

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT - инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А.
_____ « ____ » _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка клиентской части системы для автоматизации учебного процесса

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

Выполнила: Щукин Д.М. Группа: ВТу-17-4

Научный руководитель: к.т.н., доцент Табултаев С.С

Консультанты:

по экономической части: Габелашвили К.Р.

_____ « ____ » _____ 2020 г.

по безопасности жизнедеятельности: Приходько Н.Г.

_____ « ____ » _____ 2020 г.

по программному обеспечению: Майкотов М.Н

_____ « ____ » _____ 2020 г.

Нормоконтролер: Абсатарова Б.Р.

_____ « ____ » _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

Кафедра IT-инжиниринг

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Щукину Данилу Михайловичу

Тема проекта: Разработка клиентской части системы для автоматизации учебного процесса

Утверждена приказом по университету № ____ от «____» _____ 2020 г.

Срок сдачи законченного проекта «__» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Angular- среда разработки сайта, IntelliJ idea – среда программирования, создание функций объектов.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) вход без регистрации;
- б) полная синхронизация клиента и сервера;
- в) просмотр и редактирование дневника, обмен данными между пользователями.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 14 таблиц, 54 иллюстраций, презентация (22 слайда).

Основная рекомендуемая литература:

1 Партыка, Т.Л. Информационная безопасность: Учебное пособие / Т.Л. Партыка, И.И. Попов. - М.: Форум, 2018. - 88 с.;

2 Баранова, Е.К. Информационная безопасность и защита информации: Учебное пособие / Е.К. Баранова, А.В. Бабаш. - М.: Риор, 2018. - 400 с.;

3 Основы безопасности жизнедеятельности. Алексеенко В.А., Матасова И.Ю., 2001. – 187 с "Безопасность в чрезвычайных ситуациях: Учебник" под ред. Н.К. Шишкина. – М., ГУУ, 2017. - 90 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

| Раздел | Консультант | Сроки | Подпись |
|---|---|-----------------------|---------|
| Технико-экономическое обоснование проекта | к.э.н., асоц. профессор Габелашвили К.Р. | 06.04 – 25.04.2020 | |
| Безопасность жизнедеятельности | доц. Приходько Н. Г. | 06.04 – 25.04.2020 | |
| Программное обеспечение | ст.преп. Майкотов М.Н. | 04.05 – 08.05.2020 | |
| Нормоконтролер | ст.преп. Абсатарова Б.Р. | 04.05 – 08.05.2020 | |

ГРАФИК

подготовки дипломного проекта

| Наименования разделов, перечень разрабатываемых вопросов | Сроки представления научному руководителю | Примечания |
|--|---|------------|
| Анализ предметной области | 13.01.20 | |
| Выбор программного обеспечения | 23.02.20 | |
| Сравнительный анализ аналогов | 04.03.20 | |
| Написание технического задания | 26.03.20 | |
| Создание UML диаграмм | 14.04.20 | |
| Реализация программного продукта | 15.05.20 | |

Дата выдачи задания «13 января 2020г.

Заведующий кафедрой _____ А.А.Досжанова

Научный руководитель проекта _____ С.С.Табуртаев

Задание принял к исполнению студент _____ Д.М.Щукин

АНДАТПА

Оқу орындарына арналған веб-сайт - бұл бүгінгі таңда ел университеттерінің алдында тұрған көптеген мәселелерді шешетін әмбебап веб-қосымшаны жасауға бағытталған жоба.

Біздің мақсатымызға жету үшін жоба Алматыдағы университеттерде қолданылатын негізгі сайттарды қарастырады. Бұл пайдаланушыларға қажет функцияларды және олардың дизайнерлік артықшылықтарын түсінуге мүмкіндік берді. Бұл жобада бірнеше интерфейс бар. Жобаның табиғаты пайдаланушының рөліне байланысты әртүрлі функционалдық мүмкіндіктерге ие болған кезде бай. Жалпы, бұл жобаның басты мақсаты - сайттың жауап беру уақытын азайту, серверге жүктемені азайту және студент пен оқытушы арасындағы байланысты жақсарту.

АННОТАЦИЯ

Веб сайт для учебных заведений - это проект, направленный на разработку универсального веб-приложения, которое решает множество проблем, с которыми сегодня сталкиваются вузы страны.

Для достижения нашей цели в проекте рассматриваются основные сайты используемые в вузах Алматы. Это позволило понять необходимые функции нужные пользователям и их предпочтениям в дизайне. Этот проект включает в себя более одного интерфейса. По своей природе проект богат, когда речь идет о разнообразных функциональных возможностях, которые зависят от роли пользователя. В общем, основная цель этого проекта - минимизировать время отклика сайта, снизить нагрузку на сервер и улучшить коммуникацию между студентом и преподавателем.

ANNOTATION

The website for educational institutions is a project aimed at developing a universal web application that solves many of the problems that the country's universities face today.

To achieve our goal, the project considers the main sites used in universities in Almaty. This made it possible to understand the necessary functions that users need and their design preferences. This project includes more than one interface. The nature of the project is rich when it comes to a variety of functionalities that depend on the role of the user. In general, the main goal of this project is to minimize the response time of the site, reduce the load on the server and improve communication between the student and teacher.

Содержание

| | |
|--|----|
| Введение | 8 |
| 1 Аналитическая часть | 9 |
| 1.1 Описание предметной области | 9 |
| 1.2 Цель разработки..... | 9 |
| 1.3 Особенности использования методов разработки | 10 |
| 1.4 Сравнительный анализ аналогов | 17 |
| 2 Проектная часть..... | 19 |
| 2.1 Создание проекта | 19 |
| 2.2 UML диаграмма | 23 |
| 2.3 Программная часть | 25 |
| 3 Экспериментальная часть | 39 |
| 3.1 Проверка нагрузки на сайт | 39 |
| 4 Техничко-экономическое обоснование проекта | 41 |
| 4.1 Расчет трудоемкости разработки ИС..... | 41 |
| 4.2 Расчет затрат на разработку ИС | 44 |
| 4.3 Расчет эксплуатационных затрат | 49 |
| 5 Безопасность жизнедеятельности | 56 |
| 5.1 Анализ потенциально опасных и вредных факторов | 56 |
| 5.2 Расчеты..... | 60 |
| Заключение | 66 |
| Список литературы..... | 67 |
| Приложение А Листинг программы..... | 68 |

Введение

Благодаря развитию языков программирования и отличным симбиозом появляются новые способы создания облегченных, функциональных и понятных проектов для начинающих программистов. В свою очередь они вовлекают и других людей, не связанных с it индустрией людей от директоров до стажеров компаний разных направлений, но и домохозяек, детей и даже старшее поколение. Чем проще будет порог вхождение в программирование, тем более обширно станет направление в целом.

Сайт в свою очередь создается для определенных целей как компаний, так и учебных учреждений. Назначение и внешний вид зависит как от программиста, так и для кого он создается. Информационным содержание сайта можно разделить на типы: промо-сайт, корпоративный сайт, персональный сайт, интернет-магазины, поисковые системы, почтовые ресурсы, доски объявлений, форумы, хранилище видео или сервис обмена фотографиями и файлами. Сложностью создания сайта является умение не только программирования, но и иметь навыки дизайна.

Так как постепенно устаревают стандартные инструменты для создания сайтов через html, css, javascript им на смену приходят новые. Одними из примеров новых инструментов которые завоевывают нынешний it мир в веб разработке являются Spring, Docer AngularJS, Liquibase и другие. Они используются в основном в Web разработках, как крупных компаний таких как Microsoft, Facebook, Google, так и более маленьких Advoco, Harvest Group и другие. Яркими представителями в Казахстане являются Greetgo, Eram и Beeline.

В моем проекте эти инструменты позволили мне создать простой, красивый, функциональный и самое главное он понятный для большого круга программистов. В нем отражена только часть возможностей инструментов, что я использовал это позволит продолжать его усовершенствовать или изменять всем, кому будет интересна моя работа.

1. Аналитическая часть

1.1 Описание предметной области

Автоматизация учебного процесса очень важная задача не только государства, но и самих вузов и их студентов. Обучаясь в вузе, студент видит все плюсы и минусы образования и на основе знаний, полученных в нем, может совершенствовать себя и все вокруг. Одной из важных частей которую нельзя оставлять без внимания является внешний и внутренний сайт университета.

Во внешнем сайте должна быть представлена информация о вузе как для родителей и абитуриентов, так и для тех, кто уже обучается в нем. Функциональная часть таких сайтов должна сводится к чему-то более простому, что бы зайдя на сайт не искать нужную для себя информацию большое количество времени, а быстро набрав в поиске или перейдя по ссылкам найти необходимое. Но при этом визуальная составляющая таких сайтов является очень высоким показателем вуза.

Для внутреннего сайта больше важна функциональная часть, которая будет понятно для студентов и преподавателей. Большая часть функций должна быть автоматизирована, что позволило бы экономить время преподавателей. Например, отправка уведомлений об успеваемости, встроенный календарь событий, запись на получение справок и многое другое.

Проект разработан в среде разработки IntelliJ Idea. Компания JetBrains, создавшая IntelliJ Idea, позволяет студентам использовать полную версию своего продукта совершенно бесплатно, на период обучения в вузе. IntelliJ Idea поддерживает множество языков программирования, что позволяет создавать очень гибкие проекты используя вместе разные языки. Как основной язык программирования использовался Java. Так же в него было добавлено множество других компонентов, таких как Docer, Spring, Bootstrap, Postgres и другие.

Java имеет простой и понятный синтаксис, что позволяет быстро обучиться. Так же он безопасный, гибкий и может использоваться в любой среде разработки. Так же имеет автоматическую чистку памяти, многопоточность и стабильность.

Docer в проекте служит как контейнер. Он позволяет упаковать приложение со всеми его зависимостями в стандартизированный модуль для разработки, что позволяет не создавать нагрузку как это делают виртуальные машины.

Postgres используется как хранение баз данных. Он вмещает в себя базы данных не ограниченного размера, так же он имеет очень надежные механизмы транзакций и репликации.

1.2 Цель разработки

Цель разработки – разработать функциональный, понятный и автоматизированный внутренний сайт для вуза.

- создать сайт под современные запросы и стандарты учебных заведений;

- обеспечить преподавателей и студентов быстрым доступом и информацией;

- автоматизировать рассылку уведомлений.

1.3 Обоснование выбора ПО

Проект создавался в среде разработки IntelliJ Idea. Компания JetBrains, создавшая IntelliJ Idea, позволяет студентам использовать полную версию своего продукта совершенно бесплатно, на период обучения в вузе. IntelliJ Idea поддерживает множество языков программирования, что позволяет создавать очень гибкие проекты используя вместе разные языки. Как основной язык программирования использовался Java. Так же в него было добавлено множество других компонентов, таких как Docer, Spring, Bootstrap, Postgres и другие.

Java имеет простой и понятный синтаксис, что позволяет быстро обучиться. Так же он безопасный, гибкий и может использоваться в любой среде разработки. Так же имеет автоматическую чистку памяти, многопоточность и стабильность.

Docer в проекте служит как контейнер. Он позволяет упаковать приложение со всеми его зависимостями в стандартизированный модуль для разработки, что позволяет не создавать нагрузку как это делают виртуальные машины.

Postgres используется как хранение баз данных. Он вмещает в себя базы данных не ограниченного размера, так же он имеет очень надежные механизмы транзакций и репликации.

Основу проекта составляет фреймворк Angular. В 2009 году Angular создавался как side project, как помощник веб дизайнерам в создании веб-приложений. Имя “Angular” пошло от угловых скобок или “<>”, которые обрамляли все теги. Например, < div >, < span > и другие. Поскольку домен angular.com был занят – собственно, как и сейчас – создатели фреймворка переименовали Angular в GetAngular и выпустили небольшой сайт, на котором можно было узнать о всех фичах фреймворка.

Ngular - фреймворк JavaScript, который помогает разработчикам создавать приложения. Библиотека предоставляет множество фиच, которые делают простые реализации сложных задач современных приложений, таких как привязка данных, маршрутизация и анимация.[1]

Angular также представляет ряд конвенций о подходах к разработке приложений. Это может быть очень полезно для больших команд, которые должны работать вместе на одном проекте. Angular - это одна из немногих библиотек JavaScript, которые обеспечивают обширный style guide с большим количеством наглядных примеров того, как вы можете писать свой код, используя этот фреймворк.

Angular работает не только с формами. Разработчики создали множество игр при помощи Angular и такие сумасшедшие вещи, как приложения с дополненной реальностью. Однако, большинство туториалов и документации по Angular все равно содержат информацию о создании некоторых form-based приложений. Например, вы встраиваете документацию Angular в приложение, где вы создали героев и их список через форму.

Angular хорош в form-based приложениях, он подходит для больших и сложных приложений. Angular - не самый простой и не самый маленький фреймворк JavaScript. Следовательно, если вы создаёте нечто небольшое, вам лучше подобрать для работы фреймворк попроще, например, jQuery. Angular хорошо подойдёт для разработчиков приложений в средних и больших командах. Если вы разрабатываете приложение самостоятельно, может показаться, что шаблонного кода и конвенций разработки в Angular намного больше, чем вам нужно.[1]

Angular также хорошо подходит для приложений, которые должны работать в нескольких средах разработки. Если приложение должно работать на веб, а также на Windows или Mac, вы можете придерживаться одного из многочисленных туториалов для запуска Angular-приложений с популярным Electron project.

Если же есть приложение, которое нужно запустить на веб, iOS или Android, можете использовать NativeScript для рендеринга вашего приложения в мобильной среде. В некоторых случаях вы даже можете распространять код через эти платформы, экономя ценное время разработки.

Angular - не самый простой в мире фреймворк, и понадобится время, чтобы понять те концепции, на которых он построен. Но когда вы возьметесь за Angular, то сможете делать действительно крутые приложения, обходясь небольшим количеством кода. Например, вы хотите добавить привязку данных к предыдущему примеру. Для этого используется очень простой синтаксис.

Что выделяет Angular среди конкурентов:

- Упрощенная двусторонняя привязка данных. AngularJS позволяет привязывать данные к HTML с помощью выражений, а директивы AngularJS позволяют разработчикам расширять функционал HTML и создавать новые конструкции. Манипуляции с DOM и код привязки данных обернуты в простые элементы, которые можно быстро и просто вставить в HTML шаблоны.

- Angular спроектирован универсальным фреймворком, поэтому с его помощью можно создать веб-приложение почти любого типа. Если вы создаете одностраничное веб-приложение, лучше способа не найти.

- Angular входит в пакет ПО MEAN, который также включает MongoDB, Express.js и Node.js. Поэтому он позволяет управлять front end и back end проекта с помощью JS. В качестве альтернативы можно использовать для back end Ruby on Rails. С AngularJS также хорошо стыкуется ASP.NET и C#.

- Angular построен по технике functionality-first, поэтому фреймворк больше всего подходит для разработки сверху вниз. Модульная концепция Angular позволяет упростить разделение работы на разные команды в крупных проектах. В приоритете минимальное количество кода, поэтому приложения Angular, как правило, компактные и легкие в редактировании.

Плюсы Angular:

- наличие cli системы генерирует минимальный шаблон приложения, позволяет легко запускать код как в режиме разработки или в production, так и запускать тесты;

- компонентный подход позволяет создавать новые теги и редактировать уже существующие, можно вынести верстку в отдельный файл, настраивать поведение компонента, указать параметры компоненту на вход;

- typescript обладает автодополнением, так же можно посмотреть документацию метода, не покидая контекста кода, код стал самодокументированным;

- rxJS позволяет обрабатывать данные так, что написанный будет работать, как и задумывалось;

- dependency Injection позволяет полностью контролировать фреймворк, вклиниваясь в процессы работы и заменяя части кода на свои.

Минусы Angular:

- посредственная документация;

- большой объем результирующего кода;

- сложный порог входа.

2012 году разработчики C# из компании Microsoft создали язык TypeScript — надмножество JavaScript. Он предназначен для разработки больших приложений, от 100 тысяч строк. Одной из отличительных черт TypeScript от JavaScript это доработанные недостатки второго к ним относятся:

- отсутствие модульности — могут возникать проблемы из-за конфликта больших файлов;

- нелогичное поведение;

- динамическая типизация.

Плюсами Typescript являются:

- статическая типизация;

- компилируется в хороший JavaScript;

- наличие инструментов для разработчика.

- нативный код хорошо читается, в нём легко разобраться;

- TypeScript поддерживается очень многими инструментами для разработки: Visual Studio, PHP Storm и другие IDE;

- поддерживается ECMAScript 6.0;

- компилятор найдёт и выдаст ошибку несоответствия типов до начала компилирования.

Параметры функций подразделяются на необязательные и по умолчанию.

Необязательный параметр

- `function имя_функции (имя_переменной?:тип): тип_возвращаемого_значения;`

Параметр по-умолчанию

-

`function имя_функции (имя_переменной?:тип = "значение"):тип_возвращаемого_значения;`

Однотипные параметры

-

`function имя_функции (...имя_переменной?:тип):тип_возвращаемого_значения;`

По аналогии с C# знак вопроса после переменной означает, что её значение можно не передавать.

Недостатки TypeScript:

- `definitelyTyped` — в некоторых случаях отсутствуют популярные библиотеки;

- `.ts .d.ts .map` — большое количество дополнительных файлов после компилирования ts-файла;

- неявная статическая типизация. Если объявим переменную типа `any`, то никакой пользы от статической типизации не получим;

Преимущества TypeScript:

- строгая типизация. Огромный плюс — IntelliSense, который на этапе компиляции указывает на ошибки, и их можно исправить до этапа выполнения.

- прекрасно поддерживаются все основные принципы ООП;

- надмножество JavaScript'а. Достаточно скопировать код из JavaScript и вставить в TS, чтобы он заработал;

- разработка сложных решений. Благодаря поддержке модульности крупные команды разработчиков могут создавать большие приложения.

Java native library (JNL) представляет собой JAR-архив, содержащий в себе JNI-код и объекты, которые операционная система может загрузить в качестве разделяемых библиотек. Это позволяет вызывать из Java-приложения функции, реализованные платформо-зависимыми методами. Способы создания JNL — это тема отдельной большой статьи, поэтому считаем, что у вас уже есть JNL и вы хотите ею воспользоваться в своем приложении. Об особенностях использования JNL в приложениях, работающих под управлением сервера приложений, и будет эта статья.[2]

Использование JNL приводит к следующим негативным последствиям:

- код в JNL выполняется вне Java-машины, что может привести к проблемам с безопасностью;

- сама JNL, скорее всего, имеет ограниченное количество поддерживаемых платформ (комбинации «операционная система»-«процессор»). Соответственно, попытка запуска приложения, использующего JNL на платформе, которая JNL не поддерживается, закончится неудачно.

