

ОГЛАВЛЕНИЕ

ВЕДОМОСТЬ ОБЪЕМА ДИПЛОМНОГО ПРОЕКТА	
ОГЛАВЛЕНИЕ	5
ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ	7
ВВЕДЕНИЕ	8
1 ОБЗОР СОСТОЯНИЯ ВОПРОСА	9
1.1 Обзор существующих аналогов	9
1.2 Обзор текущей версии системы CATS	11
1.3 Фреймворк Angular	14
1.4 Angular CLI	14
1.5 Краткий обзор используемых источников	15
2 ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ	17
2.1 Модель вариантов использования	17
2.1.1 Список действующих лиц	17
2.1.2 Список вариантов использования	17
2.1.3 Диаграммы вариантов использования	25
3 ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ	27
3.1 Взаимодействие компонентов распределенного приложения	27
3.2 Модель данных	28
4 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	30
4.1 Архитектура проекта	30
4.2 Модули	31
4.3 Компоненты	33
4.4 Шаблоны	36
4.5 Сервисы	36
4.6 Тестирование программного обеспечения	37
4.6.1 Используемые аппаратные средства	37
4.6.2 Критическое тестирование	38
4.6.3 Углубленное тестирование	40
4.6.4 Результаты тестирования	43
5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	45
5.1 Руководство пользователя «Студент»	45
5.2 Руководство пользователя «Преподаватель»	50
6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ	57
ВЕБ-ПРИЛОЖЕНИЯ	57
6.1 Определение единовременных затрат на создание программного продукта	57
6.1.1 Определение трудоемкости разработки программного продукта	57
6.1.2 Определение себестоимости создания программного продукта	59

	6
6.1.3 Определение оптовой и отпускной цены программного продукта	61
6.1.4 Определение стоимости машино-часа работы ЭВМ	62
6.1.5 Определение ожидаемого прироста прибыли в результате внедрения программного продукта.....	65
6.2.1 Определение годовых эксплуатационных расходов при ручном решении поставленной задачи	65
6.2.2 Определение годовых затрат, связанных с эксплуатацией задачи.....	66
6.2.3 Определение ожидаемого прироста прибыли в результате внедрения программного продукта.....	68
6.3 Расчёт показателей эффективности использования программного продукта	68
7 ОХРАНА ТРУДА.....	71
7.1 Производственная санитария, техника безопасности и пожарная профилактика ...	71
7.1.1 Метеоусловия	71
7.1.2 Вентиляция и отопление.....	73
7.1.3 Освещение.....	74
7.1.4 Шум.....	74
7.1.5 Электробезопасность.....	75
7.1.6 Излучение.....	75
7.1.7 Пожарная безопасность	77
7.2 Организация рабочего места пользователя ПЭВМ.....	78
ЗАКЛЮЧЕНИЕ	80
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	81
ГРАФИЧЕСКАЯ ЧАСТЬ	83
ПРИЛОЖЕНИЕ.....	92

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

В данной пояснительной записке используются следующие условные обозначения и термины:

- 1) CLI – интерфейс командной строки;
- 2) БД – база данных;
- 3) HTTP – протокол передачи гипертекста;
- 4) REST – передача состояния представления;
- 5) JSON – текстовый формат обмена данными, основанный на JavaScript;
- 6) URL – унифицированный указатель ресурса;
- 7) DOM – объектная модель документа;
- 8) HTML – язык разметки гипертекста;
- 9) CSS – каскадные таблицы стилей;
- 10) ЭВМ – электронно-вычислительная машина.

ВВЕДЕНИЕ

Любой учебный процесс включает в себя программу обучения и оценку полученных знаний. Обычно программа обучения и расписание занятий составляются в начале определенного периода в виде электронных таблиц и распространяются в электронном и печатном виде. Лабораторные работы зачастую сдаются преподавателям посредством внешних дисковых устройств, таких как USB-накопители. Тесты, как правило, печатаются на бумаге, на ней же печатаются и ведомости для выставления оценок.

Такой подход используется уже много лет и является вполне жизнеспособным, но при этом обладает рядом недостатков. Во-первых, расходуется очень много бумаги, что делает образовательную систему проблемой для экологии, и к тому же на бумагу расходуется часть бюджета. Во-вторых, любые изменения в программе обучения требуют распространения измененных копий среди студентов и преподавателей. Различные внештатные ситуации, например болезнь, могут приостановить учебный процесс.

Все эти проблемы можно решить с помощью программных средств управления образовательным процессом. Создав единую электронную систему мы можем сделать процесс обучения более удобным и прозрачным для всех. При этом все смогут получать необходимую информацию находясь в любом месте, и тогда, даже находясь, например, в больнице, студенты будут в курсе событий и даже смогут учиться. Поэтому подобная электронная система станет очень полезной, и её разработка является актуальной задачей.

Такая система уже разработана на базе нашего учебного заведения, и она доказала свою эффективность. Однако её можно улучшить. Нашей задачей является разработка нового веб-сайта с обновлённым пользовательским интерфейсом. Это сделает систему более простой и понятной для пользователя, а с новым дизайном она будет выглядеть красиво и современно.

1 ОБЗОР СОСТОЯНИЯ ВОПРОСА

Автоматизированные системы управления обучением являются основой учебного процесса и используются для организации и проведения аудиторных и дистанционных занятий, разработки, управления и распространения учебных материалов с обеспечением совместного доступа. Создаются данные материалы в визуальной учебной среде с заданием последовательности изучения. В состав системы, как правило, входят различного рода индивидуальные задания, проекты для работы в малых группах и учебные элементы для всех студентов, основанные как на содержательной компоненте, так и на коммуникативной. В русскоязычной литературе в качестве синонима LMS встречаются термины СДО – система дистанционного обучения, а также СУО – система управления обучением. В настоящее время существует множество систем управления обучением, причем каждая из них имеет как очевидные преимущества, так и недостатки, обусловленные разработками программных продуктов под определенные цели [2].

Нашей задачей будет обновление части интерфейса для управления курсовыми проектами.

Основные требования к новому веб-приложению:

- оно должно быть одностраничным, то есть при выборе другого пункта меню страница в браузере не должна меняться и перезагружаться, при этом заголовок сайта и меню должны быть статичными, и заменяться должен компонент страницы, содержащий элементы в соответствии с выбранным пунктом меню;
- интерфейс приложения должен быть интуитивно понятным и дружелюбным;
- содержимое компонентов должно зависеть от роли текущего пользователя.

1.1 Обзор существующих аналогов

Анализ наиболее распространенных LMS (ePathLearning, CourseWork, Moodle, Sakai, E-xcellence, SharepointLMS, BlackBoard, Claroline, Dokeos, LAMS, Learn eXact, e-University, Eucalyptus, Desire2Learn, Edmodo, Education Elements, OpenClass, Schoology, Haiku learning, iSpring, Ilias, Odijoo, Scorm Cloud, Dnevnik.ru, MoyUniver.ru, Yaklass.ru, Coursera, ATutor, WebTutor, Efront и др.) позволяет сделать вывод о многообразии предлагаемых функциональных возможностей, о разных подходах к реализации, о возможности индивидуальной компоновки или даже разработки LMS, о разных схемах лицензирования и стоимости. Все это свидетельствует о том, что классификацию таких систем необходимо проводить по определенным признакам, в рамках которых будут рассматриваться однотипные системы. Набор требуемых для конкретного

пользователя признаков сформирует критерии выбора наилучшей для него системы [1].

Вследствие специфики системы образования, отличающегося от более индивидуального западного, а также по стоимостным характеристикам, в высших учебных заведениях Республики Беларусь наибольшее распространение получили системы Moodle, «Прометей», e-University (нынешнее название e-Uni) и SharePoint LMS. Moodle (англ., Modular Object-Oriented Dynamic Learning Environment), известная как виртуальная обучающая среда или как средство создания динамических веб-сайтов для учащихся, ориентирована, прежде всего, на организацию взаимодействия между преподавателем и учениками, подходит для организации традиционных дистанционных курсов и поддержки очного обучения. LMS Moodle – свободно распространяемый программный продукт с открытым исходным кодом. Этот факт одновременно является и преимуществом этой системы, и ее недостатком, поскольку, с одной стороны, позволяет создавать любые учебные курсы и занятия, а с другой – требует особых знаний в их разработке. Вследствие чего систему довольно сложно настроить, а преподавателю, не владеющему основами программирования, не просто разобраться. Для обучения работе с Moodle даже существуют учебники, что еще раз подчеркивает нелегкость в ее освоении. Также следует отметить недружественный интерфейс системы, перегруженность ссылками и множество лишней информации.

