation of the company.

System setting tasks:

- collect basic data about HR-management and document flow;
- research office work area in the company, studies of office automation;
- analyzing and comparing existing EDMS, search advantages and disadvantages in systems, making conclusion;
- choose necessary software of Information System for development;
- development efficient Electronic Documents Management System for office automation of the company.

## **2 Methods of implementing Information System**

### **2.1 Microsoft Visual Studio - Integrated development environment**

Microsoft Visual Studio is a Microsoft product line that includes an integrated software development environment and a number of other tools. These products allow you to develop both console applications and applications with a graphical interface, including support for Windows Forms technology, as well as websites, web applications, web services in both native and managed code for all platforms, Supported by Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework and Silverlight. Visual Studio is a software development tool used to create programs for Windows and web applications. The developer's kit includes an integrated development environment (IDE) and testing tools for all stages of the software development process [6].

Visual Studio includes the following components:

IDE Visual Studio includes many built-in perspectives for performing various tasks. Unique interfaces are provided for source code programming, debugging, and creating visual forms. Developers can use one of these pre-prepared layouts or customize their own perspectives. The Visual Studio software package contains Microsoft Visual Basic, Visual C ++, Visual C #, Visual F #, Visual Web Developer, and Team Foundation Server. The software supports a wide range of programming languages, including C, C ++, VB.NET, C #, F #, XML, HTML, JavaScript and CSS.

Microsoft Visual Studio provides developers with a customizable IDE and supports a wide range of popular programming languages. This is a good tool for professional software developers who want to create commercial programs. Microsoft Visual Studio is also useful for web developers who want to create dynamic web sites and interactive web applications for the .NET platform [7].

The picture shows Visual Studio 2017 with an open project and several windows of the main tools that you will most likely need:

- solution Explorer (top right) allows you to view code files, navigate and manage them. Solution Explorer allows you to streamline code by combining files into solutions and projects;
- in the editor window (center), where you will most likely spend most of your time, the contents of the file are displayed. Here you can edit the code or develop a user interface, such as a window with buttons or text fields;
- in the output window (center below), Visual Studio sends notifications, such as debugging and error messages, compiler warnings, publication status messages, and many others. Each message source has its own tab;
- team Explorer (lower right corner) allows you to track work items and share code with other users using version control technologies such as Git and the Team Foundation version control system (TFVC).

In Figure 2.1 shown open project window, where we can see all components of MS Visual Studio 2017. Next in table 2.2 presented Visual Studio 2017 versions, also in Table 2.3 shown main working load options of Microsoft Visual Studio 2017 [8].

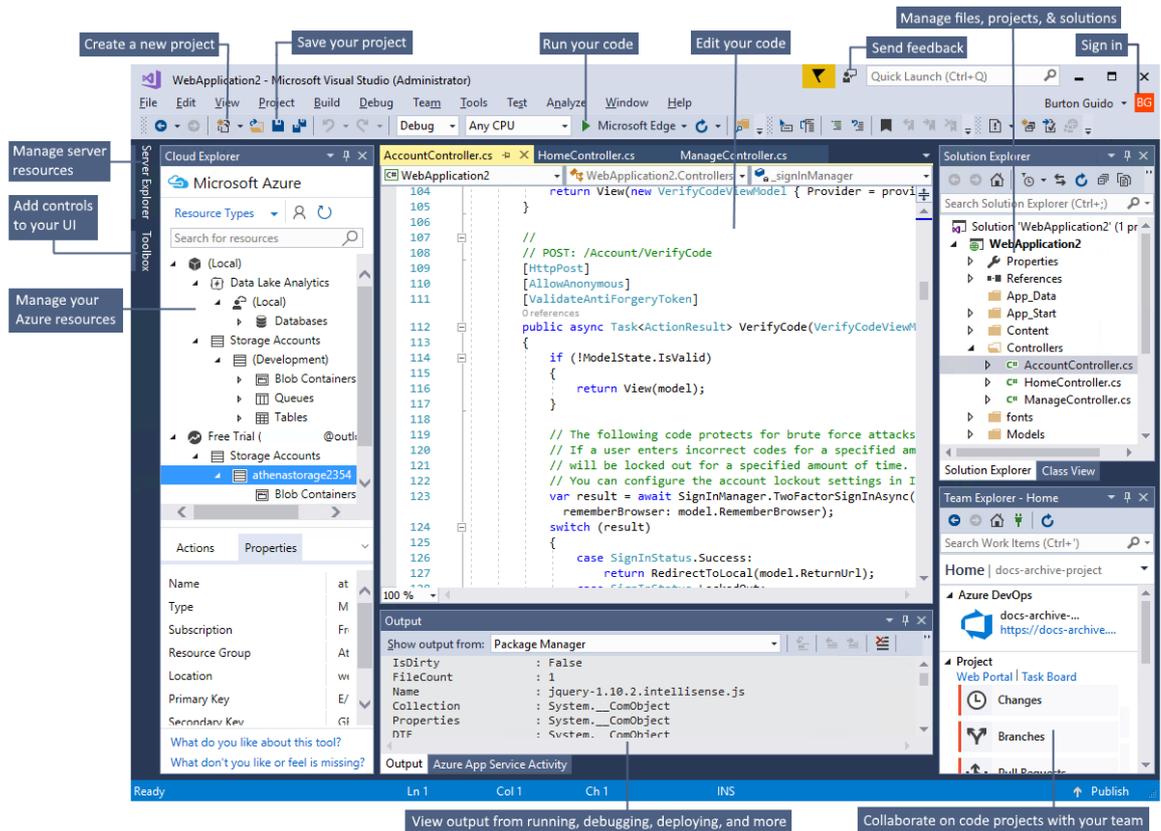


Figure 2.1 – MS Visual Studio 2017 open project windows

Table 2.2 – Microsoft Visual Studio 2017 versions

No	Version	User
1.	Visual Studio Community 2017	For non-commercial use: students, open source developers
2.	Visual Studio Professional 2017	For professional program development: used for organizations, professional developers
3.	Visual Studio Enterprise 2017	For large enterprises, for teams of any big sizes

Table 2.3 – Main working load options

No	Option name	Description
1.	.NET desktop development	allows to create C # applications: – C# Console Application; – C# Windows Forms Application; – C# WPF Application.
2.	ASP.NET and web development	allows you to create C # applications: ASP.NET Web Application (C# web-app)
3.	.Net Core cross-platform development	allows you to create C # applications: ASP.NET Core Web Application(C# web-app)
4.	Data storage and processing	Tools for working with the database

## 2.2 C# Programming language

C # (pronounced "C sharp") is a simple, modern object-oriented and type-safe programming language. C # refers to a family of languages with a C-like syntax, of which its syntax is closest to C ++ and Java.

C# is an object-oriented language, but also supports component-oriented programming. The development of modern applications is increasingly a matter of creating software components in the form of autonomous and self-descriptive packages that implement individual functionality. An important feature of such components is a programming model based on properties, methods, and events. Each component has attributes that provide declarative information about the component, as well as built-in documentation elements. C # provides language constructs that directly support this work concept. Because of this, C # is great for creating and using software components.

Below in table 2.3 shown overview of the C# main elements and description of this elements. There are 10 main elements of C# programming language [9].

Table 2.3 – Brief overview of the elements

No	Element	Description
1.	Program structure	The organizational structure of C # is based on such concepts as programs, namespaces, types, members, and assemblies.
2.	Types and variables	In C #, subdivided into simple types, enumeration types, structure types, and types that admit the value Null. Reference types in C # are divided into class types, interface types, array types, and delegate types.
3.	Expressions	Expressions are created from operands and operators. Expressions return values.
4.	Operators	An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations.
5.	Classes and objects	Classes are the most important type in C #. Objects are instances of classes. Classes are created by the description of their members, which are also described in this article.
6.	Structures	Structures are entities for storing data. They differ from classes primarily in that they are value types.
7.	Arrays	An array is a data structure containing several variables that are accessed by calculated indices.
8.	Interfaces	The interface defines a contract that can be implemented by classes and structures. An interface can contain methods, properties, events, and indexers.
9.	Delegates	The delegate type represents references to methods with a specific list of parameters and the type of return value. Delegates allow you to use methods as entities, saving them into variables and passing them as parameters.
10.	Attributes	Attributes allow programs to specify additional descriptive data about types, members, and other entities.

```

public class MyClass
{
    public string myField = string.Empty;
    public MyClass()
    {
    }
    public void MyMethod(int parameter1, string parameter2)
    {
        Console.WriteLine("First Parameter {0}, second parameter {1}", parameter1, parameter2);
    }
    public int MyAutoImplementedProperty { get; set; }
    private int myPropertyVar;
    public int MyProperty
    {
        get { return myPropertyVar; }
        set { myPropertyVar = value; }
    }
}

```

Figure 2.3 – MS Visual Studio 2017 project syntax

Visual C # is the implementation of the C # language by Microsoft. Visual C # support in Visual Studio is provided using a full-featured code editor, compiler, project templates, designers, code wizards, a powerful and convenient debugger, and many other tools. The .NET Framework class library provides access to many operating system services and other useful, correct classes, which significantly speeds up the development cycle [10].

Above presented figure 2.3 where we can see syntax of MS Visual Studio 2017 in C#. Table 2.4 has shown the table lists available value types in C#.

Table 2.4 – The table lists available value types in C#

Type	Represents	Range
bool	Value Boolean	True or False
byte	Unsigned integer 8-bit	0 to 255
char	Unicode character 16-bit	U +0000 to U +ffff
decimal	Precise decimal values with 28-29 significant digits 128-bit	$(-7.9 \times 10^{28}$ to $7.9 \times 10^{28}) / 10^0$ to 28
double	Double-precision floating point type 64-bit	$(+/-)5.0 \times 10^{-324}$ to $(+/-)1.7 \times 10^{308}$
float	Single-precision floating point type 32-bit	$-3.4 \times 10^{38}$ to $+ 3.4 \times 10^{38}$
int	Signed integer type 32-bit	-2,147,483,648 to 2,147,483,647
long	Signed integer type 64-bit	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
short	Signed integer type 16-bit	-32,768 to 32,767

## 2.3 Object-oriented programming

Object-oriented Programming (OOP) is a programming methodology based on the representation of a program as a set of objects, each of which is an instance of a certain class, and the classes form an inheritance hierarchy. The basic principles of structuring in the case of OOP are associated with various aspects of the basic understanding of the objective problem, which is required for optimal control of the relevant model:

- abstraction for the selection in the modeled object important for the solution of a specific task on the subject, ultimately - contextual understanding of the subject, formalized as a class;
- encapsulation for fast and secure organization of hierarchical manageability proper: so that the “what to do” command is simple enough without simultaneously specifying how to do it, since this is a different level of control;
- inheritance for quick and safe organization of related concepts: so that it was enough to take into account only changes at each hierarchical step, without duplicating everything else taken into account at the previous steps;
- polymorphism to determine the point at which a single control is better to parallelize, or vice versa - to put together [11].

That is, in fact, we are talking about a progressive organization of information according to the primary semantic criteria: “important / unimportant”, “key / details”, “parent / child”, “single / multiple”. Progressing, in particular, at the last stage, makes it possible to move to the next level of detail, which closes the overall process.

In figure 2.5 we can see basic structure of Procedural Oriented Programming and Object Oriented Programming. Also this figure presented difference between them [12].