Однако в некоторых случаях без JNL обойтись невозможно. В качестве примеров можно назвать:

- необходимость напрямую использовать API операционной системы из Java-приложения (например, для реализации работы с последовательными портами ввода-вывода);

- необходимость реализации тяжелых алгоритмов (таких, как перекодирование медиа).

Использование JNL в standalone-приложениях практически ничем не отличается от использования обычных Java-библиотек. Т.е. необходимым условием является расположение JNL в месте, известном загрузчику классов (обычно достаточно разместить библиотеку где-нибудь в classpath). Однако иногда бывает необходимо разместить JNL в каталоге, непосредственно доступном и системному загрузчику ОС.

А вот если приложение предназначено для работы на каком-либо сервере приложений, то просто так засунуть JNL WAR/EAR файл и задеплоить его вместе с самим приложением на сервер не получится. Причина очевидна: возможная дыра в безопасности. Приложение с JNL получает доступ к операционной системе с привилегиями сервера приложений (это как минимум). Сервер приложений имеет свою собственную систему безопасности, и обходить ее пользовательскому приложению будет как-то не по пацански.

Таким образом, если вы засунете JNL непосредственно в EAR или WAR, то приложение, конечно же, задеплоится. Но вот при попытке вызова JNI-кода вы получите исключение (скорее всего это будет `java.lang.UnsatisfiedLinkError` с диагностикой `Native Library already loaded in another classloader`).

Так что же, использование JNL на сервере приложений абсолютно невозможно? На самом деле, это не так. Просто необходимо объяснить самому серверу приложений, что JNL используется на законных основаниях (и, наверное, ее использование для самого сервера приложений будет более-менее безопасно). Ну и далее начинаются тонкости, зависящие от конкретного сервера приложений.

В любом случае, при компиляции Java-кода необходимо указать пути к используемой JNL (чтобы иметь возможность ссылаться на содержащиеся в ней Java-классы), но вот упаковывать ее внутрь EAR/WAR или JAR файла, предназначенного для деплоя, не надо. Дальнейшие действия зависят от целевого сервера приложений.

Так же в проекте использовался фреймворк Bootstrap. В нем написаны готовые стили и скрипты, для применения которых вам достаточно всего лишь прописать необходимые стилевые классы и атрибуты html-элементам.

По сути, если говорить простым языком, это файл или несколько файлов с готовым написанным кодом, которые подключаются в сайту в секции head, после чего становится возможным использование возможностей этого фреймворка.

Фреймворки создают для того, чтобы другим веб-разработчикам было легче верстать сайты. Я уже говорил вначале о том, что сегодня практически любой разработчик после создания с нуля парочки сайтов задумывается, как ему ускорить процесс разработки.

Дело в том, что если мы с вами будем делать разработку сайта с нуля, то придется позаботиться об очень многих вещах. Все css-стили, все веб-сценарии придется писать с нуля, а ведь это могут быть сотни и тысячи строчек кода. Причем вы можете совершить массу ошибок в верстке. Например, попросту ваш шаблон будет по-разному выглядеть в основных браузерах или он будет не адаптивен.

В общем-то, как раз ради адаптивной верстки и стоит использовать Bootstrap, потому что если мы говорим о фиксированных макетах, то их легко сделать даже с нуля. Просто создаем блоки, задаем им фиксированную ширину и работаем по макету.

Но в случае с адаптивной версткой все в разы сложнее. Вам нужно будет сделать так, чтобы на любых разрешениях экранов ваш сайт отображался хорошо. Для этого вам придется использовать медиа-запросы. Для крупных шаблонов таких вот запросов может понадобиться очень много, кроме того, вы же еще должны научиться их писать.[2]

В общем, при разработке с нуля адаптивного шаблона вам придется потрудиться как следует, при этом ваша квалификация в верстке должна быть достаточно высокой.

Недостатки Bootstrap

По сути, их всего два. Первый – кода обычно в библиотеке написано больше, чем если бы вы написали при разработке с нуля. Потому что когда вы делаете самостоятельно, вы реализуете только необходимый функционал и все. В Bootstrap же есть все на все случаи жизни. Даже то, что вам может не пригодиться. Но опять же, эта проблема очень легко решается тем, что вы можете сами выбирать, какие компоненты фреймворка загрузить в css-файл. Например, вы вообще можете скачать только сетку, а все остальное делать самостоятельно.

Второй недостаток – шаблонный дизайн. Да, действительно, я часто захожу на разные сайты и вижу там одинаковые кнопки. И я знаю, что они сделаны в Bootstrap, потому что уж слишком это очевидно. Но и эта проблема легко решается, потому что она будет существовать только в том случае, если вы будете использовать только готовые компоненты фреймворка и ничего никогда не кастомизировать под себя.

А если вы, например, подключите только сетку Bootstrap, то сможете воссоздать любой дизайн, при этом пользуясь очень удобной гибкой сеткой фреймворка.[3]

Nginx — третий по популярности веб-сервер в мире и один из самых успешных российских IT-стартапов. Им пользуется более 447 млн сайтов по всей планете (это почти 40% всех сайтов в мире), включая Яндекс, Facebook, Instagram, Google, Netflix, Adobe, Cloudflare, WordPress.com и многие другие.

Простыми словами: это программа для развертывания собственного веб-сервера. Может использоваться как независимый полноценный сервер, так и прокси с переадресацией веб-запросов на другие сервера.[4]

Этот сервис использует открытый исходный код и доступен любому для редактирования и подстраивания под свои нужды.

В своём первоначальном виде ПО функционировало для веб-обслуживания HTTP, но сейчас оно также служит обратным прокси-сервером, балансировщиком нагрузки HTTP, а также почтовым прокси-сервером для IMAP, POP3 и SMTP.

Nginx одновременно обрабатывает тысячи подобных запросов и распределяет нагрузку на загруженных сайтах, вроде облачных хранилищ, интернет-магазинов, поисковых систем и прочего. За счёт этого он и обрёл свою популярность.

Все это делалось на основе HTML, CSS и JavaScript.

HTML- это язык разметки гипертекста. Данный язык применяется для создания веб-страниц. Он интерпретируется (обрабатывается) браузером и отображается в виде документа в удобной для человека форме.

Именно HTML позволяет нам наделять содержимое страницы определенным смыслом, а реализуется это с помощью так называемых тэгов.

Тэги – это специальные маркеры, которые определенным образом интерпретируются браузером. Суть тэгов в том, что содержимое страницы, заключенное в разные тэги, по-разному обрабатывается браузером.

Для разных версий HTML предусмотрено разное объявление типа документа. Например:

- строгий (Strict): не содержит элементов, помеченных как «устаревшие» или «не одобряемые» (deprecated);

- переходный (Transitional): содержит устаревшие теги в целях совместимости и упрощения перехода со старых версий HTML;

- с фреймами (Frameset): аналогичен переходному, но содержит также теги для создания наборов фреймов.

CSS – это язык описания внешнего вида документа, написанного с использованием языка разметки. Название произошло от англ. Cascading Style Sheets — каскадные таблицы стилей.

Проще говоря, язык CSS предназначен для того, чтобы придавать необходимый внешний вид HTML-документам.

Придание внешнего вида документам HTML – это хоть и самый популярный, однако лишь частный случай применения языка CSS, т.к. с его помощью можно придавать вид и документам других типов: XHTML, SVG и XUL. Про них мы отдельно говорить не будем, т.к. это выходит за рамки рассматриваемого вопроса.

Итак, целью создания CSS было отделение описания логической структуры веб-страницы от ее внешнего вида. Как вы уже знаете, для описания структуры используется HTML, для описания же того, как эта логическая структура будет выглядеть, отвечает как раз CSS.

Раздельное описание логической структуры и представления документа позволяет более гибко управлять внешним видом документа и минимизировать объем повторяющегося кода, который бы неизбежно возникал при использовании HTML для описания внешнего вида документа.

С помощью CSS веб-разработчик может задавать для страницы и отдельных ее элементов различные гарнитуры и размеры шрифта, цвета элементов, отступы элементов друг от друга, расположение отдельных блоков на странице и т.д.

Разумеется, для того, чтобы использовать CSS для придания внешнего вида HTML-документу, нужно этот документ как-то связать со стилями, т.е. «сообщить» HTML-документу, что он будет оформлен с помощью CSS.

Для этого существуют различные способы подключения CSS к документу, которые дают браузеру знать, что к странице в целом, либо к каким-то отдельным ее элементам должно быть применено стилевое оформление.

Таблицы стилей могут располагаться как непосредственно внутри того, документа, к которым они будут применяться, так и находиться в отдельном файле, имеющем расширение .css.[5]

Важно понимать, что CSS-файл – это обычный текстовый файл. В нем пишутся специальные инструкции, описывающие внешний вид элемента и его позиционирование на странице а также комментарии (произвольные пояснения относительно написанных инструкций).

Стили CSS могут быть подключены к странице (либо внедрены в нее) четырьмя следующими способами:

- если таблица стилей находится в отдельном файле, то она подключается к документу с помощью специального тэга link, который должен располагаться в этом документе внутри тэга head;

- второй способ подключения применяется также в том случае, если стили находятся в отдельном файле. При этом используется директива @import, которая должна находиться в этом документе внутри тэгов style (которые, в свою очередь, должны находиться внутри тэгов head);

- третий способ используется в случае, когда стили CSS расположены внутри того документа, к которому они должны применяться. В этом случае стили должны находиться в этом документе внутри тэгов style (которые, в свою очередь, должны находиться внутри тэгов head);

- четвертый способ также применяется в случаях, когда таблица стилей описана в самом документе. При этом стилевое оформление задается для какого-то конкретного элемента веб-страницы (тэга) посредством использования его атрибута style.[6]

1.4 Сравнительный анализ аналогов

Для оценки программного изделия были выбраны критерии качества, которые оценивают реализацию требуемых функций:

- информативность;
- навигация;
- интерфейс;
- технические возможности;
- обновляемость.

Информативность – информативность сайта в таких случаях зависит от университета. Можно рекламировать продукты выпускников, курсов и кружков охватив при этом больше аудитории.

- навигация - для сайта учебных заведений информативность заключается в возможности быстро и легко получить информацию о текущих оценках и актуальном материале;

- интерфейс - в основном, все крупные учебные порталы отличить не сложно, каждый добавляет свои функции, защиту и кроссплатформенность. Но из-за недостаточной конкуренции в этой нише, часто происходят проблемы с серверами;

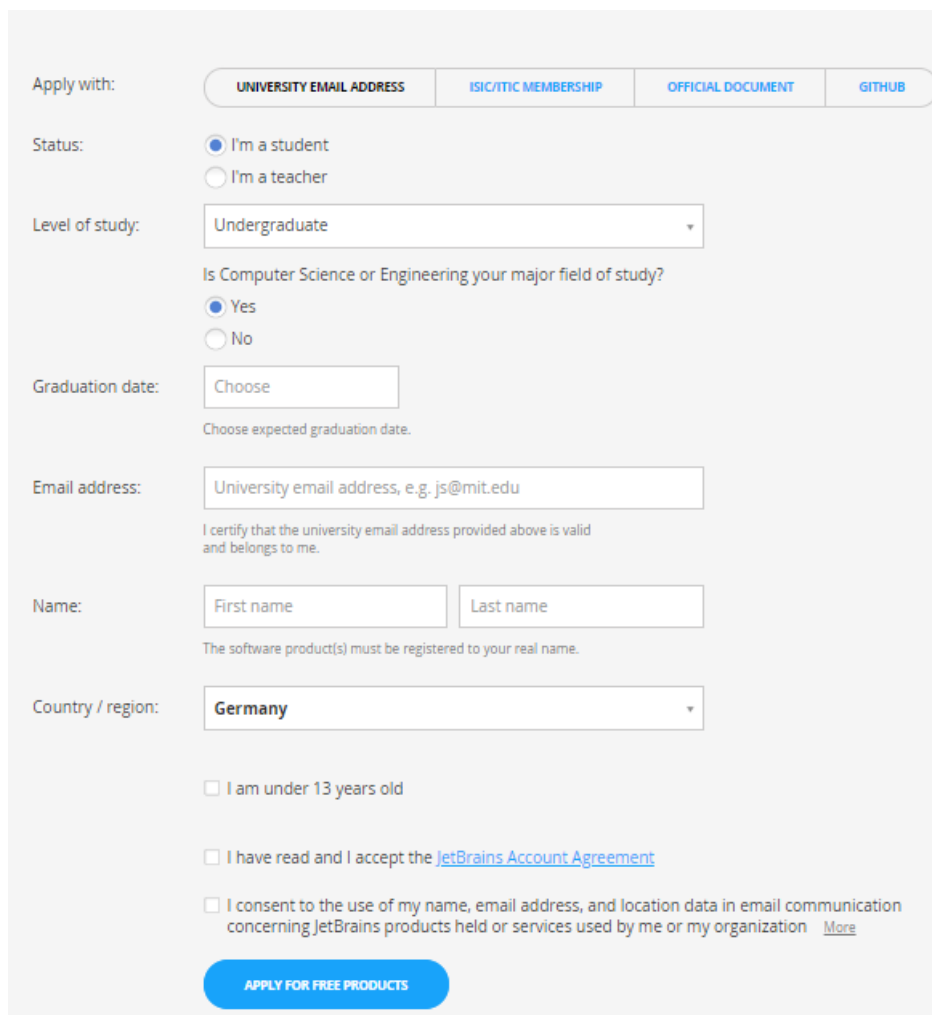
- технические возможности – в зависимости от пользователя на сайт ему будет дана одна из двух ролей. Это либо студент, либо преподаватель. От этого и будет зависеть их функции на сайте, студент может просматривать и скачивать материал, смотреть свои оценки в журнале, в то же время функции у преподавателя шире, загружать материал и редактировать дневники студентов;

- обновляемость – позволит добавлять функции, изменять дизайн тематически или сезонно, что скрасит пребывание на сайте в учебные дни.

2 Отчет по преддипломной практике на кафедре

2.1 Создание проекта

Для скачивания полной версии IntelliJ IDEA можно оформить студенческую лицензию, так же можно ее продлить. Для получения лицензии необходимо на сайте пройти регистрацию и прикрепить к ней студенческий.



The image shows a registration form for JetBrains products. It includes the following fields and options:

- Apply with:** Four tabs: UNIVERSITY EMAIL ADDRESS (selected), ISIC/ITIC MEMBERSHIP, OFFICIAL DOCUMENT, and GITHUB.
- Status:** Radio buttons for "I'm a student" (selected) and "I'm a teacher".
- Level of study:** A dropdown menu currently showing "Undergraduate".
- Is Computer Science or Engineering your major field of study?** Radio buttons for "Yes" (selected) and "No".
- Graduation date:** A text input field with "Choose" inside, and a note "Choose expected graduation date." below it.
- Email address:** A text input field with the placeholder "University email address, e.g. js@mit.edu". Below it is a certification statement: "I certify that the university email address provided above is valid and belongs to me."
- Name:** Two text input fields for "First name" and "Last name". Below them is a note: "The software product(s) must be registered to your real name."
- Country / region:** A dropdown menu currently showing "Germany".
- Checkboxes:**
 - I am under 13 years old
 - I have read and I accept the [JetBrains Account Agreement](#)
 - I consent to the use of my name, email address, and location data in email communication concerning JetBrains products held or services used by me or my organization [More](#)
- Button:** A blue button labeled "APPLY FOR FREE PRODUCTS".

Рисунок 2.1 - Регистрация лицензии

Через 2-3 дня на почту придет сообщение с подтверждением лицензии и ссылкой на ее скачивание.

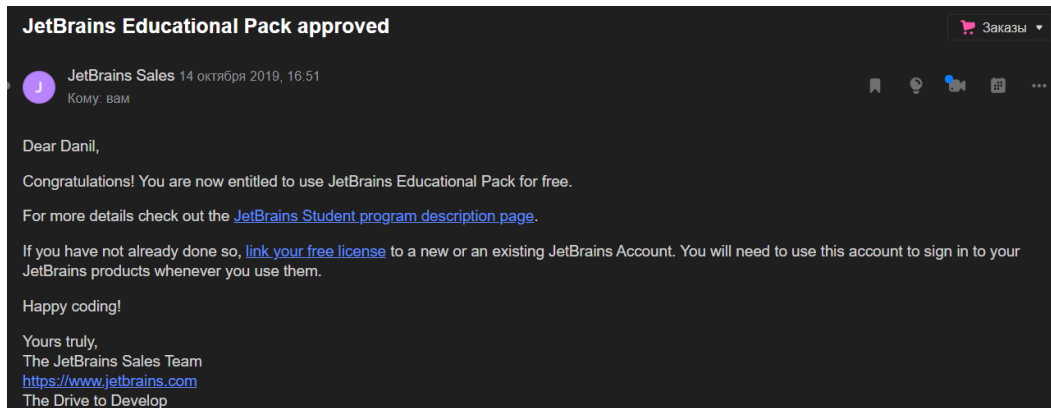


Рисунок 2.2 - Подтверждение регистрации и лицензии

Установка программы IntelliJ IDEA. После скачивания дистрибутива с официального сайта выйдет окно выбора папки куда установится программа.

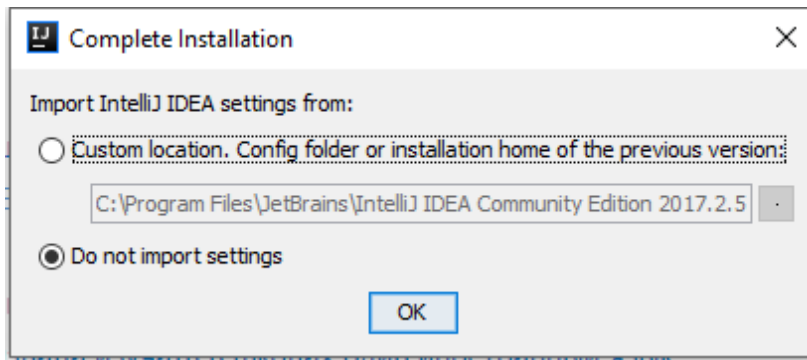


Рисунок 2.3- Место установки программы

Далее идет выбор темы. По умолчанию их 2.

