

**MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF
KAZAKHSTAN**

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY JSC

FACULTY OF INFORMATION TECHNOLOGY

Rodionova A.V.

**The development of the universal web-application for the resource
management of programming school**

DIPLOMA PROJECT

Major 5B070300 – Information Systems

Almaty 2016

**MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF
KAZAKHSTAN**

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY JSC

FACULTY OF INFORMATION TECHNOLOGY

Department of Information System

APPROVED FOR DEFENCE

Head of the Department,
Candidate of Technical Sciences,
Professor of Russian Academy
of Natural History

V.V.Serbin

« ____ » _____ 2016

**Diploma Project
EXAPLANATORY TEXT**

**Topic: The development of the universal web-application for the resource
management of programming school**

STUDENT:

GROUP IS 126 R
RODIONOVA ALEXANDRA
« ____ » _____ 201_ г

SUPERVISOR:

SENIOR LECTURER
ZHUANY SHEV I.O.
« ____ » _____ 201_

REVIEWER:

(POSITION, NAME)
« ____ » _____ 201_

(signature)

Almaty 2016

International University of Information Technology
Faculty of Information Technologies

Department of IS&MM
Major 5B070300 – Information Systems

Diploma Paper Assignment

Student

Rodionova Alexandra Valeryevna

Diploma paper (project) topic

“The development of the universal web-application for the resource management of programming school”

Approved by IITU order № ____ dated

«____» _____ 20__

Diploma paper submission date _____

Diploma paper initial data

- Analysis worksheet of similar projects
- Primeface library
- IntelliJ IDE.

Details of computations and explanations (list of issues due to be addressed)

- Study the problem statement
- Analysis of the analogical web-applications
- Review of the technologies used
- The development of the universal web-application for the resource management of programming school
- Rationale for the economic efficiency
- Compliance with the rules of labor safety

CD containing the digital version of diploma paper and attachments

- Diploma paper
- Source code
- Presentation

Consultations on diploma paper (with related project chapters named)

Chapter	Advisor/Consultant	Signature, date		
		Assignment given		Assignment received
Consultant on Economics	Berdykulova G.M. Candidate of Economic Science, Professor			
Consultant on labor safety and industrial ecology	Davletova S.K. Candidate of Biological Science, Assistant professor			
English language consultant	Pak N.S Candidate of Philological Sciences, Docent			
Compliance monitor	Syrymbayeva A.M. Master of Technical Science, Lecturer			

Date « ____ » _____ 20 ____

Research advisor _____
(signature)

Assignment received by _____
(signature)

Diploma paper writing schedule

International University of Information Technology

Student of IS126-R group 4th course

Rodionova Alexandra Valeryevna

Topic: “The development of the universal web-application for the resource management of programming school”

№	Assignment	Submission date	Comments
1.	Creation of the graduation paper writing schedule; submission to Department	30 November	Done
2.	Collection, study, processing, analyzing and generalizing data	November – December	Done
3.	Drafting and submission to the Research advisor Introduction Chapter 1 Chapter 2 Chapter 3 Chapter 4 Chapter 5 Conclusion	January – February	Done
4.	Revision of the graduation paper with due consideration of the advisor's comments	March -April	Done
5.	Submission of the completed graduation paper to the Research advisor	15 April	Done
6.	Pre-defence	4 May	Done
7.	Submission to the reviewer for approval	25 May	Done

8.	Compilation of the graduation paper presentation for the State Attestation Commission	May	Done
9.	Graduation paper presentation to SAC	25 May - 13 June	

Student Rodionova A.V.

(signature)

Research advisor Zhuanyshv I.O.

(signature)

Date «____» _____ 20____

CONTENT

LIST OF TERMS AND ABBREVIATIONS	
INTRODUCTION	
1. OVERVIEW AND ANALYSIS OF THE WEB APPLICATIONS IN EDUCATION AREA	
1.1 Description of the study.....	
1.2 IT education in Kazakhstan	
1.3 Online studying programming.....	
1.3.1 The advantages of distance learning.....	
1.3.2 The disadvantages of distance learning	
1.4 Programming for kids	
1.5 Comparison table of similar products.....	
1.6 Formulation of the problem.....	
1.7 Conclusion for chapter.....	
2. TECHNOLOGY REVIEW	
2.1. Overview of the used software	
2.1.1 Choosing the development environment.....	
2.1.2 Choosing the database schema	
2.1.3 Database modelling tool	
2.2 Overview of the used technologies.....	
2.2.1 Choose of the design pattern	
2.2.2 Features of JSF 2.2	
2.2.3 Hibernate.....	
2.3 Use cases.....	
2.3.1 Separation of roles	
2.3.2 Usage scenarios	
2.4 Conclusion for chapter.....	
3. IMPLEMENTATION OF THE WEB-APPLICATION	
3.1 Database structure of the project	
3.2 Project structure description	
3.2.1 Model part.....	
3.2.2 View part	
3.2.3 Controller part.....	
3.3 Project interface and functionality description.....	
4. ORGANIZATIONAL-ECONOMIC PART	
4.1 Selection and justification of the calculation method	
4.2 Calculation of costs and profit.....	

4.2.1	Material costs.....
4.2.2	Basic salaries and taxes
4.2.3	Computer time expenses.....
4.2.4	Cost of tools
4.2.5	Total cost
4.2.6	Profit
4.3	Conclusion for chapter.....
5.	LABOR SAFETY AND INDUSTRIAL ECOLOGY.....
5.1	Characteristics of workplace
5.2	Analysis of dangerous and harmful factors
5.2.1	Microclimate at the workspace.....
5.2.2	Noise levels at the workspace.....
5.2.3	Artificial and natural lighting
5.3	Health and safety during the work.....
5.3.1	Work and rest while using the computer
5.3.2	Requirements for the organization and the workplace equipment
5.4	Health and safety during emergencies.....
5.4.1	Electrical safety
5.4.2	Fire safety
5.5	Calculations on safety and labor protection
5.5.1	Artificial and natural illumination
5.5.2	Calculations of ventilation.....
5.5	Conclusion for chapter.....
	CONCLUSION.....
	REFERENCES
	APPENDIX A – Hibernate configuration file.....
	APPENDIX B – Navigation configuration file
	APPENDIX C – Security managing bean

АННОТАЦИЯ

В данной работе рассматриваются проблемы обучения языкам программирования на платной основе. Основными вопросами являются: создание универсального веб-приложения для управления ресурсами школы программирования в целях получения прибыли и предоставление инструмента для изучения языков программирования студентами. В качестве инструмента для создания пользовательского интерфейса и всех необходимых функций back-end-a используются возможности языка программирования Java, а также фреймворка для создания веб-приложений JSF. В результате данной работы была создана система, которая позволяет размещать все необходимые материалы для обучения студентов языкам программирования.

Данная дипломная работа состоит из введения, перечня сокращений, 5 разделов, заключения, списка литературы и приложения. Во введении приводится актуальность данной дипломной работы, а также цели и задачи. В первом разделе приводится анализ аналогов, обзор состояния сферы образования программированию в Казахстане и мире, обзор преимуществ и недостатков дистанционного обучения. Во втором разделе описываются выбранные технологии и инструменты для создания универсального и современного веб-приложения. В третьем разделе описываются детали реализации проекта, т.е. архитектура, структура базы данных, детали модулей. Затем идет экономическое обоснование проекта, охрана труда и техника безопасности, заключение, список литературы и приложения.

Данная работа состоит из 65 страниц, 21 рисунков, 8 таблиц и 14 использованных источников. В приложении приведен исходный код основных конфигурационных файлов и функций главных модулей.

Ключевые слова: ШКОЛА ПРОГРАММИРОВАНИЯ, УНИВЕРСАЛЬНЫЙ, ВЕБ-ПРИЛОЖЕНИЕ, ДИСТАНЦИОННОЕ ОБУЧЕНИЕ, JSF.

ABSTRACT

In this paper we consider the problem of the learning of programming languages on a fee basis. The main issues are: creation of a universal web-application for the resource management of programming school in order to generate profit and provide a tool for students who are studying programming languages. As a tool for creating the user interface and back-end functions we used the capabilities of the Java programming language and JSF, which is framework for creating Web applications. As a result of this work we created a system that allows anyone to place all the necessary materials for teaching programming languages.

This diploma work consists of an introduction, a list of abbreviations, 5 chapters, conclusions, references and appendices. The introduction includes the relevance of this paper, goals and objectives. The first section provides an analysis of analogues, a review of the education programming in Kazakhstan and the world, an overview of the advantages and disadvantages of distance learning. The second section describes the selected technologies and tools to create a universal and modern web applications. The third section describes the details of the project, i.e. architecture, database structure, module parts. Then there is the economic justification of the project, labor protection and safety, conclusion, references and appendices.

This work consists of 65 pages, 21 figures, 8 tables and 14 sources used. The appendices show the source code for basic configuration files and functions of the main modules.

Keywords: PROGRAMMING SCHOOL, UNIVERSAL, WEB-APPLICATION, DISTANCE LEARNING, JSF.

LIST OF TERMS AND ABBREVIATIONS

1. IDE - Integrated Development Environment
2. DBMS - Database Management System
3. ER - Entity Relationship
4. MVC - Model View Controller
5. JSF - JavaServer Faces
6. UI - User interface
7. ORM - Object-relational mapping
8. HQL - Hibernate Query Language
9. SQL - Structured Query Language
10. GUI - Graphical User Interface
11. UML - Unified Modeling Language
12. JSTL - Java Server Pages Standard Tag Library
13. MD5 - Message Digest 5 Algorithm
14. XHTML - EXtensible HyperText Markup Language
15. IT - Information Technology
16. ICT - Information and Communication Technology

INTRODUCTION

The relevance of developing the universal web-application for the resource management of programming school is caused by the current development of information and communication technologies in the Republic of Kazakhstan. In terms of the state program "Information Kazakhstan - 2020" approved by the Decree of the Republic of Kazakhstan's President by № 464 dated January 8, 2013 contains a paragraph 3.1.1 "Increasing the level of computer literacy and ICT specialists". In this part defined the task: "creating conditions for the development human capital as the main factor in the formation of information society in Kazakhstan".

The relevance of this diploma work is in the development of a web application as systems for increasing the information literacy of the population.

This web application is aimed to the wide audience (including people of different age groups), so it should be developed using modern technologies, should have user-friendly and simple interface, and be intuitively understandable.

The meaning of the practical part is to make this web-application universal in order to be reuse for different schools of programming. This web application will have the opportunity to anyone who is interested in starting their own school of programming, to place all the necessary materials (lectures, practical tasks, exercises), to receive feedback from students which will sign to their courses. In addition, anyone who is interested in the study of a particular programming language will have opportunity to subscribe to a certain course instructor, study material and lectures, do practical tasks and receive feedback from the teacher.

Novelty of the project is in universality of this web application and granting opportunity to open school of programming in order to get profit. The existing web-sites with online training courses don't give such an opportunity for the tutors.

The aim of graduate work is development of a universal web-application for the resource management of programming school.

The object of the research is training programming.

The subject is the universal web-application for programming schools.

1. OVERVIEW AND ANALYSIS OF THE WEB APPLICATIONS IN EDUCATION AREA

This chapter contains overview of the programming education area in Kazakhstan and the world. We also provided an analysis of the similar project and revealed their advantages and disadvantages in comparison with the project developed by us.

1.1. Description of the study

Today there is no more demanded and actual specialty than programming. Modern digital technologies provided such rough demand for experts in this area that they will remain demanded for many years.

Nowadays there is no longer a problem to get a full education on any subject remotely, because distance-learning system is actively developing,

Online learning of the programming languages has several significant advantages over the traditional method. Firstly, you can practice at your own pace. Secondly, you can choose the teachers and training materials you like. That in certain cases can have crucial importance. At last, the online way of training will allow you to pick up that course and level of complexity, which is necessary.

It is crucial to be able to choose the right educational materials on the difficult path of learning programming. Now in the sphere of programming there are many useful training books, every day developers from the entire world publish tens and hundreds of articles devoted to the solution of specific problems. It is very heavy to find the most necessary, fundamental knowledge in such chaos, having which it is easy to understand and acquire any other technology in the shortest possible time. That is why it is so important that now in the network appears more and more services, which help to gain necessary new knowledge.

Various schools of programming at which professionals make courses, will help not to be lost in this stream of knowledge and resources, they will help you to define the course you should move to achieve success in programming.

1.2. IT education in Kazakhstan

Nowadays, economy of Kazakhstan is dynamically developing and needs for qualified personnel for IT and telecommunications. Universities do not manage to train workers, conforming to requirements of employers. In Kazakhstan IT education needs to be differentiated as average special and the high which is received respectively in colleges, a bachelor degree, a magistracy and PHD.

Considering the IT market it is possible to tell that now there is a formation

stage. The state do a lot, first of all, financial support - a large amount of government grants are provided, which numbers increase every year. Only in the last year it has been allocated more than 30 thousand of grants, half of which goes to the technical specialties, including a quarter to the IT. Of course, financial support is insufficiently. Improvement of state standards in IT education is necessary: to give to higher education institutions freedom in formation of training programs and plans. After all, IT science in Kazakhstan is very young and dynamically developing.

If compare our market with the markets of other countries, Kazakhstan is closer to the markets of the CIS. Though it is necessary to recognize that Russia and Belarus made very serious jump forward in the field of IT in general, and IT educations, in particular. Therefore, there are a number of reasons, for example, development of software business.

Today in the Kazakhstan there are 149 higher educational institutions from which more than 60 train IT specialists. Therefore, the main competitive fight is for students of a bachelor degree, because there is no science especially commercialized in this area.

Today in Kazakhstan there is over one thousand IT companies, the turn of the growing ICT branch makes about \$1 billion and, of course, all this demands increase in number of qualitative IT specialists.

Two years ago, the market needed 4-5 thousand highly qualified specialists. Generally, the need for experts is formed in the large cities, like Almaty and Astana, and advanced industrial regions. Moreover, it is clear as each enterprise wants to automate the production to introduce more innovations and for this purpose involves IT specialists.

1.3 Online studying programming

Now people seriously think about how to applied programming and computer science became a full-fledged school discipline. Despite the fact that computer science in schools began teaching 30 years ago, the graduates of educational institutions are often unable to write their own, even the simplest calculator in one of the most popular programming languages, whether it is C # or Java.