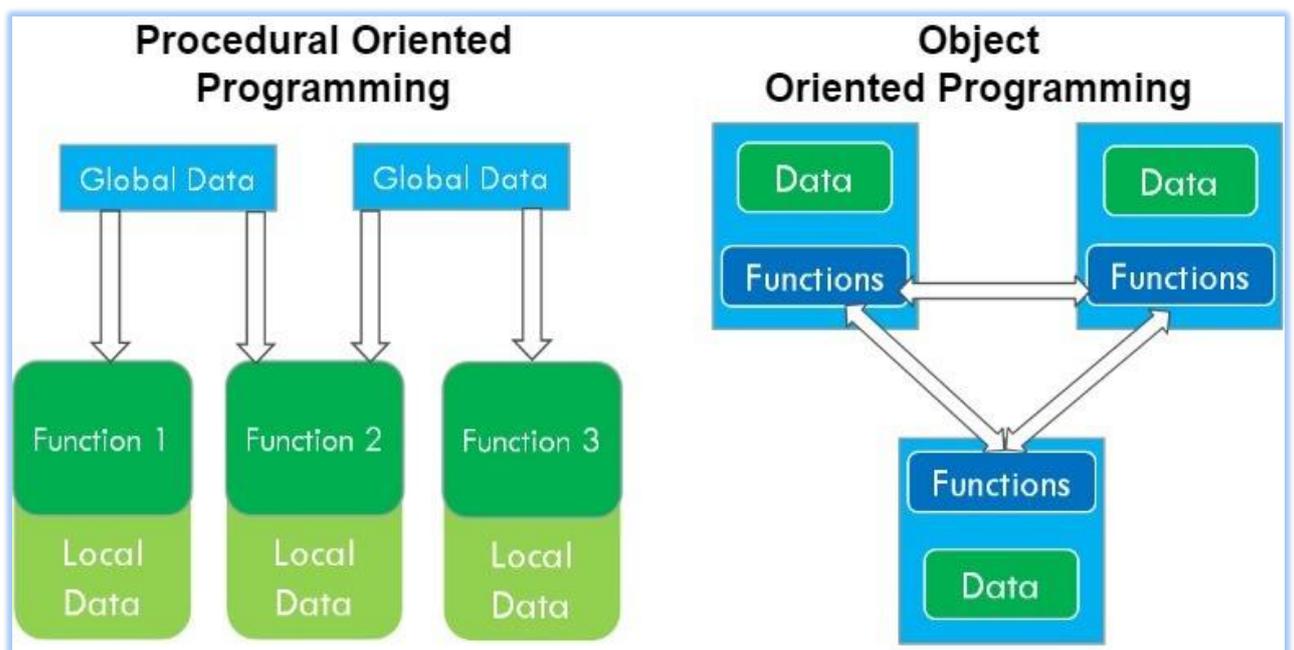


Figure 2.5 – Difference between POP and OOP

## 2.4 ASP.NET MVC 5 framework

ASP.NET MVC Framework - a framework for creating web applications that implements the Model-View-Controller pattern.

The concept of the MVC pattern (model - view - controller) involves the division of an application into three components.

The controller represents a class that provides communication between the user and the system, the view and the data store. It receives user input and processes it. And, depending on the processing results, it sends a certain output to the user, for example, in the form of a presentation. The view is the actual visual part or user interface of the application. As a rule, the html-page that the user sees by going to the site. The model represents a class that describes the logic of the data used [13].

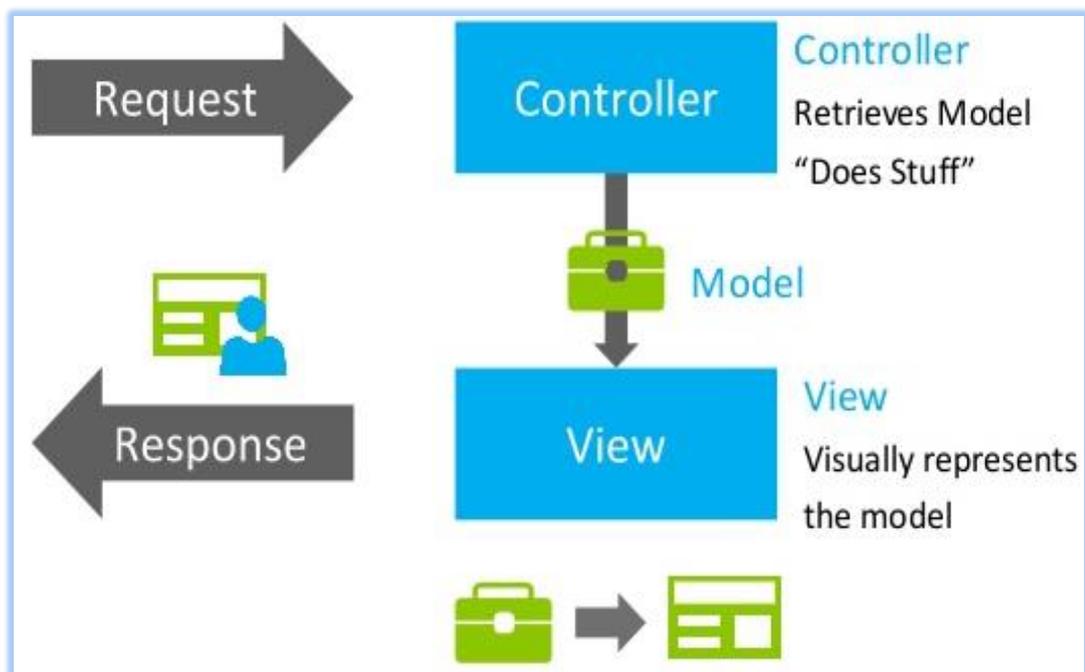


Figure 2.6 – The general scheme of interaction

In figure 2.6 has shown the general scheme of interaction, the model is an independent component – any changes to the controller or view do not affect the model. The controller and view are relatively independent components, and they can often be changed independently of each other.

Thanks to this, the concept of sharing responsibility is realized, and therefore it is easier to build work on individual components. In addition, as a result, the application has better testability. And if we assume, for example, the visual part or the frontend is important, then we can test the presentation independently of the controller. Or we can focus on the backend and test the controller.

Controller is a central component in the MVC architecture. In ASP.NET MVC 5, Views are files with a cshtml extension that contain user interface code mainly in html language. Models describe the logic of the data. All entities in the application can be divided into separate models [14].

## 2.5 PostgreSQL Data Base Management System

PostgreSQL is an object-relational database management system (ORDBMS) that has the traditional capabilities of commercial DBMS with extensions that are in the new generation DBMS.

PostgreSQL is developed by a development team scattered around the world and connected via the Internet. Development is a public project and is not managed by any company. PostgreSQL is free and completely open source software of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features: complex queries, transactional integrity, foreign keys, triggers, updatable views, multiversion concurrency control. Also, PostgreSQL can be extended by the user in many ways, for example by adding new: data types, functions, operators, aggregate functions, index methods, procedural languages [15]. In table 2.7 presented PostgreSQL data size.

Table 2.7 – PostgreSQL can handle a lot of data

Maximum database size	Unlimited
Maximum size of the table	32 TB
Maximum line size	1.6 TB
Maximum field size	1 GB
Maximum number of rows in the table	Unlimited
Maximum number of columns in table	250-1600 depending on the type of column
Maximum number of indexes in the table	Unlimited

And because of the liberal license, PostgreSQL can be used, modified, and distributed by anyone free of charge for any purpose, be it private, commercial, or academic. PL / pgSQL is a procedural language for PostgreSQL. The goal of PL / pgSQL design was to create a downloadable procedural language that:

- used to create functions and triggers;
- adds control structures to the SQL language;
- can perform complex calculations;
- inherits all user types, functions and operators;
- can be defined as a trusted language;
- easy to use.

PL / pgSQL allows you to group a block of computations and a sequence of queries within the database server, so we get the power of a procedural language and ease of use of SQL, with significant savings in overhead of client-server interaction. Additional calls between client and server are excluded. Intermediate unnecessary results are not transferred between the server and the client. It is possible to avoid multiple analyzes of a single request. As a result, this leads to a significant increase in performance compared to an application that does not use stored functions. In addition, PL / pgSQL allows you to use all data types, operators, and SQL functions.

PostgreSQL has many options. Created using the object-relational model, it supports complex structures and a wide range of built-in and user-defined data types. It provides enhanced data capacity and has earned confidence with respect for data integrity. PostgreSQL manages database privileges using the concept of roles. The role can be either a separate database user or a group of users. Roles can be owners of objects in the database (for example, tables), and can also assign privileges to access these objects for other roles. It is possible to grant one role membership in another role and, accordingly, transfer this role the rights of the role of which it will be a member. Below in figure 2.8 showed Comparing modern popular DBMS for main options [16].

Название	ASE	DB2	FireBird	InterBase	MS SQL	MySQL	Oracle	PostgreSQL
Лицензия	\$\$\$	\$\$\$	IPL <sup>2</sup>	\$\$\$	\$\$\$	GPL/\$\$\$	\$\$\$	BSD
ACID	Yes	Yes	Yes	Yes	Yes	Depends <sup>1</sup>	Yes	Yes
Referential integrity	Yes	Yes	Yes	Yes	Yes	Depends <sup>1</sup>	Yes	Yes
Transaction	Yes	Yes	Yes	Yes	Yes	Depends <sup>1</sup>	Yes	Yes
Unicode	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Schema	Yes	Yes	Yes	Yes	No <sup>5</sup>	No	Yes	Yes
Temporary table	No	Yes	No	Yes	Yes	Yes	Yes	Yes
View	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Materialized view	No	Yes	No	No	No	No	Yes	No <sup>3</sup>
Expression index	No	No	No	No	No	No	Yes	Yes
Partial index	No	No	No	No	No	No	Yes	Yes
Inverted index	No	No	No	No	No	Yes	Yes	Yes <sup>6</sup>
Bitmap index	No	Yes	No	No	No	No	Yes	No
Domain	No	No	Yes	Yes	No	No	Yes	Yes
Cursor	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
User Defined Functions	Yes	Yes	Yes	Yes	Yes	No <sup>4</sup>	Yes	Yes
Trigger	Yes	Yes	Yes	Yes	Yes	No <sup>4</sup>	Yes	Yes
Stored procedure	Yes	Yes	Yes	Yes	Yes	No <sup>4</sup>	Yes	Yes
Tablespace	Yes	Yes	No	?	No <sup>5</sup>	No <sup>1</sup>	Yes	Yes

Figure 2.8 – Comparing modern popular DBMS for main options

Indigo IT reported the results of benchmarking performance tests of the most demanded DBMS on the market - in separate test suites, the PostgreSQL open source database was three times better than MS SQL and Oracle DataBase. Testing was conducted to monitor new technologies and implement a technology development plan. For testing, the company specialists prepared test data sets identical for all DBMS. The object of the test was a 1 TB database consisting of 1 million business objects. The test duration for each DBMS is 10 hours. It was attended by the latest versions of the most sought-after customers of Indigo IT DBMS: Microsoft SQL Server, Oracle DataBase, PostgreSQL.

Total conducted 5 sets of tests: creation of complex structured documents, updating of difficult structured documents, document search, write file to database, getting file from database.

In table 2.9 presented tests results of the most demanded DBMS.

Table 2.9 – Tests results of the most demanded DBMS

DBMS	Creating documents (ms)	Updating documents (ms)	Search documents (ms)	Write file to DB (ms)	Getting file from DB (ms)
Microsoft SQL Server	66	62	11	181	162
PostgreSQL	22	20	11	179	165
Oracle DataBase	59	57	11	183	163

The strengths of PostgreSQL are:

- high-performance and reliable transaction and replication mechanisms;
- extensible embedded programming language system: PL / pgSQL, [PL / Perl, PL / Python, and PL / Tcl are supported as standard; In addition, you can use PL / Java, PL / PHP, PL / Py, PL / R, PL / Ruby, PL / Scheme, PL / sh and PL / V8, and there is also support for loading C-compatible modules;
- support from many programming languages: C \ C ++, Java, Perl, Python, Ruby, ECPG, Tcl, PHP and others.
- inheritance;
- easy extensibility [17].

## 2.6 Object-Relation Mapper Entity Framework

ORM (Object-Relational Mapping) is a programming technology that links databases with object-oriented programming language concepts, creating a “virtual object database.” There are both proprietary and free implementations of this technology. ORM libraries exist for many different programming languages. In general, the ORM technology allows you to design working with data in terms of classes, not data tables. It allows you to convert classes into data suitable for storage in a database, and the conversion scheme is determined by the developer. In addition, ORM provides a simple API interface for CRUD operations on data. Thanks to the ORM technology, there is no need to write SQL code to interact with the local database [18].

The advantages of ORM are:

- the presence of an explicit description of the database schema presented in terms of a programming language that is located and edited in one place;
- possibility to operate with elements of a programming language, i.e. classes, objects, attributes, methods, and not elements of the relational data model;
- the ability to automatically create SQL queries, which eliminates the need to use a language to describe the database structure (Data Definition Language) and the

data manipulation language (Data Manipulation Language) when designing a database and changing its schema, respectively;

- there is no need to create new SQL queries when transferring to another database management system, since the low-level ORM driver is responsible for this;
- ORM eliminates the need to work with SQL and study a significant amount of software code, which is often monotonous and error prone;
- the code generated by ORM is hypothetically tested and optimized, therefore no need to worry about testing it;
- advanced ORM implementations support mapping inheritance and composition to tables;
- ORM makes it possible to isolate the program code from the data storage details.