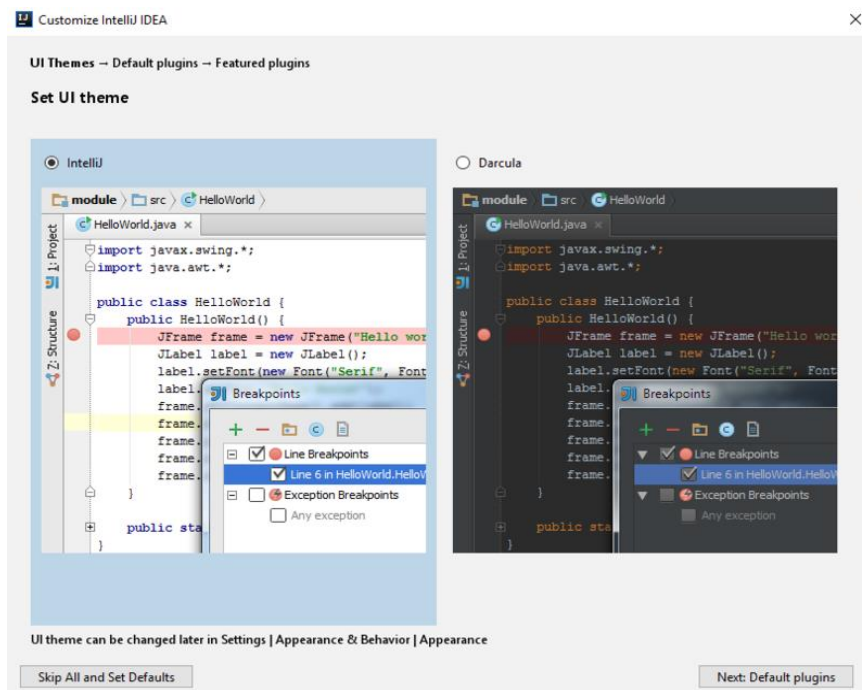


Рисунок 2.4 - Тема программы

После этого будет предложено меню вспомогательных инструментов. Их можно отключить и позже подключить те что необходимы.

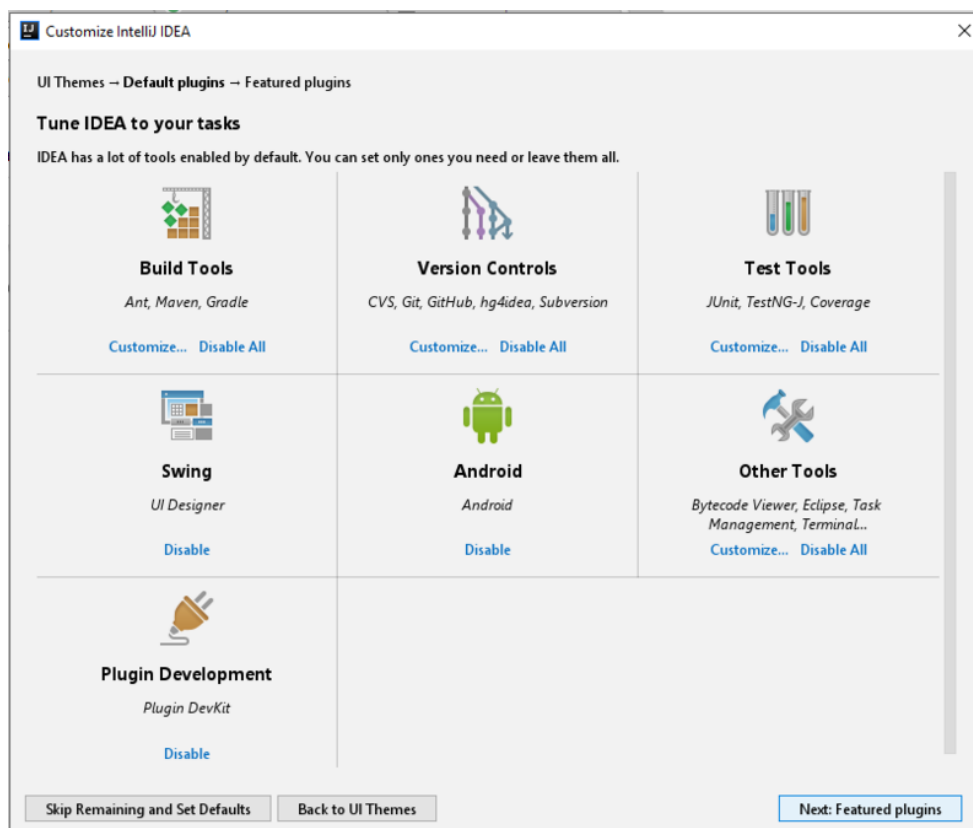


Рисунок 2.5 - Дополнительные инструменты

И в конце установи программа предлагает создать, импортировать или открыть папку с проектом.

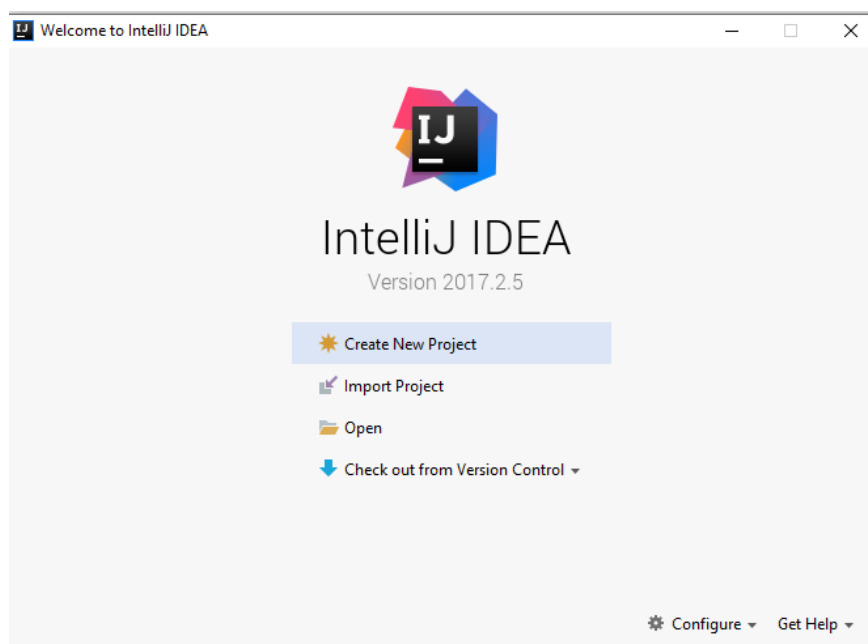


Рисунок 2.6 - Управление проектом

После установки IntelliJ IDEA, нужно выбрать язык программирования, в проекте выбрана Java.

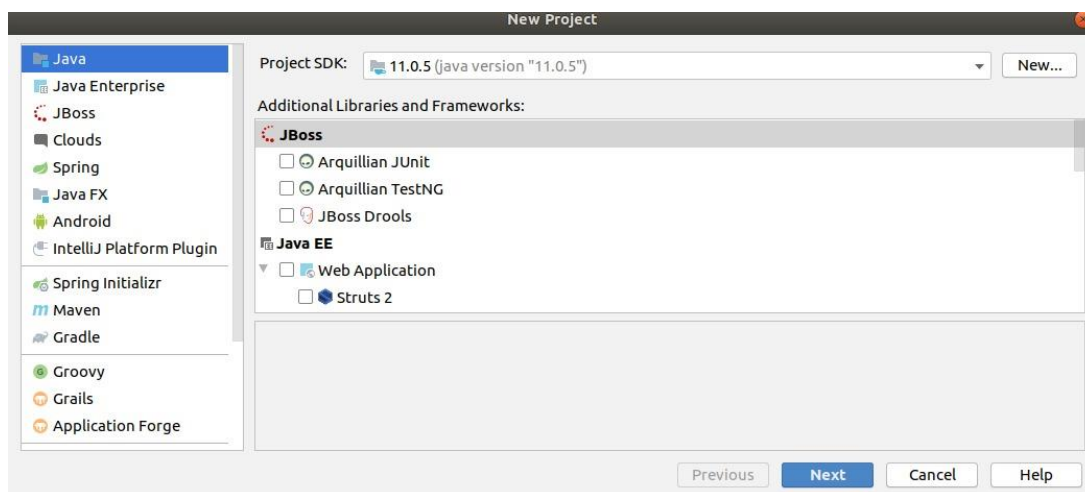


Рисунок 2.7 - Выбор языка программирования

Далее идет выбор шаблона.

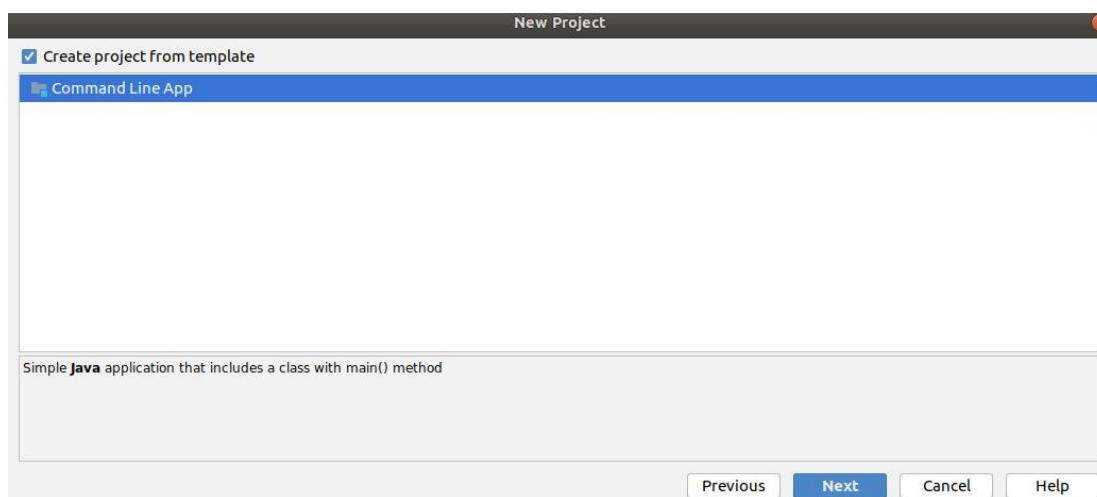


Рисунок 2.9 - Шаблон

Следующим шагом идет название проекта, его место сохранения и название внутренней папки хранения.

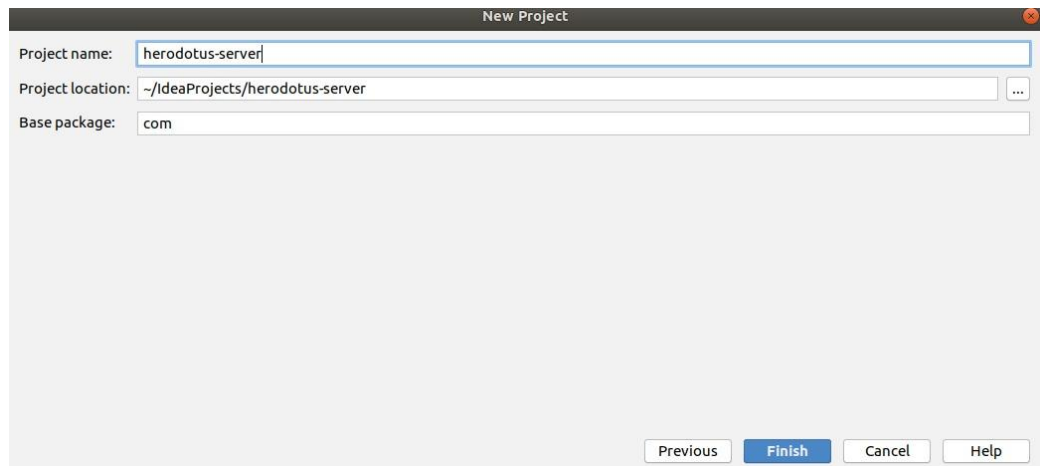


Рисунок 2.8 - Название и место хранения проекта

После всех шагов откроется проект, для которого необходимо настроить подключение к серверу с базами данных.

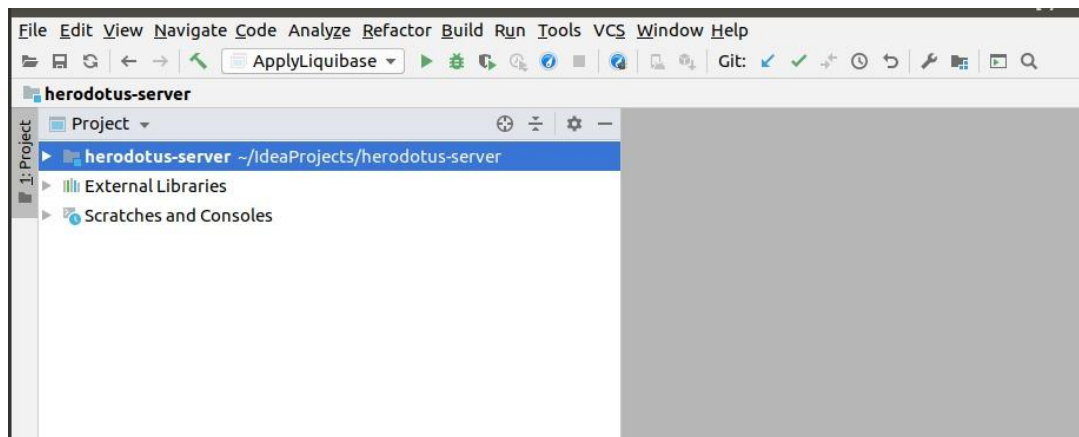


Рисунок 2.10 - Проект

2.2 UML диаграмма

В этом подразделе будут представлены диаграммы деятельности UML.

На рисунке 2.9 представлен процесс входа на сайт. Пользователь заходит в систему, далее отправляется запрос на сервер и идет подтверждение.

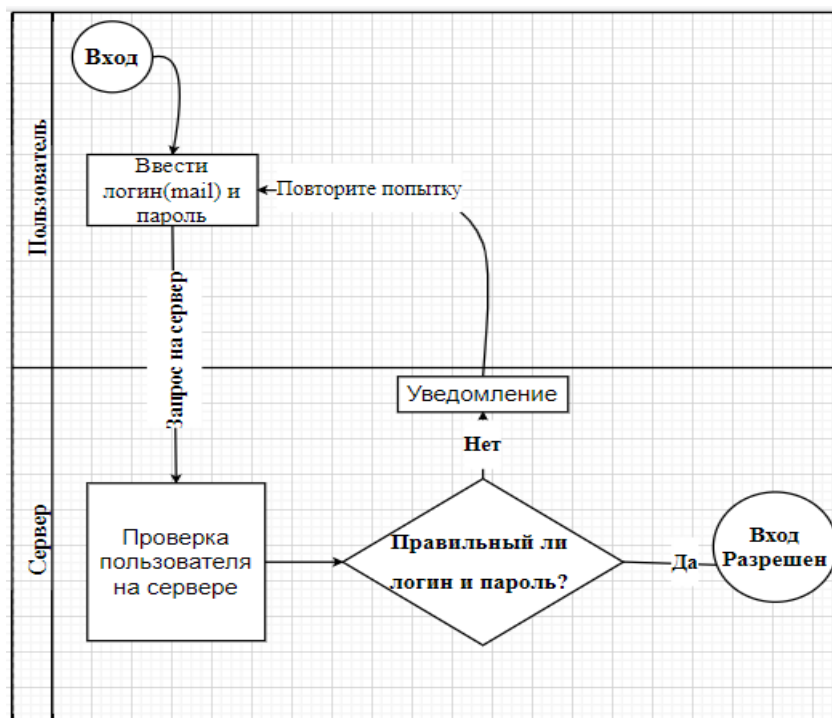


Рисунок 2.11 - Вход на сайт

На рисунке 2.10 диаграмма уведомлений через смс и почту, загрузка файлов.

```

classDiagram
    class GradeRegisterImpl {
        logger
        gradeDao
        gradeList(Integer, Integer, Integer)
        saveGrade(Grade)
        loadDisciplines(Integer, String, String)
        loadGroups(Integer, Integer)
        loadCurrentDate()
        loadValidYears()
        saveNotification(NotificationRecord)
        loadNotificationList()
        loadGradeChartData(GradeChartFilter)
    }
    class NotificationRegisterImpl {
        logger
        notificationDao
        gradeDao
        smsSender
        emailSender
        sendNotification()
    }
    class TestRegisterImpl {
        testConfig
        hello(String)
        test()
        testConfig()
        acsLogin()
        getClientFioById(String)
    }
    class SmsSender {
        ACCOUNT_SID
        AUTH_TOKEN
        main(String[])
        sendSms(String, String)
    }
    class AuthRegisterImpl {
        authDao
        checkCredential(String, String)
        getAuthInfo(String)
    }
    class EmailSender {
        main(String[])
        sendEmail(String, String)
    }
    class BeanConfigRegister
  
```

Рисунок 2.12 - Уведомление и регистрация

На рисунке 2.3 диаграмма представлена регистрация пользователя и заполнение шаблона уведомлений на 3 языках.

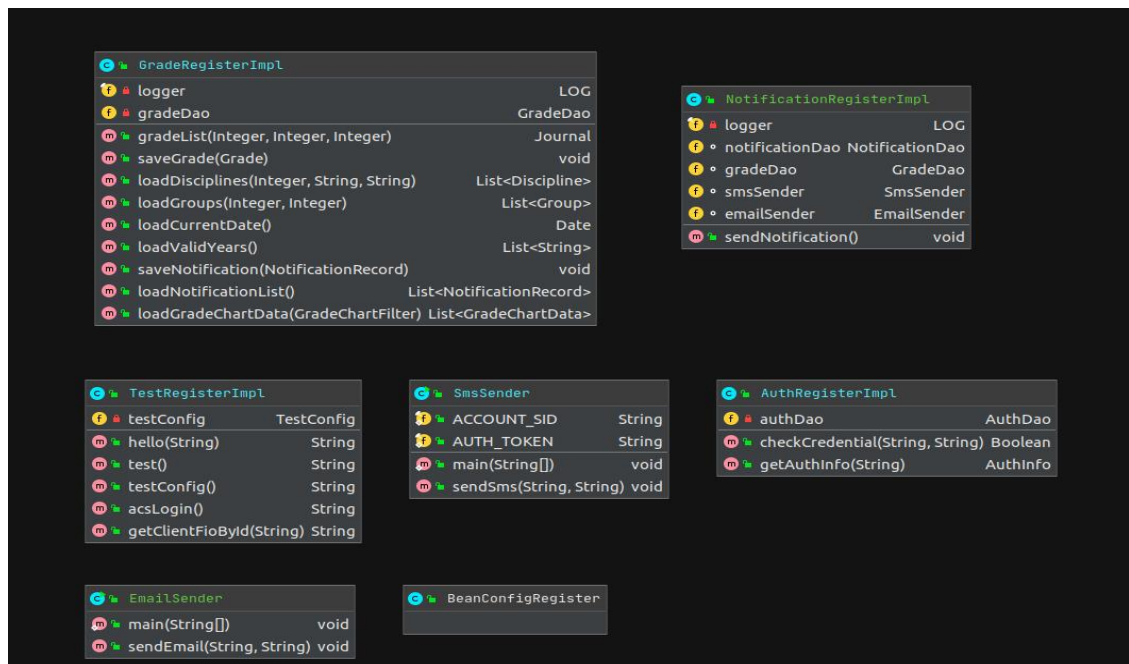


Рисунок 2.13 - Регистрация и форма уведомлений

2.3 Программная часть

Часто поиск в Интернете не всегда эффективен, и поиск нужной информации занимает несколько часов. И я понял, что вам нужно создать сайт, где люди могут делиться этой информацией с помощью статей. В конце концов, хорошо иметь сайт, на котором уже размещено много информации. При создании такого сайта я создал:

- 1) лавная страница;
- 2) панель преподавателя и студента;
- 3) вход;
- 4) дневник;
- 5) страница добавления статьи.