The problem is that each of us uses information technology every day, through the laptop or smartphone, but only a small fraction has at least a rough idea of how it works. And after all, having even basic knowledge about mathematical logic, algorithms and creation of programs, it is possible to make the life more interesting: for example, to turn the apartment into the operated "clever house" by means of the Arduino platform.

Present teenagers and children are called as "digital generation". Grown with a free access to the Internet, with smartphones and tablets, they often appear as experts in the sphere of digital technologies.

So young people today have lots of experience of interacting with new technologies, but not many of them can express themselves with creating technologies.

Increasingly, people are starting to recognize the importance of learning to code. In recent years, there have been hundreds of new organizations and websites that are helping young people learn programming.

The project of Stanford University Udacity is one of the few web-sites which offer courses of college to the public, including Introduction to Computer sciences, which is available free for free online. Udacity contains the most details works on programming, which can only be imagined. For those who needs deep knowledge of this sphere, but not superficial.

Codecademy is indisputable, the most known site for programming studying. Training begins literally with the homepage of the site, there is an interactive console which help you to learn the principle of training on these courses. Then user can choose one of the languages begin its development. The free and very well thought over guide on programming languages for beginners. Here you will find materials for studying of Rails, AngularJS, HTML&CSS, Javascript, jQuery, PHP, Python and so on.

Code Avengers is created for those people who are afraid of programming, thinking that this infinitely boring and difficult. Here you can study HTML5, CSS3 and JavaScript. Each of these courses is carefully developed to interest. At the end of each lesson, there is a small mini-game which allows to put the knowledge into practice.

Code School for those students who already finished the courses of Codecademy or Code Avengers and who are ready for getting the knowledge. Unlike the majority of the interactive training sites, Code School offers more advanced courses and will turn you from the beginner into the expert in the chosen area.

The interesting one is Teamtreehouse. These courses are more focused than the others, they are focused on one programming language. It means that the specific task is set for you, for example creation of the simple web-site or interactive web application. They give all necessary knowledge which can be necessary. Therefore, this resource is perfectly suitable for the beginning programmers.

Moreover, the last one I want to mention is LearnStreet. It is possible to begin training on this web-site with pressing of only one button. But you will not finish the courses so quickly. There are a lot of training materials on JavaScript, Python and Ruby. There are also lectures for the beginners, practical tasks, and examples of the solution of real projects.

It seems that everybody is getting into the act. However, when you become fluent with reading and writing, it is not something that you are doing just to become a professional writer. Very few people become professional writers. Nevertheless, it is useful for everybody to learn how to read and write. Again, the same thing is happening with coding. Most people won't grow up to become professional computer

scientists or programmers, but those skills which you can gain during training programming are things that people can use no matter what they're doing in their work lives. Among them are creative thinking, systematically reasoning, planning, collaboratively working.

1.3.1 The advantages of online learning

A lot of things becomes available thanks to distance learning: saving money on a payment for education, listening lectures of the well-known professors, receiving professional development at leading universities of the world, using extensive electronic libraries or participating in a webinar.

The first advantage of online learning is opportunity to be trained at any time. The student who is trained remotely can independently solve when he will study during the semester. It builds for himself the individual schedule of training. Some educational institutions give to the students opportunity to postpone training for long term and to come back to it needlessly again to pay educational services.

Second one is the opportunity to study at your own pace. Those who learn remotely need not worry about the fact that they lag behind their classmates. You can always return to the study of more complex issues, watch video lectures several times, read the correspondence with the teacher, and skip the already well-known topics. The main thing is to successfully pass final certification.

Third one is the opportunity to study in any place. Students can learn at home or office from anywhere in the world. You need only to have a computer with Internet access to start training. No need to go to school every day is a definite plus for people with disabilities, for those who are living in remote areas or raising young children.

Fourth, study without departing from the main activity. Remote education can be taught several courses at the same time with the receiving a main higher education or working full-time. There are educational organizations that organize corporate training for employees of businesses and public servants. In this case, study does not interrupt seniority, and study questions can be immediately applied in the workplace.

Fifth advantage is the availability of teaching materials. Those, who learn not remotely familiar with the problem as a lack of textbooks and problems books. Access to all relevant literature offers the student after registration if they learn remotely.

Sixth is the mobility. Communication with teachers, tutors carried out in different ways: as an online, and offline. Consult with the tutor via e-mail is sometimes more efficient and faster than appoint a personal meeting.

1.3.2 The disadvantages of online learning

Besides the advantages of the online learning, there are several disadvantages of it.

First disadvantage is the need for a strong motivation. Almost all students master course material on their own. This requires the development of will power, responsibility and self-control. To maintain the tempo of training without supervision cannot please everyone.

Second, distance education is not suitable for the development of communication skills. With distance learning students' personal contact with each other and with the teachers is minimal, if not absent. Therefore, this form of training is not suitable for the development of communication skills, confidence, teamwork skills.

Third problem is the identification of the user. While the most effective way to see to it honestly and independently whether students taking exams or tests - is video surveillance, which is not always possible. Therefore, on the final assessment students have personally come to the institution or its affiliates.

1.4 Programming for kids

Programming is very popular. Worldwide, 36 million children participate in the activities of the movement "The hour of programming." The purpose of these activities - to help children become active, rather than passive users of technology, as well as to gain knowledge that can help in the future to get a good job in our increasingly technical world.

Even if your children do not want in the future to make a living programming, a basic understanding of the principles of programming and develops thinking in the future, the child better cope with emerging challenges. Bill Gates once said: "If you learn to write programs, it develops your mind and helps you think better and creates an image of the perception of things. Overall, I think it would be useful to a person, regardless of the scope of its activities. "

Such platforms like Scratch and Alice allow children (and adults) to create their own games and animations, using simplified methods of programming. However, not all children are interested. And for these children there are many games that teach basic principles of programming in a fun and accessible way.

Lightbot - is a puzzle game on programming created Danny Yaroslavsky, a Canadian student. The goal is to get the robot to highlight all the blue tiles on the 3D-lattice. The difficulty is that it must be done in one approach, creating a series of commands to the robot. The game is designed for children from 4 to 8 years.

Brooklyn resident Raj Sidhu wanted to explain to children the basics of programming in a fun way. He chose a completely non-technical solution for solving highly technical tasks and implemented based on binary logic and conditional instructions in a family board game. Play Code Monkey Island is quite simple, and

the game takes a little time - only about 45 minutes. Players use cards with instructions such as "Every monkey to move into three cells, without stepping on the rocks" in order to move his troops of monkeys on the island. The game is built for children from 8 years.

For children from 10 years invented the game SpaceChem. SpaceChem - a combination of programming and pseudocode in terms of science fiction. However, the game has received excellent reviews since its release in 2011. This game is quite complicated, so think about how to make high school students and adults. The purpose SpaceChem - build a mechanism to deal with monsters, using the concepts of programming, as the execution of orders, loops, transitions and routines.

Robozone is a puzzle game, which in the free time created by Microsoft programmer Igor Ostrovsky. Task Robozone - hold the robot through the maze by using the limited command. The difficulty levels of the game vary: some levels are suitable for children, and some tasks make scratch your head, even experienced programmers.

1.3 Comparison table of similar products

Table 1 - Comparison of the similar products

	Teamtreehouse	CodeAcademy	Udacity	My project
Registration as a tutor	-	-	-	+
Java lessons	-	+	+	+
User friendly design	+	+	+	+
Price for usage	\$25/mo - \$49/mo	\$19.90 for PRO subscription	\$199/program	Free
Practice tasks	+	+	+	+
Video lessons	+	-	+	+

Continuation of the Table 1

Program as the project	-	+	+	+
Online compiler	-	+	-	+
Asking questions	-	+	-	+
Adding files as an answer	-	-	-	+

In the Table 1 we compared three web-application with our project: Teamtreehouse, CodeAcademy, and Udacity.

The table shows all advantages and disadvantages of these web-applications. Opportunity to be registered on the site as the teacher and to place the lectures and practical tasks was a major factor of an assessment. Among the sites neither provides such an opportunity. However, in my web application will be able to register as both students and teachers.

Another important factor was the presence of the lessons of the programming language Java, as the main language for studying in my web application is Java. All applications except Teamtreehouse provide an opportunity to learn Java. Other important factors were the presence of the forum and opportunity to ask questions directly to the teacher as an important part of the learning process. A forum for participants in the project have all the sites, but the opportunity to ask questions has only on CodeAcademy.

Also, my project will feature a built-in online compiler, which is currently available only on the website CodeAcademy.

All web applications listed in the table of comparison except Codecademy have video tutorials to study the material on my site will also be able to attach video tutorials.

Only Udacity web application designed for people of different levels of knowledge, TeamTreeHouse aimed at experienced people with practical knowledge, CodeAcademy designed for beginners, learning takes place from scratch.

Among the sites only TeamTreeHouse is paid resource, the cost of education varies from 25 to 49 dollars a month. Other products are free.

Interface design of web applications also has an important role, and all the presented products has user-friendly interface.

1.4 Formulation of the problem

The aim of graduate work is development of a universal web-application for the programming school resource management, which will allow anyone who wishes to open his own school programming, post lectures and practical exercises, as well as to register as a student and study the material for interesting programming language.

The aim of graduate work is development of the information web portal for IITU with multifunctional administrative panel, which allows to comfortably use the portal. Present website of IITU does not have such panel, boring design, hardly understandable interface and considered as poor and not interesting.

The aim is to create full web application, with attractive design and useful interface, even simplified, to provide safety and no useless buttons and not working links. It should be a tool which provide capability for tutors to add lessons and practical works, some additional information, receive feedback from students, monitor their progress, for students to learn lecture materials, do the tasks and receive comments and feedback from tutor, have opportunity to communicate with other students on forum.

The object of the research work is training of information technology in today's environment. The subject of my work is the universal web-application for programming schools.

1.5 Conclusion for chapter

To conclude, the project is target to those people, who want to open their own programming school and for those who want to learn programming. Today there is no more demanded and actual specialty than programming. Modern digital technologies provided such rough demand for experts in this area that they precisely will remain demanded still long - for many years. So, this project's relevance is justified by the current state of the IT industry.

2. TECHNOLOGY REVIEW

This part of the diploma works describes the main techniques and technologies, which we used during the development process.

2.1. Overview of the used software

The main purpose of this diploma work is to create web-application for the resource management of programming school. This web application is aimed to the wide audience (including people of different age groups), so it should be developed using modern technologies, should have user-friendly and simple interface, and be intuitively understandable.

According to the requirements, we use latest versions of software for development the application and modern technologies in order to make our application understandable for users and functionally extensible.

2.1.1 Choosing the development environment

For the development of this Web-application we choose IntelliJ Idea from Jet Brains company. This is the most intelligent and convenient development environment for Java, including support of all the latest technologies and frameworks. IntelliJ IDEA provides tools for productive work and perfect for the creation of commercial, mobile, and web applications.

There are a lot of advantages of this development environment. Among them are:

- Deep intelligence. During the coding process IntelliJ indexes the code and gives the relevant suggestions. It suggests only those types of classes, methods, fields, and keywords, which are expected in the current context. It analysis the code on-the-fly and instant complete it;
- Out-of-the-box experience. IntelliJ provides mission-critical tools such as integrated version controls systems (SVN, GIT, CVS, Mercurial, GitHub), a lot of plugins (Hibernate, JSF, GRADLE, MAVEN, Spring MVC, Junit) and etc.;
- Wide variety of supported languages (Java, JavaScript, HTML/XHTML/HAML, CSS/SASS/LESS, Python, SQL, PHP, C, C++ and etc.) It understands these languages even if when the language expression is injected into a String literal in the Java code;
- Unobtrusive intelligence. The coding assistance in IntelliJ IDEA is not about only the coding view. It also in searching over the projects, accessing a tool window, toggling for a setting. There are a lot of hot keys, that helps

the developer be more productive;

- Integration with application servers, including Tomcat, TomEE, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Resin, Jetty and Virgo;
- Tools for working with databases and SQL files, including client and convenient editor for the database schema;
- Tools to run the tests and analyze code coverage, including support for all popular frameworks for testing (JUnit, TestNG, Spock, ScalaTest and spec2);
- Provision of reliable refactoring tools;

So, as we can conclude, this development environment is perfect for our web-application, it helps us to program for the shortest possible time.

2.1.2 Choosing the database schema

Databases - is specially repository to store different types of data. Each database has a certain model (relational, document-oriented), which provides easy access to data. Database management systems (DBMS) are the special applications (or library) to manage databases of various sizes and shapes.

DBMS must ensure the relational model of data. The model itself implies a certain type of relationship between entities from different tables. To store and manipulate data, this type must have a certain database structure (table). In each column of the tables may contain data of various types. Each record consists of a plurality of attributes (columns) and a unique key stored in the same table - all of these data are interrelated as described in the relational model.

There are three major open source DBMS:

- SQLite - a very powerful embedded control system;
- MySQL - the most popular and common database;
- PostgreSQL - the most advanced database.

The main factor of using DBMS was that we build a web-application. It means that it should be available for 24 hours, 7 day a week. So, we choose MySQL DBMS, because it high reliable.

MySQL - is the most common full-fledged DBMS server. MySQL is very functional, a freely available database, which has been successfully working with a variety of websites and web applications. Despite the fact that it has not implemented all the functionality of SQL, MySQL offers a lot of tools for application development.

There are a lot of advantages of this DBMS. Among them we can highlight:

- MySQL - the most popular and common database;
- Easy to use. Installing MySQL is quite simple. Other applications, such as the GUI, make it easier to work with database;
- Rich functionality. MySQL supports most of the SQL functionality;

- Safety. A large number of features to ensure safety, which is supported by default;
- Scalability. MySQL is easy works with large amounts of data and can be easily scaled;
- Speed. The simplification of certain standards allows MySQL to significantly increase productivity.

For all these reasons, MySQL has become standard for web in the DBMS field, and now it develops opportunities to use it in any business-critical applications, that are competing with database vendors such as Oracle, IBM, Microsoft and Sybase.

For all these reasons we choose MySql DBMS and use it for free during the development process.

2.1.3 Database modelling tool

For modeling and analysis of business processes following modeling tools are widely used: Rational Rose, Oracle Designer, AllFusion Process Modeler (BPWin) and AllFusion ERwin Data Modeler (ERWin), ARIS, Power Designer.