Entity Framework is a special object-oriented technology based on the .NET framework for working with data. If traditional ADO.NET tools allow you to create connections, commands and other objects for interacting with databases, then the Entity Framework is a higher level of abstraction that allows you to abstract from the database itself and work with data regardless of the type of storage. If at the physical level we operate with tables, indexes, primary and foreign keys, but at the conceptual level that the Entity Framework offers us, we already work with objects.

Entity Framework suggests three possible ways to interact with the database:

- database first: Entity Framework creates a set of classes that reflect a specific database model;
- model first: first, the developer creates a database model, by which the Entity Framework then creates a real database on the server;
- code first: the developer creates a data model class that will be stored in the database, and then the Entity Framework generates a database, its tables using this model [19].

## **2.7 Bootstrap framework in ASP.NET MVC 5**

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

Bootstrap - a modern assistant, interface developer, designers and webmasters, available for use under an open source license. The framework is very dynamic and regularly updated, so not all of it. Functions can be correctly supported by older browsers. Bootstrap has the following patterns: fonts, buttons, forms, tags, navigation, grid, JavaScript extensions.

The modern world is very dynamic, and web programmers are often faced with the choice of making fast prototypes and running them slowly and for a long time to remake the layout, bringing everything to perfect condition. Bootstrap perfectly implements the first approach.

At the same time, HTML, JavaScript and CSS code in Bootstrap is thought out in hundreds developers from around the world - all so that ordinary webmasters and web designers can easily customize the site's grid or embed the necessary elements

into the interface. Also, Bootstrap uses the dynamic style language LESS, which extends the capabilities of CSS: developers can control colors, create nested columns and variables.

Benefits of the Bootstrap framework:

- save time - achieved through the use of ready-made classes and design. This allows you to direct the saved energy and money on the development of additional functionality;
- adaptability (high speed and optimization, standardization of interfaces) - dynamic layouts are qualitatively displayed on a variety of devices without the need for changes to the markup;
- design - uniform templates and layout elements of the layout and all pages on the site as a whole. And at the same time, Bootstrap is cross-browser and is well displayed in all Safari, Firefox, IE, EDGE browsers and those based on Chromium (Webkit - based Blink engine: Yandex.Browser, Opera, Google Chrome);
- simplicity and openness - using Bootstrap is so simple that even beginner web developers cope with it, and open source code allows you to participate in the development, modify to fit your needs or just enjoy a good free solution.

ASP.NET MVC 5 projects by default already contain all the necessary Bootstrap files. The Bootstrap functionality is contained in the bootstrap.css style file and in the bootstrap.js script. When using Bootstrap it is very convenient to position the interface in the form of a table or grid, using rows and columns. So, if we look in the browser at the standard Index.cshtml view, which goes by default, we will see a similar positioning: the elements under the heading are arranged in three columns and comprise one line [20].

In figure 2.10 presented Bootstrap framework components.

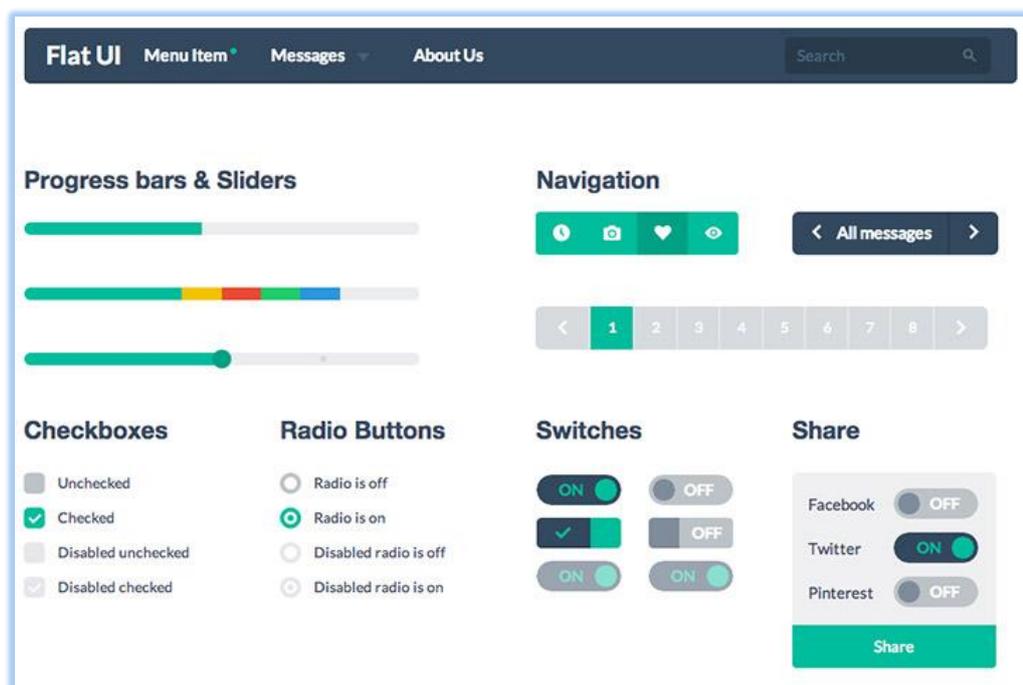


Figure 2.10 – Bootstrap framework components

## 3 Software for Information system

### 3.3 Structure of software

Software is the most important component of any information system. In general, information software is a program whose function is to solve certain problems. No system, even ideally designed, can function without software.

The structure of the software is presented in 3.1-picture.

It consists of:

#### 1. System software and OS.

Document Management System «Discipline» is implemented for workstations, such as, Windows 7/8/10, Linux. Works in any modern web-browsers: Internet Explorer, Microsoft Edge, Chrome, Opera, Mozilla, etc.

#### 2. Programming tools.

The program is implemented in the C# programming language. Integrated development environment is Microsoft Visual Studio, also using frameworks ASP.NET MVC 5 and Bootstrap, Object-Relation Mapping technology Entity Framework. Query language PL/pgSQL and Data Base Management System PostgreSQL.

#### 3. Application software.

Document Management System «Discipline» on the administrator's side includes:

- function “Create private offices for new users”. This function allows to create new users with self-private offices;
- function “Permission of access for different users”. With this function, Administrator of Document Management System can open access for users with different roles;
- function “Process review and monitoring”. With this function, Administrator of Document Management System can review of document flow process and monitoring each step of process.

Document Management System «Discipline» on the client side includes:

- function “Create own company and department in system ”. With this feature, client can create self-company and inside departments for document flow;
- function “Create and send documents to employees”. Client can quickly and without problems send any documents to employees;
- function “Post documents to the entire company or department”. This feature allows client post one or more documents to a lot of employees in own company or departments.

The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers. These software products comprise of programs written, which interact with the hardware at a very basic level. System software serves as the interface between the hardware and the end users.

In figure 3.1 presented basic structure of software Information System.

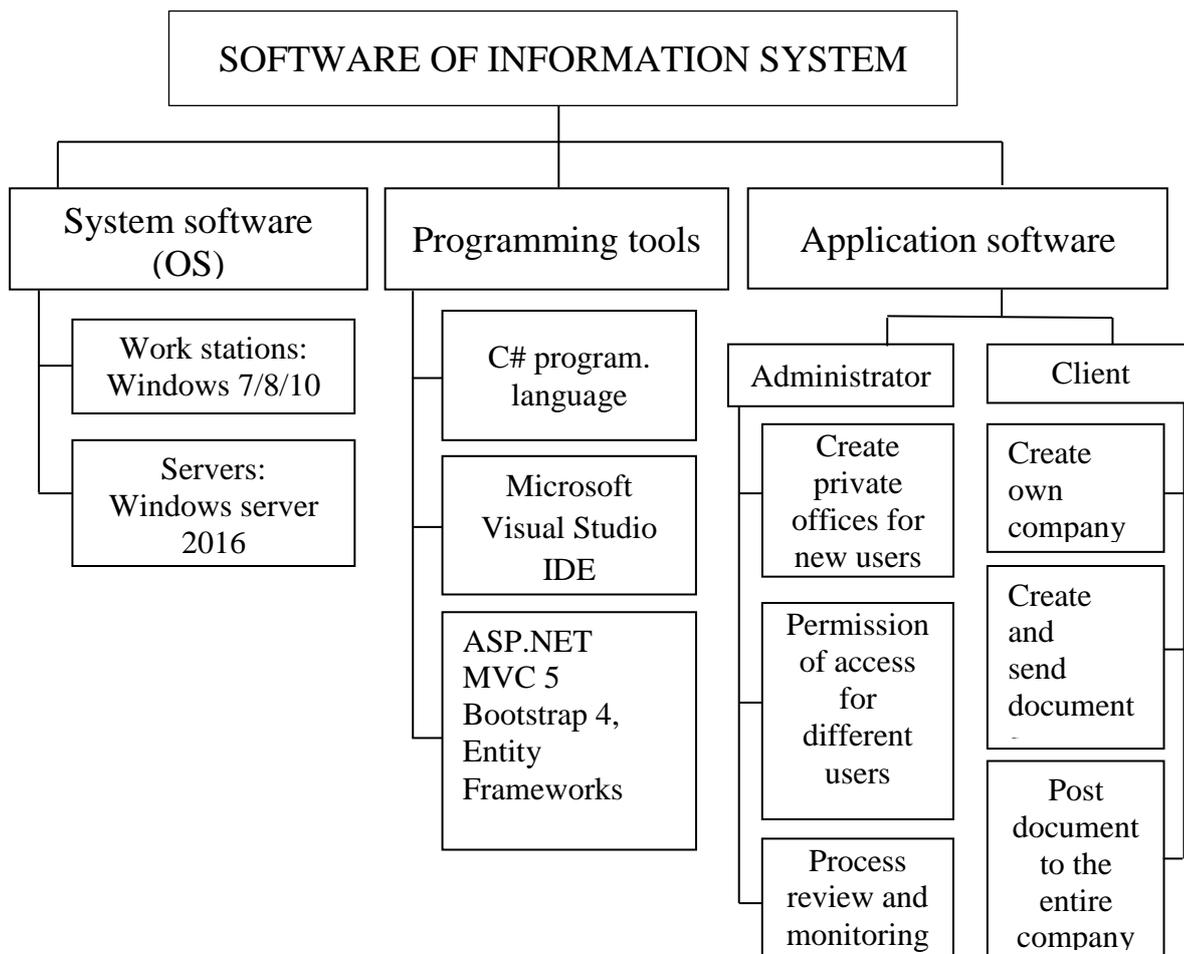


Figure 3.1– Structure of software IS

### 3.2 Description of the developed application software

3.2.1 General Information. The program is implemented in the C# programming language. Integrated development environment is Microsoft Visual Studio, also using frameworks ASP.NET MVC 5 and Bootstrap, Object-Relation Mapping technology Entity Framework. Query language PL/ pgSQL and Data Base Management System PostgreSQL.

Document Management System «Discipline» is implemented for workstations, such as, Windows 7/8/10, Linux. Works in any modern web-browsers: Internet Explorer, Microsoft Edge, Chrome, Opera, Mozila.

3.2.2 Functional purpose. This program is intended for implementation on a Document Management System «Discipline». The system performs document flow between company employees and individual departments.

3.2.3 Used technical tools. Following technical tools were used for realization and implementation of Document Management System “Discipline”:

- CPU: Intel(R) Core(TM) i5-7200U 2.50GHz;
- RAM: 8 Gb;
- hard drive: 1 Tb;

- removable Flash Drive disk: 32 Gb USB 2.0;
- network adapter: Realtek 8821CE Wireless LAN 802.11ac PCI-E NIC;
- video adapter:AMD Radeon RX 530 Graphics Card Video;
- printer Hewlett Packard dj 940c;
- standard keyboard and mouse.

3.2.4 Call and Download. The program is called through an Internet browser, such as Microsoft Internet Explorer. After launching the browser, the path in the address bar is specified as <https://www.discipline.kz>, after that shows the initial page of the Document Management System.