LMS «Прометей» является российской разработкой и позволяет организовать учебные курсы, учебные материалы, тесты с 10 видами вопросов, чат, форум, сообщения и многое др. К сожалению, в системе не предусмотрена возможность поддержки курсового и дипломного проектирования, а также отсутствует адаптивность обучения к текущим знаниям студента и его психофизиологическим особенностям. LMS e-University реализована белорусскими разработчиками и внедрена на нескольких факультетах БГУ. В системе предусмотрено создание учебных курсов и тестов по ним. Однако вследствие того, что система уже не поддерживается разработчиком, ее интерфейс и набор функциональности значительно устарели. В настоящее время на замену этому программному продукту вышла система e-Uni, более ориентированная на корпоративный сектор. SharePoint – это коллекция программных продуктов и компонентов от компании Microsoft для разработки сайтов. Таким образом, разработка LMS с использованием данных продуктов требует отдельных навыков и умений, которыми преподаватели вряд ли обладают. Этим моментом воспользовались некоторые разработчики сайтов и предлагают свои услуги с использованием SharePoint. Поэтому полнота функциональности разработанных таким образом LMS будет зависеть от фантазии разработчика, а легкость настройки – от его мастерства. Следует также добавить, что описанные выше системы «Прометей», e-University и SharePoint LMS являются платными разработками и обладают одинаковыми недостатками: высокой стоимостью и отсутствием гибкости, т. е.

купленный продукт – это «вещь в себе» и не подлежит самостоятельной модификации, а лишь за дополнительную оплату [3].

1.2 Обзор текущей версии системы CATS

Для начала необходимо проанализировать существующую систему, чтобы решить, что можно в ней улучшить.

Данное веб-приложение разработано на базе платформы ASP.NET MVC и CSS-фреймворка Bootstrap.

Платформа ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. В зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) – это визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

Схема взаимодействия компонентов паттерна MVC изображена на рисунке 1.1.

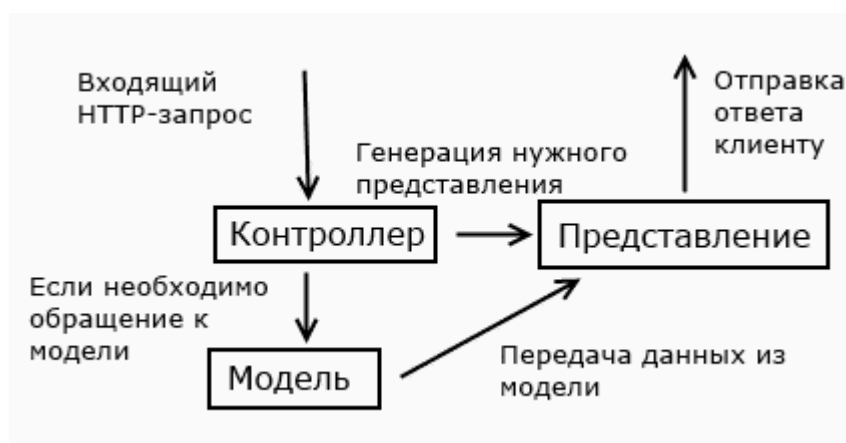


Рисунок 1.1 – Схема паттерна MVC

Можно выделить следующие недостатки данной концепции:

1) Усложнен механизм разделения программы на модули. В концепции MVC наличие трех блоков (Model, View, Controller) прописано жестко. Соответственно каждый функциональный модуль должен состоять из трех блоков, что в свою очередь,

несколько усложняет архитектуру функциональных модулей программы.

2) Усложнен процесс расширения функционала. Проблема очень схожа с вышеописанной. Недостаточно просто написать функциональный модуль и подключить его в одном месте программы. Каждый функциональный модуль должен состоять из трех частей, и каждая из этих частей должна быть подключена в соответствующем блоке.

3) Отсутствие механизма хранения состояния. MVC Framework не предлагает механизма хранения состояния, и разработчику необходимо реализовывать его самостоятельно, используя скрытые поля, cookie-файлы или хранение данных в URL.

4) Сложности создания библиотек компонентов. Отсутствие механизма элементов управления, присутствующего в WebForms, затрудняет создание компонентов для повторного использования. При существующем подходе к созданию представлений инкапсуляция полной логики повторно используемого компонента в отдельную сборку затруднена.

Интерфейс системы представлен на рисунке 1.2.

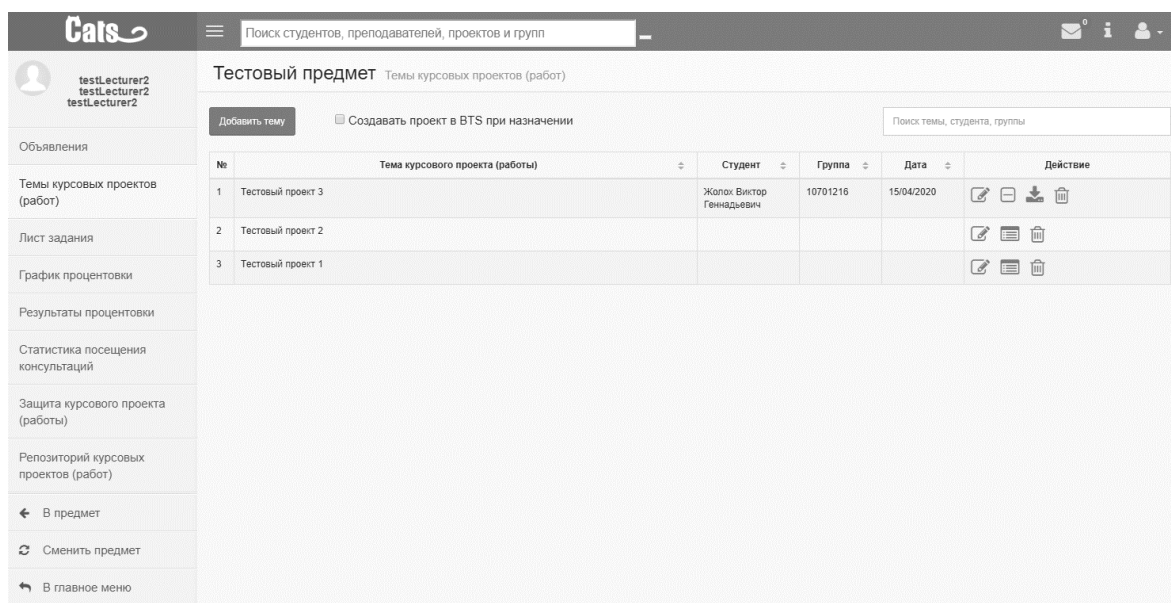


Рисунок 1.2 – Интерфейс системы

Сразу же следует отметить, что дизайн является слегка устаревшим. Гораздо лучше будет выглядеть интерфейс, выполненный в концепции Material Design [4], разработанной компанией Google.

Интерфейс системы является адаптивным, например, при изменении размеров окна браузера некоторые элементы меняют свое положение, тем самым сохраняя пользовательский интерфейс приемлемым и пригодным для использования. Однако некоторые элементы всё же плохо подстраиваются под размер экрана. На рисунке 1.3 видно, что часть интерфейса «уехала» за пределы окна.