Разработка Алгоритма.

Разработка алгоритма (а также его обоснование и, при необходимости, модификация) значительно усложняются, если разработчик не придерживается с самого начала определенной дисциплины, которая на каждом этапе разработки четко определяет подцели и прослеживает отношения между ними. Такая дисциплина, которая широко используется в последние годы, является методом поэтапного развития. Суть этого метода заключается в том, что алгоритм разрабатывается «поэтапно» (обычно «сверху вниз»), начиная с его спецификации, полученной в результате анализа проблемы. На каждом этапе принимается небольшое количество решений, что приводит к постепенному уточнению (уточнению) контрольной и информационной структуры алгоритма. Таким образом, получается последовательность более подробных спецификаций алгоритма, близких к финальной версии программы.

Этот метод позволяет разбить алгоритм на части (модули), каждая из которых решает независимую (обычно небольшую) подзадачу. Это позволяет сосредоточить усилия на решении подзадачи, реализованной в виде отдельной процедуры или функции. Связь между модулями осуществляется через соответствующие вызовы (вызовы), а передача информации от одного модуля к другому осуществляется через параметры и глобальные переменные. Наряду с использованием метода пошаговой разработки необходимо также учитывать следующие факторы, которые могут существенно повлиять на разрабатываемый алгоритм.

Алгоритм:

- 1) функциональности и возможностях моего проекта;
- 2) продумывание проекта;
- 3) выбор языка программирования, на котором будет написан проект;
- 4) материал для проекта;
- 5) создание сайта.

Создание проекта

Необходимо определить, для чего предназначен сайт, т.е. какие задачи он должен решить: дать общее представление о компании или выделить на многосторонней основе любой аспект человеческой деятельности, увеличить продажи по традиционным каналам или организовать веб-трейдинг, проводить рекламную или маркетинговую кампанию. Это один из самых важных шагов, не только создание сайта как такового, но и самый важный этап интернет-маркетинга. После определения целей сайта, необходимо четко и максимально тщательно представлять и описывать целевую аудиторию сайта, так как это влияет на способ представления информации. Определение целевой аудитории сайта так же важно, как и определение цели сайта, стадии. Не всегда целевая аудитория компании из офлайн будет напрямую проецироваться на онлайн. Зная целевую аудиторию и интернет-аудиторию, можно сделать некоторые предположения о том, кто будет основными посетителями сайта. Определение и максимально подробное описание целевой аудитории сайта позволяет разработать правильный дизайн для проекта, а также принять правильное направление для написания текстов. Очень важно общаться с аудиторией на понятном им языке. И это требование в Интернете в разы важнее, чем в реальной жизни. На заключительном этапе этого этапа они приблизительно определяют, по каким поисковым запросам сайт должен появиться в результатах поискового запроса, и видят интернет-проекты конкурентов будущего веб-проекта.

Определение поисковых запросов или, другими словами, состав семантического ядра довольно сложное и в идеале должно быть сделано профессионалами. Для этого вам просто нужно записать все те слова и фразы, которые вы лично используете для поиска конкурентов и интервью с менеджерами по продажам, какие вопросы чаще всего возникают у клиентов вашей компании. После этого будет очень полезно сравнить ваши догадки с шаблоном запросов, которые нам задают поисковые системы.

Дополнительная информация о состоянии Интернета на вашем рынке предоставит обзор других сайтов. Необходимо описать как можно более подробно:

- цель сайта и его целевая аудитория;
- структура сайта и количество страниц в каждом разделе.

Разработка на стороне клиента

В настоящее время как насчет того, чтобы сосредоточиться на коде, связанном с серверной частью и клиентской частью. Для каждой ситуации код контрастирует в целом:

- У них разные цели и причины;
- Обычно они не используют один язык программирования (особый случай может быть использован как в клиентской, так и в серверной частях);
- Они продолжают работать в различных рабочих условиях.

Код, который продолжает выполняться в программе, известный как код на стороне клиента, в основном связан с улучшением внешнего вида и поведения созданного сайта. Это включает в себя определение и стилизацию части пользовательского интерфейса, формат, маршрут, утверждение структуры и т. Д. Серверная часть фактически занимается определением: какой контент возвращается в программу в свете просьб. Серверный код управляет такими действиями, как утверждение отправленной информации и вопросов, использование баз данных для хранения и восстановления информации и отправка правильной информации клиенту, верной форме.

Код на стороне клиента составлен с использованием HTML, CSS и JavaScript, выполняется внутри интернет-браузера и имеет ограниченный доступ или не имеет доступа к рабочей структуре (считая ограниченный доступ к структуре записи). Веб-инженеры не могут контролировать, какую программу использует каждый клиент для просмотра веб-сайта: программы предоставляют противоречивые измерения сходства с выделенными фрагментами кода на стороне клиента, а часть вызова программирования на стороне клиента - это изысканный подход к контрастам поддержки программ.

Серверный код написан на языке программирования Java. Код на стороне сервера имеет полный доступ к рабочей среде сервера, и мы выбрали, какой язык программирования (и какую адаптацию) мы могли бы использовать. С целью, что выбрал язык программирования является Java.

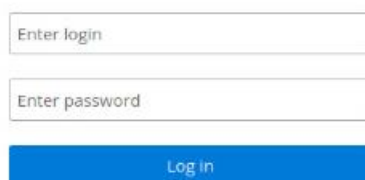
Каждый инженер чаще всего составляет свой код, используя веб-системы. Как мы уже писали ранее, мы выбрали Spring System. Веб-структуры - это множество возможностей, элементов, правил и других сборок кода, предназначенных для решения обычных проблем, ускорения продвижения и распутывания различных видов заданий, возникающих в конкретном регионе. Более того, еще раз, поскольку как клиентская, так и серверная части используют системы, зоны совершенно разные, и, следовательно, структуры дополнительно экстраординарны. Системы на стороне клиента реорганизуют формат и представление информации, в то время как структуры на стороне сервера дают большую «обычную» полезность веб-сервера, которая может

вам так или иначе понадобится для актуализации себя (например, поддержка сеанса, поддержка клиента и подтверждение простой доступ к макетам библиотеки баз данных и т. д.).

Проектная часть

Вход будет производиться уже с зарегистрированных пользователей, которые будут храниться в базе данных серверной части.

HERODOTUS



Enter login

Enter password

Log in

Рисунок 2.14 - Вход на сайт

HERODOTUS



student1

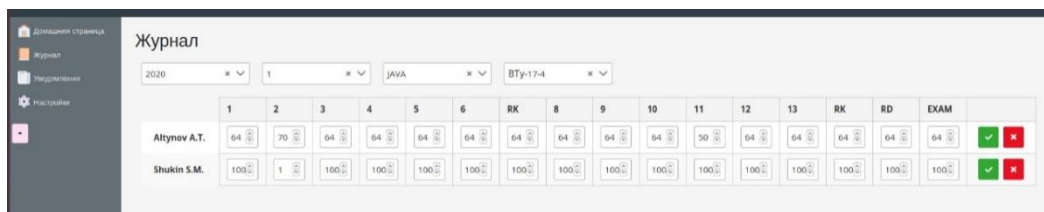
...

login or password invalid

Log in

Рисунок 2.15 - Ошибка при входе

Журнал оценок преподавателя и студента для разных устройств:



| | 1 | 2 | 3 | 4 | 5 | 6 | RK | 8 | 9 | 10 | 11 | 12 | 13 | RK | RD | EXAM | |
|--------------|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|---|
| Altynov A.T. | 64 | 70 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 50 | 64 | 64 | 64 | 64 | 64 | ✓ |
| Shukin S.M. | 100 | 1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | ✓ |

Рисунок 2.16 - Журнал преподавателя через браузер

HERODOTUS test Lecturer test Sign out

Домашняя страница Журнал Уведомления Настройки

Журнал

2020 1 JAVA Vtu-17-4

| | 1 | 2 | 3 | 4 | 5 | 6 | RK | 8 | 9 | 10 | 11 | 12 | 13 | RK | RD | EXAM |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Altynov A.T. | 50.0 | 50.0 | 50.0 | 50.0 | 50.0 | 50.0 | 52.0 | 50.0 | 50.0 | 50.0 | 50.0 | 50.0 | 50.0 | 53.0 | 50.5 | 54.0 |
| Shukin S.M. | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

Рисунок 2.17 - Журнал группы через браузер

HERODOTUS

Журнал

2020 1

> TP

> JAVA

> C#

> KS

> AIOKS

> ORKSP

© 2020 All rights reserved

Рисунок 2.18 - Журнал студента через мобильный телефон

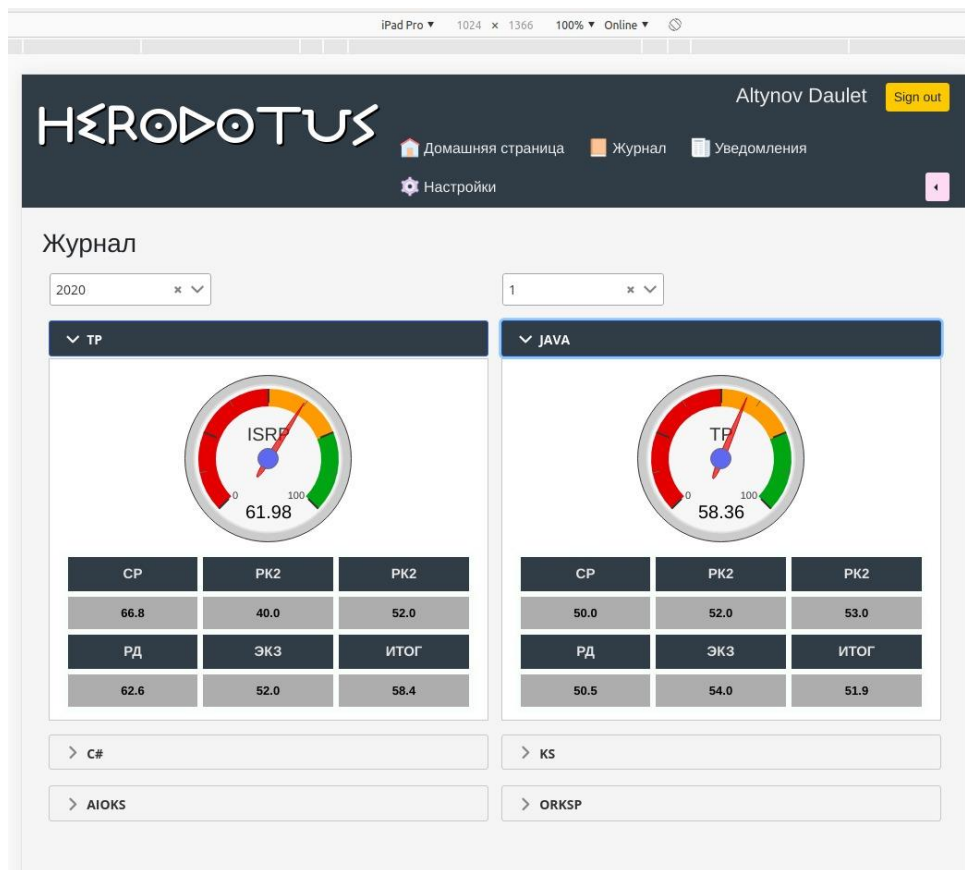


Рисунок 2.19 - Журнал оценок за семестр

Помимо основных функций так же есть смена пароля на разных устройствах:

The screenshot shows the password change form on a smartphone browser. The form includes three input fields for 'Old password', 'New password', and 'Repeat new password', followed by a blue 'Change password' button with a checkmark icon.

Рисунок 2.20 - Смена пароля через браузер на смартфон

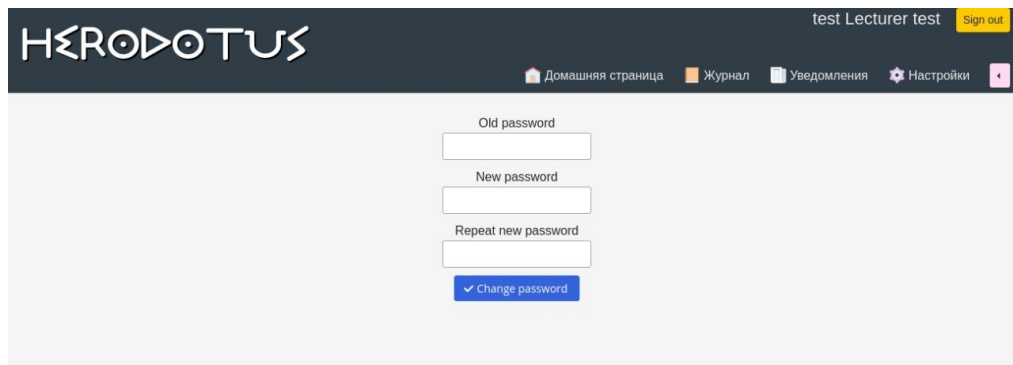


Рисунок 2.21 - Смена пароля через браузер на ноутбуке

Планировщик, который отправляет каждую минуту неотправленные уведомления и меняет статус на отправлено

```

@Component
public class NotificationScheduler implements HasScheduled {
    @Autowired
    NotificationRegister notificationRegister;
    @Scheduled("repeat every 1 minute")
    public void sendNotification() {
        notificationRegister.sendNotification();
    }
}

```

Рисунок 2.21 - Планировщик

Таблица на рисунке 2.22 библиотека liquibase используется для регистрации изменений в базе данных.

| databasechangelog | |
|-------------------|--------------|
| id | varchar(255) |
| author | varchar(255) |
| filename | varchar(255) |
| dateexecuted | timestamp |
| orderexecuted | integer |
| exectype | varchar(10) |
| md5sum | varchar(35) |
| description | varchar(255) |
| comments | varchar(255) |
| tag | varchar(255) |
| liquibase | varchar(20) |
| contexts | varchar(255) |
| labels | varchar(255) |
| deployment_id | varchar(10) |

Рисунок 2.22 - Регистрация изменений в базе

Таблица на рисунке 2.23 библиотека liquibase используется для регулирования нескольких изменений в одно время.

| ▼ databasechangeloglock | |
|----------------------------|--------------|
| id | integer |
| locked | boolean |
| lockgranted | timestamp |
| lockedby | varchar(255) |
| databasechangeloglock_pkey | (id) |
| databasechangeloglock_pkey | (id) UNIQUE |

Рисунок 2.23 - Регулирование нескольких изменений

Таблица на рисунке 2.24 показывает сущность факультета учебного заведения

| ▼ department | |
|----------------------------|----------------------------------|
| id | bigint = nextval('departmen...') |
| name | text |
| person_id | integer |
| institute_id | integer |
| actual | smallint = 1 |
| department_pk | (id) |
| department_institute_id_fk | (institute_id) → institute (id) |
| department_person_id_fk | (person_id) → person (id) |
| department_id_uindex | (id) UNIQUE |
| department_pk | (id) UNIQUE |

Рисунок 2.24 - Сущность кафедры

Таблица на рисунке 2.25 показывает сущность предметов учащихся.

| ▼ discipline | |
|-----------------------------|-----------------------------------|
| id | bigint = nextval('disciplin...') |
| name | text |
| department_id | integer |
| actual | smallint = 1 |
| discipline_pk | (id) |
| discipline_department_id_fk | (department_id) → department (id) |
| discipline_id_uindex | (id) UNIQUE |
| discipline_pk | (id) UNIQUE |

Рисунок 2.25 - Сущность дисциплины

Таблица на рисунке 2.26 показывает сущность функции отдельных ролей.

| ▼ func | |
|------------------|----------------------------------|
| id | bigint = nextval('func_id_s...') |
| name | text |
| actual | smallint = 1 |
| func_pk | (id) |
| func_id_uindex | (id) UNIQUE |
| func_name_uindex | (name) UNIQUE |
| func_pk | (id) UNIQUE |

Рисунок 2.26 - сущность функций отдельных ролей

Таблица на рисунке 2.27 показывает оценки учащегося за период времени.

```
▼ grade
  id bigint = nextval('grade_id...')
  lecturer_discipline_id integer
  student_id integer
  grade integer
  grade_type_id integer
  createdat date = now()
  modifiedat date = now()
  actual smallint = 1
  semester integer
  year integer
  grade_pk (id)
  grade_grade_type_id_fk (grade_type_id) → grade_type (id)
  grade_lecturer_and_discipline_id_fk (lecturer_discipline_id) → lecturer_and_discipline (id)
  grade_student_id_fk (student_id) → student (id)
  grade_id_uindex (id) UNIQUE
  grade_pk (id) UNIQUE
```

Рисунок 2.27 - Оценки за определенный период

Таблица на рисунке 2.28 определяет тип балла, который получил обучающийся.

```
▼ grade_type
  id bigint = nextval('grade_typ...')
  type text
  actual smallint = 1
  grade_type_pk (id)
  grade_type_id_uindex (id) UNIQUE
  grade_type_pk (id) UNIQUE
  grade_type_type_uindex (type) UNIQUE
```

Рисунок 2.28 - Тип балла

Таблица на рисунке 2.29 представляет из себя группу в которой обучаются обучающиеся.

```
▼ group
  id bigint = nextval('group_id...')
  name text
  count integer
  specialty_id integer
  actual smallint = 1
  short_education boolean = false
  group_pk (id)
  group_specialty_id_fk (specialty_id) → specialty (id)
  group_id_uindex (id) UNIQUE
  group_name_uindex (name) UNIQUE
  group_pk (id) UNIQUE
```

Рисунок 2.29 - Группа студентов

Таблица на рисунке 2.30 показывает институт как подраздел университета.

```

▼ [table icon] institute
  [key icon] id bigint = nextval('institute...')
  [table icon] name text
  [key icon] person_id integer
  [key icon] university_id integer
  [table icon] actual smallint = 1
  [key icon] institute_pk (id)
  [key icon] institute_person_id_fk (person_id) → person (id)
  [key icon] institute_university_id_fk (university_id) → university (id)
  [key icon] institute_id_uindex (id) UNIQUE
  [key icon] institute_pk (id) UNIQUE

```

Рисунок 2.30 - Институт

Таблица на рисунке 2.31 представляется показывает сущность преподавателей, имеющие характерные черты.

```

▼ [table icon] lecturer
  [key icon] id bigint = nextval('lecturer_...')
  [key icon] department_id integer
  [key icon] person_id integer
  [table icon] actual smallint = 1
  [key icon] lecturer_pk (id)
  [key icon] lecturer_department_id_fk (department_id) → department (id)
  [key icon] lecturer_person_id_fk (person_id) → person (id)
  [key icon] lecturer_id_uindex (id) UNIQUE
  [key icon] lecturer_pk (id) UNIQUE

```

Рисунок 2.31 - Сущность преподавателей

Таблица на рисунке 2.32 представляется из себя сущность связь преподаватели и дисциплина дабы избавиться от связей многие ко многим.

```

▼ [table icon] lecturer_and_discipline
  [key icon] id bigint = nextval('lecturer_...')
  [key icon] lecturer_id integer
  [key icon] discipline_id integer
  [table icon] actual smallint = 1
  [key icon] lecturer_discipline_pk (id)
  [key icon] lecturer_discipline_discipline_id_fk (discipline_id) → discipline (id)
  [key icon] lecturer_discipline_lecturer_id_fk (lecturer_id) → lecturer (id)
  [key icon] lecturer_discipline_id_uindex (id) UNIQUE
  [key icon] lecturer_discipline_pk (id) UNIQUE

```

Рисунок 2.32 - Связь преподавателя и дисциплины

Таблица на рисунке 2.33 представляется из себя сущность уведомлений, которые отправляются из системы.

| notification | |
|--------------------------------------|------------------------------------|
| id | bigint = nextval('notificat...') |
| ru | text |
| en | text |
| type_id | integer |
| role_id | integer |
| state | text |
| publishDate | date |
| publishType | integer |
| kz | text |
| notification_pk | (id) |
| notification_notification_type_id_fk | (type_id) → notification_type (id) |
| notification_publish_type_id_fk | (publishType) → publish_type (id) |
| notification_role_id_fk | (role_id) → role (id) |
| notification_id_uindex | (id) UNIQUE |
| notification_pk | (id) UNIQUE |

Рисунок 2.33 - Сущность уведомлений

Таблица на рисунке 2.34 представляется из себя сущность типов уведомлений для того чтобы понимать какой вид уведомлений нужно отправлять.

| notification_type | |
|-----------------------------|----------------------------------|
| id | bigint = nextval('notificat...') |
| name | text |
| notification_type_pk | (id) |
| notification_type_id_uindex | (id) UNIQUE |
| notification_type_pk | (id) UNIQUE |

Рисунок 2.34 - Сущность типов уведомлений

На рисунке 2.35 главная таблица где хранятся пользователи.

| person | |
|------------------|----------------------------------|
| id | bigint = nextval('person_id...') |
| surname | text |
| name | text |
| patronymic | text |
| birthdate | date |
| gender | text |
| address | text |
| email | text |
| phone | text |
| actual | smallint = 1 |
| login | text |
| password | text |
| person_pk | (id) |
| person_id_uindex | (id) UNIQUE |
| person_pk | (id) UNIQUE |

Рисунок 2.35 - Таблица пользователей

Таблица на рисунке 2.36 представляется из себя сущность связь пользователи и роли дабы избавиться от связей многие ко многим.