For our project we choose PowerDesigner, because it's easy to use and has a lot of useful tools for developing. PowerDesigner is a comprehensive solution for modeling and developing applications and business processes for organizations that need a fast, consistent and effective cost-create or re-engineering of business applications.

PowerDesigner allows you to remove these obstacles to effective development projects: the differences in the training of the project participants, heterogeneous platforms and a wealth of development languages. This allows you to focus on business needs to create applications throughout the development process - from system analysis and design, and up to code generation for immediate application.

Last version of the PowerDesigner offers new opportunities for business process modeling, object modeling based on the UML, and supports both traditional and emerging modeling techniques within a single graphical environment developed. This significantly reduces the time and costs of the project, which should run on different platforms and tool environments.

Key features of PowerDesigner:

- Scalability. MySQL is easy works with large amounts of data and can be easily scaled;
- Business Process Modeling. PowerDesigner allows non-technical specialists to design and simulate business processes, focusing on the business problem and relying on them to known terms, using a simple and intuitive graphical non-technical model;
- Data modeling. PowerDesigner allows you to design and generate the database schema through the two-level (conceptual and physical) relational

database modeling that supports classical methods of database design. It also has built-in data warehouse modeling tools;

- Object Modeling. PowerDesigner offers a complete technology systems analysis and design using UML standard (business process diagrams, sequence diagrams, class diagrams and etc.) On the basis of class diagrams PowerDesigner automatically generates and reverse engineer the code for a popular tool environments, such as JavaTM (including EJB 2.0), SQL, XML, Web Services, C ++, PowerBuilder, Visual Basic and others, by adjustable generator;
- Repository PowerDesigner. A complete repository of information detailed in the chosen database. Repository is scalable and has a friendly interface for users, who is working remotely. It provides the following features: role-based access to the models and submodels, version control, configuration management, integration models, statements of changes between models and versions, full-text search across the repository.

According to all the features above we choose the PowerDesigner and use it to create ER models of our application and also to generate SQL code for our database.

2.2 Overview of the used technologies

During the development of this diploma work we tried to use the latest techniques and technologies. The main objective was to make our code reusable, easy to change and complement.

2.2.1 Choose of the design pattern

Design patterns are used for solving the common problems in the field of software development. Design patterns are not ready-made solutions that can be transformed directly into code; they are a general description of solving the problems, which can be used in different situations. There are a lot of them for different types of projects:

- Memento;
- Observer;
- Template method;
- Mediator;
- Abstract factory;
- Builder;
- MVC;
- Singleton and others.

One of the most well-known and commonly used design patterns is a pattern

"Model - View - Controller" (MVC).

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

Model is the data and rules used to manage data, which represent the application management concept. In any application, the entire structure is modeled as data that are processed in a certain way. The model allows the controller representation of data that the user has uploaded. The data model will be the same regardless of whether we want to represent them to the user. Therefore, we choose any of the available views for displaying data. The model contains the most important part of the logic of our application logic that solves the problem with which we are dealing.

View provides various ways to represent the data obtained from the model. It may be a template that is filled with data. There may be several different types, and the controller selects a best fit for the current situation.

Web application typically consists of a set of controllers, models and types. The controller may be arranged as a core, which receives all requests and causes the controller to perform other actions as appropriate.

In fact, views and the controllers is the user interface. Moreover, if the components typically can be reused in other parts of the system, the controller is often specific to the particular case.

This pattern is perfect for the web-applications like ours. So, we use MVC to develop our program.

2.2.2 Features of JSF 2.2

There are many frameworks for developing web applications that have been created for different purposes and with different capabilities.

According to the survey of Java developers, which was made in 2014, we can estimate, which WEB framework is most popular. As we can see in the Figure 2.2.1 on the first place is Spring MVC, it took 40% of the votes, JSF on the second place with 21%, on the third place is Vaadin, and on the forth is Google Web Toolkit.

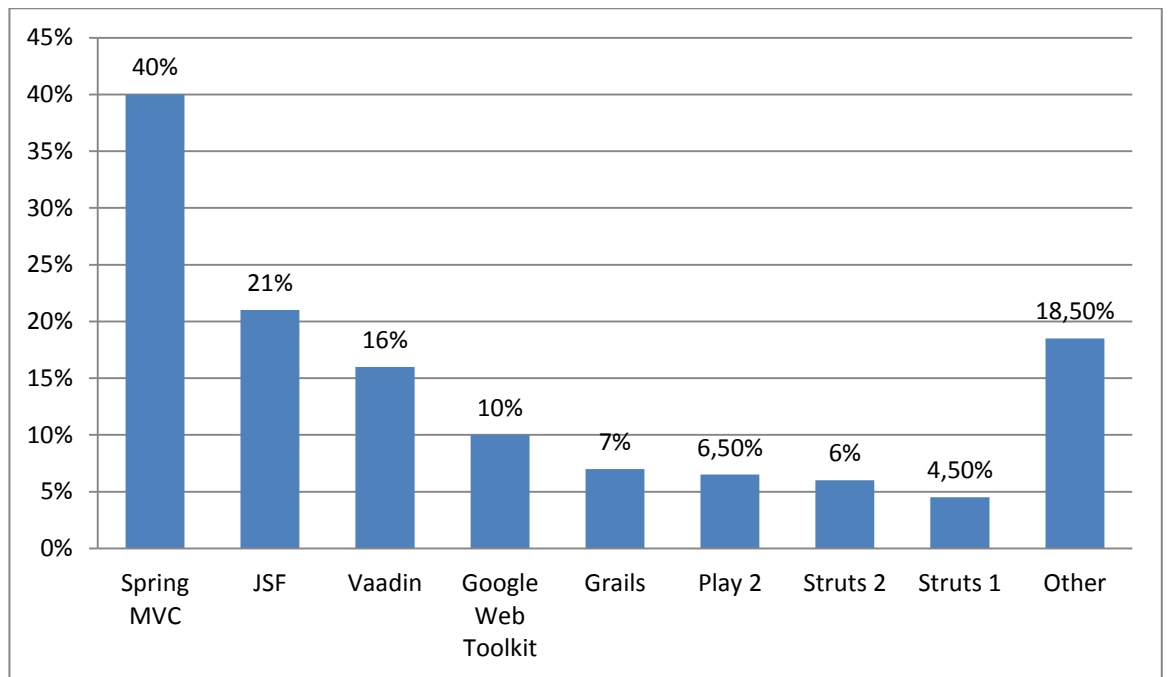


Figure 2.2.1 Web frameworks in use

As the view part of our project, which we build on the base of MVC model, we will use JSF.

JavaServer Faces (JSF) - is a framework for web applications written in Java programming language. It serves to facilitate the development of user interfaces for Java EE-application. Unlike other MVC-frameworks that are managed queries, the JSF approach is based on the use of components.

Status of UI components is saved when the user requests a new page, and then restored, if the request is repeated. To display data is commonly used JSP, Facelets, but JSF can be adapted to other technologies, such as XUL.

Java Server Faces technology includes:

- Special JSP tag library for JSP expressions interface JSP page. In JSF 2.0 as a view handler uses Facelets technology that has come to replace the JSP;
- API set to represent a component of the user interface (UI) and manage their state, event handling and validation of input data, the definition of navigation, as well as support for internationalization (i18n) and accessibility.

Java Server Faces technology enhances the existing standard concept of the user interface (UI) and Web-tier concepts without reference to a specific developer markup language, protocol or client device.

Combining the functionality of a user interface component with special Renderers, developers can design custom tags for a given client device.

There are several JSF libraries, which include:

- PrimeFaces;
- MyFaces;

- Tomahawk;
- Trinidad;
- Tobago;
- Orchestra;
- ICEFaces;
- OpenFaces;
- RichFaces.

On the Figure 2.2.2 presented a graph generated by Google Trends comparing the popularity PrimeFaces with the competitors. For many, PrimeFaces is the de-facto standard of JSF Component Libraries.

As we can see, during the last seven years the popularity of PrimeFaces library has grown relative to the others. Nowadays, most of the developers use PrimeFaces library for creating their web-applications. At the beginning, RichFaces was the most popular, and it still have several advantages, but PrimeFaces is rapidly developing library.

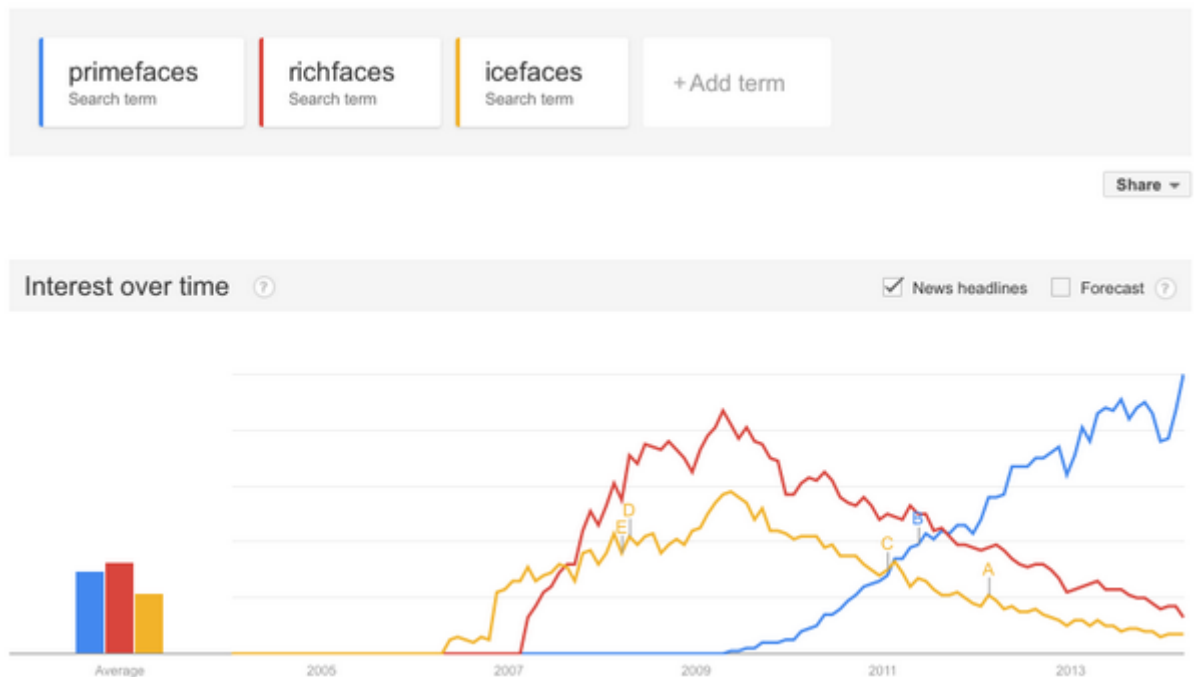


Figure 2.2.2 Comparison PrimeFaces with the compatitors

The advantages of the PrimeFaces are as following:

- Simplicity and Performance. PrimeFaces is a lightweight library, all decisions made are based on keeping PrimeFaces as lightweight as possible. Usually adding a third-party solution could bring a overhead however this is not the case with PrimeFaces. It is just one single jar with no dependencies and nothing to configure;
- Ease of Use. Components in PrimeFaces are developed with a design

principle which states that "A good UI component should hide complexity but keep the flexibility" while doing so;

- Strong Community Feedback. PrimeFaces community continuously helps the development of PrimeFaces by providing feedback, new ideas, bug reports and patches.

So, in our web-application we will use the open-source JSF library – PrimeFaces. All of the view pages use these components to represent understandable and user-friendly interface.

2.2.3 Hibernate

The last important technology which we use is Hibernate. Hibernate is the library for the Java programming language, designed to solve the problems of object-relational mapping (ORM). It is free open-source software, distributed under the GNU Lesser General Public License. This library provides an easy-to-use framework for displaying an object-oriented data model in traditional relational databases.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieve the developer from 95% of common data persistence related programming tasks. Hibernate sits between traditional Java objects and database server to handle all the work in persisting those objects based on the appropriate ORM mechanisms and patterns.

There are several advantages of using Hibernate technology, among them are:

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code;
- Provides simple APIs for storing and retrieving Java objects directly to and from the database;
- Hibernate does not need an application server to work;
- Minimize database access with smart fetching.

For operating with the Hibernate library we use HQL - Hibernate Query Language. Hibernate Query Language (HQL) is an object-oriented query language, similar to SQL, but instead of operating on tables and columns, HQL works with persistent objects and their properties. HQL queries are translated by Hibernate into conventional SQL queries which in turns perform action on database.

2.3 Use cases

For identifying the needs of our target user we use method of the Use cases. Use case diagrams are used to present a graphical overview of the functionality of a system. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.

Associations between actors and use cases are drawn as the solid lines. An association shows how an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system.

2.3.1 Separation of roles

In our project, we identified three main roles:

- Admin;
- Tutor;
- Student

Each of these roles has its own possible actions in our system. Starting from the admin, we can describe, what exactly each role can do in our web application.

Admin is the main manager of the system. He monitors the actions performed by other users and can change the information on the web-site. His duties include:

- Registration of the user in the system. Only admin can create user profiles in our web-application, no user can be registered by himself;
- Managing user profiles. Admin can block the users, who violate the rules of the system. Also, admin can change information about the users, including username, full name, email and phone number;
- Admin can see through the all courses, which were added by the tutors, but can't change it or delete it.

The second role in our system is tutor. The main feature of our project is the ability to be the part of the system in the role of the tutor. Tutor has several opportunities:

- Tutor can add courses with the lecture materials (including pictures, videos, links and files) and tasks for the certain lecture;
- Tutor can receive feedback from the students. Tutor can see how the students do the tasks and read messages from them;
- Tutor can manage groups for certain courses. He can choose for which student his course will be available;
- Tutor can put marks on the homework of the student.

The last role is the student itself. This role also has its own opportunities, including:

- Student can follow the course in order to see its lecture materials and tasks;
- Student can do the task, which were added by the tutor, and send the result

in the form of archive of the project files;

- Student can send the messages to the tutor in order to ask questions;
- Student can compile his work in the inner compiler of the web-application.

So, as we can see, we strictly delineate the roles and possible actions of the registered users. Role division minimizes the risks of personal data loss or unauthorized disclosure of educational materials. From a security point of view, the distinction between the roles will help us to protect our customers from fraud.