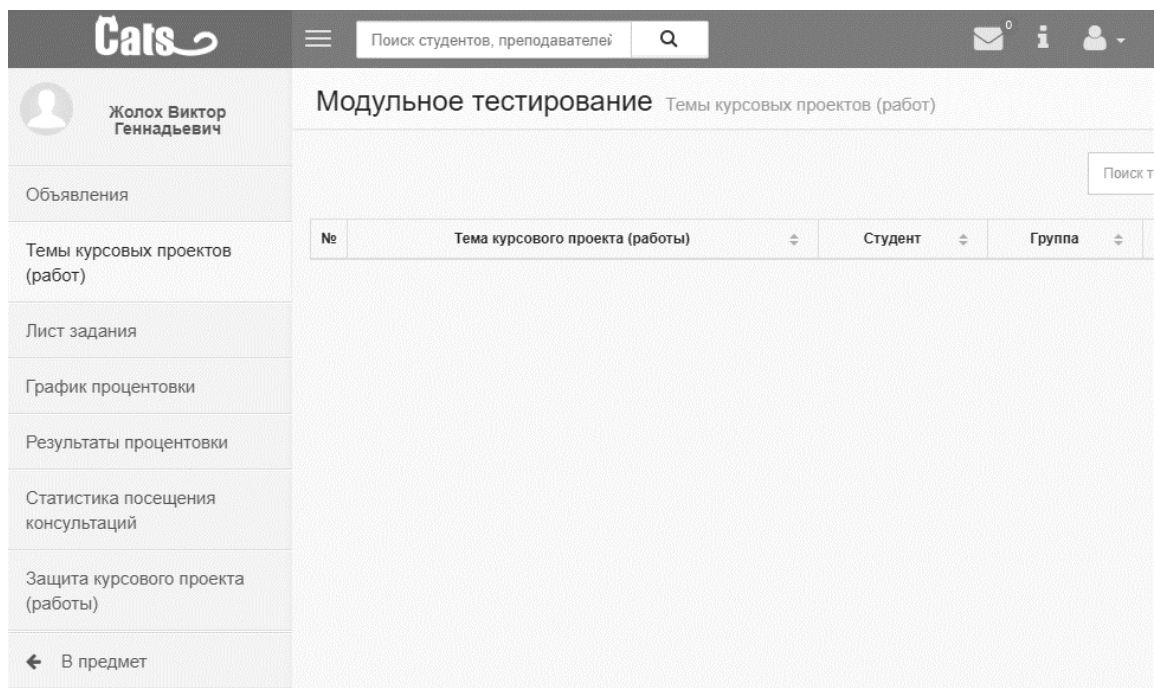


Рисунок 1.3 – Проблема адаптивности интерфейса

В меню управления курсовыми проектами представлены следующие пункты:

- объявления;
- темы курсовых проектов (работ);
- лист задания;
- график процентки;
- результаты процентки;
- статистика посещения консультаций;
- защита курсового проекта (работы);
- репозиторий курсовых проектов (работ).

В новой версии интерфейса в отношении данного меню было решено внести следующие нововведения:

- 1) Убрать из названий пунктов пояснение в скобках, к примеру пункт «Темы курсовых проектов (работ)» будет называться «Темы курсовых проектов».
- 2) Убрать пункт «Объявления», так как теперь объявления курсового проекта будут отображаться в разделе управления предметом.
- 3) На странице «Результаты процентки» при выставлении результата в новой версии нужно будет кроме процента выполнения указывать еще и дату выставления оценки, а также можно будет написать комментарий.

1.3 Фреймворк Angular

В качестве фреймворка, на котором будет построен новый веб-интерфейс был выбран Angular.

Angular представляет фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время Angular это не новая версия AngularJS, а принципиально новый фреймворк.

Почему был выбран данный фреймворк:

1) Angular предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

2) Material Design. Компоненты, реализующие данную концепцию дизайна доступны «из коробки» в пакете @angular/material.

3) Надежность. В связи с тем, что Angular – это продукт Google, он использует все преимущества его инфраструктуры тестирования. Каждое изменение в Angular проверяется на соответствие с каждым проектом Angular в пределах Google. Это значит, что еще до любого публичного релиза фреймворк уже используют сотни проектов, а такой подход увеличивает шансы на отсутствие непредусмотренных критических изменений или регрессий.

4) Angular имеет мощную поддержку. В интернете существует множество многократно используемых инструментов, библиотек и примеров кода для Angular, и большое количество этих инструментов либо были оптимизированы под работу с Angular, либо сейчас находятся в процессе оптимизации.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript. При желании можем писать приложения на Angular с помощью таких языков как Dart или JavaScript. Однако TypeScript все-таки является основным языком для Angular.

1.4 Angular CLI

Angular CLI – официальный инструмент для инициализации и работы с Angular-проектами. Он позволяет значительно упростить разработку приложения.

CLI можно использовать для быстрой генерации Angular-проекта, выполнив следующую команду в интерфейсе командной строки: `ng new <название_проекта>`. После этого будет создан проект со структурой каталогов и набором конфигураций и файлов кода.

Angular CLI предоставляет полную цепочку инструментов для разработки приложений на локальном компьютере. Таким образом, не нужно устанавливать локальный сервер для обслуживания проекта - можно просто использовать команду `ng serve`.

Angular CLI предоставляет команду `ng generate`, которая помогает разработчикам создавать базовые артефакты Angular, такие как модули, компоненты, директивы, пайпы и сервисы.

Другие команды Angular CLI, которые могут пригодиться при разработке:


- `add`: добавляет поддержку внешней библиотеки в проект;
- `build (b)`: компилирует приложение Angular в каталог с именем `dist/` по указанному пути;
- `config`: извлекает или устанавливает значения Angular конфигурации;
- `doc (d)`: открывает официальную документацию Angular (angular.io) в браузере и выполняет поиск по заданному ключевому слову;
- `e2e (e)`: создает и обслуживает приложение Angular, затем запускает сквозные тесты с использованием Protractor;
- `help`: список доступных команд и их краткое описание;
- `lint (l)`: запускает инструменты проверки кода приложения Angular в данной папке проекта;
- `run`: запускает пользовательскую цель, определенную в проекте;
- `test (t)`: запуск модульных тестов в проекте;
- `update`: обновляет приложение и его зависимости;
- `version (v)`: вывод Angular версии CLI;
- `x18n`: извлекает `i18n` из исходного кода.

1.5 Краткий обзор используемых источников

Главным источником является официальная документация фреймворка Angular на английском языке [5]. Здесь содержится вся необходимая для разработчика информация: инструкция по установке и запуску, описание доступных интерфейсов, примеры использования различных интерфейсов и компонентов (рисунок 1.4).

A

Search

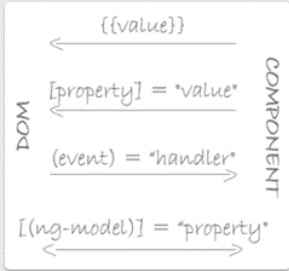


Data binding

Without a framework, you would be responsible for pushing data values into the HTML controls and turning user responses into actions and value updates. Writing such push and pull logic by hand is tedious, error-prone, and a nightmare to read, as any experienced front-end JavaScript programmer can attest.

Angular supports *two-way data binding*, a mechanism for coordinating the parts of a template with the parts of a component. Add binding markup to the template HTML to tell Angular how to connect both sides.

The following diagram shows the four forms of data binding markup. Each form has a direction: to the DOM, from the DOM, or both.



The diagram illustrates four forms of data binding between the DOM and a Component:

- Interpolation:** A single-headed arrow points from the Component to the DOM with the markup `{{value}}`.
- Property Binding:** A single-headed arrow points from the Component to the DOM with the markup `[property] = "value"`.
- Event Binding:** A single-headed arrow points from the DOM to the Component with the markup `(event) = "handler"`.
- Two-Way Binding:** A double-headed arrow connects the DOM and the Component with the markup `[(ng-model)] = "property"`.

This example from the `HeroListComponent` template uses three of these forms.

src/app/hero-list.component.html (binding)

```
<li>{{hero.name}}</li>
<app-hero-detail [hero]="selectedHero"></app-hero-detail>
<li (click)="selectHero(hero)"></li>
```

Рисунок 1.4 – Веб-страница официальной документации Angular

Источник [6] содержит документацию по material-компонентам, которые содержатся в пакете @angular/material. Здесь перечислены все доступные компоненты, их интерфейсы и большое количество полезных примеров.

Концепция Material Design подробно описана в источнике [4]. Здесь находятся инструкции для разработчиков, которым необходимо следовать, чтобы данная концепция была соблюдена.

2 ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

На этапе логического моделирования необходимо определить роли пользователей системы и основные сценарии её использования.

2.1 Модель вариантов использования

На диаграммах вариантов использования отображается взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему. Из диаграмм вариантов использования можно получить довольно много информации о системе. Этот тип диаграмм описывает общую функциональность системы.