▼ **person_and_role**

- id bigint = nextval('person_an...')
- person_id integer
- role_id integer
- actual smallint = 1
- person_role_pk (id)
- person_role_person_id_fk (person_id) → person (id)
- person_role_role_id_fk (role_id) → role (id)
- person_role_id_uindex (id) UNIQUE
- person_role_pk (id) UNIQUE

Рисунок 2.36 - Связь пользователей и ролей

Таблица на рисунке 2.37 представляется из себя сущность типов времени отправки для того чтобы понимать, когда уведомления нужно отправлять.

▼ **publish_type**

- id bigint = nextval('publish_t...')
- name text
- publish_type_pk (id)
- publish_type_id_uindex (id) UNIQUE
- publish_type_pk (id) UNIQUE

Рисунок 2.37 - Сущность типов времени отправки

Таблица на рисунке 2.38 представляется из себя сущность ролей для того чтобы понимать какой пользователь вошел в систему и какие роли он выполняет.

▼ **role**

- id bigint = nextval('role_id_s...')
- name text
- actual smallint = 1
- role_pk (id)
- role_id_uindex (id) UNIQUE
- role_name_uindex (name) UNIQUE
- role_pk (id) UNIQUE

Рисунок 3.38 - Сущность ролей

Таблица на рисунке 2.39 представляется из себя сущность связь роли и функции дабы избавиться от связей многие ко многим.

| role_and_func | |
|----------------------|---------------------------------|
| id | bigint = nextval('role_and...') |
| role_id | integer |
| func_id | integer |
| actual | smallint = 1 |
| role_func_pk | (id) |
| role_func_func_id_fk | (func_id) → func (id) |
| role_func_role_id_fk | (role_id) → role (id) |
| role_func_id_uindex | (id) UNIQUE |
| role_func_pk | (id) UNIQUE |

Рисунок 3.39 - Сущность связи роли и функции

Таблица на рисунке 2.40 представляется из себя сущность специальность чтобы понимать по какой специальности учится обучающийся.

| specialty | |
|---------------------------|----------------------------------|
| id | bigint = nextval('specialty...') |
| name | text |
| institute_id | integer |
| code | text |
| actual | smallint = 1 |
| specialty_pk | (id) |
| specialty_institute_id_fk | (institute_id) → institute (id) |
| specialty_code_uindex | (code) UNIQUE |
| specialty_id_uindex | (id) UNIQUE |
| specialty_pk | (id) UNIQUE |

Рисунок 2.40 - Сущность специальностей

Таблица на рисунке 2.41 представляется из себя сущность студент которые имеют характерные черты в отличие от пользователей.

| student | |
|--------------------------|----------------------------------|
| id | bigint = nextval('student_i...') |
| person_id | integer |
| group_id | integer |
| max_study_course | integer |
| current_study_id | integer |
| grant | boolean |
| actual | smallint = 1 |
| student_pk | (id) |
| student_group_id_fk | (group_id) → group (id) |
| student_person_id_fk | (person_id) → person (id) |
| student_id_uindex | (id) UNIQUE |
| student_person_id_uindex | (person_id) UNIQUE |
| student_pk | (id) UNIQUE |

Рисунок 2.41 - Сущность студент

Таблица на рисунке 2.42 представляет из себя сущность университет, который содержит в себе все остальные единицы.

```
▼ university
  id bigint = nextval('universit...')
  name text
  address text
  person_id integer
  actual smallint = 1
  university_pk (id)
  university_person_id_fk (person_id) → person (id)
  university_id_uindex (id) UNIQUE
  university_name_uindex (name) UNIQUE
  university_pk (id) UNIQUE
```

Рисунок 2.42 - Сущность университет

3. Экспериментальная часть

3.1 Проверка нагрузки на сайт

Несмотря на то, что тестирование проходит на каждом из этапов реализации проекта, финальное тестирование необходимо. Вот некоторые из наиболее важных моментов. Во всех современных браузерах сайт работает; Размещены ли все необходимые материалы; Все ли программные компоненты работают вместе плавно и четко. Когда отладка завершена, наступает момент тестирования сайта. Вопреки распространенному мнению, после того, как сайт выложен, работа с ним не прекращается. Если ваша цель состоит в том, чтобы превратить ваш сайт в инструмент маркетинга, то будьте готовы к тому, что вам нужно: выкладывайте новые материалы, продвигайте сайт, проводите собеседования с посетителями и добавляйте новые функциональные возможности, которые им необходимы.

Основные функции, представленные в программе, тестировались не только нами, но и пользователя. Проверялась адаптивность сайта к любому экрану от телефонов и планшетов до мониторов с разрешением QHD. Функции входа без регистрации, функции электронного дневника.

Помимо основных видимых функций производили проверку на отправку писем на почту и телефон, как оценок, так и уведомлений.

Тестирование уведомлений



Рисунок 3.1 - уведомление по оценкам

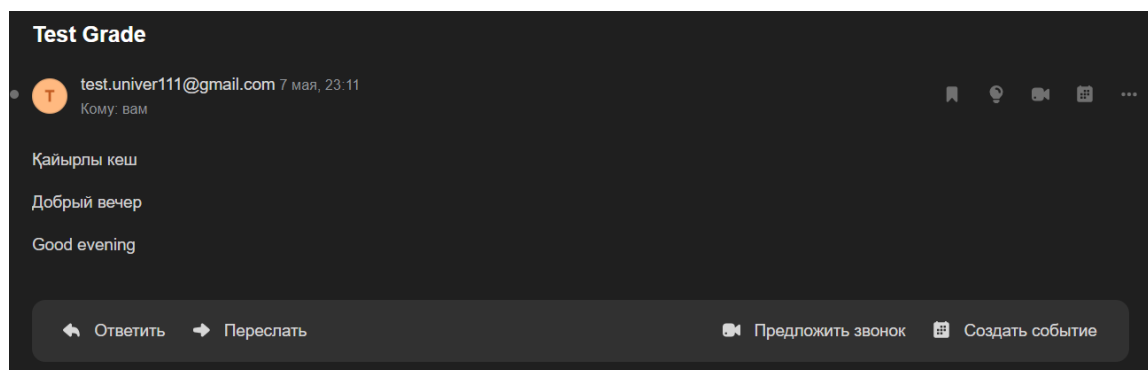


Рисунок 3.2 - уведомление через почту

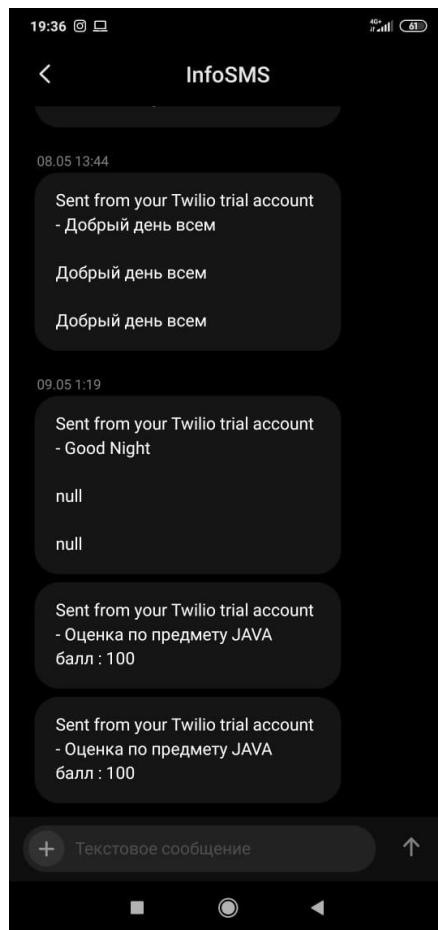


Рисунок 3.3 - уведомление через смс

4. Экономическая часть

Тема данного дипломного проекта – «Разработка клиентской части системы для автоматизации учебного процесса».

Автоматизация учебного процесса позволяет сократить затраты, время и улучшить коммуникацию между преподавателем и студентом. Данный проект в перспективе не имеет ограничений в функциональность. На данный момент помимо основных функций в нем так же есть смс оповещение, что позволяет более ускоренно и эффективно предупреждать как студентов, так и преподавателей, например, о наступающей рубежной недели, праздниках или о предстоящих сейсмотренировка, пожарной безопасности.

В данном разделе приводится рассмотрение экономической составляющей реализации данного проекта, отражающей временные, трудовые и финансовые затраты на реализацию данного проекта.

4.1 Расчет трудоемкости разработки ИС

Расчета затрат на разработку программного обеспечения является объем программного продукта, используемые программы и компоненты. Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой:

$$V_0 = \sum_{i=1}^n V_i \quad (4.1)$$

где V_i - объем отдельной функции ПО;

n - общее число функций.

В экономической части дипломной работе в качестве единиц измерения объема ПО используется строка исходного кода (LOC).

Расчет объема ПП рассчитывается от количества строк кода, функциональности проекта и среды разработки. На стадии технико-экономического обоснования проекта могут быть получены только примерные оценки на основе похожих проектов, существующих на рынке, или путем применения действующих нормативов.

$$V_y = \sum_{i=1}^n V_{yi} \quad (4.2)$$

где V_{yi} - уточненный объем отдельной функции ПО (LOC).

По уточненному объему ПО и нормативам затрат труда в расчете на единицу объема определяются нормативная и общая трудоемкость разработки ПО.

Для подсчета LOC использовалось расширение для среды разработки IntelliJ Idea 2019. В соответствии с ним, проект состоит из 15 компонентов и 3 директив, общее число строк всех этих компонентов – 4053.

Нормативная трудоемкость разработки ПО (T_n) определяется на основании принятого к расчету объема (V_y) и категории сложности, которая уточняется с учетом тяжести и новизны и различных компонентов в проекте.

Разрабатываемая информационная система относится ко 3 категории сложности, тогда нормативная трудоемкость её разработки составит $T_n=19$ чел./дн.

Коэффициент сложности рассчитывается по формуле:

$$K_c = 1 + \sum_{i=1}^n K_i \quad (4.3)$$

где K_i - коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n - количество учитываемых характеристик.

Тогда, коэффициент сложности составит:

$$K_c = 1 + 0,06 + 0,07 + 0,12 = 1,25$$

Поправочный коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_m), определяется удельным весом этих модулей в общем объеме проектируемого продукта (см. таблицу 2 [1]) $K_m=0,8$.

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО (K_n). Разрабатываемая ИС относится к Б-категории новизны ПО и является развитием определенного параметрического ряда ПО, не используется на основе новых типов ПК и в средах новых ОС. Значение коэффициента новизны соответствующее этим факторам равно $K_n=0,9$.

Разработка информационной системы для управления проектированием реализуется одним исполнителем.

Процесс разработки данной ИС состоит из стадий разработки:

- техническое задание (ТЗ);
- эскизный проект (ЭП);
- технический проект (ТП);
- рабочий проект (РП);
- внедрение (ВН).

Для таких проектов рассчитывается общая трудоемкость с учетом стадий разработки:

$$T_0 = \sum_{i=1}^n T_i \quad (4.4)$$

где T_i – трудоемкость разработки ПО на i -й стадии (чел./дн.);

n – количество стадий разработки.

Трудоёмкость стадий определяется на основе нормативной трудоемкости с учетом сложности, новизны и степени использования в

разработке стандартных модулей ПО и удельного веса трудоемкости каждой стадии:

$$T_{yi} = T_H \times d_{cmi} \times K_C \times K_m \times K_H$$

где T_{yi} – уточненная трудоемкость разработки ПО на i -й стадии;

T_H – нормативная трудоемкость;

d_{cmi} – удельный вес трудоемкости i -й стадии разработки ПО в общей трудоемкости разработки ПО (определяется по таблице 4 [1] в зависимости от степени новизны);

K_C – коэффициент, учитывающий сложность ПО, вводится на всех стадиях;

K_m – коэффициент, учитывающий степень использования стандартных модулей ПО, вводится только на стадии рабочего проекта;

K_H – коэффициент, учитывающий степень новизны ПО, вводится на всех стадиях.

Трудоемкость стадий ПО рассчитывается по следующим формулам:

Трудоемкость стадии ТЗ:

$$T_{уз} = T_H \times K_C \times d_3 \times K_H = 19 \cdot 1,25 \cdot 0,1 \cdot 0,9 = 2,13 \text{ чел/дн}$$

Трудоемкость стадии ЭП:

$$T_{уз} = T_H \times K_C \times d_3 \times K_H = 19 \cdot 1,25 \cdot 0,08 \cdot 0,9 = 1,71 \text{ чел/дн}$$

Трудоемкость стадии ТП:

$$T_{ум} = T_H \times K_C \times d_p \times K_H = 19 \cdot 1,25 \cdot 0,09 \cdot 0,9 = 1,92 \text{ чел/дн}$$

Трудоемкость стадии РП:

$$T_{ур} = T_H \times K_C \times d_3 \times K_H \times K_m = 19 \cdot 1,25 \cdot 0,58 \cdot 0,9 \cdot 0,8 = 9,91 \text{ чел/дн}$$

Трудоемкость стадии ВН:

$$T_{ув} = T_H \times K_C \times d_b \times K_H = 19 \cdot 1,25 \cdot 0,15 \cdot 0,9 = 3,2 \text{ чел/дн}$$

Общая трудоемкость определяется как сумма трудоемкостей по стадиям:

$$T_y = T_{уз} + T_{уз} + T_{ум} + T_{ур} + T_{ув} = 2,13 + 1,71 + 1,92 + 9,91 + 3,2 = 18,87 \text{ чел/дн}$$

Таблица 4.1 Трудоемкость разработки программного продукта

| Стадии разработки | Содержание работ | Трудоемкость чел/дн |
|------------------------|---|---------------------|
| 1. Техническое задание | Постановка задачи. | 2,13 |
| | Сбор исходных материалов. | |
| | Выбор и обоснование критериев эффективности и качества разрабатываемой программы. | |
| | Определение требований к программе. | |
| | Разработка технико-экономического обоснования разработки программы. | |
| | Выбор языков программирования. | |
| | Согласование и утверждение технического задания | |

Продолжение таблицы 4.1

| | | |
|-----------------------|---|-------|
| | Предварительная разработка структуры входных и выходных данных. | |
| 2. Эскизный проект | Разработка макета ИС | 1,71 |
| | Разработка общего описания алгоритма решения задачи. | |
| | Разработка технико-экономического обоснования | |
| 3. Технический проект | Разработка прототипа ИС | 1,92 |
| | Определение формы представления входных и выходных данных. | |
| | Разработка структуры программы. | |
| 4. Рабочий проект | Программирование и отладка программы. | 9,91 |
| | Разработка, согласование и утверждение программы и методики испытаний. | |
| 5. Внедрение | Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление. | 3,2 |
| | Передача программы на сопровождение | |
| Итого | | 18,87 |

Трудоемкость в чел/ч при 8-часовом рабочем дне $T_y = 18,87 \cdot 8 = 150,96$ чел/ч

4.2 Расчет затрат на разработку ИС

Расчет полных затрат на разработку проектного решения в виде информационных технологий (C_{pi}) осуществляется по формуле:

$$C_{pi} = Z_{фот} + Z_{сзи} + M_i + P_{ci} + P_{mi} + P_{рки} + P_{зи} + P_{ни} \quad (4.5)$$

где $Z_{фот}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{сзи}$ – отчисления по социальному налогу, тенге;

M_i – затраты на материалы, тенге;

P_{ci} – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

P_{mi} – затраты, связанные с эксплуатацией техники, тенге;

$R_{нкi}$ – затраты на научные командировки, тенге;

$P_{зи}$ – прочие затраты, тенге;

$R_{ни}$ – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ($Z_{фот}$) рассчитывается по формуле:

$$Z_{фот} = Z_{oi} + Z_{ди}, \quad (4.6)$$

где Z_{oi} – основная заработная плата, тенге;

$Z_{ди}$ – дополнительная заработная плата, тенге.

Основная заработная плата исполнителей по конкретному ПО рассчитывается по формуле:

$$Z_{oi} = \sum_{i=1}^n T_{чи} \times T_{ч} \times \Phi_{п} \times K, \quad (4.7)$$

где n - количество исполнителей, занятых разработкой конкретного ПО;

$T_{чи}$ - часовая тарифная ставка i -го исполнителя (тенге/тыс. тенге);

$\Phi_{п}$ - плановый фонд рабочего времени i -го исполнителя (дней);

$T_{ч}$ - количество часов работы в день (час);

K - коэффициент премирования.