2.3.2 Usage scenarios

At this part of the diploma work we will introduce the main usage scenarios of the system. In the first Figure 2.3.1 we present the student-administrator interaction.

As we can see from the figure there are several functions which provide communication between administrator and student. Student provides personal information to the administrator in order to be registered in the system. Administrator can change personal information about student and also change his status to “Blocked” without any requests from the student side.

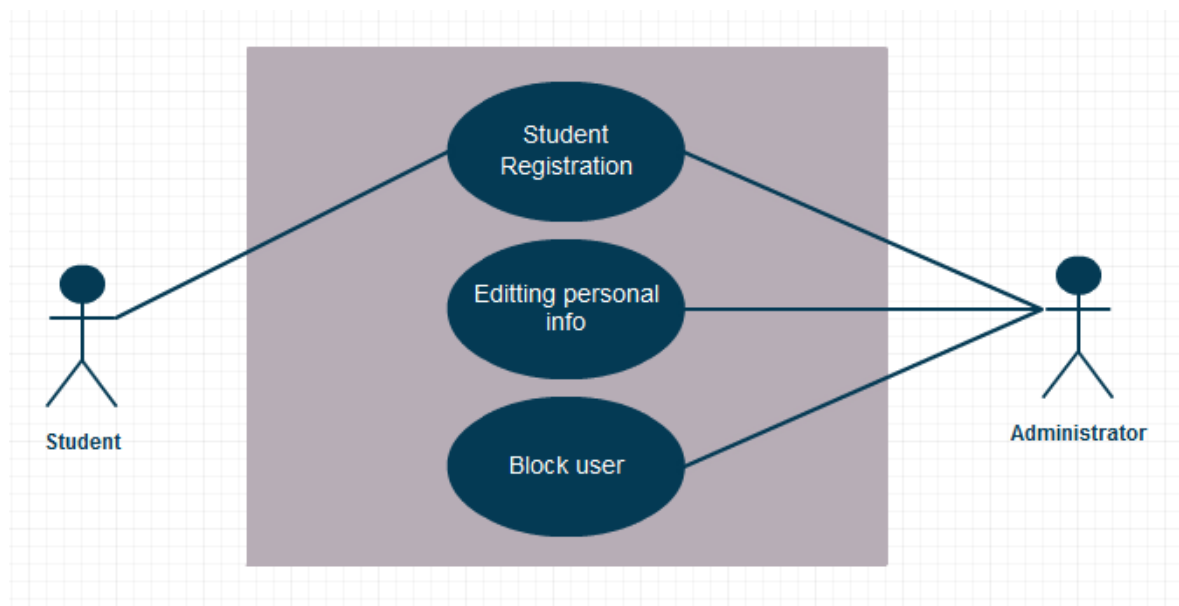


Figure 2.3.1 Student-administrator interaction

The second Figure 2.3.2 describes the interaction between administrator and the system.

This figure describes which actions are available to administrator. As was described in part 2.3.1 administrator can manage users' info and status, register new users and see through all the courses. All of these actions occur through interaction with the system, since the administrator always receives a response from the system.

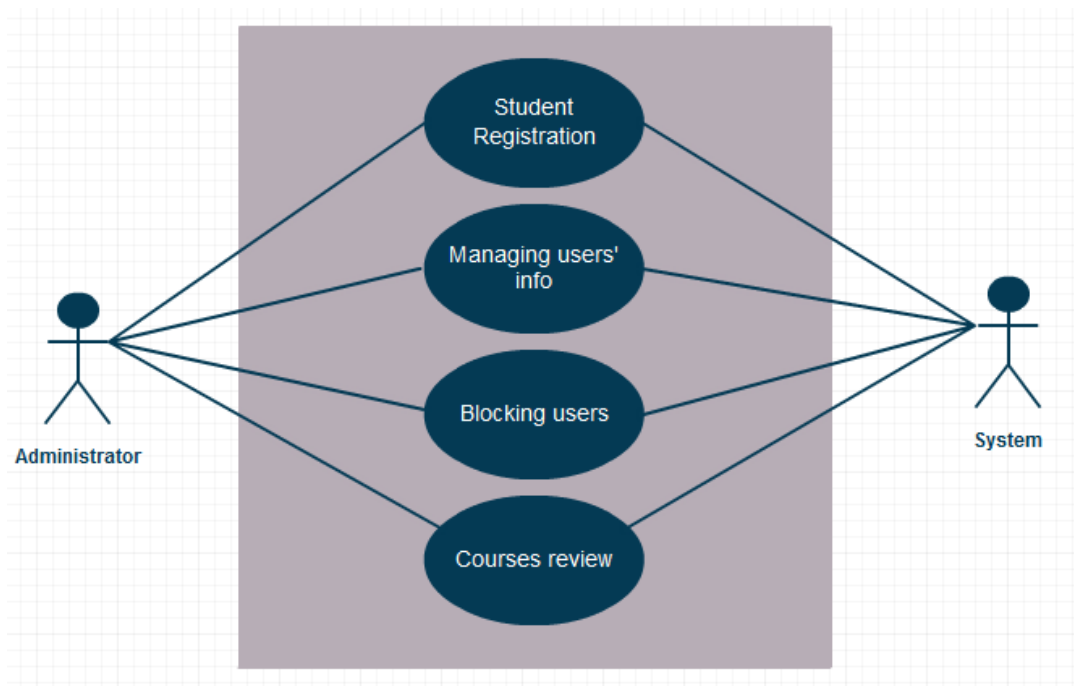


Figure 2.3.2 Administrator-system interaction

At the figure 2.3.3 we can see how tutor interact with the system. As we see above, tutor can do several actions in the system. He can add courses and lectures to them, form the groups, set marks to the students' homework and receive feedback from them.

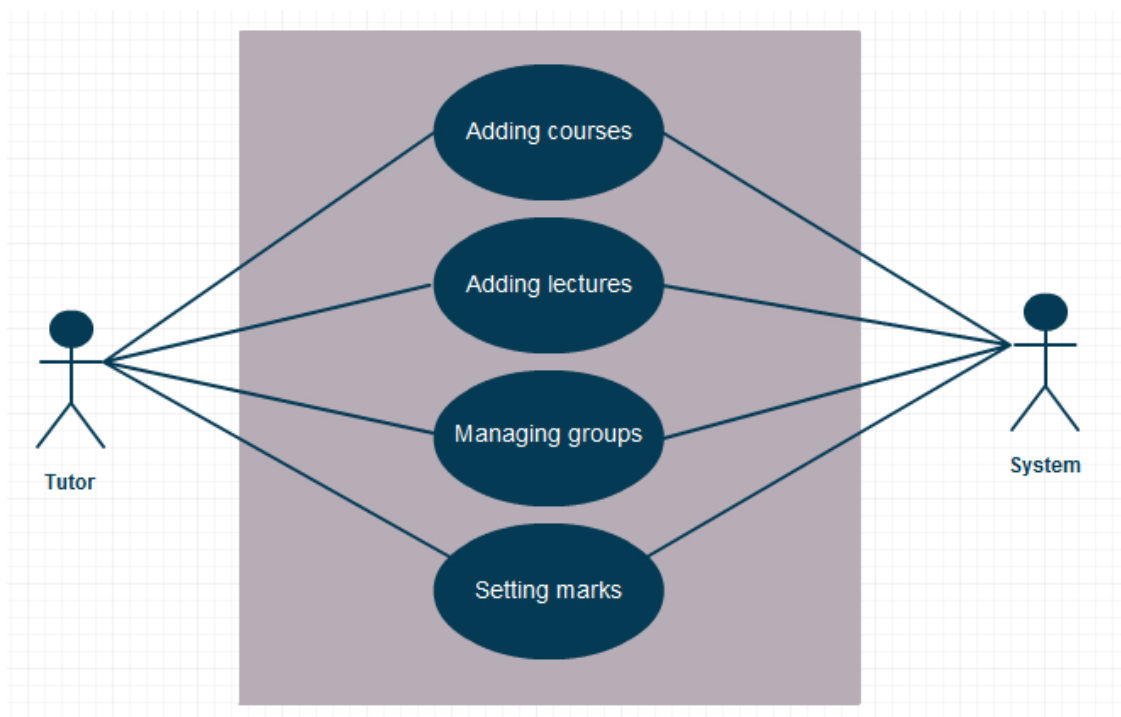


Figure 2.3.3 Tutor-system interaction

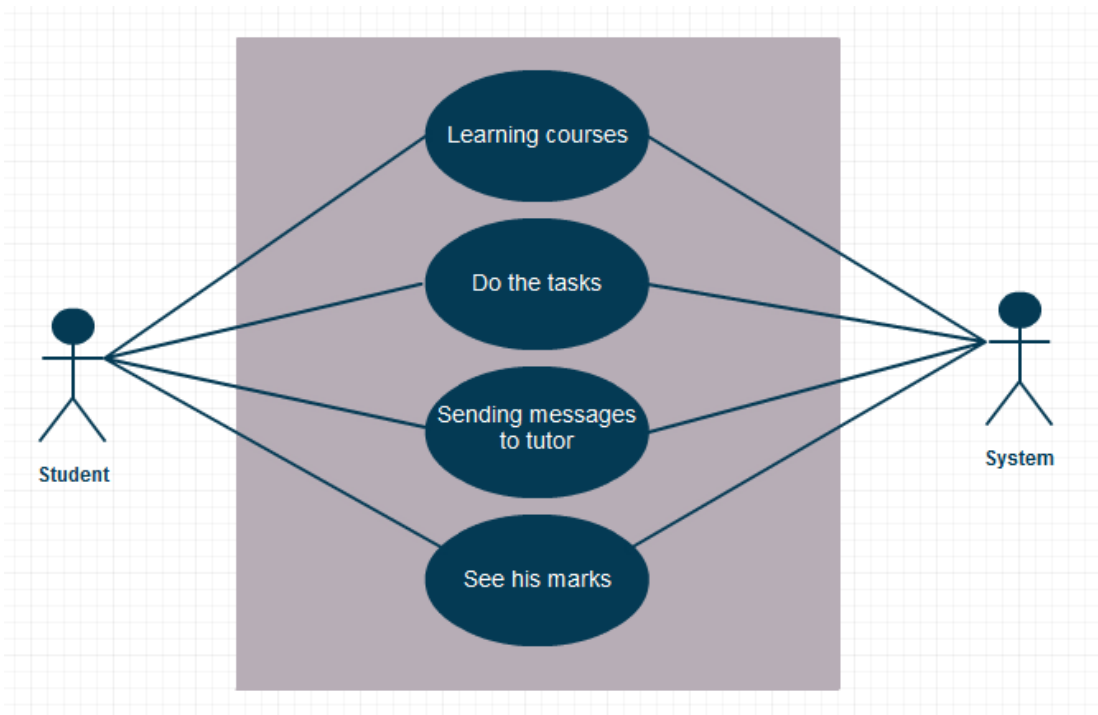


Figure 2.3.4 Student-system interaction

As we can see from the Figure 2.3.4 student do not have a lot of actions with the system. He can only follow the courses, see the lectures, do the tasks, send messages to the tutor and see his own marks.

There are no actions between student and tutor without the connection with the system. And also, there are no actions between tutor and administrator.

2.4 Conclusion for chapter

To conclude, during the development of the web-application we use modern technologies and techniques in order to create understandable and user friendly interface and secure and reliable tool for the tutors and students.

We carried out some analysis to justify our choice of the development tools and environment, type of database schema, tool for database architecture, coding style and languages. We provide the results of these analyses in the diagrams to visually show the advantages of selected technologies.

3. IMPLEMENTATION OF THE WEB-APPLICATION

In this part of the diploma work we will describe detailed implementation of our web-application for resource management of programming school. Also, we will provide samples of code, schemas and screenshots of our application.

3.1 Database structure of the project

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

Figure 3.2.1 is ER Diagram, when you can clearly see all of the relationships between entities of our database.

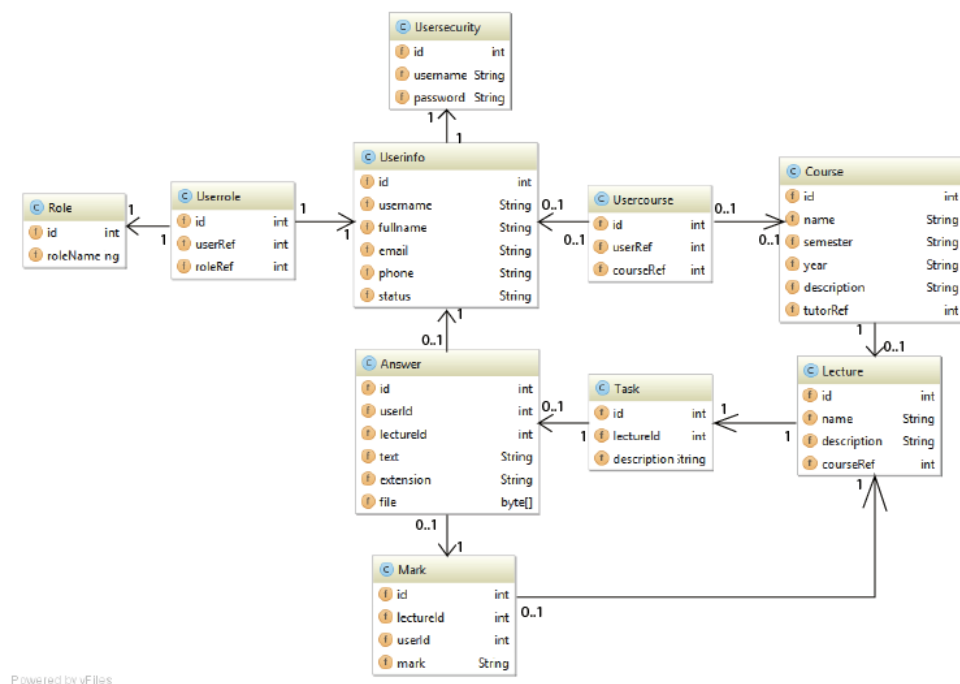


Figure 3.2.1 – ER Diagram

The table, which contains the information about user (username, full name, email, phone) is UserInfo table. We use this table to get personal information about tutors and students. The table UserSecurity contains username and encrypted passwords, which are needed for authentication. We use this table for authentication

process in our application. For the encryption we use MD5 encryption standard.