Пользователи, менеджеры проектов, аналитики, разработчики, специалисты по контролю качества и все, кого интересует система в целом, могут, изучая диаграммы вариантов использования, понять, что система должна делать.

Достоинствами модели вариантов использования являются:

- определение пользователей и границы системы;
- эффективность общения заказчика и разработчика;
- использование при написании тестов;
- использование при написании пользовательской документации.

2.1.1 Список действующих лиц

В диаграммах вариантов использования пользователи, которые взаимодействуют с системой, называются акторами.

При анализе работы с курсовыми проектами были выделены следующие действующие лица:

- студент (авторизованный пользователь, использует систему для получения необходимой информации и для защиты курсового проекта);
- преподаватель (авторизованный пользователь, занимается распределением тем курсовых проектов и аттестацией студентов).

2.1.2 Список вариантов использования

Сценарий использования (вариант использования, прецедент использования) определяет взаимодействия между внешними агентами и системой, направленные на

достижение цели. Актор представляет собой роль, которую играет человек или вещь, взаимодействуя с системой.

Сценарий использования должен:

- описывать, что именно нужно сделать, чтобы актер достиг своей цели;
- не затрагивать деталей реализации;
- иметь достаточный уровень детализации.

В сценариях будут также определены требования к данным, поступающим извне.

В таблице 2.1 перечислены варианты использования для роли «Студент».

Таблица 2.1 – Варианты использования для роли «Студент»

UC1	
Название	Авторизация в роли студента
Роль	Студент
Цель	Авторизоваться в системе как студент
Результат	Пользователь авторизован и наделен правами студента
Триггер	Пользователь зашел в систему и нажал на кнопку "Начать работу"
Главный поток	1. Пользователь вводит логин и пароль учителя. Ограничения: - логин(3-30 символов, латинские буквы, цифры и знак подчеркивания), - пароль(6-30 символов, латинские буквы, цифры и знак подчеркивания). 2. Логин и пароль верно отображаются. Пароль должен отображаться в виде точек и того количества символов, сколько ввёл пользователь. 3. Пользователь нажимает на кнопку "Вход". 4. Логин и пароль верны, пользователь авторизован.
Альтернативные потоки	4.1 Логин или пароль неверны, пользователь не авторизован. 4.2 Появляется сообщение о том, что пароль или логин неправильные. Пользователь снова вводит логин и пароль.
UC2	
Название	Выбор проекта в пункте меню "Курсовые проекты"
Роль	Студент
Цель	Зайти в курсовой проект и получить доступ к функциям этого раздела
Результат	Пользователь получил доступ к функциям выбранного проекта
Триггер	Пользователь авторизовался как студент
Главный поток	1. Пользователь выбирает в меню "Курсовые проекты". 2. Во всплывающем подменю пользователь выбирает нужный проект. 3. Разделы (темы курсовых проектов, лист задания, график процентовки, результаты процентовки, статистика посещения консультаций, защита курсового проекта) верно отображаются.

Продолжение таблицы 2.1

UC3/1	
Название	Выбор темы курсового проекта
Роль	Студент
Цель	Выбрать тему курсового проекта
Результат	Тема выбрана и ожидает подтверждения преподавателем
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	1. Пользователь нажимает на кнопку "Выбрать тему" для определенного элемента в списке тем. 2. Открывается диалоговое окно для подтверждения операции. 3. Пользователь нажимает на кнопку "Выбрать". 4. Для выбранной темы в столбце Студент отображается ФИО пользователя.
Альтернативные потоки	3.1 Пользователь нажимает "Отмена". 4.1 Тема курсового проекта остается неизменной.
UC3/2	
Название	Скачать лист задания
Роль	Студент
Цель	Скачать сгенерированный лист задания как файл Word
Результат	Лист задания сохранен на выбранный диск
Триггер	Пользователь зашел в раздел Темы курсовых проектов или в раздел Лист задания
Главный поток	1. Пользователь нажимает на кнопку "Скачать лист задания". 2. Открывается системное диалоговое окно для выбора места, куда будет сохранен файл. 3. Пользователь выбирает нужный путь и нажимает на кнопку "Сохранить". 4. Файл сохранен в выбранное место.
Альтернативные потоки	3.1 Пользователь нажимает "Отмена". 4.1 Пользователь перенаправлен обратно на страницу Темы курсовых проектов или Лист задания.
UC4	
Название	Добавление работы
Роль	Студент
Цель	Добавить готовый курсовой проект на защиту
Результат	Работа успешно добавлена на защиту
Триггер	Пользователь зашел в раздел Защита курсового проекта
Главный поток	1. Пользователь нажимает на кнопку "Добавить работу". 2. Открывается форма с полем Комментарий(1-255 символов) и кнопкой "Добавить файл"(можно добавить только один файл). 3. Пользователь верно заполняет поле и прикрепляет файл. 4. Пользователь нажимает на кнопку "Добавить работу". 5. Работа отображается в списке.
Альтернативные потоки	3.1 Пользователь неправильно заполняет поля. 3.2 Кнопка "Добавить работу" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.

В таблице 2.2 перечислены варианты использования для роли «Преподаватель».

Таблица 2.2 – Варианты использования для роли «Преподаватель»

UC5/1	
Название	Добавление темы курсового проекта
Роль	Преподаватель
Цель	Добавить тему курсового проекта
Результат	Тема курсового проекта успешно добавлена
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	<ol style="list-style-type: none"> 1. Пользователь нажимает на кнопку "Добавить тему". 2. Открывается форма с полем Тема курсового проекта(обязательное поле, 3-255 символов) и чекбоксами для выпора студенческих групп. 3. Пользователь верно заполняет поле и выбирает группы. 4. Пользователь нажимает на кнопку "Сохранить". 5. Новая тема отображается в списке.
Альтернативные потоки	<ol style="list-style-type: none"> 3.1 Пользователь неправильно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC5/2	
Название	Редактирование темы курсового проекта
Роль	Преподаватель
Цель	Редактировать тему курсового проекта
Результат	Тема курсового проекта успешно отредактирована, изменения отображаются в списке
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	<ol style="list-style-type: none"> 1. Пользователь нажимает на кнопку "Редактировать" для определенной темы. 2. Открывается форма с теми же полями, что и в UC5/1. Поля заполнены значениями соответствующей темы. 3. Пользователь верно заполняет поле и выбирает группы. 4. Пользователь нажимает на кнопку "Сохранить". 5. Отредактированная тема отображается в списке.
Альтернативные потоки	<ol style="list-style-type: none"> 3.1 Пользователь неправильно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC5/3	
Название	Удаление темы курсового проекта
Роль	Преподаватель
Цель	Удалить тему курсового проекта
Результат	Тема курсового проекта успешно удалена
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	<ol style="list-style-type: none"> 1. Пользователь нажимает на кнопку "Удалить" для определенной темы. 2. Открывается диалоговое окно для подтверждения операции. 3. Пользователь нажимает на кнопку "Удалить". 4. Удаленная тема не отображается в списке.

Продолжение таблицы 2.2

Альтернативные потоки	3.1 Пользователь нажимает на кнопку "Отмена". 4.1 Тема по-прежнему отображается в списке.
UC5/4	
Название	Назначение темы курсового проекта
Роль	Преподаватель
Цель	Назначить тему студенту
Результат	Тема курсового проекта назначена выбранному студенту
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	1. Пользователь нажимает на кнопку "Назначить" для определенной темы. 2. Открывается диалоговое окно со списком студентов, которым можно назначить данную тему. 3. Пользователь нажимает на кнопку "Назначить" для определенного студента. 4. Тема назначена, и для нее в списке в столбце Студент указаны ФИО выбранного студента.
Альтернативные потоки	3.1 Пользователь нажимает на кнопку "Отмена". 4.1 Тема не назначена.
UC5/5	
Название	Отмена назначения темы курсового проекта
Роль	Преподаватель
Цель	Отменить назначение темы
Результат	Тема не назначена студенту
Триггер	Пользователь зашел в раздел Темы курсовых проектов
Главный поток	1. Пользователь нажимает на кнопку "Отменить назначение" для определенной темы. 2. Открывается диалоговое окно для подтверждения операции. 3. Пользователь нажимает на кнопку "Удалить". 4. Тема становится неназначенной, и столбец Студент становится пустым для данной темы.
Альтернативные потоки	3.1 Пользователь нажимает на кнопку "Отмена". 4.1 Тема остается назначенной.
UC6/1	
Название	Редактирование листа задания
Роль	Преподаватель
Цель	Редактировать лист задания
Результат	Лист задания успешно отредактирован
Триггер	Пользователь зашел в раздел Лист задания