Базой для расчета основной заработной платы являются общая трудоемкость, плановая численность работников и плановые сроки разработки ПО.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности. Зарплата frontend разработчика согласно headhunter.kz начинается с 130 000 тенге в месяц при 8-часовом рабочем дне.

Таблица 4.2 Сведения по работникам, задействованным в проекте

| Специалист - Исполнитель | Количество, человек | Зарботная плата в месяц, тенге |
|-----------------------------|------------------------|-----------------------------------|
| Программист | 1 | 130 000 |
| Итого | | 130 000 |

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40-часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах ($\Phi_{р}$):

$$T_{ч} = \frac{T_{м}}{\Phi_{р}} = \frac{130000}{8 \cdot 21} = 773,8 \text{ тг/ч}$$

где $T_{ч}$ - часовая тарифная ставка (тыс. тенге);

$TМ$ - месячная тарифная ставка (тыс. тенге).

Таблица 4.3 Затраты на оплату труда

| Категория работника | Трудоемкость разработки ПП, чел.×ч | Часовая ставка, тг/ч | Сумма, тг |
|-------------------------------|------------------------------------|----------------------|-----------|
| Разработчик | 150,96 | 773,8 | 116 812 |
| ИТОГО затраты на оплату труда | | | 116 812 |

$Z_{oi} = 116\ 812$ тенге.

Дополнительная заработная плата составляет в среднем 10% от основной заработной платы и рассчитывается по формуле:

$$Z_{di} = Z_{oi} \times H_d / 100 = 116\ 812 \cdot 0,1 \approx 11\ 681 \text{ тенге}$$

где H_d - коэффициент дополнительной заработной платы разработчиков.

$$Z_{\text{ФОТ}} = Z_{oi} + Z_{di} = 116\ 812 + 11\ 681 = 128\ 493 \text{ тенге}$$

Социальный налог для юридических лиц рассчитывается:

$$Z_{\text{СН}} = (Z_{\text{ФОТ}} - \text{ОПВ} - \text{ВОСМС}) \cdot 0,095 - \text{СО}$$

где ОПВ – обязательный пенсионный взнос (10% от заработной платы);

ВОСМС – взнос на ОСМС (1% от заработной платы);

СО – социальные отчисления (3,5% от заработной платы с вычетом пенсионного взноса).

$$\text{ОПВ} = 128\ 493 \cdot 0,1 = 12\ 849$$

$$\text{ВОСМС} = 128\ 493 \cdot 0,02 = 2\ 569$$

$$\text{СО} = (128\ 493 - 12\ 849) \cdot 0,035 = 4\ 047$$

$$\text{СН} = (128\ 493 - 12\ 849 - 2\ 569) \cdot 0,095 - 4\ 047 = 6\ 695 \text{ тенге}$$

$$Z_{\text{сзи}} = \text{СН} + \text{СО} + \text{ВОСМС} = 6\ 695 + 4\ 047 + 2\ 569 = 13\ 311 \text{ тенге}$$

Величина затрат на материалы на основании исходных данных определяется по формуле:

$$M_i = \frac{Z_{\text{осн}} \cdot H_{\text{мз}}}{100} \quad (4.8)$$

где $H_{\text{мз}}$ - норма расхода материалов от основной заработной платы (3-5%).

Величина затрат на материалы при норме расхода – 3%:

$$M_i = \frac{116\ 812 \cdot 3}{100} = 3\ 504 \text{ тенге}$$

Расходы по статье «Спецоборудование» включают затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного ПО, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию:

$$P_{ci} = \sum_{i=1}^n C_{ci} \quad (4.9)$$

где C_{ci} - стоимость конкретного специального оборудования (тыс. тенге);

n - количество применяемого специального оборудования.

Таблица 4.4 Затраты на спецоборудование

| Наименование материального ресурса | Единица измерения | Количество израсходованного материала | Цена за единицу, тг | Сумма, тг |
|--|-------------------|---------------------------------------|---------------------|-----------|
| Ноутбук Lenovo Legion Y530 | шт | 1 | 278 000 | 278 000 |
| Мышь компьютерная Seven gx-950 | шт | 1 | 4 000 | 4 000 |
| Принтер лазерный HP LaserJet Pro M125a | шт | 1 | 45 000 | 45 000 |
| ОС Windows 10 Pro (ключ активации) | шт | 1 | 8000 | 8000 |
| intellij idea (лицензия на пользователя в месяц) | шт | 1 | 10000 | 10000 |
| Итого | | | | 345 000 |

Научные командировки не предусмотрены.

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате:

$$P_{zi} = Z_{oi} \cdot \frac{H_{пз}}{100} \quad (4.10)$$

где $H_{пз}$ - норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$P_z = 116\,812 \cdot 0,2 = 23\,362 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на

общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу (H_{nr}) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации:

$$P_{ni} = Z_{oi} \cdot \frac{H_{nr}}{100}$$

где P_{ni} - накладные расходы на конкретную ПО (тыс. тенге);

H_{nr} - норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

$$P_{ni} = Z_{oi} \cdot \frac{H_{nr}}{100} \quad (4.11)$$

Таблица 4.5 Затраты на разработку

| Затраты на разработку | Условное обозначение | Значение, тенге | В процентах от общей суммы |
|--------------------------------|----------------------|-----------------|----------------------------|
| Фонд оплаты труда | $Z_{фот}$ | 128 493 | 30% |
| Социальные отчисления и налоги | $Z_{сзи}$ | 13 311 | 3,1% |
| Материалы | M_i | 3 504 | 0,9% |
| Спецоборудование | P_{ci} | 345 000 | 41,6% |
| Прочие затраты | P_{zi} | 23 362 | 5,4% |
| Накладные расходы | P_{ni} | 81 768 | 19% |
| Итого | C_{ni} | 595 438 | 100% |

Рентабельность и прибыль по создаваемому ПО (P_{ci}) определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. В случае разработки ПО для использования внутри организации оценка программного продукта производится по действующим правилам и показателям внутреннего хозрасчета (по ценам, устанавливаемым для расчета за услуги между подразделениями). Прибыль рассчитывается по формуле:

$$P_{oi} = C_{ni} \cdot \frac{U_{pni}}{100}$$

где P_{oi} - прибыль от реализации ПО заказчику (тыс. тенге);

U_{pni} - уровень рентабельности ПО (%), в дипломной работе брать 40-60%;

C_{ni} - себестоимость ПО (тыс. тенге).

$$P_{oi} = 595\,438 \cdot 0.4 = 238\,175 \text{ тенге}$$

Прогнозируемая цена ПО без налогов (C_{pi}):

$$C_{pi} = C_{ni} + P_{oi} = 595\,438 + 238\,175 = 833\,613 \text{ тенге}$$

Прогнозируемая отпускная цена (C_{01}):

$$C_{01} = C_{ni} + \text{НДС}$$

Ставка налога на добавленную стоимость НДС в РК на 2020 год составляет 12% от отпускной цены ПО.

$$C_{01} = 833\,613 + \frac{833\,613 \cdot 12}{100} = 933\,646 \text{ тенге}$$

Организация-разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по нормативу ($H_0=10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{0i} = C_{ni} \cdot \frac{H_0}{100} = 595\,438 \cdot 0,1 = 59\,543 \text{ тенге}$$

Затраты на сопровождение ПО (P_{Ci}). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_C=20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле:

$$P_{Ci} = C_{ni} \cdot \frac{H_C}{100} = 595\,438 \cdot 0,2 = 119\,087 \text{ тенге}$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение будет:

$$K = 933\,646 + 59\,543 + 119\,087 = 1\,112\,276 \text{ тенге.}$$

4.3 Расчет эксплуатационных затрат

Годовые эксплуатационные текущие затраты в условиях функционирования информационных технологий (C) рассчитываются по формуле:

$$C = ЗП + ОТ + А + М + НР \quad (4.13)$$

где ЗП - годовые затраты на оплату труда специалистов при выполнении ими своих функций в рамках автоматизируемого процесса после внедрения ИТ, тенге;

ОТ - отчисления по социальному налогу, тенге;

А - затраты на амортизацию, тенге;

М - годовые материальные затраты на сопровождение программного продукта, тенге;

НР - накладные расходы, тенге.

Годовые затраты по заработной плате специалистов после внедрения ИТ определяются по формуле:

$$ЗП = \frac{O_c \cdot Ч_c \cdot 12}{\Phi_{р.в.}} \cdot t_{общ} \cdot 12 \cdot (1 + K_d) \quad (4.14)$$

где O_c - оклад специалиста, тенге/мес.;

$Ч_c$ - численность специалистов, участвующих в процессе, чел.;

$\Phi_{р.в.}$ - годовой фонд рабочего времени, час;

$t_{общ}$ - трудоемкость решения задач в условиях функционирования ИТ в месяц, час;

K_d - коэффициент дополнительной заработной платы.

Трудоемкость решения задач в условиях функционирования ИТ в месяц вычисляется следующим образом:

$$t_{общ} = \sum_{\beta=1}^n t_{\beta} \cdot K_{\beta} \quad (4.15)$$

где t_{β} - затраты времени на решение β -й задачи, час;

K_{β} - количество решаемых β -х задач в месяц, ед.

В ходе эксплуатации системы предполагается появление серверных ошибок, на обработку одной ошибки в среднем уходит 2 часа, на протяжении месяца эксплуатации может возникать до 40 ошибок, помимо этого ожидается внесение изменений в интерфейс и доработка компонентов по требованию руководства. Всего 15 компонентов, на доработку одного уходит в среднем 2 часов.

$$t_{общ} = 2 \cdot 40 + 15 \cdot 2 = 80 + 30 = 110 \text{ часов}$$

$$ЗП = \frac{130\,000 \cdot 1 \cdot 12}{1968} \cdot 165 \cdot 12 \cdot 1 = 1\,569\,512 \text{ тенге}$$

Материальные затраты определяются по формуле:

$$M = \sum_{\mu=1}^n Ц_{\mu} \cdot N_{\mu} \quad (4.16)$$

где $Ц_{\mu}$ - цена μ -го вида единицы материальных затрат в условиях функционирования ИТ, тенге.

N_{μ} - используемое количество μ -го вида материальных затрат в месяц, ед.

В условиях функционирования разрабатываемой ИС, не предполагается материальных затрат.

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются в процентах к балансовой стоимости оборудования и рассчитываются по формуле:

$$A = C_{обор} \cdot H_a \quad (4.17)$$

где H_a — норма амортизации (25 %);

$C_{\text{обор}}$ — первоначальная стоимость оборудования;

$$A = (278\,000 \cdot 0,25) + (8\,000 \cdot 0,25) + (45\,000 \cdot 0,25) = 82\,750$$

Отчисления по социальному налогу и накладные расходы рассчитываются так же, как и при разработке информационных технологий.

$$CH = (3П - ОПВ - ВОСМС) \cdot 0,095 - CO$$

$$ОПВ = 1\,569\,512 \cdot 0,1 = 156\,951 \text{ тенге}$$

$$ВОСМС = 1\,569\,512 \cdot 0,02 = 31\,390 \text{ тенге}$$

$$CO = (1\,569\,512 - 156\,951) \cdot 0,035 = 49\,439 \text{ тенге}$$

$$CH = (1\,569\,512 - 156\,951 - 31\,390) \cdot 0,095 - 49\,439 = 81\,772 \text{ тенге}$$

$$OT = CH + CO + ВОСМС = 81\,772 + 49\,439 + 31\,390 = 162\,601 \text{ тенге}$$

$$HP = 3П \cdot \frac{H_{HP}}{100} = 1\,569\,512 \cdot 0,7 = 1\,098\,658 \text{ тенге}$$

Таблица 4.7 Эксплуатационные затраты

| Затраты на эксплуатацию | Условное обозначение | Значение, тенге | В процентах от общей суммы |
|--------------------------------|----------------------|-----------------|----------------------------|
| Заработная плата | ЗП | 1 569 512 | 54,6% |
| Социальные отчисления и налоги | OT | 162 601 | 5,6% |
| Материальные затраты | М | 0 | 0% |
| Амортизационные отчисления | А | 49 439 | 1,7% |
| Накладные расходы | HP | 1 098 658 | 38,1% |
| Итого | | 2 880 210 | |

$$C = 2\,880\,210 \text{ тенге}$$

4.4 Расчет результатов от создания и использования ИС

Информационная система для учебных заведений предназначена в первую очередь для улучшения образовательного процесса. Она позволяет сократить затраты, время и улучшит коммуникацию между преподавателем и студентом.

Для оценки экономии от использования разрабатываемой ИС необходимо сравнить эксплуатационные расходы с ее применением и без.

Статьи затрат при применении ИС включают в себя:

- заработная плата специалиста, осуществляющего поддержку и сопровождение системы;
- износ оборудования;
- накладные расходы.

Данная система не предполагает расхода каких-либо материалов.

В организации имеется специалист (месячная заработная плата – 150 000 тенге), основной задачей которых является осуществление контроля работы со студентами.

$$ЗП = 3\,000\,000 \text{ тенге}$$

$$ОПВ = 3\,000\,000 \cdot 0,1 = 300\,000$$

$$СО = (3\,000\,000 - 300\,000) \cdot 0,035 = 94\,500$$

$$ВОСМС = 3\,000\,000 \cdot 0,02 = 60\,000$$

$$СН = (ЗП - ОПВ - ВОСМС) \cdot 0,095 - СО = 250\,800$$

$$ОТ = СО + СН + ВОСМС = 94\,500 + 60\,000 + 250\,800 = 405\,300 \text{ тенге}$$

В их распоряжении находятся два персональных компьютера (стоимость одного ~ 175 000 тенге), принтер (70 000 тенге) и прочая компьютерная периферия (~30 000 тенге); итого затрат на оборудование – 300 000 тенге.

Износ оборудования рассчитывается исходя из 25 % амортизационных отчислений за год.

$$А = (350\,000 \cdot 0,25) + (80\,000 \cdot 0,25) + (20\,000 \cdot 0,25) = 112\,500 \text{ тенге}$$

На заправку картриджей, бумагу прочую канцелярию в месяц уходит порядка 25 000 тенге.

Общие накладные расходы составят:

$$НР = ЗП \cdot \frac{Н_{нр}}{100} = 3\,000\,000 \cdot 0,7 = 2\,100\,000 \text{ тенге}$$

Статьи затрат без применения ИС включают в себя:

- заработная плата специалистов – кураторов проектов, которые осуществляют управление проектами;
- износ используемого ими оборудования;
- расход материалов (к примеру канцелярия – бумага, картридж и т.д.)
- накладные расходы.

Таблица 4.8 Годовые эксплуатационные затраты

| Статьи | Без применения ИС (1 сотрудник – куратор проекта) | С применением ИС (1 сотрудник – программист, сопровождающий систему) |
|-----------------------------------|---|---|
| Годовая заработная плата | 3 000 000 | 1 569 512 |
| Социальные отчисления и налоги | 250 800 | 162 601 |
| Расходуемые материалы | 200 000 | 0 |

Продолжение таблицы 4.8

| | | |
|----------------------------|-----------|-----------|
| Амортизационные отчисления | 112 500 | 49 439 |
| Накладные расходы | 2 100 000 | 1 098 658 |
| Всего | 5 663 300 | 3 323 318 |

Ожидаемая условно-годовая экономия определяется по формуле:

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i \quad (4.18)$$

где $\mathcal{E}_{\text{уг}}$ - величина экономии, тенге;

C_1 и C_2 – показатели текущих затрат по базовому и внедряемому вариантам, тенге;

$\sum \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, тенге.

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 = 2\,339\,982 \text{ тенге}$$

4.5 Расчет основных показателей экономической эффективности

Так как разработанная информационная система несет более социальный эффект, чем экономический, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле:

$$\mathcal{E}_r = \mathcal{E}_{\text{уг}} - K \cdot E_H \quad (4.19)$$

где \mathcal{E}_r - ожидаемый годовой экономический эффект, тенге;

$\mathcal{E}_{\text{уг}}$ — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения, тенге;

E_H - нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

$$E_H = \frac{1}{T_H} \quad (4.20)$$

где T_H — нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений принимается исходя из срока морального старения -технических средств и проектных

решений ИС ($T_H=1,2,3\dots n$), для программных продуктов срок окупаемости принимаем равным 4 года.

$$E_H = \frac{1}{4} = 0,25 \quad (4.21)$$

$$\Delta_r = 2\,339\,982 - 1\,112\,276 \cdot 0,25 = 1\,811\,913 \text{ тенге}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\Delta_{yr}}{K} \quad (4.22)$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

Δ_{yr} — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

$$E_p = \frac{2\,339\,982}{1\,112\,276} = 2,1$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p} \quad (4.23)$$

где E_p - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{2,1} = 0,48 \text{ года} \approx 5,8 \text{ месяцев}$$

Таблица 4.9 Показатели сравнительной экономической эффективности от внедрения программного продукта

| Наименование показателей | Значение |
|--|-----------|
| Условная годовая экономия затрат, тенге | 1 811 913 |
| Коэффициент экономической эффективности капитальных вложений (E_p) | 2,1 |
| Срок окупаемости капитальных вложений (T_p) | 0,48 |

Таким образом, разработанная информационная система позволила упростить и сэкономить проект. Ожидаемый годовой экономический эффект составил 1 811 913 тенге. Приложение окупится в первые 5,8 месяцев использования.

Вывод

Суммарные затраты на разработку программного продукта составляют 1 811 913 тенге, цена реализации – 3 323 318 тенге с учетом НДС. Основную долю в себестоимости программного продукта составляют машинное время – 1 098 658 тенге или 87,6%, и фонд оплаты труда – 162 601 тенге или 8,4%.

Гибкость проекта позволяет полностью удовлетворить потребности клиентов в вопросах функций и дизайна. Затраты в начале полностью оправдают возможности проекта.

5 Безопасность жизнедеятельности

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

5.1.2 Характеристики рабочего места

Рабочее помещение находится вблизи развязки и кафе так же с солнечной стороны это является высоким источником шума, вредных газов, и временами не правильной освещенностью. Но это все решено за счет кондиционеров, светоотражающей тонировке и окнами, имеющий двойное остекление.

Таблица 5.1 Описание помещения

| Название | Характеристика |
|-------------------------------------|--|
| Этаж | 6 этаж из 7 этажей |
| Размер помещения | Ширина 12м, длина 23м, высота 3м |
| Окно | 2 двойных окна (3м х 2м), двойной стеклопакет. |
| Освещение искусственное | 15 светильников по 2 люминесцентные лампы |
| Стены | Светлого оттенка |
| Разряд работы зрительного характера | IV разряда |
| Кондиционер | 2 кондиционера |

Помимо этого, работники могут столкнуться с физически опасными и вредными производственными факторами, такими как повышенный шум, пожар, статическое электричество, высокая температура в помещении, неправильное освещения на рабочих местах, отсутствие или недостаток естественного света, поражение электрическим током, электромагнитное излучение и другие.