3.2.5 Input Data. The input data of the diploma work are documents of the company and each department. Also official documents of employees for post or send necessary documentations.

3.2.6 Output Data. The output data of diploma work is document flow and data base of company documents in Document Management System «Discipline».

### 3.3 Work algorithm with Programming tools

3.3.1 Work algorithm with Microsoft Visual Studio – IDE and C#. We can separately use a text editor and a compiler, manually compile and run the program in a console or terminal, however using a different development environment or IDE is a more convenient way. They, as a rule, contain a built-in text editor, have a connection with the compiler, allowing you to compile and run the program with a single mouse click, as well as many other various support features. In this case, for programming under Windows, as a development environment and for creating applications in C #, we will use the free and full-featured Visual Studio 2017 Community. In figure 3.2 presented open VS and creating new project.

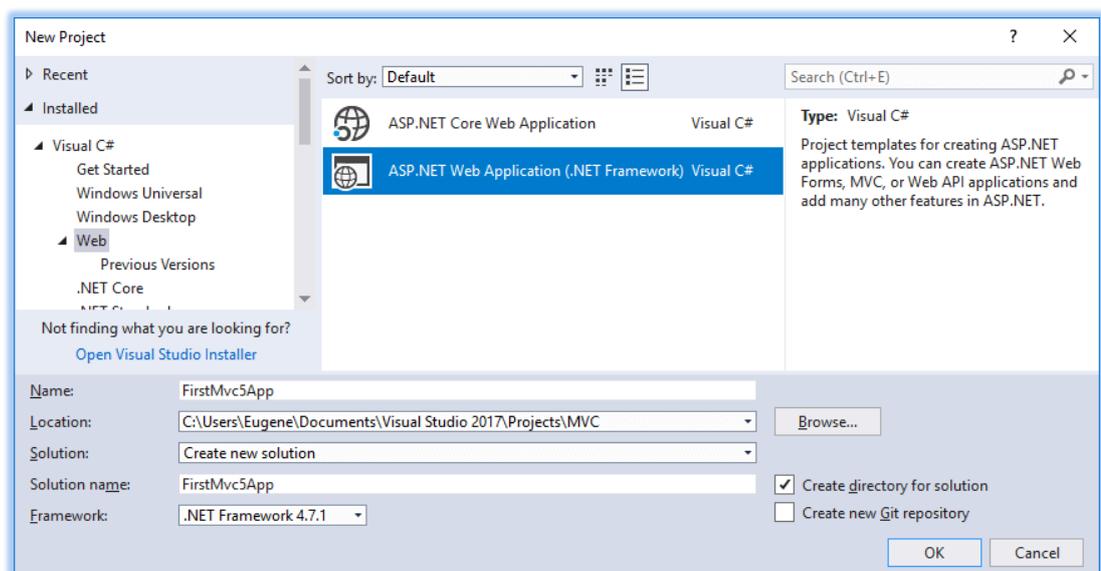


Figure 3.2 – Open VS and creating new project

3.3.2 Work algorithm with ASP.NET MVC framework. Open Visual Studio 2017 and in the File menu select the New Project ... item. The project creation dialog opens before us, where we need to go to the Web sub-tab and select the ASP.NET Web Application template (.NET Framework). Below in figure 3.3 presented choose MVC template for web-developing. Figure 3.4 change Authentication options.

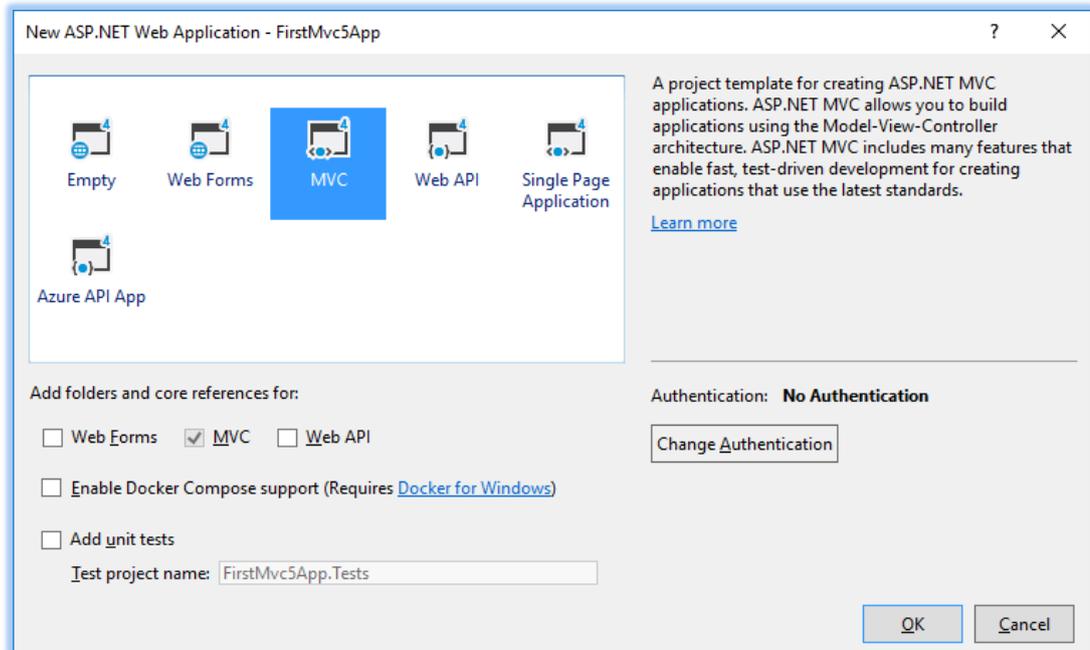


Figure 3.3 – Choose MVC template for web-developing

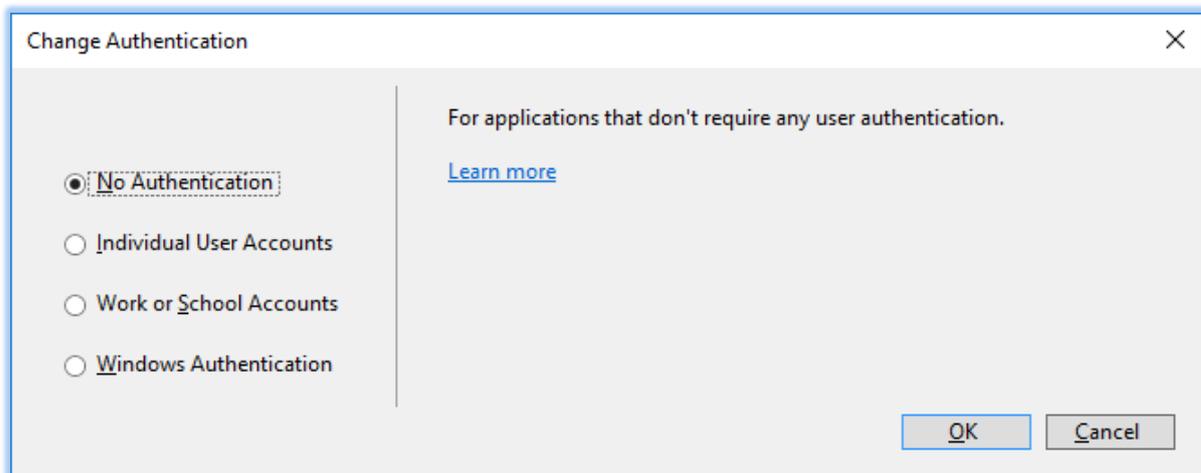


Figure 3.4 – Change Authentication

Click OK, and a new project is created. It already contains a branched structure and has some default content. Run the project for execution, and we will see some content that already exists by default in the application. Figure 3.5 presented default content in VS for first execution.

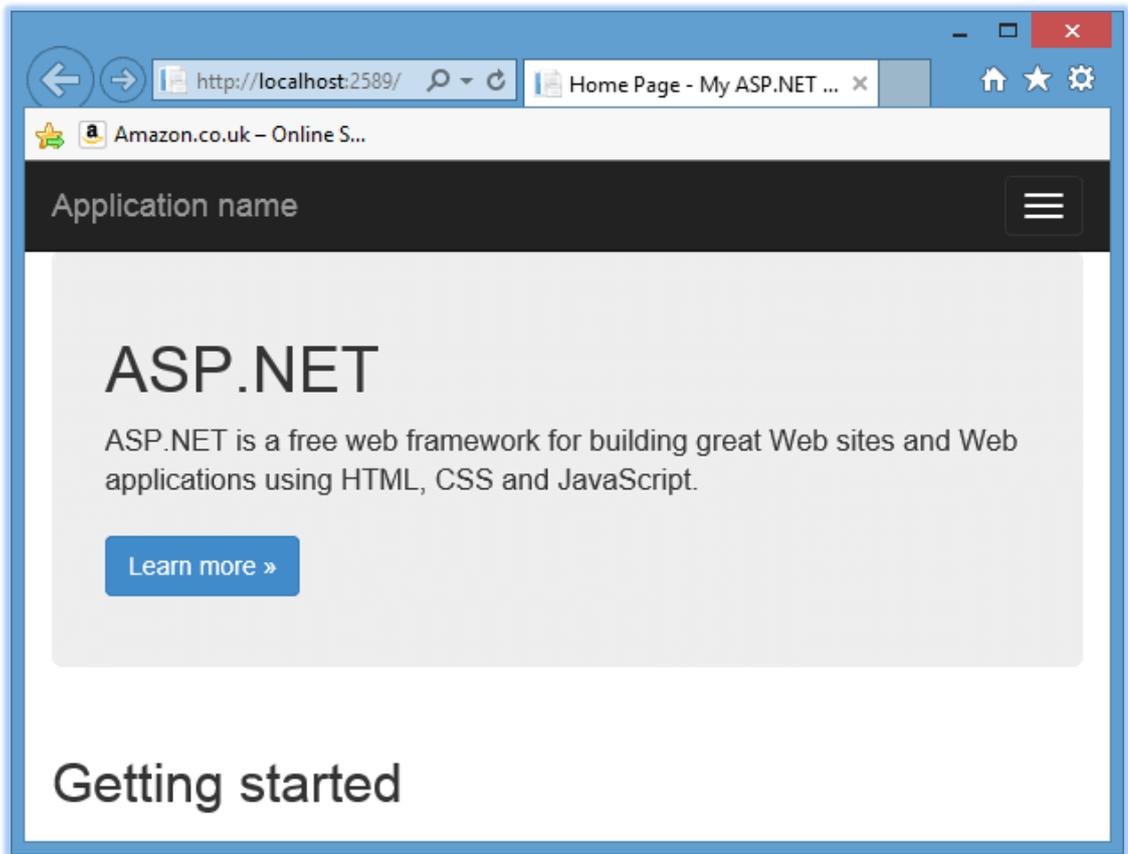


Figure 3.5 – Default content in VS for first execution

In figure 3.6 presented all this functionality is provided by the following project structure. Appendix A presented project-script «Web.config».

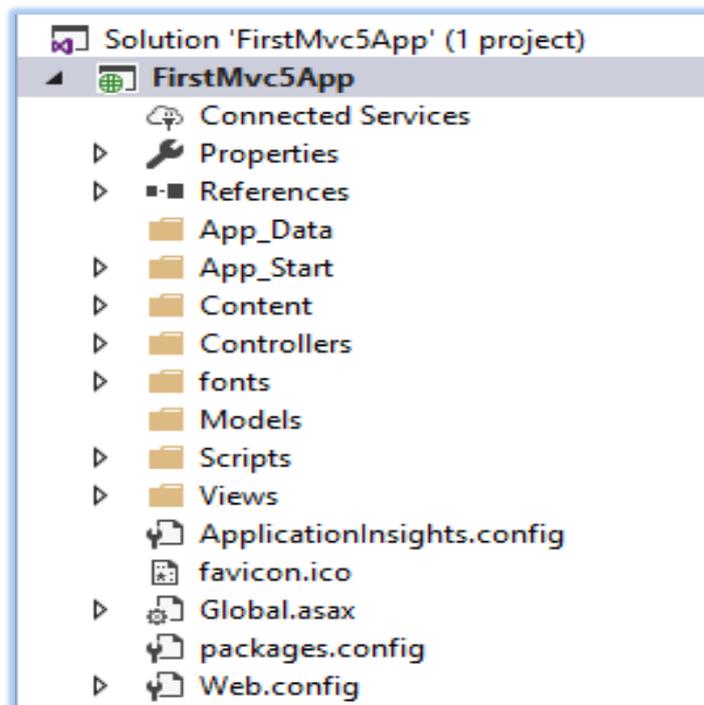


Figure 3.6 – MVC 5 project structure

3.3.3 Work algorithm with DBMS PostgreSQL 4. PostgreSQL is one of the most popular database management systems. PostgreSQL is supported for all major operating systems - Windows, Linux, MacOS.