Продолжение таблицы 2.2

Главный поток	<p>1. Пользователь выбирает из выпадающего списка тему курсового проекта.</p> <p>2. Пользователь нажимает на кнопку "Редактировать".</p> <p>3. Открывается форма с выпадающим списком выбора шаблона, полем Название шаблона(1-30) и вкладками: Входные данные(поле 0-999), Содержание(поле 0-999), Графические материалы(поле 0-999), Сведения о ВУЗе(поля Название учебного заведения(0-255); Название факультета(0-255); Фамилия, инициалы заведующего кафедры(0-30)), Даты(поля Дата выдачи задания(ДД.ММ.ГГГГ), Сроки сдачи курсового проекта(ДД.ММ.ГГГГ)).</p> <p>4. Пользователь верно заполняет все поля и нажимает кнопку "Сохранить".</p> <p>5. Лист задания успешно отредактирован.</p>
Альтернативные потоки	<p>3.1 Пользователь неверно заполняет поля.</p> <p>3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.</p>
UC6/2	
Название	Редактирование листа задания на основе шаблона
Роль	Преподаватель
Цель	Заполнить поля формы редактирования листа задания из сохраненного шаблона
Результат	Поля заполнены значениями из шаблона
Триггер	Пользователь нажал кнопку "Редактировать" в разделе "Лист задания"
Главный поток	<p>1. Пользователь выбирает шаблон из выпадающего списка.</p> <p>2. Поля формы заполняются значениями из шаблона.</p>
UC6/3	
Название	Сохранение шаблона
Роль	Преподаватель
Цель	Сохранить шаблон с данными формы
Результат	Шаблон успешно сохранен
Триггер	Пользователь нажал кнопку "Редактировать" в разделе "Лист задания"
Главный поток	<p>1. Пользователь верно заполняет поля формы.</p> <p>2. Пользователь верно заполняет поле Название шаблона.</p> <p>3. Пользователь нажимает на кнопку "Сохранить шаблон"</p> <p>4. Шаблон успешно сохранен.</p>
Альтернативные потоки	<p>3.1 Пользователь неверно заполняет поля.</p> <p>3.2 Кнопка "Сохранить шаблон" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.</p>
UC7/1	
Название	Добавление этапа процентовки
Роль	Преподаватель
Цель	Добавить этап процентовки
Результат	Этап процентовки успешно добавлен

Продолжение таблицы 2.2

Триггер	Пользователь зашел в раздел График процентовки
Главный поток	1. Пользователь нажимает на кнопку "Добавить этап". 2. Открывается форма с полями Название этапа(обязательное поле, 3-100), Процент выполнения(обязательное поле, целое число от 0 до 100), Дата(обязательное поле, ДД.ММ.ГГГГ). 3. Пользователь верно заполняет все поля. 4. Пользователь нажимает на кнопку "Сохранить". 5. Этап успешно сохранен и отображается в списке.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC7/2	
Название	Редактирование этапа процентовки
Роль	Преподаватель
Цель	Редактировать этап процентовки
Результат	Этап процентовки успешно отредактирован
Триггер	Пользователь зашел в раздел График процентовки
Главный поток	1. Пользователь нажимает на кнопку "Редактировать" для определенного этапа. 2. Открывается такая же форма, как и в UC7/1. Поля формы заполнены значениями данного этапа. 3. Пользователь верно заполняет все поля. 4. Пользователь нажимает на кнопку "Сохранить". 5. Этап успешно отредактирован и отображается в списке.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC7/3	
Название	Удаление этапа процентовки
Роль	Преподаватель
Цель	Удалить этап процентовки
Результат	Этап процентовки успешно удален
Триггер	Пользователь зашел в раздел График процентовки
Главный поток	1. Пользователь нажимает на кнопку "Удалить" для определенного этапа. 2. Открывается диалоговое окно для подтверждения операции. 3. Пользователь нажимает на кнопку "Удалить". 4. Этап процентовки удален и не отображается в списке.
Альтернативные потоки	3.1 Пользователь нажимает на кнопку "Отмена". 4.1 Этап процентовки не удален и отображается в списке.
UC8/1	
Название	Редактирование результата процентовки
Роль	Преподаватель
Цель	Редактировать результат процентовки
Результат	Результат процентовки успешно изменен
Триггер	Пользователь зашел в раздел Результаты процентовки

Продолжение таблицы 2.2

Главный поток	1. Пользователь нажимает на ячейку в таблице результатов процентовки. 2. Открывается форма с полями Результат(целое число от 0 до 100) и Комментарий(0-255). 3. Пользователь верно заполняет поля. 4. Пользователь нажимает на кнопку "Сохранить". 5. Результат процентовки сохранен и отображается в таблице.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC8/2	
Название	Редактирование оценки
Роль	Преподаватель
Цель	Редактировать оценку
Результат	Оценка успешно изменена
Триггер	Пользователь зашел в раздел Результаты процентовки
Главный поток	1. Пользователь нажимает на ячейку в столбце Оценка. 2. Открывается форма с полем Оценка(обязательное поле, целое число от 1 до 10). 3. Пользователь верно заполняет поля. 4. Пользователь нажимает на кнопку "Сохранить". 5. Оценка сохранена и отображается в таблице.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC9/1	
Название	Добавление даты консультации
Роль	Преподаватель
Цель	Добавить дату консультации
Результат	Дата консультации успешно добавлена
Триггер	Пользователь зашел в раздел Статистика посещения консультаций
Главный поток	1. Пользователь нажимает на кнопку "Добавить дату консультации". 2. Открывается форма с полем Дата консультации(обязательное поле, ДД.ММ.ГГГГ). 3. Пользователь верно заполняет поле. 4. Пользователь нажимает на кнопку "Сохранить". 5. Дата консультации добавлена и отображается как столбец таблицы.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC9/2	
Название	Удаление даты консультации
Роль	Преподаватель
Цель	Удалить дату консультации

Продолжение таблицы 2.2

Результат	Дата консультации успешно удалена
Триггер	Пользователь зашел в раздел Статистика посещения консультаций
Главный поток	1. Пользователь нажимает на заголовок столбца таблицы с датой консультации. 2. Открывается диалоговое окно для подтверждения операции. 3. Пользователь нажимает на кнопку "Удалить". 4. Дата консультации удалена и не отображается в таблице.
Альтернативные потоки	3.1 Пользователь нажимает на кнопку "Отмена". 4.1 Дата консультации не удалена и отображается в таблице.
UC9/3	
Название	Редактирование посещаемости консультации
Роль	Преподаватель
Цель	Редактировать посещаемость консультации
Результат	Посещаемость успешно отредактирована
Триггер	Пользователь зашел в раздел Статистика посещения консультаций
Главный поток	1. Пользователь нажимает на ячейку в таблице посещения консультаций. 2. Открывается форма с полями Пропущено часов(целое положительное число) и Комментарий(0-255). 3. Пользователь верно заполняет поля. 4. Пользователь нажимает на кнопку "Сохранить". 5. Посещаемость сохранена и отображается в таблице.
Альтернативные потоки	3.1 Пользователь неверно заполняет поля. 3.2 Кнопка "Сохранить" неактивна, под неправильно заполненными полями появляется сообщение с предупреждением.
UC10	
Название	Скачать курсовые проекты архивом
Роль	Преподаватель
Цель	Скачать архив курсовых работ
Результат	Архив с курсовыми работами сохранен на выбранный диск
Триггер	Пользователь зашел в раздел Защита курсового проекта
Главный поток	1. Пользователь нажимает на кнопку "Скачать архивом". 2. Открывается системное диалоговое окно для выбора места, куда будет сохранен файл. 3. Пользователь выбирает нужный путь и нажимает на кнопку "Сохранить". 4. Файл сохранен в выбранное место.
Альтернативные потоки	3.1 Пользователь нажимает "Отмена". 4.1 Пользователь перенаправлен обратно на страницу Защита курсового проекта.