5.1.2 Физически вредные и опасные факторы

Основные виды опасных и вредных факторов:

- физические;
- химические;
- психофизические;
- биологические.

К физическим вредным и опасным факторам относятся:

- повышенные уровни электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения;

- повышенный уровень статического электричества и запыленности воздуха рабочей зоны;

- повышенное содержание положительных аэронов и пониженное содержание отрицательных аэроионов в воздухе рабочей зоны;

- повышенный уровень блескости и озлобленности;

- неравномерность распределения яркости в поле зрения;

- повышенная яркость светового изображения;

- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Химические вредные и опасные факторы следующие:

- повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола и формальдегида.

Психофизиологические вредные и опасные факторы:

- напряжение зрения и внимания;

- интеллектуальные, эмоциональные и длительные статические нагрузки;

- монотонность труда;

- большой объем информации, обрабатываемый в единицу времени;

- нерациональная организация рабочего места.

Работа в офисе для программистов и других специальностей, связанных с компьютером характерны следующие физические проблемы: переутомление глаз, головная боль, тянущие боли в мышцах шеи, рук и спины, снижение концентрации внимания.

Одной из причин служит то, физиологически выстроенная зрительная система человека, воспринимающая объекты в отраженном свете, но не приспособлена к работе с искусственным изображением, на примере дисплея. Изображение на дисплее подсвечивается, мерцает, состоит пикселей и чаще всего цветовая гамма не далека от идеала. Помимо этого, частая работа с текстом дает дополнительную нагрузку, так как для набора текста необходимо переводить взгляд с экрана на клавиатуру, которые находятся на разном расстоянии и обладают разной освещенностью.

Длительная и интенсивная работа на компьютере может стать источником тяжелых профессиональных заболеваний, таких, как травма повторяющихся нагрузок (ТПН), представляющая собой постепенно накапливающиеся недомогания, переходящие в заболевания нервов, мышц и сухожилий руки.

После многолетней работы в таком режиме и при неправильно организованном рабочем месте, у специалистов возникают проблемы со зрением, онемение, ослабление мышц, сонливость, раздраженность.

Химическая же опасность заключается в использовании недорогих синтетических материалов в «чистом» виде или в виде «добавок», а также чернил для принтеров, факсов, копиров. Дешевая мебель, пластиковые стеновые панели, ковровые покрытия, рабочие элементы оборудования,

оконные профили выделяют множества токсичных веществ. Так же от копировального оборудования выделяется озон, который в большом количестве становится ядовитым.

Психофизически факторы в целом зависит от правильного шумоподавления оборудования, правильного освещения. Помимо этого, однообразная работа, перегрузки так же способны привести к профессиональным болезням.

Биологические опасности связаны с не правильной системой кондиционирования и вентиляции. Не правильный расчет, не своевременная чистка ведет к распространению вирусов, бактерий, спор грибка. Это может привести к развитию разных инфекций специалистов, в том числе и трудно излечимых.

5.1.3 Здоровье и безопасность на рабочем месте

Для поддержания установленных стандартов, параметров микроклимата и качества воздуха в помещении с большим количеством компьютеров и другого оборудования, вентиляция является конструкцией системы воздушного потока, включающей в себя определение относительно воздушного потока для вентиляции помещения и охлаждения самих ПК. Существует общая и местная вентиляция. Минимальный расход определяется исходя из расхода 50-60 м³ / ч на человека, однако не менее двух модификаций воздуха в час.

Таблица 5.2 Параметры микроклимата

| Сезон года | Параметры микроклимата | Величина |
|------------|-------------------------|---------------|
| Летний | Температура в помещении | 20-26°С |
| | Относительная влажность | 40-60% |
| | Скорость воздуха | до 0,1м/сек |
| Зимний | Температура в помещении | 22-24°С |
| | Относительная влажность | 40-60% |
| | Скорость воздуха | 0,1-0,2 м/сек |

Так же важным неотъемлемым элементом организации труда является освещенность. В помещении есть 2 источника свет: естественный и искусственный. Для регулирования этих двух источников в помещениях предусмотрены регулируемые жалюзи, светоотражающая тонировка. Люминесцентные лампы используются в основном для искусственного освещения. Они имеют длительный срок службы (до 10000 ч), высокую светоотдачу (до 75 лм / Вт и более), низкую яркость относительно освещаемой поверхности, близкую к естественному спектральному составу излучаемого света, что обеспечивает хорошее световосприятие для глаз.

Уровень шума на рабочем месте не может превышать 50 дБА. Шумоподавление достигается за счет увеличения ограждающей изоляции,

герметизации по периметру оконных точек подключения звукоизоляции, пересечения или прохода со стенами. Защита от шума на рабочем месте, а также от внутренних источников шума снаружи, за счет снижения шума у источника, использования в отношении коллективной и индивидуальной защиты.

Большая нагрузка в IT сфере ложится на зрение и мозг. Что бы не привезти к последствиям необходимо:

- во-первых, многое зависит от экрана монитора, поэтому необходим экран с хорошей цветопередачей, 15дюймов, и правильное расстояние между глазами и монитором (45-60 см);

- во-вторых, правильно распределение света в рабочей зоне. На экране не должно быть бликов, свет должен падать только на документы или предметы в рабочей зоне (клавиатура мышь и тд.)

- максимальная длительность работы за компьютером, для взрослого человека, не должно превышать 2 часов, с обязательными перерывами не меньше 15 минут.

- уменьшить время работы в интернете, связано это с быстрой сменой цветов, контрастности и большим потоком информации;

- по чаще слушать спокойную музыку в перерывах, для расслабления не только головы, но и зрения;

- так же необходим специальный стул, для снижения нагрузки на позвоночник, он должен быть газпатроне, без подлокотников;

- стол в свою очередь должен иметь выдвижную доску для клавиатуры и мыши.

5.1.4 Антропогенное воздействие на окружающую среду и экологическая безопасность.

Персональные компьютеры не загрязняют окружающую среду меньше, чем автомобили. ПК деионизирует окружающую среду и снижает влажность. Для производства компьютера используется от 24 кг до 240 кг ископаемого топлива, 22 кг химикатов, 1,5 тонны воды. Использование ископаемого топлива способствует глобальному потеплению, и многие из химических веществ являются токсичными. В частности, химические вещества, используемые в производстве компьютеров, материнских плат, мониторов, процессоров и угрожают здоровью рабочих на фабриках, а также влияют на резервуары с питьевой водой вокруг полигона.

Респираторные заболевания носят преимущественно аллергический характер. Это связано с тем, что корпус и системная плата мониторинга питания нагреваются и выделяют вредные вещества в воздух, особенно если ваш компьютер новый. Компьютер создает вокруг себя электростатическое поле, которое притягивает пыль и, в свою очередь, осаждается на свет пользователя.

Чтобы решить эти проблемы, необходимо обеспечить правильное использование компьютеров, и вы должны использовать материалы и биопластик при производстве корпусов и компонентов и минимизировать использование опасных веществ.

5.2 Расчеты

5.2.1. Расчет шума

Шум пагубно влияет на условия труда оказывая вредное влияние на организм человека. Работая в длительном воздействие шума, испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенная утомляемость, потеря аппетита, боли в ушах и др. Такое воздействие на органы человека влияют не только на психологическое состояние (стресс, раздраженность и др.), но и физически на тело недоедание, сонливость. Под воздействием шума, снижение концентрации внимания, нарушенных физиологических функций появляется усталость из-за повышения энергозатратности и умственные нагрузки, ухудшается речевая функция. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума выше 80 дБ на человека, приводит к частичной или полной потере слуха. В таблице 5.3 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Таблица 5.3 - Предельные уровни звука, дБ, на рабочих местах.

| Категория напряженности труда | Категория тяжести труда | | | |
|-------------------------------|-------------------------|---------|---------|---------------|
| | Легкая | Средняя | Тяжелая | Очень тяжелая |
| Мало напряженный | 80 | 80 | 75 | 75 |
| Умеренно напряженный | 70 | 70 | 65 | 65 |
| Напряженный | 60 | 60 | - | - |
| Очень напряженный | 50 | 50 | - | - |

В основном высокий уровень шум в помещении создают кулеры на компьютерах, кондиционеры, принтеры. Для устранения необходимости и целесообразности снижения шума, необходимо знать уровень шума на рабочем месте.

Шум, возникающий от нескольких источников, работающих одновременно, рассчитывается на основе принципа энергетической суммирования отдельных источников излучения:

$$L_{\Sigma} = 10 \log \sum_{i=1}^{i=n} 10^{0,1L_i}, \quad (5.1)$$

где L_i – уровень звукового давления;
 i -го источника шума; n – количество источников шума.

Результаты решения сравниваются с допустимыми параметрами уровня шума, если он превышает, то необходимо принять меры по его снижению. Например, покрыть стены и потолок звукопоглощающими материалами, грамотное расположение оборудования и расположения сотрудника на рабочем месте.

Уровни шума оборудования:

- в стандартном системном блоке это жесткий диск 40дБ и кулеры 45дБ;
- монитор 17дБ;
- в зависимости от типа клавиатуры (в офисах мембранные) 17дБ;
- принтер-сканер 45дБ;

Подставив в формулу все эти значения получим:

$$L_{\Sigma} = 10 \cdot \lg(10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ}$$

Полученный результат не превышает допустимую норму шума в 65дБ. А так как чаще всего одновременно будут работать только системный блок (находящийся под столом), клавиатура и монитор, то значения будут ещё ниже. Пример расположения на рисунке 5.1.

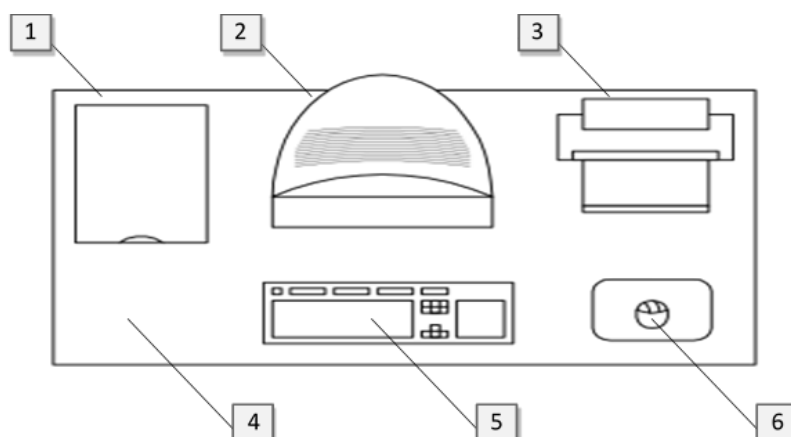


Рисунок 5.1

Так же для работы необходимо соблюдать:

- высота рабочей поверхности в рекомендуемом диапазоне 68-76 см;

- высота сиденья над уровнем пола находится в пределах 42-55 см;
- установить экран на расстоянии 50-60 см от глаз;
- голова не должна быть наклонена более чем на 20 °, плечи должны быть расслаблены, локти - под углом 80 ° - 100 °, предплечья и кисти рук - в горизонтальном положении.

5.2.2 Расчет освещения

Исходные данные:

- 1) Параметры помещения (L x B x H), м: 12 x 23 x 3;
- 2) Высота окна hок, м: 2;
- 3) Высота начала окна hн.ок., м: 0,8;
- 4) Уровень условной рабочей поверхности hпов, м: 0,65;
- 5) Разряд зрительной работы: V, в;
- 6) Световой пояс: г. Алматы, IV;
- 7) Нзд, м: 24;
- 8) Расстояние до рядом стоящего здания, P, м: 12;
- 9) ρср: 0,5;

5.2.3 Расчет естественного освещения

Расчет естественного освещения заключается в определении площади световых проемов.

Формула общей площади окна при боковом освещении:

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0 \cdot K_{зд} \cdot K_3}{100 \cdot \tau_0 \cdot r_1}, \quad (5.2)$$

Где S_n – площадь пола помещения, м²:

$$S_n = B \cdot L = 12 \cdot 23 = 276 \text{ м}^2 \quad (5.3)$$

e_n – нормированное значение КЕО:

$$e_n^{IV} = e_n \cdot m \cdot c$$

Значение коэффициента

$e_n = 1.2$ для работ средней точности V подразряда. Исходя из этого, $m = 0.9$; $c = 0.75$. (СНиП РК 2.04-05-20013, с.20)

$$e_n^{IV} = 1,2 \cdot 0,9 \cdot 0,75 = 0,81$$

K_3 – коэффициент запаса по таблице 3.11: $K_3 = 1,2$; (СНиП РК 2.04-05-20013, с.24)

τ_0 – общий коэффициент светопропускания с $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$,

τ_1 – коэффициент светопропускания материала: для стеклопакета 0,8 (СНиП РК 2.04-05-20013, 25)

τ_2 - коэффициент, учитывающий потери света в переплетах светопроёма: для металлических одинарных 0,75(СНиП РК 2.04-05-20013, с.25)

τ_3 - коэффициент, учитывающий потери света в несущих конструкциях, при боковом освещении: для железобетонных форм 0,8(СНиП РК 2.04-05-20013, с.25)

τ_4 - коэффициент, учитывающий потери света в солнцезащитных устройствах: для убирающихся регулируемых жалюзи 1

$$\text{Тогда } \tau_0 = 0,8 * 0,75 * 0,8 * 1 = 0,48$$

η_0 - световая характеристика окон:

Отношение длины помещения к его глубине:

$$23/(12-1)=2,09;$$

$$l=12-1=11$$

$$h_1 = h_{ок} + h_{н.ок} - h_{пов} = 2,09 + 0,8 - 0,65 = 2,24 \text{ м} \quad (5.4)$$

где h_1 - высота от уровня рабочей поверхности до верхней точки окна.

$$B/h_1 = 12/2,24 = 5,35$$

$$l/B = 11/12 = 0,91$$

$$L/B = 23/12 = 1,91$$

$$\rho_{ср} = 0,5;$$

$$r_1 = 1,5$$

$K_{зд}$ -коэффициент, учитывающий затенение окон противостоящими зданиями:

$$P/H_{зд} = 12/24 = 0,5$$

$$K_{зд} = 1,7$$

Подставим все значения в расчетную формулу и вычислим необходимую общую площадь окон при одностороннем боковом освещении:

$$S_0 \approx 7,54 \text{ м}^2$$

Отсюда следует, что для соответствия нормативам естественного освещения общая площадь окон должна быть равна 7,16 м². В нашем случае она равна 7 м², таким образом, существующей площади окон недостаточно для обеспечения нормированного освещения. Поэтому мы выполним расчет искусственного освещения.

5.2.4 Расчет искусственного освещения методом коэффициента использования

Исходные данные:

- количество источников искусственного освещения – 15 светильника ЛСП64-2м (1120x276 мм), в каждом по 2 люминесцентных лампы мощностью 65 Вт, световой поток лампы $\Phi_{л} = 3320$ лм.

- нормируемая освещенность – 400 лк.
- коэффициенты отражения от потолка, стен и пола соответственно равны:

$\rho_{\text{пот}}=70\%$, $\rho_{\text{ст}}=50\%$, $\rho_{\text{пол}}=30\%$, (СНиП РК 2.04-05-20013, с.43)

Вычислим высоту подвеса светильника над рабочей поверхностью:

$$H=h-h_p-h_c \quad (5.5)$$

где h_c - расстояние от светильника до перекрытия, $h_c=0,05$ м;

h_p – высота рабочей поверхности над полом, $h_p=0,7$ м;

h – высота помещения, $h=3$ м;

$$H=3-0,7-0,05=2,25\text{м};$$

Самым выгодным расстоянием между светильниками определяется как:

$$L=\lambda *H, \quad (5.6)$$

где $\lambda=1,2/1,4$

$$L=1,2*2,25=2,7\text{м}$$

Расстояние от стены до ближайшего светильника определяется по формуле:

$$l_1=(0,4/0,5) *L$$

$$l_1=0,4*2,7=1,08$$

Определяем индекс помещения:

$$i=(1*s)/(H(1+s)) \quad (5.7)$$

$$i=(6*4)/(2,25*(6+4))= 1,06$$

Коэффициент использования в данном случае равен $\eta=50\%$, коэффициент запаса равен $k_z=1,2$ (СНиП РК 2.04-05-20013,с.58)

Определим количество люминесцентных ламп по формуле:

$$N=(E*k_z*S_{oc}*Z)/(n*\Phi_{л}*\eta) \quad (5.8)$$

где S_{oc} – площадь помещения

k_z – коэффициент запаса;

Z - коэффициент минимальной освещенности, определяемый

отношением;

E_{cp}/E_{min} , значение которого для люминесцентных ламп – 1,1; (СНиП РК 2.04-05-20013,с.61)

n – количество ламп в светильнике;

$\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л}=3320$ лм;

η – коэффициент использования, $\eta=50\%$;(СНиП РК 2.04-05-20013, с.62)

$$N=(400*1,2*24*1,1)/(2*3320*0.5)= 3,81\approx 4;$$

Вывод

В данной части дипломной работы было рассмотрено правильная организация рабочего места, правильное освещение в помещении, вредность шума, микроклимат. Чаще всего в помещения даже с частью стен из стекла не хватает естественного освещения и необходимо правильно интегрировать его с искусственным. Необходимо держать рабочее место в порядке, следить за приборами уровень шума которых может доставлять дискомфорт. Чаще проветривать помещение и знать основы первой помощи.

Заключение

Во время создания дипломного проекта укрепил теоретические навыки, полученные во время обучения в университете. Получил знания в веб-программировании и дизайне. Изучил новые методы программирования через Angular, TypeScript и другие инструменты для Java. Все это стало результатом создания проекта " Herodotus". В нём удалось объединить органичность, простоту интерфейса, красочный современный дизайн с упрощенным и понятным кодом.

Толчком для создания проекта стали аналоги, которые были изучены и доработаны уже в этом проекте. Сайт для учебных заведений имеет дневник, уведомления и отправкой документов сразу на почту группе или отдельному ученику, или преподавателю.

В проект так же было внедрено оповещение через смс. Была подключена облачная коммуникационная платформа. Так же был сделан расчет по затратам, скорость отправки на одно и несколько устройств сразу. Результатом стала цена в 47тенге за смс и его получением в течении 5-10 минут, после отправки с сервера.

Для простого переноса проекта на любой персональный компьютер использовалась программа развертывания Docker. С ее помощью был объединена клиентская и серверная часть на двух устройствах, что позволило быстро и эффективно использовать запросы на сервер и получением правильно отображения результата в клиентской части.