Open PostgreSQL and create new Data Base for project. In figure 3.7 presented home page PostgreSQL.

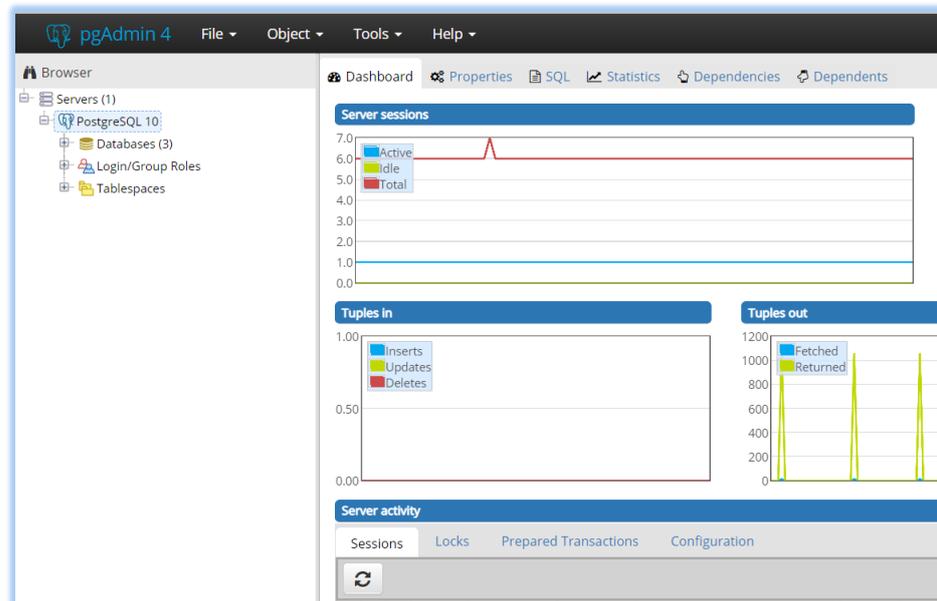


Figure 3.7 – Home page PostgreSQL

To create a database, use the CREATE DATABASE command, followed by the name of the database. To execute queries, we will use the pgAdmin graphical client, although you can also use the psql console client. Figure 3.8 presented Logical database ER - model. Below figure 3.9 has shown creating data base.

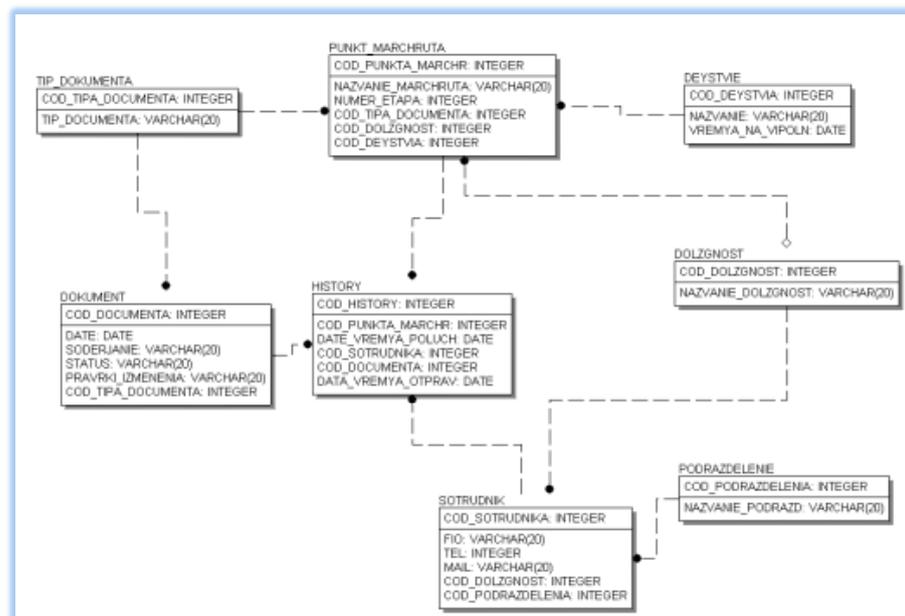


Figure 3.8 – Logical database ER- model

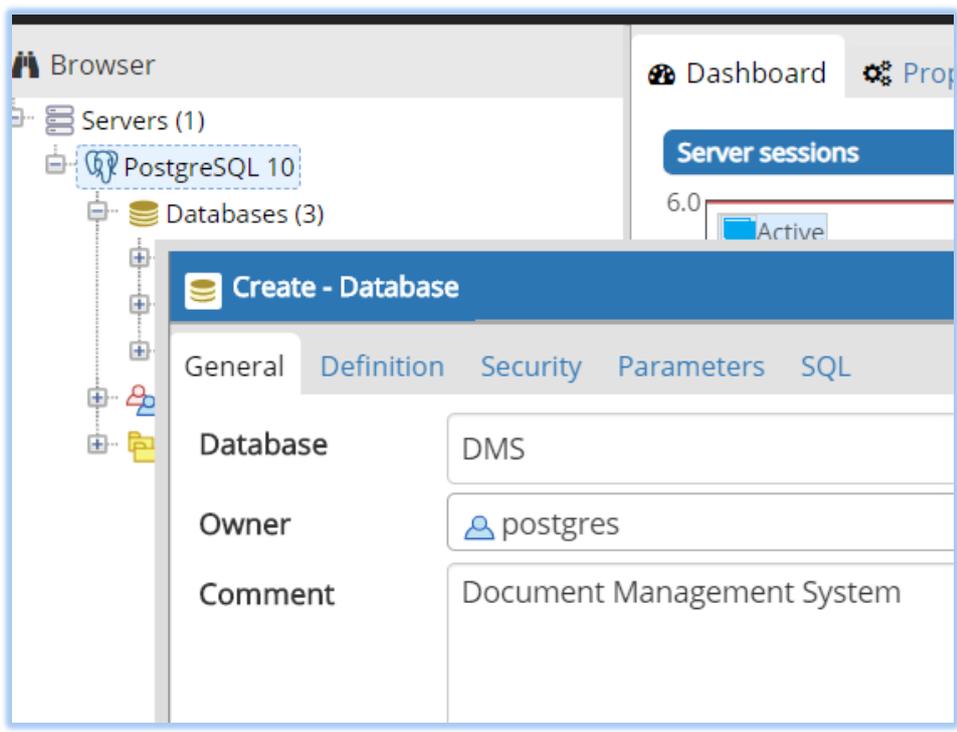


Figure 3.9 – Creating Data Base

To create a new database, open pgAdmin. In the left part of the program, select any database, for example, the standard postgres database, and click on it with the right mouse button. In the menu that appears, select the Query Tool ... item, and in the central part of the program a field for entering the SQL code will open. To execute the code, click on the lightning icon, and after that the usersdb database will be created. To see our database, right-click on the Databases node in the left part and select Refresh in the context menu. An update will occur, and we will see the database created. Figure 3.10 presented table «Companies»

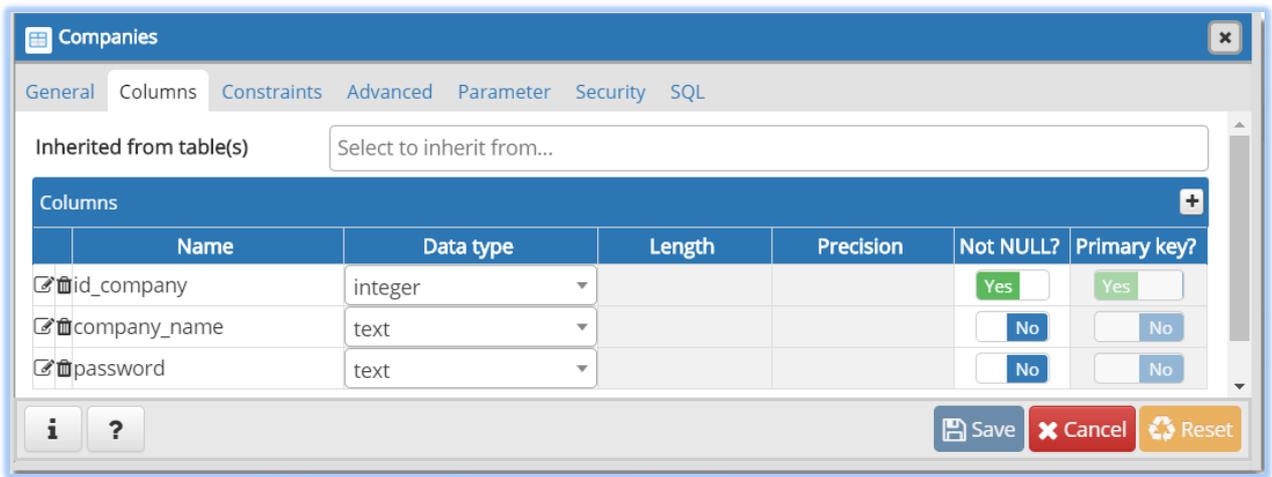


Figure 3.10 – Table «Companies»

To create a database, use the CREATE DATABASE command, followed by the name of the database. To create a new database, open pgAdmin. In the left part of the program, select any database. In figure 3.11 presented table «Documents». Appendix B presented all tables which used in development of data base in PostgreSQL Data Base Management System.

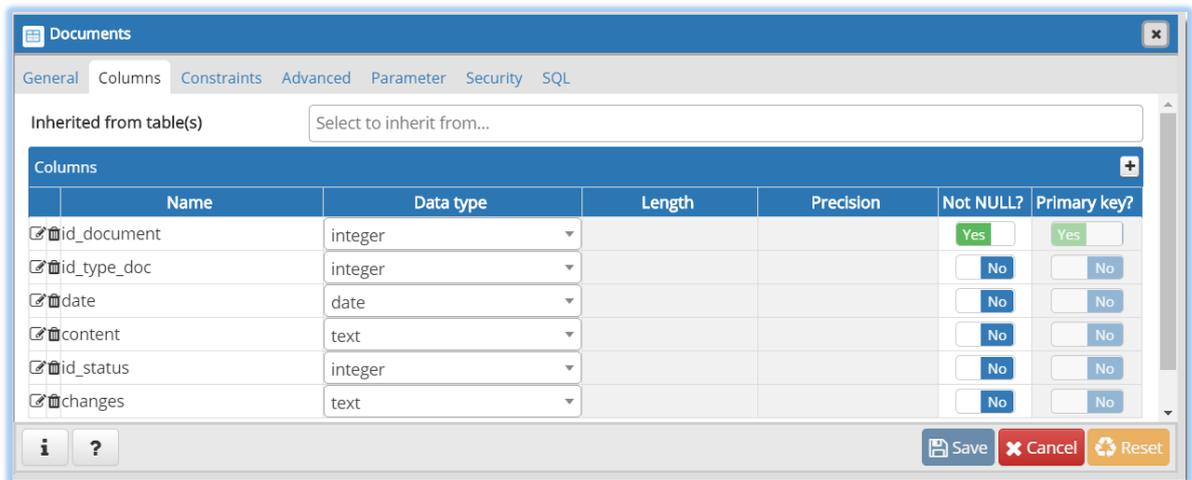


Figure 3.11 – Table «Documents»

3.3.4 Work algorithm with ORM Entity Framework. There is recommended to use the Entity Framework for working with data in ASP.NET MVC, although its use is not necessarily and entirely dependent on the preferences of the developer. The advantage of this framework is that it allows you to abstract from the structure of a specific database and conduct all data operations through the model. In figure 3.12 presented Installation Entity Framework.