2.1.3 Диаграммы вариантов использования

Информация в графическом виде воспринимается гораздо проще, чем в текстовом, поэтому для сложных вариантов использования, связанных с защитой курсового проекта, разработаем соответствующую диаграмму (рисунок 2.1).

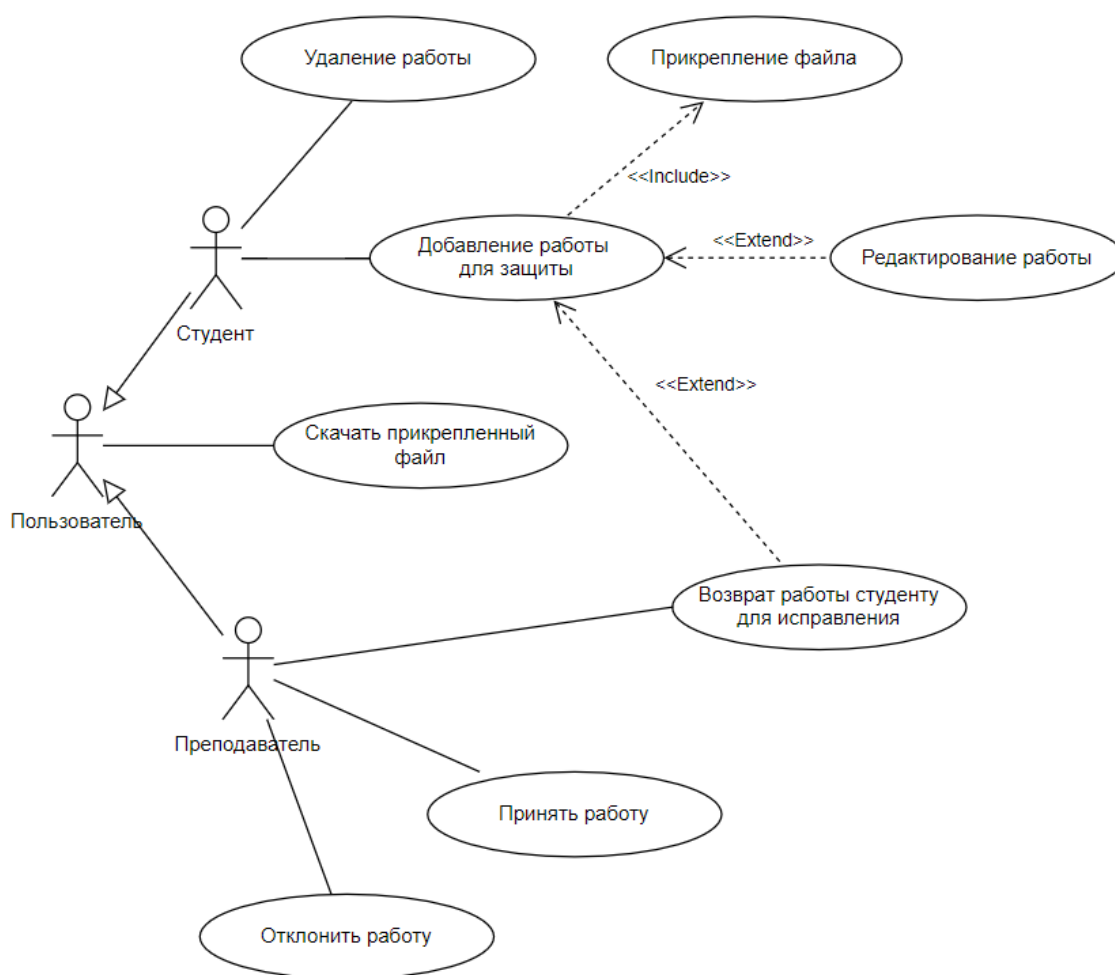


Рисунок 2.1 – Диаграмма вариантов использования прецедента «Защита курсового проекта»

Процесс защиты курсового проекта начинается с добавления студентом своей работы. Для этого он должен прикрепить файл и написать комментарий к своей работе. После добавления работы, как студент, так и преподаватель могут скачать прикрепленный файл. Студент может редактировать свою работу (изменить комментарий или прикрепить другой файл), а также удалить её.

На диаграмме между прецедентами «Возврат работы студенту для исправления» и «Добавление работы для защиты» существует связь «Наследование». Дело в том, что для того, чтобы вернуть работу, преподаватель также обязательно должен прикрепить файл и написать комментарий. В итоге работа студента автоматически удаляется, и вместо неё появляется файл с комментарием от преподавателя. После этого студент должен снова добавить работу для защиты, при этом работы, добавленные преподавателем, не удаляются.

Если преподаватель принимает работу студента, то она отправляется в архив, при этом преподаватель всегда может отклонить работу, если передумает.

3 ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ

3.1 Взаимодействие компонентов распределенного приложения

Приложение-сервер разработано с использованием технологии ASP.NET и использует базу данных Microsoft SQL Server. Оно предоставляет нам внешний интерфейс в виде веб-сервисов, использующих протокол HTTP. Разработанное нами приложение-клиент будет посылать запросы этим сервисам. Такая архитектура взаимодействия компонентов распределенного приложения называется REST (от англ. Representational State Transfer – «передача состояния представления»). Схема архитектуры представлена на рисунке 3.1.

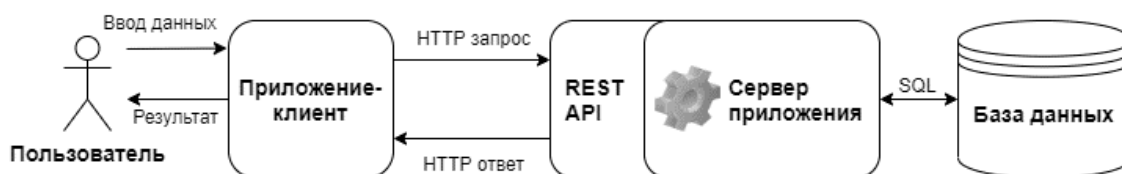


Рисунок 3.1 – REST архитектура

REST используется для создания легковесных, поддерживаемых и масштабируемых веб-сервисов. Сервис, построенный на REST архитектуре, называется RESTful-сервисом.

REST – это способ получить доступ к ресурсам, которые находятся в определённой среде. Например, у вас может быть сервер с важными документами или фотографиями. Всё это – ресурсы. Если клиенту, скажем, веб-браузеру, нужны какие-то из этих ресурсов, ему необходимо отправить запрос на сервер для получения доступа к ним. REST определяет, как может осуществляться доступ к этим ресурсам.

Ключевые составляющие реализации RESTful:

- Ресурсы. Допустим, у нас есть сервер с записями о сотрудниках, а адрес веб-приложения – `http://server.com`. Чтобы получить доступ к записи сотрудника, мы можем выполнить команду `http://server.com/employee/1`, которая говорит серверу предоставить запись сотрудника под номером 1.
- Методы запросов. Они говорят, что вы хотите сделать с ресурсом. Браузер использует метод GET, чтобы проинформировать удалённую сторону о том, что он хочет получить данные. Кроме GET есть много других методов вроде POST, PUT и DELETE. В примере с `http://server.com/employee/1` выше браузер на самом деле использует метод GET, поскольку он хочет получить данные о сотруднике.
- Заголовки запроса. Это дополнительные инструкции, посылаемые вместе с

запросом. Они могут определять тип необходимого ресурса или подробности авторизации.

- Тело запроса. Это данные, отправляемые вместе с запросом. Данные обычно отправляются, когда выполняется POST-запрос к REST веб-сервису. Зачастую в POST-запросе клиент говорит серверу, что он хочет добавить на него ресурс. Следовательно, тело запроса содержит подробную информацию о ресурсе, который необходимо добавить на сервер.

- Тело ответа. Это основная часть ответа. В нашем примере на запрос `http://server.com/employee/1` сервер мог бы прислать XML-документ с данными о сотруднике в теле ответа.

- Коды ответа. Эти коды возвращаются сервером вместе с ответом. Например, код 200 обычно означает, что при отправке ответа не произошло никакой ошибки.

3.2 Модель данных

Рассмотрим модель данных, относящихся к разделу курсового проектирования. Физическая модель данных представлена в виде диаграммы таблиц БД на рисунке 3.2.