MD5 stands for Message Digest 5. This 128-bit hashing algorithm is designed to create a "fingerprint" or message digest of arbitrary length and then verify their authenticity. It is widely used to test the integrity of the information and store your passwords in a closed form.

The table which is responsible for authorization process is UserRole. This table contains the reference to the user's id and the reference to the Role table. So, we can connect the particular user to his role, and he will see only those parts of the system, which are available for his role.

The table Role contains the number of available roles. Among them are: student, tutor and administrator.

We have Course table which contains all the courses that has been created by any tutor in the system. We can understand which tutor creates each course using the table UserCourse. It includes the reference to the user and reference to the course. The relationship in this table is many to many, so the tutor can have several courses and one course can have many tutors.

The table Lecture contains all the lectures with the attribute which referred to the Course table, so we can understand to which course particular lecture is applies. The Task table also contains the id of the lecture, so we can bind the task to the lecture which it belongs to.

In the table HomeWork we store the homework of the students. After they send the executed task we save it in the table HomeWork with the comment of the student, which will be available to the tutor with the homework. The student can attach the files and tutor will receive them.

We save the marks which were send to the student in the table Marks. It contains the reference to the student, to the particular lecture and task of this lecture. So the marks will be available for the tutor to change and for student to see.

3.2 Project structure description

3.2.1 Model part

As we use MVC, our project in the IntelliJ Idea is divided into several parts. As a model part we use JavaBeans which are created by the Hibernate Persistence library. These models are stored in the Model folder (Figure 3.2.1). These beans have the property of persistence when its properties, fields, and state information are saved

to and retrieved from storage. Component models provide a mechanism for persistence that enables the state of components to be stored in a non-volatile place for later retrieval.

The mechanism that makes persistence possible is called serialization. Object serialization means converting an object into a data stream and writing it to storage. Any applet, application, or tool that uses that bean can then "reconstitute" it by deserialization. The object is then restored to its original state.

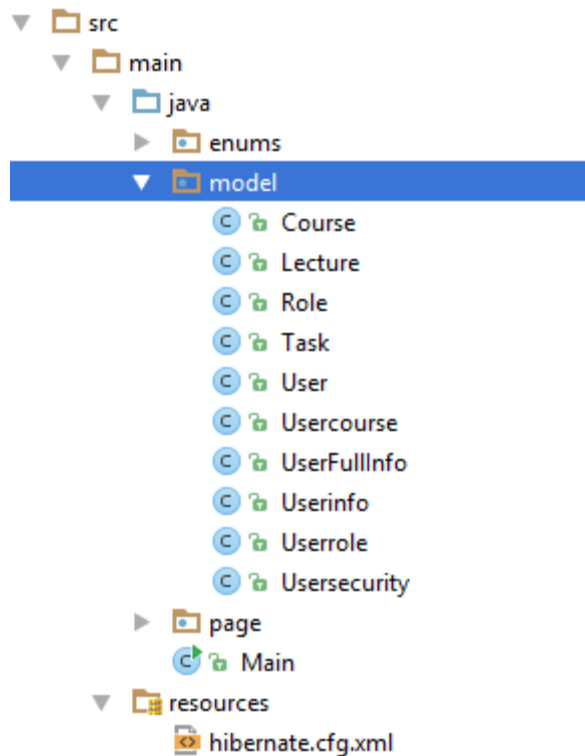


Figure 3.2.1 Location of model classes

So, all the tables, which are stored in our database, have a representation as a models in our project.

3.2.2 View part

As a view part of the Model-View-Controller we use .XHTML files. This is one of the types which can be used with JSF.

Unlike JSF 1.x versions, which use JavaServer Pages (JSP) for views, JSF 2.0 mandates support for Facelets as the view technology for JSF pages. Like JSP, Facelets are an implementation of View Declaration Language (VDL), which allows developers to declare UI components in different presentation technologies using HTML templates. However, because the Facelets view technology has been designed

specifically to leverage the features of JSF, Facelets provide JSF developers with a simpler, more powerful programming model than JSP. That is why beginning with JSF 2.0 Facelets will replace JSP (JSF 2.0 has retained JSP support only for backward compatibility).

Facelets pages are authored using XHTML, and they provide good expression language (EL) support. Facelets also leverage the concept of XML namespaces to support these tag libraries:

- JSF HTML Tag Library;
- JSF Core Tag Library;
- JSTL Core Tag Library;
- JSTL Functions Tag Library;
- JSF Facelets Tag Library.

The idea behind authoring Facelets using XHTML is to make them portable across diverse development platforms. Because JSF is authored using XHTML pages, which conform to DTD, Facelets have the .xhtml extension. To use Facelets in a JSF application, you must set the following context parameters in web.xml (Figure 3.2.2):

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

Figure 3.2.2 Parameters in web.xml

So, as we describe above, we use files with .xhtml extension to represent the interface for our users. These files are in the View folder. The view folder is in the Web folder along with the CSS and JS resource files and all the necessary pictures and fonts. (Figure 3.2.3)

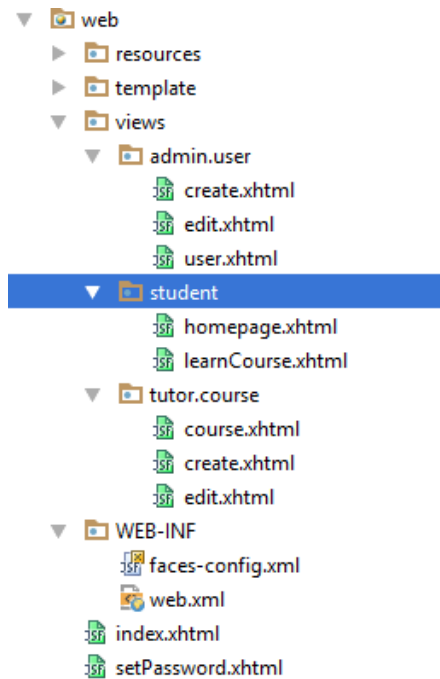


Figure 3.2.3 Web folder in the project

For standardizing the form of pages, we use the templating. In web application, most pages are follow a similar web interface layout and styling, for example, same header and footer. In JSF 2.0, you can use Facelets tags to provide a standard web interface layout easily. We use 4 Facelets tags to build page from a template:

- `ui:insert` – Used in template file, it defines content that is going to replace by the file that load the template. The content can be replace with “`ui:define`” tag;
- `ui:define` – Defines content that is inserted into template with a matching “`ui:insert`” tag;
- `ui:include` – Similar to JSP’s “`jsp:include`”, includes content from another XHTML page;
- `ui:composition` – If used with “`template`” attribute, the specified template is loaded, and the children of this tag defines the template layout; Otherwise, it’s a group of elements, that can be inserted somewhere. In addition, JSF removes all tags “outside” of “`ui:composition`” tag.

3.2.3 Controller part

As a controller part of our web-application we use Managed Beans. A typical JavaServer Faces application includes one or more managed beans, each of which can be associated with the components used in a particular page. This section introduces the basic concepts of creating, configuring, and using managed beans in an

application. A managed bean is created with a constructor with no arguments, a set of properties, and a set of methods that perform functions for a component.

There are several scope annotations in managed beans. Scope annotations set the scope into which the managed bean will be placed. If scope is not specified then bean will default to request scope. Each scope is briefly discussed below:

- @RequestScoped. Bean lives as long as the HTTP request-response lives. It get created upon a HTTP request and get destroyed when the HTTP response associated with the HTTP request is finished;
- @NoneScoped. Bean lives as long as a single EL evaluation. It get created upon an EL evaluation and get destroyed immediately after the EL evaluation;
- @ViewScoped. Bean lives as long as user is interacting with the same JSF view in the browser window/tab. It get created upon a HTTP request and get destroyed once user postback to a different view;
- @SessionScoped. Bean lives as long as the HTTP session lives. It get created upon the first HTTP request involving this bean in the session and get destroyed when the HTTP session is invalidated.

At most of our managed beans we use SessionScoped annotation, because information which is stored in this managed bean should be accessed from different pages as long as session is still active.

Our managed beans is stored in the Page folder, because they provide all necessary information to display for the users (Figure 3.2.4).

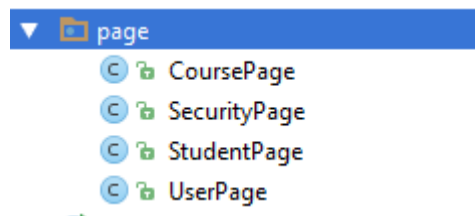


Figure 3.2.4 Location of the Managed beans

3.3 Project interface and functionality description

There will be screenshots and the descriptions of these screenshots...

4. ORGANIZATIONAL-ECONOMIC PART

Nowadays, programming is very popular profession. Now it is clear that with the development of technologies knowledge does not necessarily get at the university, you can get them from the comfort of home. Therefore, web site development for school management software is a hot topic.

The aim of this diploma work is the opportunity for anyone, who is interested in starting their own programming school, places all the necessary materials on the web-site and to get feedback from students. Students will have the opportunity to sign up for courses they are interested in and learn programming languages.

In accordance with the tasks described above, in the next stage of graduate design we will calculate the cost-effectiveness of web-application for resource management of programming school. For evaluating the effectiveness, it is necessary to solve the following problems:

- calculation of the cost of developing the site;
- calculation of the profit from the web-site;
- payback of the educational web-site;

4.1. Selection and justification of the economic efficiency calculation method

Economic efficiency is the ratio of the cost of the development, implementation and operation of the system and profit from its introduction. Production efficiency indicators system should give a comprehensive evaluation of the use of all enterprise resources, and contain all the general economic indicators.

Basic methods for determining the main financial investments in information technology are as follows:

- NPV (Net present value);
- IRR (Internal rate of return);
- Payback.

Calculation of cost-effectiveness will be calculated based on the payback period of development. Methods of assessing the effectiveness of the project suggest the need to evaluate "expenses" and "profit" part of the project.

4.2 Analysis of dangerous and harmful factors

The development process of the web-application for resource management of the programming school includes the following stages:

- Research the field;
- Requirements;
- Development of the system's back-end;

- Development of the system's front-end;
- Testing of the system;
- Deployment.

Total time of the system development is 3 month. The information about the time distribution of the development stages over this period is shown in the Table 4.1.

Table 4.1 – The time distribution of the development stages

Stage	Time, days	Time, %
Research	8	8.69
The system requirements	10	11.1
Back-end	32	34.78
Front-end	20	21.74
Testing	15	16.3
Deploying	7	7.6
Total	92	100

The components of the development cost are as follows:

- Material costs;
- Basic salaries and taxes;
- Computer time cost;
- Cost of tools;
- Overheads.

4.2.1 Material costs

Material cost is the cost of the raw materials, which were used during the development process (auxiliary materials, energy, and other costs which are part of the product). In our case the materials were used in the system development are shown on the Table 4.2.

Table 4.2 – the materials information

Name	Price, tg	Quantity, unit	Cost, tg
Paper	500	1	500
Pen	80	10	800

Continuation of Table 4.2

Flash drive	2000	2	4000
Board	5000	1	5000
Marker	200	3	600
Total			10900

As we can see from the table, the total cost of all raw materials is 10900 tenge. The most expensive things are board and flash cards.

4.2.2 Basic salaries and taxes

The basic salary is the salaries of all employees, which are involved in the system development. We have only two employees, which will implement the system. One of them is the system developer, and other is the system tester.

The basic salary is calculated by the following formula:

$$S_{basic} = \sum_{i=1}^n S_{avg,i} , \quad (4.1)$$

where $S_{avg,i}$ - the average salary of the i -employee;

n - the number of employees.

As we have already mentioned above the development time duration is 3 month or 92 days. The monthly salary of the developer is 80000 tenge. The average developer daily salary can be calculated as follows:

$$S_{day} = \frac{80000}{20} = 4000 \text{ tenge/day}$$

Total developer salary is:

$$S_{developer} = 92 * 4000 = 368000 \text{ tenge.}$$

The monthly salary of the tester is 70000 tenge. So, according to formula he average tester daily salary is:

$$S_{day} = \frac{70000}{20} = 3500 \frac{\text{tenge}}{\text{day}}$$

Total salary of the tester is:

$$S_{tester} = 92 * 3500 = 322000 \text{ tenge.}$$

Now, we can calculate the basic salary, which is a sum of the developer salary and the tester salary.

$$S_{basic} = S_{developer} + S_{tester} = 368000 + 322000 = 690000 \text{ tenge} \quad (4.2)$$

where $S_{\text{developer}}$ – developer's salary, tg
 S_{tester} – tester's salary

Taxes are calculated as the percent of the total salary of all employees. In our case, the percent is 18% (10% for pension contributions and 8% for income tax).

$$T = 0.2 * S_{\text{total}} = 0.2 * 690000 = 138000 \text{ tenge} \quad (4.3)$$

where S_{total} – total salary

4.2.3 Computer time expenses

Computer time expenses are dependent on several factors:

- Work time on a computer;
- Cost of 1 hour work on a computer.

Cost of 1 hour work on a computer consists of equipment amortization and cost of electricity and can be calculated in the following way:

$$C_{\text{hour}} = 0.3 \frac{\text{kW}}{\text{hour}} * 20.78 \frac{\text{tenge}}{\text{kW}} = 6.234 \text{ tenge/hour} \quad (4.4)$$

Only first stage of the development, i.e. requirements, does not include work time on a computer. The time can be calculated by the next formula:

$$T_{\text{pc}} = 0.9 * T_{\text{res}} + 0.6 * T_{\text{back}} + 0.65 * T_{\text{front}} + 0.7 * T_{\text{test}} + 0.6 * T_{\text{depl}}, \quad (4.5)$$

where T_{res} - the amount of time spent on research;

T_{back} - the amount of time spent on back-end development;

T_{front} - the amount of time spent on front-end development;

T_{test} - the amount of time spent on testing;

T_{depl} - the amount of time spent on deployment.

So, work time on a computer is:

$$T_{\text{pc}} = 0.9 * 9 + 0.9 * 35 + 0.8 * 22 + 0.7 * 16 + 0.6 * 8 = 73.2 \text{ days}$$

Transform this value to hours, we get:

$$T_{\text{pc}} = 73.2 * 8 = 585.6 \text{ hours}$$

Cost of electricity is calculated by the following formula:

$$C_{\text{electricity}} = T_{\text{pc}} * C_{\text{hour}} = 585.6 * 6.234 = 3650.63 \text{ tenge} \quad (4.6)$$

In our case we consider only the operation expenses. The formula which we will use is:

$$A_e = \frac{C_e * N_a * T_{\text{pc}}}{365 * 100}, \quad (4.7)$$

where C_e - the equipment cost;

N_a - the amortization norm;

T_{pc} - the usage time of the equipment.