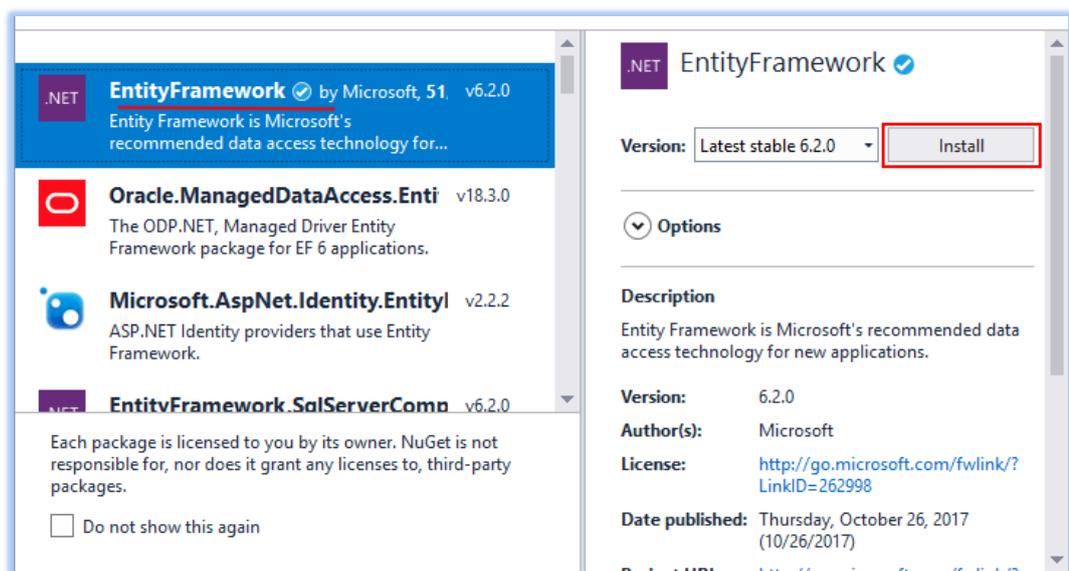


Figure 3.12 – Installation Entity Framework

### 3.4 Realization of Information system

Development of Document management System «Discipline» have index page interface which presented in figure 3.13. Also there are registration and login forms, which can provide open personal cabinet. Figure 3.14 presented interface. This data save in data base.

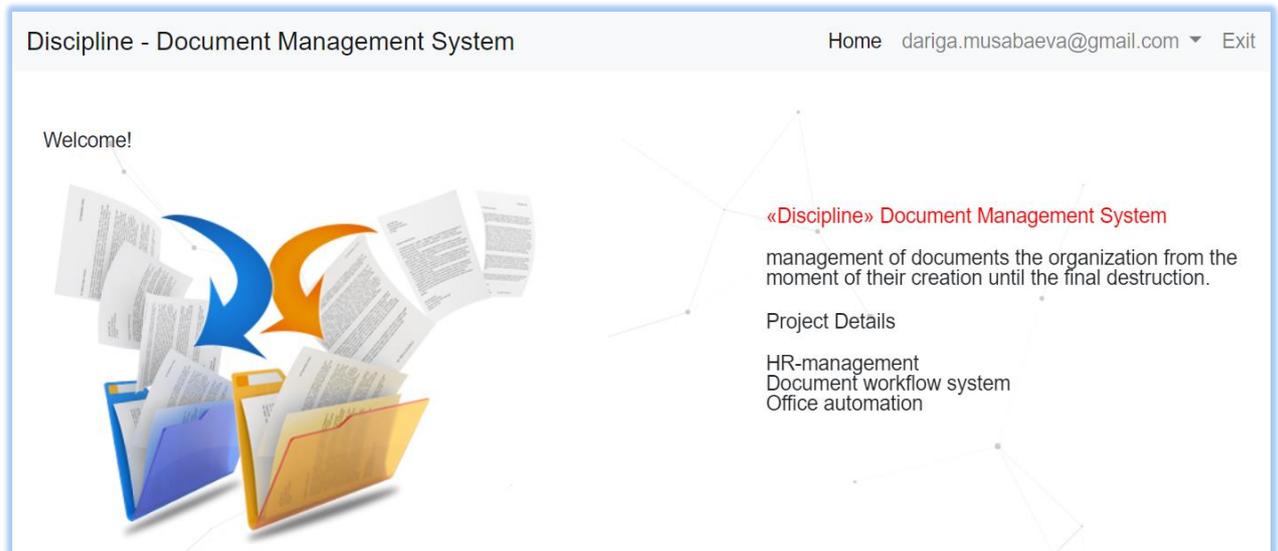


Figure 3.13 – Main page interface of system

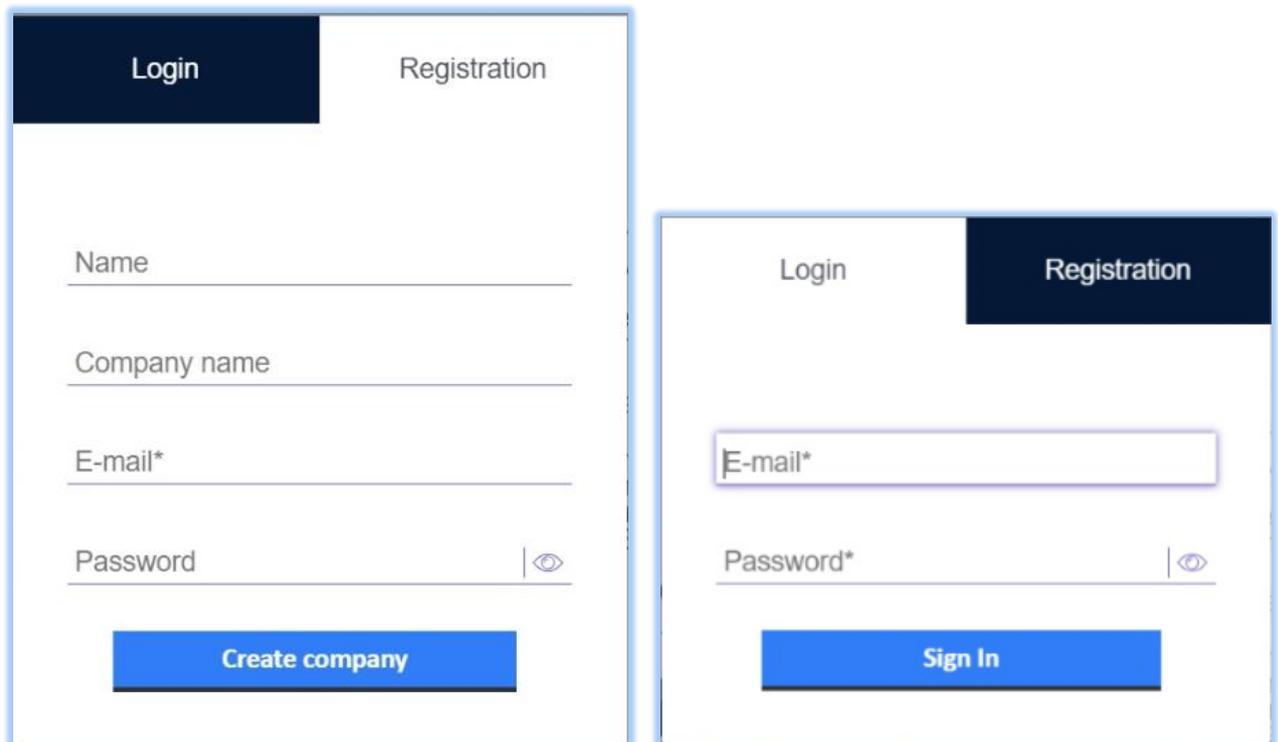


Figure 3.14 – Registration and Login forms in system

Also next options of system are create and open personal cabinet. It's presented in Figure 3.15.

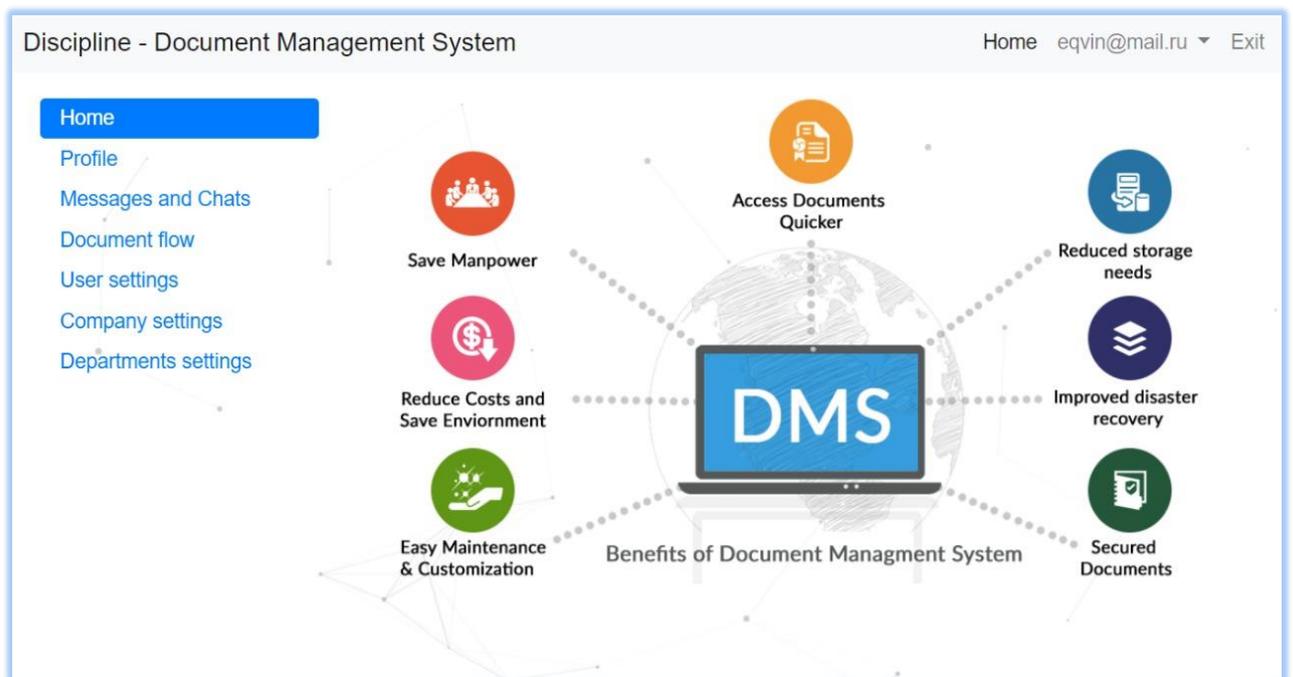


Figure 3.15 – Home and personal office for users

Below presented another options of system figure confirm registration for users and allows document flow in figure 3.16. Status documents presented in figure 3.17. Also there are Setting for users and company. We can see all company users and create or delete users.