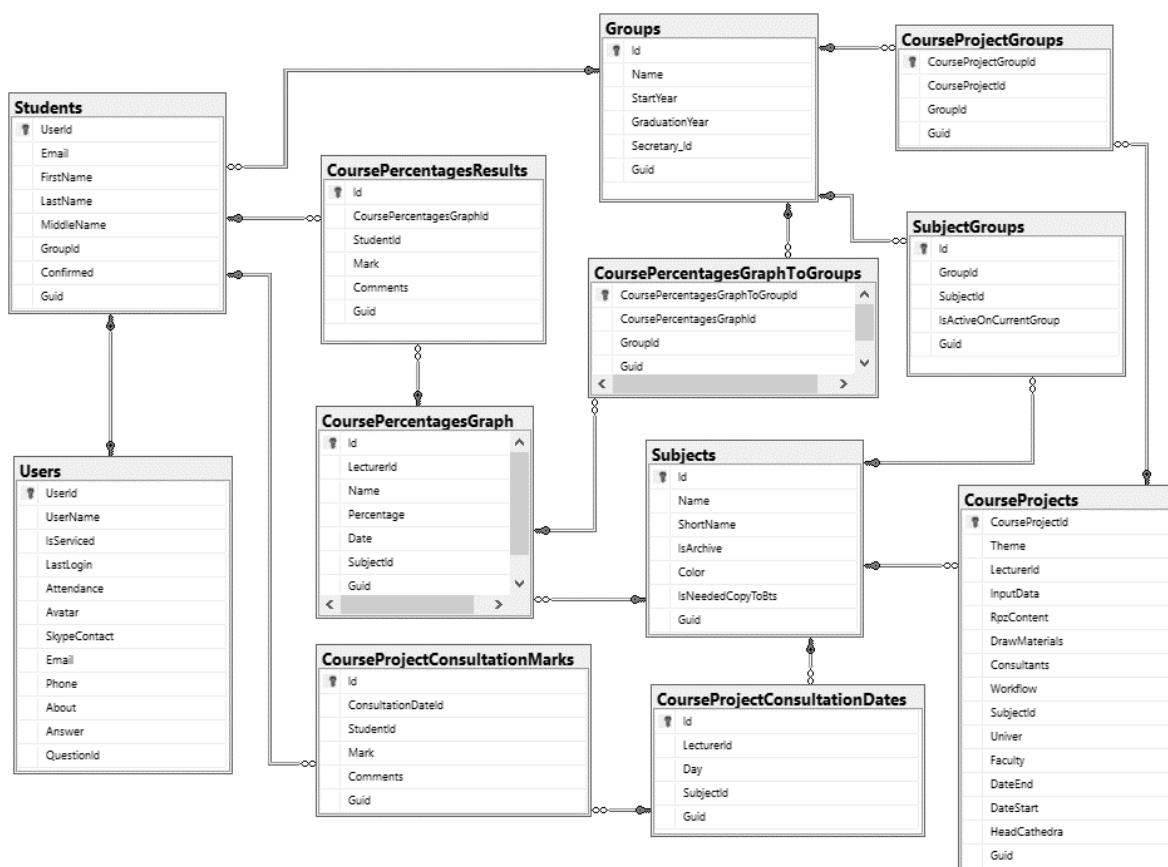


Рисунок 3.2 – Диаграмма таблиц базы данных

Таблица CourseProjects отражает сущность «Курсовой проект» и имеет столбцы Theme (Тема), InputData (Входные данные), RpzContent (Содержание), DrawMaterials (Графические материалы), Univer (Название ВУЗа), Faculty (Название факультета), HeadCathedra (ФИО заведующего кафедры). Таблица имеет два внешних ключа, связывающих её с таблицами Lecturers (Преподаватели) и Subjects (Предметы).

Таблица CourseProjectGroups содержит внешние ключи CourseProjectId и GroupId. С её помощью реализуется связь «многие ко многим», это означает что одна студенческая группа может иметь несколько курсовых проектов, так же, как и курсовой проект может быть у нескольких групп.

Таблицы CoursePercentagesGraph, CoursePercentageGraphToGroups и CoursePercentagesResults хранят информацию об этапах процентовки и их результатах для каждого студента; этапы процентовки отличаются для каждой пары группы и предмета.

Таблица CourseProjectConsultationDates хранит даты консультаций, а таблица CourseProjectConsultationMarks – оценки по каждому этапу для каждого студента.

Данные из БД наше Angular-приложение будет получать от приложения-сервера в структурированном виде в формате JSON.

На рисунке 3.3 показан ответ сервера на запрос на получение списка тем курсовых проектов.

```
▼ Items: [{Id: 2216, Theme: "Тестовый проект 3", Lecturer: "testLecturer2 testLecturer2 testLecturer2",...},...]  
▼ 0: {Id: 2216, Theme: "Тестовый проект 3", Lecturer: "testLecturer2 testLecturer2 testLecturer2",...}  
    ApproveDate: "2020-04-22T23:01:06.407"  
    ApproveDateString: "22-04-2020"  
    Group: "10701216"  
    Id: 2216  
    Lecturer: "testLecturer2 testLecturer2 testLecturer2"  
    LecturerId: null  
    SelectedGroupsIds: null  
    Student: "Жолох Виктор Геннадьевич"  
    StudentId: 8960  
    Subject: null  
    SubjectId: null  
    Theme: "Тестовый проект 3"  
▶ 1: {Id: 1211, Theme: "Тестовый проект 2", Lecturer: "testLecturer2 testLecturer2 testLecturer2",...}  
▶ 2: {Id: 1210, Theme: "Тестовый проект 1", Lecturer: "testLecturer2 testLecturer2 testLecturer2",...}  
Total: 3
```

Рисунок 3.3 – Ответ сервера

URL этого запроса:

<http://localhost:8080/api/courseProject/?count=1000000&filter=%7B%22subjectId%22:%224112%22,%22searchString%22:%22%22%7D&filter%5BsubjectId%5D=4112&page=1&sorting%5BId%5D=desc>

4 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Хорошо проработанная архитектура нужна любому ПО, и сложному, и шаблонному. С ее помощью экономится время, усилия и деньги.

Прежде чем приступить к разработке приложения, надо получить представление о назначении каждой его части. Это позволит сделать приложение более плавным и быстрым, а код более читабельным и пригодным для поддержки.

4.1 Архитектура проекта

Архитектура приложения обусловлена использованием Angular и является стандартной для программ, разработанных на базе данного фреймворка. Она состоит из таких частей как Компонент (Component), Шаблон (Template), Директива (Directive), Сервис (Service) и Модуль (Module) (рисунок 4.1).