TypeScript использовался вместо JavaScript так как у первого более широкий функциональный спектр и меньшее время отклика, так же снизить нагрузку на оперативную память.

На сайте не предусмотрена регистрация, но есть возможность смены пароля. Это позволит обезопасить аккаунты пользователей и исключит появление ботов на сервере.

Список используемой литературы

- 1 Дженнифер Нидерст Роббинс. Web-дизайн. Справочник. – М.:Кудиц-Пресс, 2017. – 816 с.
- 2 Дэвид Флэнаган. JavaScript. Подробное руководство. – СПб.: Символ, 2016. – 992 с.
- 3 Дронов В.А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. – СПб.: BHV, 2016. – 688 с.
- 4 Аманбаев У.А. Экономика предприятия. – А.: Бастау, 2012. – 100
- 5 Буров В.П. Бизнес-план фирмы. – М.: Инфра-М, 2016. – 50 с.
- 6 Куатова Д.Я. Экономика предприятия. – А.: Экономика, 2018. – 89
- 7 СНиП РК 2.04-15-20013«Естественное и искусственное освещение»
- 8 СН РК 2.04-03-2015 Защита от шума и вибрации
- 9 Безопасность жизнедеятельности. Сычев Ю.Н. (ЕАОИ; 2015, 311с.)
- 10 Безопасность жизнедеятельности. Калюжный Е.А., Михайлова С.В. и др. (АГПИ, 2017, 316с.)
- 11 Безопасность жизнедеятельности. Белов С.В., Ильницкая А.В., Козьяков А.Ф. и др. (2015, 616с.)
- 12 Пауэлл, Т. Web-дизайн / Т. Пауэлл. – 3-е изд., перераб. и доп., 2016. – 1084 с.
- 13 Голенищев, Э.П. Информационное обеспечение систем управления: учеб. пособие для вузов / Э.П. Голенищев, И.В. Клименко. – Ростов н/Д : Феникс, 2015. – 352 с.
- 14 Кузнецов, М.В.. Практика разработки Web-сайтов / М.В. Кузнецов, И.В. Симдянов, С.В. Гольшев, 2016. – 960 с.
- 15 Томсон Л. Разработка Web-приложений на Angular /Л. Томсон, Л. Веллинг. – 2-е изд., испр.: ООО «ДиаСофт», 2017. – 672 с.
- 16 Хокинс, С. Администрирование Web-сервера и руководство по электронной коммерции / С. Хокинс. – М. : Издательский дом «Вильямс», 2015. – 336 с.
- 17 Евдокимов Н. В. Основы контентной оптимизации. / Н. В Евдокимов. - изд. ООО И. Д. Вильяме, 2017 год, 160 с.
- 18 Редкол. Тихонов А. Н. Интернет-порталы: сборник научных статей, выпуск 2, / А. Н. Тихонов - изд. Просвещение, 2015 год, 499 с.
- 19 Кирсанов Д., Веб-дизайн./ Д. Кирсанов.,:Символ-плюс, 2015. –376 с.
- 20 Казарин О.В. Безопасность программного обеспечения компьютерных систем./ О.В. Казарин - Москва, МГУЛ, 2015, 212 с.
- 21 Аксенов К. А., Клебанов Б. И. Работа с CASE-средствами ВРwin и ERwin: учебное пособие./ К. А. Аксенов, Б. И.Клебанов 2016-50с.
- 22 Вальтер, Штефен Создание приложений для Windows 10 с использованием TypeScript / Штефен Вальтер. - М.: ДМК Пресс, 2017. - 344 с.

Приложение А

(ЛИСТИНГ ПРОГРАММЫ)

```
// журнал студента
<h3>Журнал</h3>
<!-- STUDENT -->
<div [fxLayout]="column" *ngIf="student">
  <div [fxLayout]="row" class="ui-g">
    <div [fxLayout]="column" class="min-width ui-g-12 ui-md-6 ui-lg-3 ui-sm-6">
      <p-dropdown *ngIf="years" [options]="years" [(ngModel)]="selectedYear"
        placeholder="Select a Year"
        [showClear]="true"></p-dropdown>
    </div>
    <div [fxLayout]="column" class=" min-width ui-g-12 ui-md-6 ui-lg-3 ui-sm-6">
      <p-dropdown *ngIf="semesters" [options]="semesters"
        [(ngModel)]="selectedSemester"
        (ngModelChange)="loadJournalForStudent()"
        placeholder="Select a semester" [showClear]="true"></p-dropdown>
    </div>
  </div>
  <div [fxLayout]="row">
    <p-accordion [activeIndex]="indexActive" [multiple]="true" class="accordion" fxFlex
      [fxLayout]="column">
      <div [fxLayout]="row" class="ui-g">
        <div [fxLayout]="column" class=" min-width ui-g-12 ui-md-6 ui-lg-3 ui-sm-6"
          *ngFor="let grade of grades; let i = index">
          <p-accordionTab header="{{ grade.label }}">
            <div fxLayout="row" fxLayoutAlign="center center">
              <google-chart [data]="data[i]"></google-chart>
            </div>
            <div [fxLayout]="row">
              <table class="darkTable">
                <thead>
                  <tr>
                    <th *ngFor="let head1 of cols">
                      <span>{{ head1 }}</span>
                    </th>
                  </tr>
                </thead>
                <tbody>
                  <tr>
                    <td>{{ grade.grades.sr.toFixed(1)}}</td>
                    <td>{{ grade.grades.rk1.toFixed(1)}}</td>
                    <td>{{ grade.grades.rk2.toFixed(1)}}</td>
                  </tr>
                </tbody>
                <thead>
                  <tr>
                    <th *ngFor="let head2 of cols2">
                      <span>
                        {{ head2 }}
                      </span>
                    </th>
                  </tr>
                </thead>
              </table>
            </div>
          </p-accordionTab>
        </div>
      </div>
    </p-accordion>
  </div>
</div>
```

```

        </th>
    </tr>
</thead>
<tbody>
<tr>
    <td>{{ grade.grades.rd.toFixed(1)}}</td>
    <td>{{ grade.grades.ex.toFixed(1)}}</td>
    <td>{{ grade.grades.sum.toFixed(1)}}</td>
</tr>
</tbody>
</table>
</div>
</p-accordionTab>
</div>
</div>
</p-accordion>
</div>
</div>
<!-- -->
//журнал учителя
<!-- TEACHER -->
<div [fxLayout]="column" *ngIf="teacher">

    <div *appScreenLarge [fxLayout]="row" class="space">

        <div [fxLayout]="column" class="year-col" style="margin: 10px">
            <p-dropdown *ngIf="years" [options]="years" [(ngModel)]=selectedYear
                placeholder="Select a Year"
                [showClear]="true"></p-dropdown>
        </div>

        <div [fxLayout]="column" class="semester-col" style="margin: 10px">
            <p-dropdown
                *ngIf="semesters"
                [(ngModel)]=selectedSemester
                [options]="semesters"
                (ngModelChange)="getDisciplines()"
                placeholder="Select a semester" [showClear]="true"></p-dropdown>
        </div>

        <div [fxLayout]="column" class="discipline-col" style="margin: 10px">
            <p-dropdown
                *ngIf="disciplines"
                [(ngModel)]=selectedDiscipline
                [options]="disciplines"
                (ngModelChange)="getGroup()"
                placeholder="Select a Discipline"
                optionLabel="name"
                [showClear]="true"></p-dropdown>
        </div>

        <div [fxLayout]="column" class="group-col" style="margin: 10px">
            <p-dropdown *ngIf="groups" [options]="groups" [(ngModel)]=selectedGroup
                (ngModelChange)="getJournal()"

```

```

placeholder="Select a Group" optionLabel="name" [showClear]="true"></p-dropdown>
  </div>
</div>
<div *appScreenSmall [fxLayout]="column" class="space">
  <div [fxLayout]="row" class="year-col">
    <p-dropdown *ngIf="years" [options]="years" [(ngModel)]="selectedYear"
      placeholder="Select a Year"
      [showClear]="true"></p-dropdown>
  </div>
  <div [fxLayout]="row" class="semester-col">
    <p-dropdown *ngIf="semesters" [options]="semesters"
      [(ngModel)]="selectedSemester"
      (ngModelChange)="getDisciplines()"
      placeholder="Select a semester" [showClear]="true"></p-dropdown>
  </div>
  <div [fxLayout]="row" class="discipline-col">
    <p-dropdown *ngIf="disciplines" [options]="disciplines"
      [(ngModel)]="selectedDiscipline"
      (ngModelChange)="getGroup()" placeholder="Select a Discipline"
      optionLabel="name"
      [showClear]="true"></p-dropdown>
  </div>
  <div [fxLayout]="row" class="group-col">
    <p-dropdown *ngIf="groups" [options]="groups" [(ngModel)]="selectedGroup"
      (ngModelChange)="getJournal()"
      placeholder="Select a Group" optionLabel="name" [showClear]="true"></p-
dropdown>
  </div>
</div>
<div *ngIf="studentGrades.length > 0" [fxLayout]="row" class="table-row">
  <p-table [value]="studentGrades" dataKey="id" editMode="row">
    <ng-template pTemplate="header">
      <tr>
        <td style="width:9em; margin-left: 5px"></td>
        <th scope="col">1</th>
        <th scope="col">2</th>
        <th scope="col">3</th>
        <th scope="col">4</th>
        <th scope="col">5</th>
        <th scope="col">6</th>
        <th scope="col">RK</th>
        <th scope="col">8</th>
        <th scope="col">9</th>
        <th scope="col">10</th>
        <th scope="col">11</th>
        <th scope="col">12</th>
        <th scope="col">13</th>
        <th scope="col">RK</th>

```

```

<th scope="col">RD</th>
  <th scope="col">EXAM</th>
</tr>

</ng-template>
<ng-template pTemplate="body" let-rowData let-ri="rowIndex">

  <tr>
    <th scope="row" style="padding-left: 20px">{{ rowData.fio }}</th>

    <td *ngFor="let grade of rowData.grades; let i = index" pEditableColumn>

      <p-cellEditor *ngIf="i !== rowData.grades.length-1">
        <ng-template pTemplate="input">
          <input pInputText type="number" [max]="100" [min]="0"
            [(ngModel)]="grade.grade"
              (blur)="saveGrade(grade)"
              [style]="{'width':'100%'}" required>
        </ng-template>
        <ng-template pTemplate="output">
          {{ grade.grade.toFixed(1) }}
        </ng-template>
      </p-cellEditor>
      <p-cellEditor *ngIf="i === rowData.grades.length-1">
        <ng-template pTemplate="input">
          <input pInputText type="number" [disabled]="true" [max]="100" [min]="0"
            [(ngModel)]="getRD(rowData.grades)"
              [style]="{'width':'100%'}" required>
        </ng-template>
        <ng-template pTemplate="output">
          {{ getRD(rowData.grades).toFixed(1) }}
        </ng-template>
      </p-cellEditor>
    </td>
  <td pEditableColumn>
    <p-cellEditor>
      <ng-template pTemplate="input">
        <input pInputText type="number" [max]="100" [min]="0"
          [(ngModel)]="rowData.grades[rowData.grades.length-1].grade"
            (blur)="saveGrade(rowData.grades[rowData.grades.length-1])"
            [style]="{'width':'100%'}" required>
      </ng-template>
      <ng-template pTemplate="output">
        {{ rowData.grades[rowData.grades.length - 1].grade.toFixed(1) }}
      </ng-template>
    </p-cellEditor>
  </td>
</tr>
</ng-template>

```



```
</p-table>
</div>
</div>
<!-- -->

//дизайн журнала
:host {
  .grade-option {
    margin: 5px;
    align-items: flex-end;
  }
  ::ng-deep .ui-accordion .ui-accordion-header:not(.ui-state-disabled).ui-state-active a {
    background-color: #303d46 !important;
  }
  .table-row {
    margin: 10px;
  }
  .min-width {
    min-width: 320px !important;
  }
  .accordion {
    width: 100%;
  }
  table.darkTable {
    font-family: "Arial Black", Gadget, sans-serif;
    border: 4px solid #000000;
    background-color: #ADADAD;
    width: 100%;
    height: 100%;
    text-align: center;
    border-collapse: collapse;
  }
  table.darkTable td, table.darkTable th {
    border: 6px solid #FBFFFC;
    padding: 10px 3px;
    max-width: 10px !important;
  }
  table.darkTable tbody td {
    font-size: 13px;
    font-weight: bold;
    color: #000000;
  }
  table.darkTable tr:nth-child(even) {
    background: #303d46;
  }
  table.darkTable thead {
    background: #303d46;
    border-bottom: 3px solid #000000;
  }
}
```

```
table.darkTable thead th {
  font-size: 15px;
  font-weight: bold;
  color: #E6E6E6;
  text-align: center;
}
}
//функции журнала
export class StudentGrades {
  id: number;
  fio: string;
  grades: Array<Grade> = new Array<Grade>();
}
export class GradeChart {
  sr: number;
  rk1: number;
  rk2: number;
  rd: number;
  ex: number;
  sum: number;
}
@Component({
  selector: 'app-journal',
  templateUrl: './journal.component.html',
  styleUrls: ['./journal.component.scss']
})
export class JournalComponent implements OnInit {
  studentGrades: Array<StudentGrades> = new Array<StudentGrades>();
  journal: Journal = new Journal();
  authInfo: AuthInfo;
  names: Array<string> = new Array<string>();
  chartGrades: GradeChartData[] = [];
  disciplines: Discipline[] = [];
  public selectedDiscipline: Discipline = new Discipline();
  groups: Group[] = [];
  public selectedGroup: Group = new Group();
  years: Year[] = [];
  semesters: Semester[] = [{label: '1', value: '1'}, {label: '2', value: '2'}];
  public selectedYear: string;
  public selectedSemester: string;
  teacher = false;
  student = false;
  studentsFilter: StudentJournalFilter;
  data: any[] = [];
  grades: GradeChartData[] = [];
  cols: any[] = [];
  cols2: any[] = [];
  filter: GradeChartFilter;
  indexActive = -1;
```

```
constructor(private service: JournalService,  
            private messageService: MessageService,  
            private authService: AuthorizationService) {  
  this.cols = [  
    'CP',  
    'PK2',  
    'PK2'  
  ];  
  
  this.cols2 = [  
    'РД',  
    'ЭКЗ',  
    'ИТОГ'  
  ];  
}  
async ngOnInit() {  
  const login = localStorage.getItem('login');  
  this.authService = await this.authService.getAuthInfo(login);  
  this.student = this.authService.privileges.includes('CAN_VIEW_JOURNAL');  
  this.teacher = this.authService.privileges.includes('CAN_MANAGE_JOURNAL');  
  const yearsList = await this.service.loadValidYears();  
  const date = await this.service.loadCurrentDate();  
  this.yearsInit(yearsList, date);  
}  
private yearsInit(yearsList: string[], date: Date) {  
  yearsList.forEach(year => {  
    this.years.push({label: year, value: year});  
  });  
  const currYear = new Date(date).getFullYear().toString();  
  this.selectedYear = currYear;  
}  
async getJournal() {  
  if (this.selectedGroup) {  
    this.studentsFilter = new StudentJournalFilter();  
    this.studentsFilter.discipline = this.selectedDiscipline.id;  
    this.studentsFilter.personGuid = this.authService.personGuid;  
    this.studentsFilter.groupId = this.selectedGroup.id;  
    this.studentsFilter.year = this.selectedYear;  
    this.studentsFilter.semester = this.selectedSemester;  
    this.journal = await this.service.loadJournal(this.studentsFilter);  
    this.journal.gradeList.forEach(x => {  
      if (!this.names.includes(x.fio)) {  
        this.names.push(x.fio);  
      }  
    });  
    // tslint:disable-next-line:prefer-for-of  
    for (let i = 0; i < this.names.length; i++) {  
      const sg: StudentGrades = new StudentGrades();  
      sg.id = i;  
    }  
  }  
}
```

```
sg.fio = this.names[i];
  this.journal.gradeList.forEach(x => {
    if (x.fio === this.names[i]) {
      sg.grades.push(x);
    }
  });
  this.studentGrades.push(sg);
}
} else {
  this.journal = null;
}
}
}
async getGroup() {
  if (this.selectedDiscipline) {
    this.groups = await this.service.getGroups(this.authInfo.personGuid,
this.selectedDiscipline.id);
  }
}
}
async getDisciplines() {
  if (this.selectedSemester && this.selectedYear) {
    console.log(this.selectedYear);
    console.log(this.selectedSemester);
    this.disciplines = await this.service.getDisciplines(this.authInfo.personGuid,
this.selectedYear, this.selectedSemester);
  }
}
}
async loadJournalForStudent() {
  await this.loadGrades();
  const source = timer(0, 60000);
  const abc = source.subscribe(async () => {
    this.filter = new GradeChartFilter();
    this.filter.personId = this.authInfo.personGuid;
    this.filter.year = this.selectedYear;
    this.filter.semester = this.selectedSemester;
    const temp = await this.service.loadGradeChartData(this.filter);
    this.grades = temp;
    temp.forEach(y => {
      this.data.forEach(x => {
        if (x.dataTable[1][0] === y.label) {
          x.dataTable[1][1] = y.grades.sum;
        }
      });
    });
  });
}
}
private async loadGrades() {
  this.filter = new GradeChartFilter();
  this.filter.personId = this.authInfo.personGuid;
  this.filter.year = this.selectedYear;
```

```
this.filter.semester = this.selectedSemester;
this.grades = await this.service.loadGradeChartData(this.filter);
this.grades.forEach(x => {
  this.data.push({
    chartType: 'Gauge',
    dataTable: [
      ['Label', 'Value'],
      [x.label, x.grades.sum],
    ],
    options: {
      title: 'Tasks',
      greenFrom: 75, greenTo: 100,
      yellowFrom: 50,
      yellowTo: 75,
      redFrom: 0, redTo: 50,
      max: 100
    },
  });
});
}
async saveGrade(grade: Grade) {
  console.log(grade);
  if (grade.grade) {
    await this.service.saveGrade(grade).then(() => {
      this.messageService.add({severity: 'success', summary: 'Success', detail: 'Grades is
updated'});
      this.getJournal();
    }).catch(resp => {
      this.messageService.add({severity: 'error', summary: 'Error', detail: resp.error});
    });
  }
}
getRD(grades: Grade[]): number {

  let arr = 0;
  let count = 0;
  let rk1 = 0;
  let rk2 = 0;

  grades.forEach(x => {
    if (x.type === GradeEnum.SIMPLE) {
      arr += x.grade;
      count++;
    }
    if (x.type === GradeEnum.FIRST_CONTROL) {
      rk1 = x.grade;
    }
    if (x.type === GradeEnum.SECOND_CONTROL) {
```

Продолжение приложения А

```
rk2 = x.grade;  
    }  
  });  
  const sr = arr / count;  
  
  return (((rk1 + rk2) / 2) * 0.2) + (sr * 0.8);  
}  
}
```