The employees use the personal computers with the following characteristics:

- Intel Core i7 processor;
- 8 GB RAM;
- 1000 GB HDD;
- NVIDIA GeForce GT 630M video card.

The price of the computer is 320000 tenge. The total equipment cost is 640000 tenge.

The amortization expenses are:

$$A_e = \frac{640000 * 20 * 73.2}{365 * 100} = 25670.14 \text{ tenge}$$

Computer time expenses are the sum of the amortization cost and electricity cost:

$$C_{cte} = 25670.14 + 3650.63 = 29320.77 \text{ tenge}$$

4.2.4 Cost of tools

During the development of the web-application we used IntelliJ IDEA Ultimate from JetBrains toolbox. Monthly subscription costs US \$49.90. We buy it for 3 month, so the cost of the usage according to the dollar to tenge exchange rate was:

$$C_{t1} = (49.90 * 3) * 345 = 51646.5 \text{ tenge}$$

To place our web-site in the web we had to buy domain and hosting. We choose universal hosting on the host.kz platform. It has 1000 mb of the disk space, 100 e-mails, 10 MySQL databases, 10 aliases, 10 additional domains. It's price is 1300 tenge per month, we rent in on 1 year. And we also bought the domain name. On the ".kz" domain zone it's costs 3360 tenge per year.

$$C_{t2} = (1300 * 12) + 3360 = 18960 \text{ tenge}$$

Other tools for developing an application are free.

4.2.5 Total cost

The total development cost of the whole system is 938827.27 tenge. The distribution of costs you can see on Table 4.3 and Figure 4

Table 4.3 – The cost distribution

Cost name	Cost amount, tg
Material costs	10900
Total salary	690000

Continuation of Table 4.3

Taxes	138000
Computer time expenses	29320.77
Cost of tools	70606.5
Total	938827.27

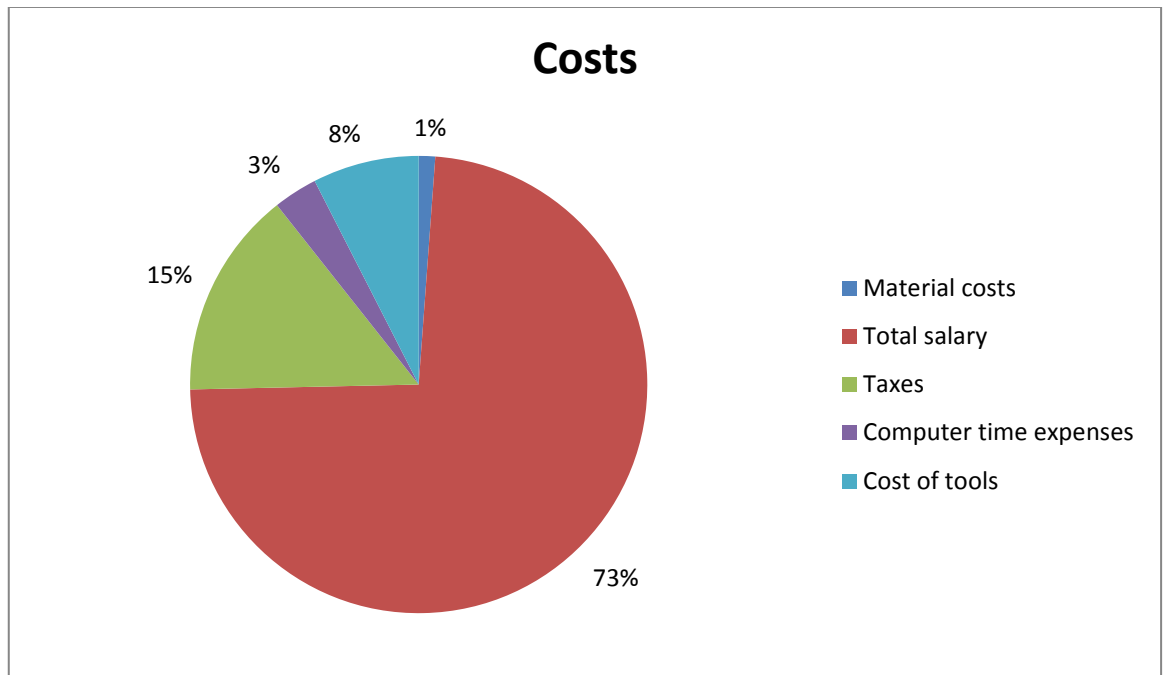


Figure 3.1 – the total development cost

So, as we can see according to the pie chart, most of the budget goes to the salary of the workers. The second one is taxes. After that goes cost of tools for developing and maintenance the application. The least expenses are the material costs.

4.2.6 Profit

For many people, who wants to learn programming it is too expensive to pay for the university. A lot of people want to learn programming not for the profession, but like the hobby, so they want to find good teachers in the web, and pay not big money for it.

From the other side, there are a lot of those, who know programming good, and they want to get money from it. They want to teach the student, but not all of them have the teacher education, so they cannot work at the university or school.

In our web-application for the resource management of programming school we give an opportunity for those who knows programming to teach others and for those who wants to know programming – learn from teachers, which they choose.

For the teachers registration is free, they can add courses to their profiles; add lectures and tasks without any expenses. However, students will pay for the courses for which they wish to subscribe.

After registration in the system, courses for which he has paid become available to the student. He can study lectures, complete assignments, communicate with the teacher in the application.

We, as the owners of the system, will get 20% from each subscription.

Assume, that we have 5 teachers on the web site, each teacher has three courses, one course will costs US \$9.99 per month, second one – US \$14.99 and the third one US \$4.99. Each teacher has 15 students, assigned to each course.

So our profit from these courses, according to the dollar to tenge exchange rate will be:

$$C_p = (9.99 + 14.99 + 4.99) * 5 * 15 * 345 * 0.2 = 155094.75 \text{ tenge}$$

The payback period is 6.05 months. It means that we need about half a year in order to recoup the preliminary expenses.

4.3 Conclusion for the chapter

Base on the calculations above, we can say that the project is perspective. Project payback of less than half a year, in the long term the resulting revenue will only grow with the expansion of the audience.

The universality of this website for resource management of programming school makes possible to reuse the code, which may reduce further investment in development.

According to the calculations in this chapter, we can say that the bulk of the investment will go to pay for developer productivity. The project itself does not require a significant investment, it should be less than 1 million tenge.

5. Labor safety and industrial ecology

Occupational safety is the measures aimed at ensuring human security, the preservation of his health, the development of methods and means of protection by reducing the impact of harmful and hazardous factors to acceptable values, development of measures to limit the damage in the aftermath of emergencies.

The aims of the OSH are detection and study of the environmental factors that adversely affect human health and the weakening of these factors to the safe limits or excluding them if possible. The workplace must provide safeguards against the possible effects of hazardous and harmful factors of production.

This section of the graduation project is devoted to the following issues:

- determination of optimal conditions for programmer's work;
- analysis of dangerous factors;
- requirements for the work equipment;
- safety during emergencies.

5.1. Characteristics of workplace

The aim of this graduate work is to develop the universal web-application for resource management of programming school. The users can work with this application using stationary PC or laptop, so the work place could differ.

During work with a computer person is exposed to a number of dangerous and harmful production factors: electromagnetic, infrared and ionizing radiation, noise and vibration, static electricity, and others.

Significant mental and neuro-emotional stress, high intensity visual work and a big load on the muscles of the hands characterize working with the computer. The great importance of the OSH is the rational design and layout of the workplace, which is responsible for maintaining optimal working position for user.

Also, it is necessary to reach the correct mode of work and rest while working with a computer. Otherwise, the users mark a significant voltage on visual apparatus with the appearance of complaints of dissatisfaction, headaches, irritability, sleep disturbance, fatigue and pain in the eyes, in the lower back, neck and hands. Therefore, it is crucial for workplace to be safe and suitable for continuous work.

We will consider the typical room where the user can work with this web-application (Figure 5.1.1). The room parameters are following:

- It is located in the second floor of the dwelling house;
- The height of the room is 2.5 m;
- The area of the room is 10 m²;
- There is a window with height 1.8 m and width 1.4 m;

- There are two exits, one of it leads to another room and others leads to the hallway;
- There are two electrical outlets;
- One lamp of artificial light;
- The workplace is 1.8m to 2.5m;

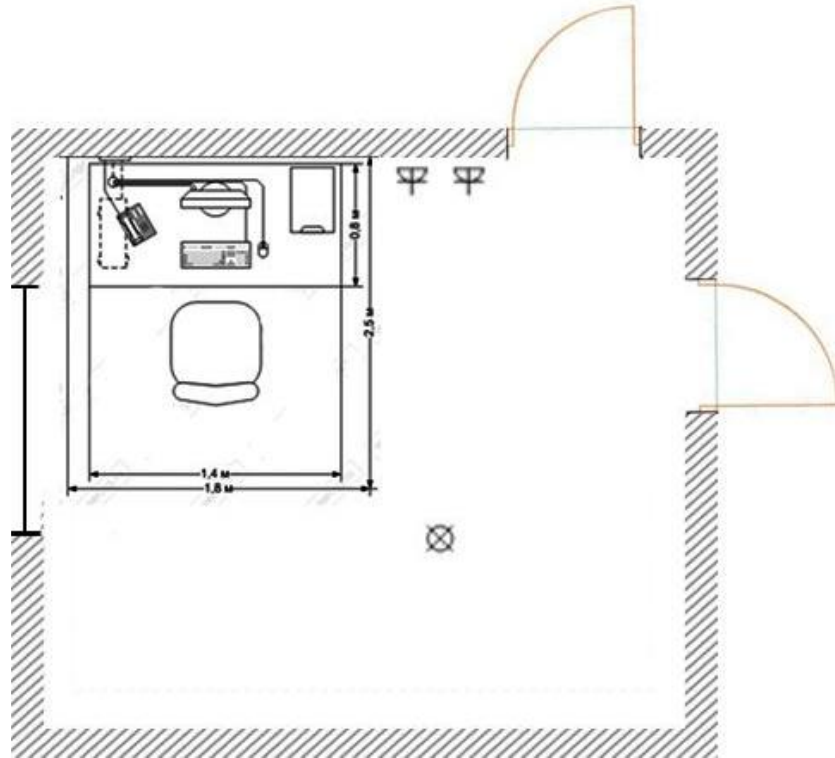


Figure 5.1.1 – Scheme of the workplace

According to the requirements for rooms during the work at the computer, the area at one workplace for adult users has to make not less than 6m², and volume not less than 20 m³.

According to Labor Code of the Republic of Kazakhstan [1] during working hours, the temperature, lighting and ventilation in the room where the workspace are located, must comply with sanitary and epidemiological requirements. So, for maintenance of optimum values of a microclimate, room has to be equipped with systems of heating and air conditioning. According to Sanitary-epidemiological requirements to working conditions with sources of physical factors [4], air temperature should be in the cold season is not over 22-24°C, in the warm season 23-25°C. Relative humidity should be 40-60%. The floor surface in rooms has to be flat, without hollows, non slipping, convenient for damp cleaning, possess anti static properties.

So, we can conclude, that the room which is described above is meets all the

necessary characteristics. The volume and the area of the workplace are not less than correct values. The floor surface is flat and convenient for damp cleaning.

5.2 Analysis of dangerous and harmful factors

There are several types of dangerous and harmful production factors. Among them are:

- Physical
- Psychophysical;
- Chemical;

During work time, the programmer may be subject to the destructive influence of the following physical factors:

- High or low temperature; (less than 20°C or bigger than 25°C)
- High or low humidity; (less than 40% or bigger than 60%)
- The lack of brightness in the workplace;
- Noise exceeding permissible limits;
- Elevated levels of electromagnetic fields;
- Increased level of static electricity;
- Risk of electric shock;
- Excessive dust and air pollution;
- Increased level of ionizing radiation.

The psychologically harmful factors affecting on the programmer during his work are as follows:

- Neuro-emotional overload;
- Mental stress;
- Overstrain of the visual analyzer.

The last factor is chemical. It means that the concentration of chemical substances (carbon dioxide, ozone and etc.) can be higher, so it violates the sanitary norms.

5.2.1 Microclimate at the workspace

The other very important thing is the creating of optimal conditions for the human body heat exchange with the environment, i.e. microclimate.

Computers cause substantial heat generation, which may lead to an increase in temperature and decrease in relative humidity. In the room with the computers must

be met certain parameters climate. The Sanitary-epidemiological requirements established values of microclimate parameters, creating a comfortable environment [4]. These rules are set depending on the time of year, the nature of the labor process and the nature of the production facilities.

To ensure comfortable conditions are used as organizational methods (rational organization of work, depending on the time of day and year, alternating work and leisure), and technical equipment (ventilation, air conditioning, heating system).

According to the “Sanitary-epidemiological requirements to working conditions with sources of physical factors (computers and video terminals), affecting human” approved by Order of the Republic of Kazakhstan Minister of health on January 21, 2015 № 38 [4] optimum microclimate parameters for the working premises values are given in Table 5.2.1.

Table 5.2.1. Optimal climate standards for premises

Period of the year	Air temperature, max, °C	Air humidity, %	Rate of air, m/s
Cold	22-27	40-60	0.1
	23-21	40-60	0.1
Warm	23-25	40-60	0.1
	22-24	40-60	0.2

5.2.2 Noise levels at the workspace

Noise impairs the working environment by providing a harmful effect on the human body. Working in conditions of prolonged noise exposure experienced irritability, headaches, dizziness, memory loss, fatigue, loss of appetite, pain in the ears, and so on. Such violations in a number of organs and body systems can cause negative changes in the emotional state of a person up to the stress.

Under the influence of the noise is reduced concentration, violates the physiological functions, there is fatigue due to increased energy costs and mental stress, deteriorating voice switching. All this reduces human productivity, quality and safety.

According to Sanitary-epidemiological requirements to working conditions

with sources of physical factors [4], the noise level in the room should not exceed 60 dB, the noise level at the workplaces with noisy units of computers should not exceed 75 dB.