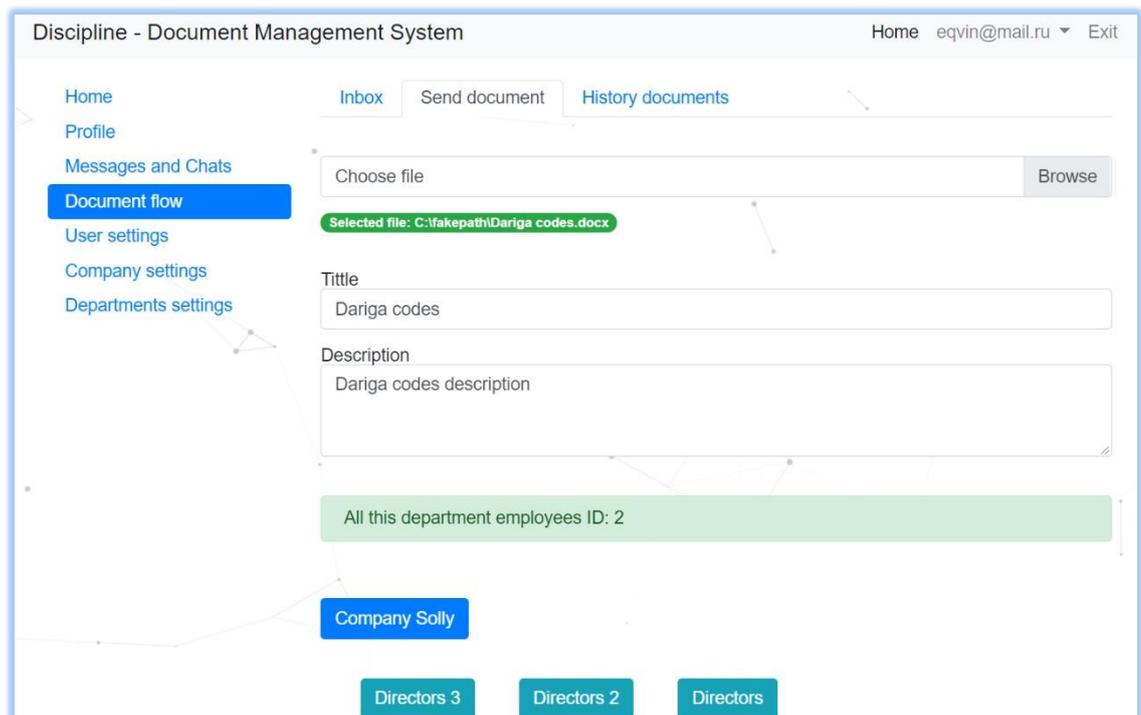


Figure 3.16 – Send document (document flow)

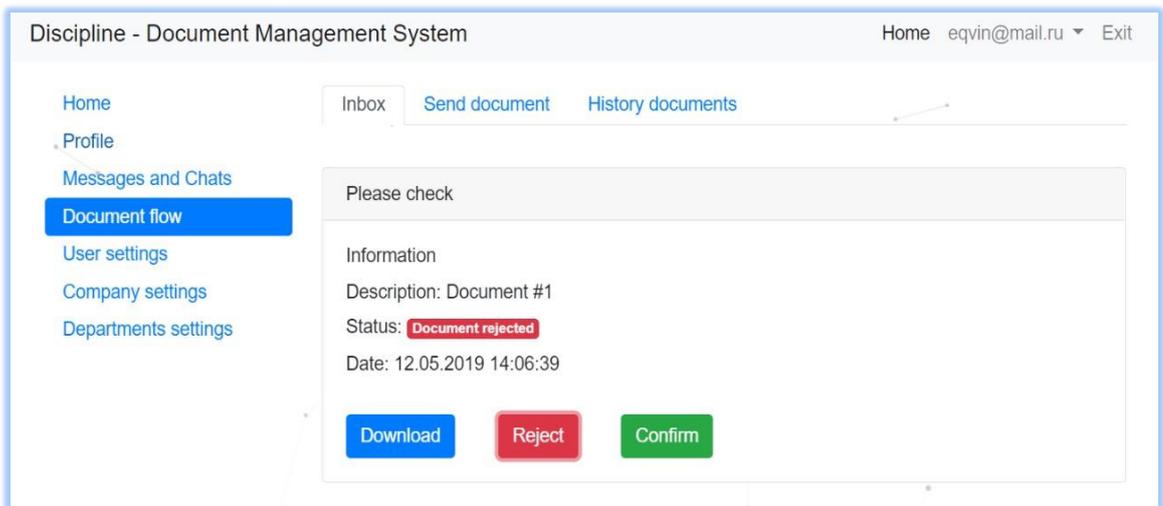


Figure 3.17 – Status of documents

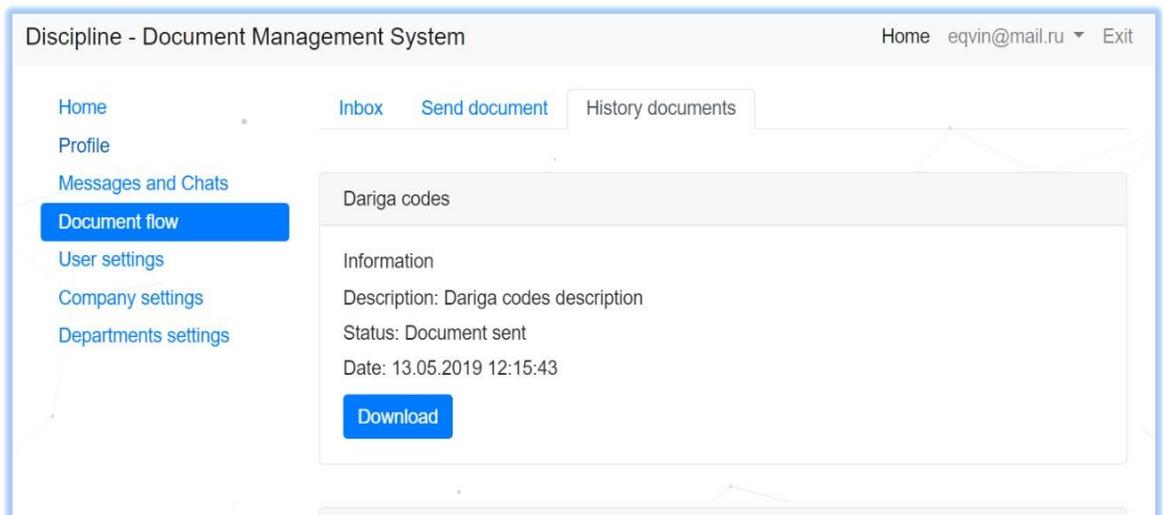


Figure 3.18 – History documents

History of all documents presented in figure 3.18. Also have settings of all users and chats between users presented in figure 3.19.

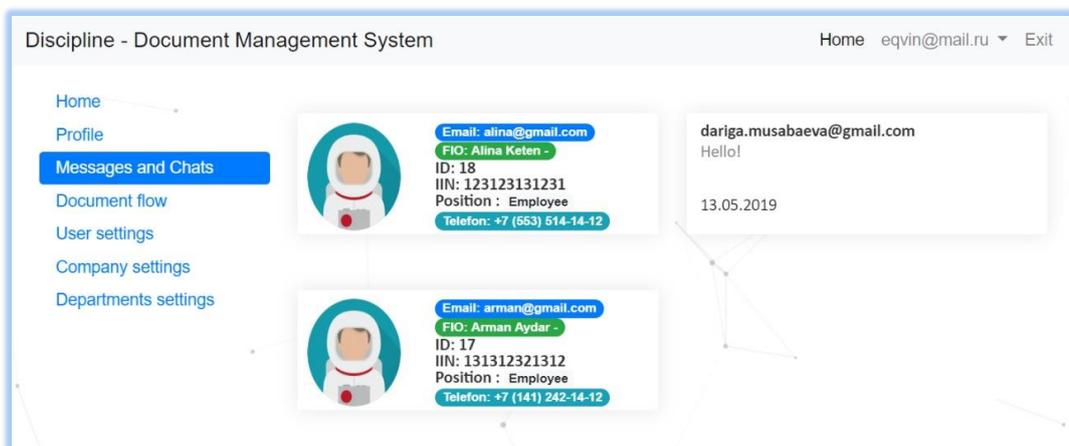


Figure 3.19 – Messages and chats

## CONCLUSION

The main purpose of the EDMS (Electronic Document Management System) is to organize the storage of electronic documents, as well as work with them (in particular, their search both by attributes and by content). In the EDMS, changes in documents, deadlines for the execution of documents, movement of documents should be automatically monitored, and all versions and sub-versions should be monitored. A comprehensive EDMS can cover the entire office management cycle of an enterprise or organization — from setting a task to creating a document to writing it off into an archive, ensuring centralized storage of documents in any format, including complex composition documents. EDMS can integrate disparate document flows of geographically distant enterprises into a single system. They allow provide flexible management of documents both through hard route determination and free routing of documents. In the EDMS, a rigid separation of user access to various documents can be implemented, depending on their competence, position and authority assigned to them. In addition, the EDMS must be configured to the existing organizational structure and system of enterprise workflow, as well as integrate with existing corporate systems.

In first part of this work was collect basic data (Knowledge Base) about subject area, especially, HR-management sphere. There were overview office work and document flow in company, a lot of information about office automation. Also was research of basic principles and purpose, function and effectiveness, requirement, classification and types of Electronic Document Management System.

Second part of this work was about Methods of implementing Information System. This part shows basic technologies and methods which used for development program: about Integrated Development Environment, Data Base, special frameworks, etc.

Last third part of this work shows programming realization, structure of software IS and logical scheme. Most important work steps implementation, work implemented in C# programming language with Visual Studio IDE and ASP.NET MVC web framework, Data Base Management System PostgreSQL and Object Relation Mapper Entity framework.

## REFERENCES

- 1 Werner J. M., DeSimone R. L. Human Resource Development 6E / South-Western, Cengage Learning – 2006. – P. 136-158. // Electronic version in web-site // <https://landing.tefexpress.co.uk/human-resource-development.pdf>
- 2 Senyuçel Z. Managing the Human Resource in the 21st Century / Wiley – 2013. – 298 p. // Electronic version in web-site // <https://bookboon.com/en/hrm/managing-the-human-ressource-ebook>
- 3 Green W. B. Electronic Document Management System (EDMS) first editions / Academic Press – 2006. – P. 179-183.
- 4 Sutton M. J. D., Document Management for the Enterprise: Principles, Techniques, and Applications 1st edition/ Wiley – 2010. – 192 p.
- 5 Smallwood R. F Managing Electronic Records: Methods, Best Practices, and Technologies / John & Sons Inc – 2013. – P. 166-168. // Electronic version in web-site // <https://www.bookdepository.com/Managing-Electronic-Records>
- 6 Chowdhury K. Mastering Visual Studio 2017 / Packt – 2017. – 249 p.
- 7 Wilson R. Visual Studio 2017: An In depth Guide Into the Essentials of Visual Studio / CreateSpace Independent Publishing – 2017. – 46 p.
- 8 Japs Ph., Dewey B. Building Web Applications with Visual Studio 2017 / Apress – 2011. – 357 p.
- 9 Hejlsberg A., Torgersen M., Wiltamuth S., Golde P. C# Programming Language 4th Edition / Addison-Wesley Professional – 2010. – P. 241-242.
- 10 Evjen B, Watson K., Nagel C., Skinner M. Professional C # 4.0 and .NET 4 / Wiley – 2016. – 955 p.
- 11 Gastón C. Learning Object-Oriented Programming / Packt Publishing – 2015. – P. 122-124. // Electronic version in web-site // <http://dinus.ac.id//OOP.pdf>
- 12 Clark D. Beginning C # Object-Oriented Programming / Apress – 2011. – P. 333-334. Electronic version in web-site // <https://www.redbrick.dcu.ie/OOP.pdf>
- 13 Sanderson S. Pro ASP.NET MVC Framework / Apress – 2009. – 478 p.
- 14 Galloway J., Wilson B., Scott A. K., Matson D. Professional ASP.NET MVC 5 / Wrox – 2014. – P. 336-337.
- 15 Regina O., Leo S. PostgreSQL: Up and Running / O'Reilly Media – 2012. – 106 p. // Electronic version in web-site // <http://hsd1.ga.PostgreSQL.pdf>
- 16 Shazad A., PostgreSQL Developer's Guide / Packt – 2015. – 137 p.
- 17 Geschwinde E., Schonig H. PostgreSQL Developer's Handbook / Sams – 2002. – 137 p.
- 18 Lerman J., Miller R. Programming Entity Framework: Code First / O'Reilly Media – 2011. – P. 84-86.
- 19 Naylor L. ASP.NET MVC with Entity Framework and CSS / Apress – 2016. – P. 580-582.
- 20 Shenoy A., Sossou U. Learning Bootstrap / Packt – 2014. – P. 212-213.

## Appendix A

### ASP.NET MVC project-script « Download File And File Status»

```
public FileResult DownloadFile(string fileName, int id)
{
    using (DisDBContext db = new DisDBContext())
    {
        DisDBContext db2 = new DisDBContext();
        #region Current User
        var newUser = User as CustomPrincipal;
        #endregion
        if (newUser != null)
        {
            user _userAdmin = db.Users.FirstOrDefault(m => m.id == newUser.UserId &&
m.email == newUser.UserEmail);
            if (_userAdmin != null)
            {
                if (id != 0)
                {
                    document currentDocument = db.Documents.FirstOrDefault(m => m.id == id);
                    currentDocument.status = 1;
                    db.SaveChanges();
                }
                string dir = fileName.Substring(0, fileName.LastIndexOf("/") + 1);
                string file = fileName.Substring(fileName.LastIndexOf("/") + 1);

                var filepath = System.IO.Path.Combine(Server.MapPath(dir), file);
                return File(filepath, MimeMapping.GetMimeMapping(dir), file);
            }
        }
        return null;
    }
}

public JsonResult DocumentStatus(int id, int status)
{
    using (DisDBContext db = new DisDBContext())
    {
        BaseModel model = new BaseModel();
        #region Current User
        var newUser = User as CustomPrincipal;
        user _userAdmin = db.Users.FirstOrDefault(m => m.id == newUser.UserId && m.email
== newUser.UserEmail);
        #endregion
        if (_userAdmin != null)
        {
            document currentDocument = db.Documents.FirstOrDefault(m => m.id == id);
            currentDocument.status = status;
            db.SaveChanges();
            return Json(id, JsonRequestBehavior.AllowGet);
        }
        else { return Json("error", JsonRequestBehavior.AllowGet); }
    }
}
```

## Appendix B

### ASP.NET MVC project-script « Upload File»

```
[HttpPost]
public JsonResult UploadFile(string description, string title, string send_id)
{
    using (DisDBContext db = new DisDBContext())
    {
        #region Current User
        var newUser = User as CustomPrincipal;
        if (newUser != null)
        {
            user _userAdmin = db.Users.FirstOrDefault(m => m.id == newUser.UserId &&
m.email == newUser.UserEmail);
            if (_userAdmin != null)
            {
                #endregion
                foreach (string file in Request.Files)
                {
                    var upload = Request.Files[file];
                    if (upload != null)
                    {
                        // получаем имя файла
                        string fileName = System.IO.Path.GetFileName(upload.FileName);

                        var path = Server.MapPath("~/Files/Documents");
                        Directory.CreateDirectory(path);
                        if (send_id == "company")
                        {
                            send_id = "company_" + _userAdmin.company_id.ToString();
                        }

                        document documentFile = new document();
                        documentFile.owner_id = _userAdmin.id.ToString();
                        documentFile.send_id = send_id;
                        documentFile.description = description;
                        documentFile.title = title;
                        documentFile.status = 0;
                        documentFile.date = DateTime.Now;
                        db.Documents.Add(documentFile);
                        db.SaveChanges();

                        fileName = fileName.Replace(fileName.Substring(0,
fileName.LastIndexOf('.')), documentFile.id.ToString());

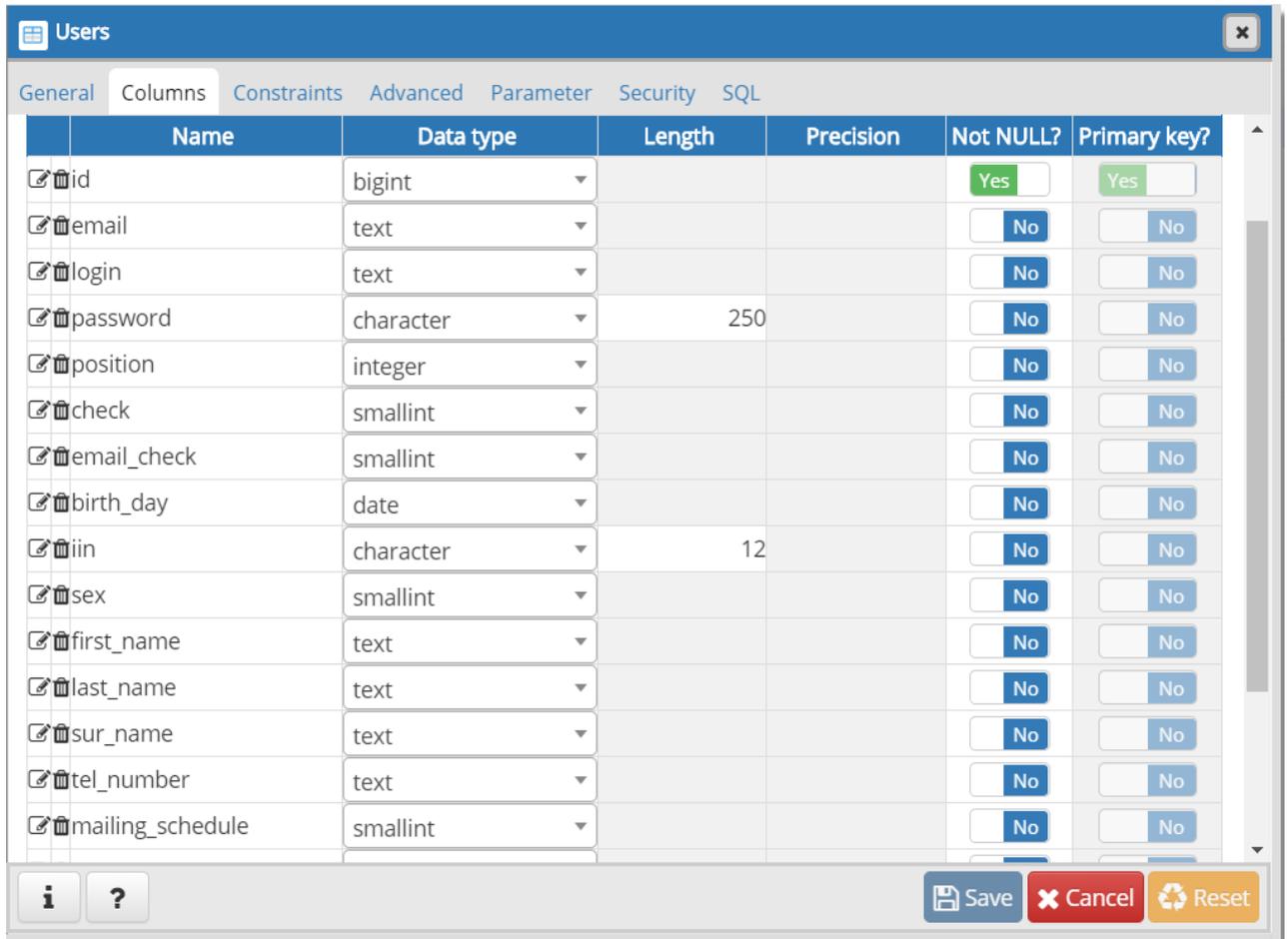
                        upload.SaveAs(Server.MapPath("~/Files/Documents/" + fileName));
                        documentFile.file_name = fileName;
                        documentFile.file_path = "~/Files/Documents/" + fileName;
                        db.SaveChanges();
                    }
                }
            }
        }
    }
}
```

## Appendix B continuation

```
        return Json("File uploaded");
    }
}
}
return Json("Error");
}
}
public JsonResult DepartmentUser(int dptId)
{
    using (DisDBContext db = new DisDBContext())
    {
        BaseModel model = new BaseModel();
        #region Current User
        var newUser = User as CustomPrincipal;
        user _userAdmin = db.Users.FirstOrDefault(m => m.id == newUser.UserId && m.email
== newUser.UserEmail);
        #endregion
        if (_userAdmin != null)
        {
            foreach (user Users in db.Users.Where(p => (p.department_id == dptId && p.id !=
_userAdmin.id)).OrderByDescending(o => o.id))
            {
                if (Users == null)
                {
                    break;
                }
                model.Users.Add(Users);
            }
            return Json(model);
        }
        else { return Json("error", JsonRequestBehavior.AllowGet); }
    }
}
}
```

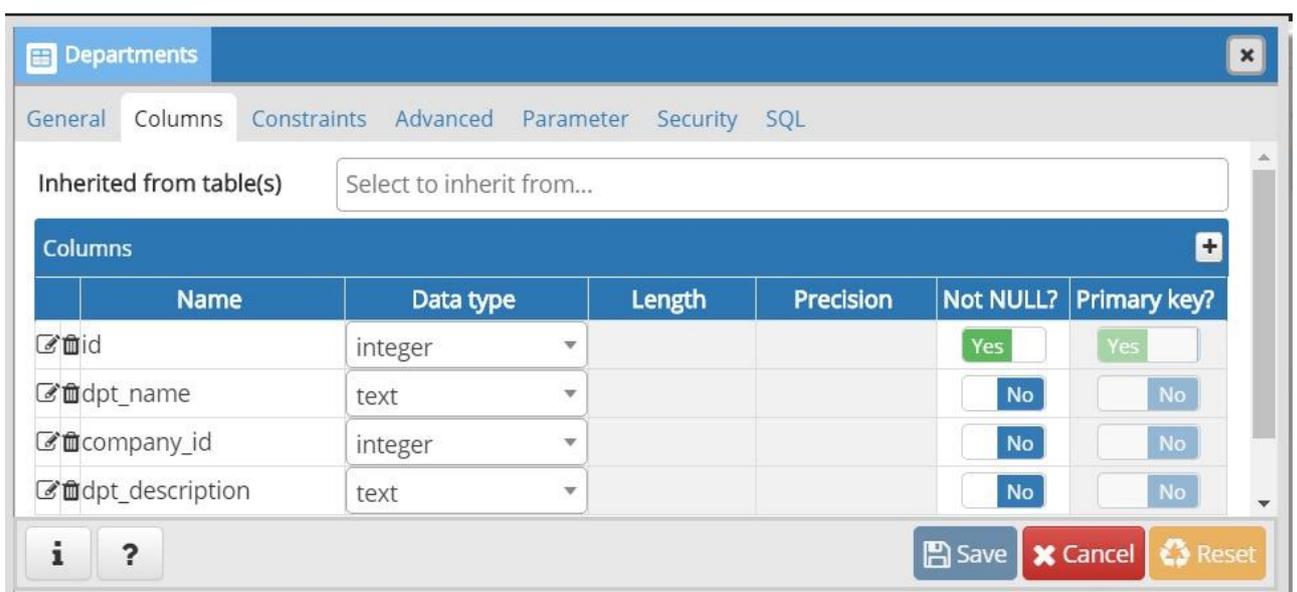
## Appendix C

### Development tables of data base in DBMS PostgreSQL



Name	Data type	Length	Precision	Not NULL?	Primary key?
id	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
email	text			<input type="checkbox"/>	<input type="checkbox"/>
login	text			<input type="checkbox"/>	<input type="checkbox"/>
password	character	250		<input type="checkbox"/>	<input type="checkbox"/>
position	integer			<input type="checkbox"/>	<input type="checkbox"/>
check	smallint			<input type="checkbox"/>	<input type="checkbox"/>
email_check	smallint			<input type="checkbox"/>	<input type="checkbox"/>
birth_day	date			<input type="checkbox"/>	<input type="checkbox"/>
iin	character	12		<input type="checkbox"/>	<input type="checkbox"/>
sex	smallint			<input type="checkbox"/>	<input type="checkbox"/>
first_name	text			<input type="checkbox"/>	<input type="checkbox"/>
last_name	text			<input type="checkbox"/>	<input type="checkbox"/>
sur_name	text			<input type="checkbox"/>	<input type="checkbox"/>
tel_number	text			<input type="checkbox"/>	<input type="checkbox"/>
mailing_schedule	smallint			<input type="checkbox"/>	<input type="checkbox"/>

Figure C.1 – Table «Users»



Inherited from table(s)

Name	Data type	Length	Precision	Not NULL?	Primary key?
id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
dpt_name	text			<input type="checkbox"/>	<input type="checkbox"/>
company_id	integer			<input type="checkbox"/>	<input type="checkbox"/>
dpt_description	text			<input type="checkbox"/>	<input type="checkbox"/>

Figure C.2 – Table «Departments»

## Appendix C continuation

Employees

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_employee	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	last_name	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	first_name	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	e-mail	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_position	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_division	integer			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.3 – Table «Employees»

Positions

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_position	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	position_name	text			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.4 – Table «Positions»

Documents

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_document	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	id_type_doc	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	content	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_status	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	changes	text			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.5 – Table «Documents»

## Appendix C continuation

Routes

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_route	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	route_name	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	stage	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_type_doc	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_position	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_action	integer			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.6 – Table «Users»

Status

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_status	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	status_name	text			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.7 – Table «Status»

Histories

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_history	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	date_of_rec	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	time_of_rec	abstime			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_employee	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_doc	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date_dep	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	time_dep	abstime			<input type="checkbox"/> No	<input type="checkbox"/> No

Figure C.8 – Table «Histories»