In the Table 5.2.2 we specified limit sound levels depending on the category of severity and intensity of work that are safe for health and performance.

Table 5.2.2. The sound levels at the workspaces, dB

Category labor intensity	Category labor heaviness			
	Light	Medium	Hard	Max
Weak intensity	80	80	75	75
Medium intensity	70	70	65	65
Strong intensity	60	60	-	-
Max intensity	50	50	-	-

To conclude, noise level at the workplace does not exceed the upper limit of normal noise level according to GOST 12.1.003 [8].

5.2.3 Artificial and natural lighting

Properly designed and executed production lighting improves the conditions of visual work, reduces fatigue, increases productivity, beneficial effect on the production environment, providing a positive psychological impact on the employee, improves safety and reduces injuries.

Lack of light leads to eye strain, reduces attention, leading to premature onset of fatigue. Extremely bright light is blinding, irritation and pain in the eyes. Wrong direction of the light in the workplace can create harsh shadows, glare, disorient working. All of these causes can lead to accidents or occupational diseases, therefore, it is so important the correct calculation of illumination.

There are three types of lighting - natural, artificial and combined (both natural and artificial).

Natural lighting - lighting of premises with daylight. Natural light varies widely depending on time of day, time of year, the nature of the area and a number of other factors.

Artificial lighting is used when working at night and during the day when natural light is not enough (cloudy weather, the short daylight hours).

Lighting, in which the lack of natural light supplemented by artificial lighting is called combined.

When carrying out a high visual accuracy (the smallest size object distinction 0.3 ... 0.5 mm), the coefficient of natural lighting should not be less than 1.5%, while the visual medium accuracy (the smallest size object distinction 0.5 ... 1.0 mm) coefficient should not be below 1.0%. As artificial light sources commonly used fluorescent lamps, which are combined in pairs in the lamps, which must be positioned above the work surface evenly.

In addition the entire field of view should be fairly evenly lit - it is basic hygiene requirements. In other words, the level of room lighting and brightness of the computer screen should be about the same, because a bright light in the area of peripheral vision considerably increases eye strain and leads to fatigue.

According to all the information above, it can be conclude, that during the development of this diploma I has not been influenced by dangerous and harmful factors. Temperature was comfortable, noise did not exceed the permissible limits, and I was not exposed to stress.

5.3 Health and safety during the work

For the programmer, it is very important not only to comply with the rules of occupational safety during the work at the computer, but before work and after work.

There are some actions, which should be done before starting a work:

- Arrive at work early to avoid rush and, as a consequence, falls and injuries;
- Inspect the work area and equipment. Remove all unnecessary items;
- Clear the display screen of a personal computer from dust. Adjust the height and angle of the screen;
- Adjust seat height. Check serviceability of equipment;
- On seen the shortcomings and faults immediately report to the head and to the troubleshooting and resolution of the head does not get to work.

Safety requirements during operation:

- Follow and comply with sanitary norms [4], modes of work and rest;

- Observe the rules of operation of computer equipment in accordance with the instructions for use [2];
- It is necessary for the entire working day to keep in order and cleanliness of the workplace. During the working period the display screen should be clear of dust at least once.
- Prevent ingress of moisture on the surface of the system unit (CPU), monitor, work surface keyboard, disk drives, printers, and others devices;
- Produce own opening and repair of equipment.

Safety requirements after work:

- Remove all unnecessary items from the workplace;
- Turn off and disconnect the equipment.

5.3.1 Work and rest while using the computer

During operation of the computer for a long time, the programmer is forced to reside in a sitting position. In this position, the body is constantly lacks in physical activity. Basic programmer's activity is expressed in the movement of the hands, which affects the muscles of the back and shoulders, the position of the spine and legs. It is necessary to limit the time spent in a sitting position, to avoid the emergence of diseases. The programmer must alternate between time spent at the computer intermittently. During leisure, you must perform some muscle load on the musculoskeletal system, to knead her back and shoulders.

Types of work on the computer are separated into 3 groups: Group A - Work on reading information from the screen; Group B - work on putting information; Group B - creative work in a mode of dialogue with the computer.

At a 8-hour shift the regulated breaks should be established as follows:

- For the first category of works in 2 hours from the beginning of the work and 2 hours after the lunch break of 15 minutes each;
- For the second category of works - 2 hours from the start of the work and 1.5-2 hours after the lunch break lasting 15 minutes each, or lasting 10 minutes after every hour of work;
- For the third category of works – 1.5-2 hours from the start of the work shift and after 1.5-2 hours after the lunch break lasting 20 minutes, or lasting 15 minutes after every hour of work.

The duration of continuous operation on a computer without a regulated break must not exceed 2 hours. Independent breaks lasting 1-3 minutes are also effective.

The working on the diploma applies to the third category, so I alternated the work and breaks according to the schedule. Each hour for 15 minutes, I knead the spine, relax the muscles of the long-term work in a sitting position.

5.3.2 Requirements for the organization and the workplace equipment

Workplace of the programmer should occupy an area of not less than 6 m, the height of the room should not be less than 4 m, and the volume - not less than 20 m³ per person. Desk can be of any design that meets modern requirements of ergonomics and allows programmer to conveniently place all the necessary equipment on the working surface.

Depth of the table surface should be 800 mm (allowed 600 mm), width - respectively 1600 mm and 1200 mm. Table working surface should not have sharp corners and edges, have a matte or semi-matt factor. Desk should be legroom height of 600 mm, width - not less than 500 mm, depth at the level of the knees - not less than 450 mm, and at the level of elongated legs - not less than 650 mm.

Fast and accurate reading of the information provided at the location of the plane of the screen below the user's eye, preferably perpendicular to the normal line of sight (the normal line of sight 15 degrees down from horizontal).

The keyboard must be placed on the table surface at a distance of 100-300 mm from the edge facing the user. To provide physiologically safe working position apply tilt-work chairs with seat and back, adjustable height and angle of inclination, as well as the distance back from the front edge of the seat.

Correct sitting position while using the computer is shown in the Figure 5.3.1.



Figure 5.3.1 – The proper sitting position while using the computer

The room, which was used for the writing of this diploma work, is fully consistent with the requirements for the organization of the workplace described above. According to GOST 12.2.032-78 [9] ergonomic chair and table fully provide

the proper sitting position.

5.4 Health and safety during emergencies

5.4.1 Electrical safety

Working room can be classified as Class 1 for electrical hazards, it means this room without heightened risk of (dry, dust-devil, with normal air temperature, insulated floors and a small number of earthed devices).

Electrical safety in the working room provided technical ways and means of protection, as well as organizational and technical measures.

Let's consider the main causes of electric shock in the workplace:

- Touching the metal electrical parts (chassis, computer peripherals), which may be under stress as a result of damage to the insulation;
- Unauthorized use of electrical appliances;
- The lack of coaching staff on the rules of electrical safety.

The main event for the electrical safety is the instruction and training on safe working practices, as well as verification of knowledge of safety rules and regulations in accordance with the position in relation to the work performed.

5.4.2 Fire safety

Fire safety - state of the object, which prevents the possibility of fire, and in the case of its occurrence prevents exposure to hazardous factors and it protected property.

Fire safety is the state of security of the person, property, society and the state from fires. There is also set of instruction and rules to eliminate fire:

- Every employee should be trained in fire safety;
- Each door should be able to be opened easily in the direction of escape;
- Every employee should be trained in fire safety;
- Each door should be able to be opened easily in the direction of escape;
- Fire extinguishers shall be subjected to external inspection at regular intervals;
- All employees should smoke only in those areas designed especially for this purpose;
- The room must be equipped with a fire alarm system and firefighting system;
- The evacuation plans must be freely available for public access;
- The workplace must be kept clean;
- Electrical switches, cable lines must be protected by stationary or mobile fire units in accordance with "Electrical Regulations";

- Every employee should be trained in fire safety;
- All electrical installations must be equipped with protective devices against short circuit and other abnormal conditions that can lead to fire;
- Electrical systems and equipment must meet requirements for existing “Fire safety of buildings and structures” [5]
- Fire equipment serviceability should be controlled and maintained.

The source of the fire in the workplace can be a power failure. To prevent fire, it is necessary to use equipment that meets the requirements of fire safety. The rooms and corridors of jobs should not clutter up the various items: paper, equipment, etc. All paper waste from printers and other waste paper should be removed in a timely manner.

The workplace can not be used for the installation of fire extinguishing using water, foam, dry chemical powder. To combat the small local fire should be used carbon dioxide extinguishers, which must be located in easily accessible locations. An advantage of carbon dioxide fire extinguishing agents is that they have high extinguishing efficiency, and do not damage the electronic equipment. In addition, carbon dioxide is not a conductor of electricity, which is important when putting out a fire in the room where the equipment is installed, the consumed electric power.

Detection and warning in case of fire are the sensors which should be installed in the exhaust duct, in an underground space, in the storage media, and other areas.

5.5 Engineering calculations on safety and labor protection

5.5.1 Artificial and natural illumination

We will start from the calculations of illumination. Calculation of lighting made for a room with area of 10 m², width of 2 m, and height of 2.5 m. We use the method of the luminous flux.

To determine the number of luminaires determine the luminous flux incident on a surface according to the formula [7]:

$$F = \frac{E \times K \times S \times Z}{n}, \quad (5.5.1)$$

where E - required illumination, lux;

K - safety factor;

S - area of the premises, m², in our case it is equal 10 m²;

Z - the ratio of average luminance to the minimum (usually taken as equal to 1.1-1.2, let Z be equal 1.1);

n - utilization ratio of lighting system.

The value of n is defined by the table of coefficients using different fixtures.

To do this, we calculate the index according to the formula premises: [7]

$$I = \frac{S}{H \times (A+B)}, \quad (5.5.2)$$

where S - area of the premises, m², in our case it is equal 10 m²;

H - height of suspension, 2.1 m;

A - width of premises, 5 m;

B - length of premises, 2 m.

We determine the utilization rate based on the values of the reflection coefficients and the index premises, n = 35. [7]

So, the luminous flux incident on a surface is equal 14142.8571, according to the formula (5.5.1).

Calculate the required number of lamps according to the formula:

$$N = \frac{F}{F_1}, \quad (5.5.3)$$

where F – total luminous flux;

F₁ - luminous flux of one lamp;

To conclude, according to calculations, our workplace needs 4 lamps. For lightning we choose fluorescent lamp, the light output of which F₁ = 4320 lux.

5.5.2 Calculations of ventilation

We will continue with the calculations of ventilation. The heat dissipation from artificial light sources is calculated by the following formula

$$Q_{\text{light}} = 860 * N_{\text{light}}, \quad (5.5.4)$$

where N_{light} is a total power of artificial light sources, kW.

In our case the heat dissipation equals to:

$$Q_{\text{light}} = 860 * 0.6 = 516 \text{ kcal/h}$$

The heat dissipation from people is calculated as follows:

$$Q_{\text{staff}} = Q_{\text{human}} * n, \quad (5.5.5)$$

where Q_{human} – the heat dissipation from a single person (55 kcal/h);

n - the number of people.

In our case the heat dissipation from people is:

$$Q_{\text{staff}} = 55 * 1 = 55 \text{ kcal/h}$$

The solar radiation, i.e. the heat coming from the sun through the window is calculated by the formula:

$$Q_{\text{window}} = F * q, \quad (5.5.6)$$

where F - a window area (2.52 m^2);

q - the heat amount per area unit ($25 \text{ kcal/h} \cdot \text{m}^2$).

The solar radiation is:

Heat loss due to construction fences is calculated by the following formula:

$$Q_{cf} = k * F * (T_{indoor} + T_{outdoor}), \quad (5.5.7)$$

where F - the construction fences area, m^2 ;

$T_{outdoor}$ - the outdoor temperature, K;

T_{indoor} - the indoor temperature, K.

Now, we can calculate the extra heat amount in the room:

$$Q_{extra} = 516 + 55 + 63 - 190.75 = 443.25 \text{ kcal/h}$$

In case of presence of the extra heat we need to calculate amount of air needed to create the normal working conditions. The amount is calculated by the formula:

$$L = \frac{Q_{extra}}{C * y * (T_{indoor} - T_{outdoor})}, \quad (5.5.8)$$

where C - the specific air heat ($0.24 \text{ kcal/kg} \cdot \text{K}$);

y - the air density, kg/m^3 ;

T_{indoor} - temperature indoor, K;

$T_{outdoor}$ - temperature outdoor, K;

The amount of air is:

$$L = \frac{443.25}{0.24 * 1.185 * 20} = 77.927 \text{ m}^3/\text{h}$$

The coefficient of air exchange is calculated by the formula:

$$K = \frac{L}{V_{room}}, \quad (5.5.10)$$

where V_{room} - the room volume, m^3 .

We will consider only staff room, which has the following characteristics: the area is 10 m^2 and height is 2.5 m .

The coefficient of air exchange is:

$$K = \frac{77.927}{2.5 * 10} = 3.11$$

Because of the coefficient is high, we can conclude, that the best option is an air conditioning, i.e. automatic maintaining indoor air condition in accordance with certain requirements (temperature, humidity, air mobility) regardless of changes in ambient air conditions and the conditions in the room.

5.6 Conclusion

In this chapter of the diploma work were set out requirements for the workplace of programmer. Created conditions should ensure a comfortable work. Based on the study of the literature on this issue, we have determined the optimum size of the desktop and chair, size of the working surface, and calculated the optimum illumination of industrial premises, as well as the calculation of ventilation.

Compliance with all the above conditions will create the optimal organization of the working space of the programmer, will keep a good performance throughout the working day, and increase programmer productivity that will contribute to the rapid development and introduction of new production technology.

All calculations match up the standards:

- Microclimate meets sanitary-epidemiological requirements [4] to working conditions;
- The noise level does not exceed the upper limit of normal GOST 12.1.003-83;
- Fire and electrical safety meets SNIP RK 2.02-05-2009 requirements [5] in production areas;
- Artificial and natural lightning meets sanitary-epidemiological requirements [4].

CONCLUSION

The goal of this diploma work is development of a universal web-application for the resource management of programming school. We have chosen this application in order to satisfy the needs of the tutors for their own schools of programming and the needs of the students to learn programming.

The implementation of the application is useful for the increasing of the computer literacy of the population and training future ICT specialists.

The structure of the diploma project includes the analysis of the similar projects all over the world, a review of the education programming in Kazakhstan and the world, an overview of the advantages and disadvantages of distance learning, economic plan of how get profit from this web-application and spread it among the schools and students.

In addition, the implementation of the project, its specifics and development of application were described in details. It includes the technical characteristics, architecture with the description of the functionality.

The project was realized using capabilities of the Java programming language and JSF framework for developing web-applications. During the implementation of the project we analyzed the methods, approaches, market situation of the education sphere, provide labor safety rules and economic justification. Also the project works online, we include description of the architecture and structure, specifically classes, methods, databases and operations.

So, the web-application for the resource management of programming schools is developed within this project and perspectives of its integration in the educational system were considered.

REFERENCES

1. "Labor Code of the Republic of Kazakhstan" 2016 (from November 23, 2015 № 414-V).
 2. "ST Republic of Kazakhstan 12.0.004-2010". Occupational safety standards system.
 3. Sanitary regulations "Sanitary-epidemiological requirements to ensure radiation safety" dated on March 27, 2015 № 261 on Radiation Safety of the Population.
 4. "Sanitary-epidemiological requirements to working conditions with sources of physical factors (computers and video terminals), affecting human" approved by Order of the Republic of Kazakhstan Minister of health on January 21, 2015 № 38.
 5. SNIP RK 2.02-05-2009. "Fire safety of buildings and structures" (with changes and additions as of 01.10.2015).
 6. "Rules of Fire Safety of Republic of Kazakhstan" on October 9, 2014 № 1077.
 7. Light calculation using method of utilization rates, URL: <http://www.technoluxtm.ru/useful/raschet-osveshennosti>
 8. GOST 12.1.003-83 Occupational safety standards system. Noise. General safety on 06 April 2015.
 9. GOST 12.2.032-78 Occupational safety standards system. Workplace when working sitting. General ergonomic requirements on 16 January 2015
<http://dev.mysql.com/doc/>
<https://www.jetbrains.com/help/idea/2016.1/meet-intellij-idea.html>
-
1. Alexander C., Ishikawa S., Silverstein M. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977.
 2. Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. Pattern-Oriented Software Architecture, Volume 1, A System of Patterns. Wiley, 1996, 476 pages
 3. Александреску А. Современное проектирование на C++ (серия C++ in Depth). - М.: Издательский дом "Вильямс", 2008.-336с
 4. Влссидес Дж. Применение шаблонов проектирования. Дополнительные штрихи. - М.: Издательский дом "Вильямс", 2003.-144с
 5. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - СПб.: Питер, 2001.-368с
 6. Кериевски Дж. Рефакторинг с использованием шаблонов. - М.: Издательский дом "Вильямс", 2006.-400с
 7. Коплиен Дж. Программирование на C++. Классика CS. - СПб.: Питер, 2005.-479с
 8. Ларман К. Применение UML и шаблонов проектирования, 2-е издание. - М.: Издательский дом "Вильямс", 2004.-624с

9. Эккель Б. Философия C++. Практическое программирование C++. Том 2. - СПб.: Питер, 2004.-608с

<http://zeroturnaround.com/rebellabs/top-4-java-web-frameworks-revealed-real-life-usage-data-of-spring-mvc-vaadin-gwt-and-jsf/>

core javaserver faces 3rd edition David Geary Cay S. Horstmann

https://ru.wikipedia.org/wiki/JavaServer_Faces

<http://www.primefaces.org/whyprimefaces>

APPENDIX A

Hibernate configuration files

Hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/diploma</property>
        <property name="hibernate.connection.autocommit">>true</property>
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="connection.username">****</property>
        <property name="connection.password">****</property>
        <mapping class="model.Role"/>
        <mapping class="model.Userinfo"/>
        <mapping class="model.Userrole"/>
        <mapping class="model.Usersecurity"/>
    </session-factory>
</hibernate-configuration>
```

Persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="2.0">
    <persistence-unit name="diplomaPersistence">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <class>model.Role</class>
        <class>model.Userinfo</class>
        <class>model.Userrole</class>
        <class>model.Usersecurity</class>
        <class>model.Course</class>
        <class>model.Lecture</class>
        <class>model.Task</class>
        <class>model.Usercourse</class>
        <class>model.Answer</class>
        <class>model.Mark</class>
        <properties>
            <property name="hibernate.connection.url"
```

```
value="jdbc:mysql://localhost:3306/diploma"/>
    <property name="hibernate.connection.driver_class"
value="com.mysql.jdbc.Driver"/>
    <property name="hibernate.connection.username" value="****"/>
    <property name="hibernate.connection.password" value="****"/>
    <property name="hibernate.archive.autodetection" value="class"/>
    <property name="hibernate.show_sql" value="true"/>
    <property name="hibernate.format_sql" value="true"/>
    <property name="hbm2ddl.auto" value="update"/>
    <property name="hibernate.connection.autocommit" value="true"/>
  </properties>
</persistence-unit>
</persistence>
```

APPENDIX B

Navigation configuration file

Faces-config.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">
  <navigation-rule>
    <from-view-id>*/</from-view-id>
    <navigation-case>
      <from-outcome>logout</from-outcome>
      <to-view-id>/index.xhtml</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <from-view-id>*/</from-view-id>
    <navigation-case>
      <from-outcome>userList</from-outcome>
      <to-view-id>/views/admin/user/user.xhtml</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <from-view-id>*/</from-view-id>
    <navigation-case>
      <from-outcome>userList</from-outcome>
      <to-view-id>/views/admin/user/user.xhtml</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <from-view-id>*/</from-view-id>
    <navigation-case>
      <from-outcome>userList</from-outcome>
      <to-view-id>/views/admin/user/user.xhtml</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <from-view-id>*/</from-view-id>
    <navigation-case>
      <from-outcome>userList</from-outcome>
      <to-view-id>/views/admin/user/user.xhtml</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
```



```

<from-view-id>*</from-view-id>
<navigation-case>
  <from-outcome>userCreate</from-outcome>
  <to-view-id>/views/admin/user/create.xhtml</to-view-id>
  <redirect/>
</navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <from-outcome>userEdit</from-outcome>
    <to-view-id>/views/admin/user/edit.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <from-outcome>courseList</from-outcome>
    <to-view-id>/views/tutor/course/course.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <from-outcome>editCourse</from-outcome>
    <to-view-id>/views/tutor/course/edit.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <from-outcome>pushLecture</from-outcome>
    <to-view-id>/views/tutor/course/create.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>

```

```

        <from-outcome>setPassword</from-outcome>
        <to-view-id>/setPassword.xhtml</to-view-id>
        <redirect/>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <from-view-id>*</from-view-id>
    <navigation-case>
        <from-outcome>index</from-outcome>
        <to-view-id>/index.xhtml</to-view-id>
        <redirect/>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <from-view-id>*</from-view-id>
    <navigation-case>
        <from-outcome>myCourses</from-outcome>
        <to-view-id>/views/student/homepage.xhtml</to-view-id>
        <redirect/>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <from-view-id>*</from-view-id>
    <navigation-case>
        <from-outcome>learnCourse</from-outcome>
        <to-view-id>/views/student/learnCourse.xhtml</to-view-id>
        <redirect/>
    </navigation-case>
</navigation-rule>
</faces-config>

```

APPENDIX C

Security managing bean

SecurityPage .class

```
@ManagedBean
@SessionScoped
public class SecurityPage implements Serializable {

    final static Logger logger = Logger.getLogger(SecurityPage.class);
    private EntityManagerFactory myEntityManagerFactory;
    public SecurityPage() {
        myEntityManagerFactory =
Persistence.createEntityManagerFactory("diplomaPersistence");
    }
    private EntityManagerFactory getEntityManagerFactory() {
        return myEntityManagerFactory;
    }
    public String login;
    public String password;
    public User user;
    public Userrole userRole;
    public Role role;
    public String newPassword;
    public String newRepeatPassword;
    public Usersecurity usersecurity;

    public String getAuth() throws NoSuchAlgorithmException {
        String s = null;
        if (this.login != null) {
            if (this.login.length() == 0) {
                return null;
            }
            if (checkUserLogin(this.login)) {
                if (checkUserSecurity(this.login, this.password)) {
                    this.user = getUserDataByLogin(this.login);
                    if (user != null) {
                        this.role = user.getRoleList().get(0);
                        if (user.getStatus().equals("ACTIVE")) {
                            if (!password.equals("qwerty123")) {
                                s = getHomePage(role.getRoleName());
                            } else {

```

```

        s = "setPassword";
    }
    } else {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Your account is blocked"));
    }
    }
    } else {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("WRONG PASSWORD"));
    }
    } else {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("USER DOES NOT EXIST"));
    }
    }
    return s;
}

```

```

public boolean checkUserLogin(String login) {
    boolean check = false;
    Object o = null;
    EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
    try {
        o = entityManager.createQuery("select u from Userinfo u where u.username =
:login").setParameter("login", login).getSingleResult();
    } catch (NoResultException nre) {
    }
    if (o != null) {
        check = true;
    }
    entityManager.close();
    return check;
}

```

```

public boolean checkUserSecurity(String login, String password) {
    boolean check = false;
    Object o = null;
    EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
    try {

```

```

        o = entityManager.createQuery("select u from Usersecurity u where
u.username = :login and u.password = :password")
        .setParameter("login", login)
        .setParameter("password", getMd5(password))
        .getSingleResult();
    } catch (NoResultException nre) {
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    if (o != null) {
        check = true;
    }
    entityManager.close();
    return check;
}

```

```

public List<Role> getRoleListByUserId(int userId) {
    EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
    List<Userrole> userroles = (List<Userrole>)
entityManager.createQuery("SELECT o FROM Userrole AS o where o.userRef =
:userId").setParameter("userId", userId).getResultList();
    List<Role> entities = new ArrayList<>();
    for (Userrole ur : userroles) {
        entities.add((Role) entityManager.createQuery("SELECT o FROM Role AS o
where o.id = :ur").setParameter("ur", ur.getRoleRef()).getSingleResult());
    }
    entityManager.close();
    return entities;
}

```

```

public User getUserDataByLogin(String login) {
    EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
    Userinfo userEntity = (Userinfo) entityManager.createQuery("select u from
Userinfo u where u.username = :login").setParameter("login",
login).getSingleResult();
    List<Role> roleList = getRoleListByUserId(userEntity.getId());
    User user = new User(userEntity.getId(), userEntity.getUsername(),
userEntity.getFullname(), userEntity.getEmail(), userEntity.getPhone(),
userEntity.getStatus(), roleList);
    entityManager.close();
}

```

```

    return user;
}

public String logout() {
    login = null;
    password = null;
    user = null;
    return "logout";
}

public String getHomePage(String role) {
    RoleName roleName = getType(role);
    switch (roleName) {
        case ADMIN:
            logger.trace("admin");
            return "views/admin/user/user.xhtml?faces-redirect=true";
        case TUTOR:
            logger.trace("tutor");
            return "views/tutor/course/course.xhtml?faces-redirect=true";
        case STUDENT:
            logger.trace("student");
            return "views/student/homepage.xhtml?faces-redirect=true";
    }
    return "views/404.xhtml?faces-redirect=true";
}

public boolean isAdmin() {
    for (Role r : user.getRoleList()) {
        if (r.getRoleName().equals(RoleName.ADMIN.getTypeValue())) {
            return true;
        }
    }
    return false;
}

public boolean isTutor() {
    for (Role r : user.getRoleList()) {
        if (r.getRoleName().equals(RoleName.TUTOR.getTypeValue())) {
            return true;
        }
    }
    return false;
}

```

```

    }

    public boolean isStudent() {
        for (Role r : user.getRoleList()) {
            if (r.getRoleName().equals(RoleName.STUDENT.getTypeValue())) {
                return true;
            }
        }
        return false;
    }

    public String setPassword() throws NoSuchAlgorithmException {
        if (newPassword.equals(newRepeatPassword)) {
            usersecurity = getUserSecurityByLogin(login);
            usersecurity.setPassword(newPassword);
            changePassword();
            return "index";
        } else {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Passwords do not match. Please, try again.));
            return null;
        }
    }

    public Usersecurity getUserSecurityByLogin(String login) {
        EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
        Usersecurity usersecurity = (Usersecurity)
entityManager.createQuery("SELECT o FROM Usersecurity AS o where o.username
= :username")
            .setParameter("username", login)
            .getSingleResult();
        entityManager.close();
        return usersecurity;
    }

    public void changePassword() throws NoSuchAlgorithmException {
        EntityManager entityManager =
getEntityManagerFactory().createEntityManager();
        entityManager.getTransaction().begin();
        entityManager.createQuery("update Usersecurity r set r.password = :newPass
where r.username = :username")

```

```

        .setParameter("newPass", getMd5(usersecurity.getPassword()))
        .setParameter("username", usersecurity.getUsername())
        .executeUpdate();
entityManager.getTransaction().commit();
entityManager.close();
}

```

```

public static String getMd5(String pass) throws NoSuchAlgorithmException {

    MessageDigest md = MessageDigest.getInstance("MD5");
    md.update(pass.getBytes());

    byte byteData[] = md.digest();
    StringBuffer sb = new StringBuffer();
    for (int i = 0; i < byteData.length; i++) {
        sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
    }
    System.out.println("Digest(in hex format):: " + sb.toString());
    StringBuffer hexString = new StringBuffer();
    for (int i = 0; i < byteData.length; i++) {
        String hex = Integer.toHexString(0xff & byteData[i]);
        if (hex.length() == 1) hexString.append('0');
        hexString.append(hex);
    }
    return hexString.toString();
}

public String getLogin() {
    return this.login;
}

public void setLogin(String login) {
    this.login = login;
}

public String getPassword() {
    return this.password;
}

public void setPassword(String password) {
    this.password = password;
}

```



```

    }

    public String getNewPassword() {
        return newPassword;
    }

    public void setNewPassword(String newPassword) {
        this.newPassword = newPassword;
    }

    public String getNewRepeatPassword() {
        return newRepeatPassword;
    }

    public void setNewRepeatPassword(String newRepeatPassword) {
        this.newRepeatPassword = newRepeatPassword;
    }

    public User getUser() {
        return this.user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public Userrole getUserRole() {
        return userRole;
    }

    public void setUserRole(Userrole userRole) {
        this.userRole = userRole;
    }

    public Role getRole() {
        return role;
    }

    public void setRole(Role role) {
        this.role = role;
    }
}

```