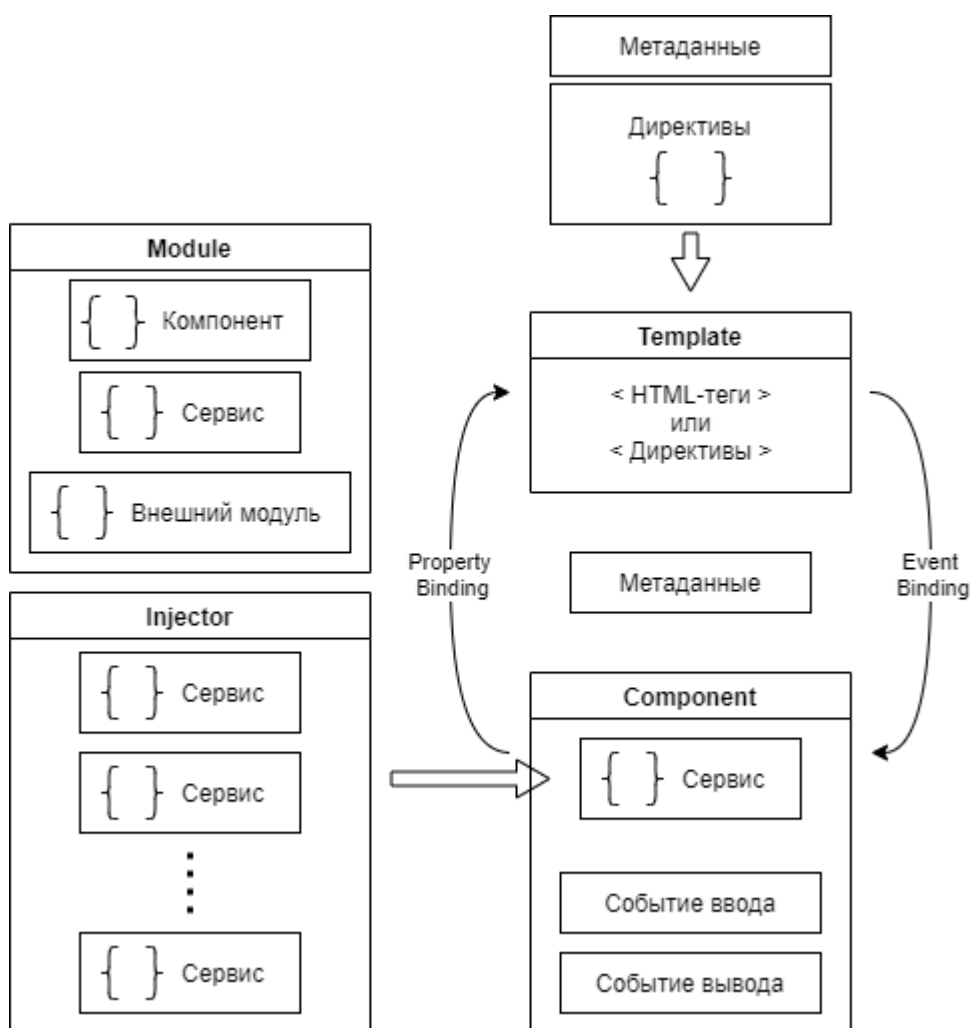


Рисунок 4.1 – Архитектура Angular

Angular поддерживает следующие типы привязки данных:

Интерполяция (interpolation): используется для просмотра значения свойства компонента на экране. Синтаксис: `{{ имя_переменной }}`.

Привязка свойства (property binding): используется для обновления свойства DOM, а также для входных событий. Синтаксис: `[свойство_DOM] = 'имя_переменной'`.

Привязка событий (event binding): используется для генерации события из определенного элемента, такого как нажатие кнопки или событие `keyup` элемента управления вводом и т. д. Синтаксис: `(событие) = 'вызов_обработчика_события'`.

Двустороннее связывание данных (two-way data binding): чаще всего используется при работе с формами на основе шаблонов. Свойство сопоставляется с элементом управления шаблона. В результате при каждом обновлении значения на экране оно будет автоматически обновляться в компоненте и наоборот. Синтаксис двусторонней привязки данных: `[(ngModel)] = 'свойство'`.

Angular использует механизм внедрения зависимостей.

Внедрение зависимостей – это способ предоставить новому экземпляру класса полностью сформированные зависимости, которые ему требуются. Инжектор (injector) содержит список сервисов, которые вы собираетесь использовать в приложении. Всякий раз, когда какой-либо компоненту требуется некоторый сервис, инжектор передает экземпляры сервиса этому компоненту.

Шаблоны в Angular являются динамическими. Когда Angular отображает их, он преобразует DOM в соответствии с инструкциями, данными в директивах.

Директива (directive) – это класс с декоратором `@Directive`.

Angular предоставляет два типа директив:

- Структурная директива: меняет структуру шаблона DOM. Например, `*ngFor`, `*ngSwitch`, `*ngIf` и т. д. являются структурными директивами.
- Директива атрибута: Директива атрибута обновляет атрибут конкретного элемента управления HTML, например `[ngClass]` это директива атрибута.

Компонент – это директива с шаблоном. Декоратор `@Component` это, по сути, декоратор `@Directive`, который расширен за счет шаблонно-ориентированных функций [7].

4.2 Модули

Приложения Angular являются модульными, и Angular имеет собственную систему модульности, которая называется `NgModule`.

Каждое приложение Angular имеет как минимум один класс с декоратором `@NgModule`, это корневой модуль, условно называемый `AppModule`.

Чтобы использовать любой компонент в приложении, вы должны объявить его в соответствующем модуле.

Реализация модуля AppModule представлена на рисунке 4.2.

```
@NgModule({
  declarations: [
    AppComponent,
    ProjectsComponent,
    ProjectsListComponent,
    PercentagesComponent,
    PercentagesListComponent,
    PercentageResultsComponent,
    PercentageResultsListComponent,
    TaskSheetComponent,
    VisitStatsComponent,
    VisitStatsListComponent,
    EditPercentageDialogComponent,
    AddDateDialogComponent,
    ConfirmDialogComponent,
    AddStageDialogComponent,
    AddProjectDialogComponent,
    AssignProjectDialogComponent,
    EditTaskSheetComponent,
    DefenseComponent,
    VisitingPopoverComponent,
    AddJobDialogComponent,
    PlagiarismCheckDialogComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    BrowserAnimationsModule,
    FormsModule,
    ReactiveFormsModule,
    MatModule,
    StoreModule.forRoot(appReducers),
    StoreDevtoolsModule.instrument(),
  ],
  providers: [
    {provide: MAT_DATE_LOCALE, useValue: 'ru-RU'}
  ],
  entryComponents: [EditPercentageDialogComponent, AddDateDialogComponent, ConfirmDialogComponent,
    AddStageDialogComponent, AddProjectDialogComponent, AssignProjectDialogComponent,
    EditTaskSheetComponent, VisitingPopoverComponent, AddJobDialogComponent,
    PlagiarismCheckDialogComponent],
  bootstrap: [AppComponent]
})
export class AppModule {
}
```

Рисунок 4.2 – Реализация модуля AppModule

Наиболее важные свойства @NgModule:

- Декларации (declarations): свойство содержит список компонентов, которые вы определяете для этого модуля. В нашем случае в AppModule объявлены все компоненты приложения.
- Экспорт (exports): если вы хотите использовать компонент или директиву этого модуля в другом модуле, вам нужно добавить сюда этот компонент или директиву. Как правило, корневой модуль или AppModule не содержит свойства экспорт.

- Импорт (imports): если вы хотите использовать внешние модули (библиотеки), такие как FormsModule, RouterModule и т. д., то вам нужно добавить модуль в это свойство. Внешний модуль может быть как встроенным в Angular, так и определяемым пользователем. В нашем проекте присутствует модуль MatModule, в котором мы импортируем все необходимые нам модули из пакета @angular/material. Таким образом, когда в AppModule мы импортируем MatModule, в проекте становятся доступны соответствующие компоненты и директивы.
- Самозагрузка (bootstrap): нужно указать имя компонента, который вы хотите загрузить, когда приложение загружается в браузер. Обычно это имя корневого компонента. Кроме того, только корневой модуль содержит это свойство [7].

4.3 Компоненты

Компонент является основным строительным блоком пользовательского интерфейса (UI). Каждый компонент сопоставлен с шаблоном (рисунок 4.1).

Angular-компонент содержит свойства, методы, конструктор, а также входные события, выходные события и методы жизненного цикла, такие как ngOnInit, ngOnDestroy и т. д. Angular создает, обновляет и уничтожает компоненты при перемещении пользователя по приложению.

Важные свойства декоратора @Component:

Селектор (selector): имя, данное в этом свойстве, используется на HTML-странице в качестве тега для загрузки этого компонента на экран. Например, чтобы загрузить компонент с селектором app-percentages на экран, вам нужно использовать <app-percentages> на странице HTML.

templateUrl : используется для сопоставления внешней HTML-страницы с этим компонентом.

styleUrls : используется для вставки списка файлов CSS, которые вы хотите использовать для этого компонента [7].

Реализация компонента PercentagesComponent представлена на рисунке 4.3.

Данный компонент реализует интерфейс OnInit таким образом, чтобы при инициализации компонента получить из локального хранилища уникальный идентификатор текущего предмета и выполнить запрос на получение необходимых данных с сервера.

Метод retrievePercentages() предназначен для получения списка этапов процентки с сервера.

Метод addStage() является обработчиком события клика по кнопке в шаблоне компонента и используется для создания нового этапа процентки.

```

@Component({
  selector: 'app-percentages',
  templateUrl: './percentages.component.html',
  styleUrls: ['./percentages.component.less']
})
export class PercentagesComponent implements OnInit {

  @Input() courseUser: CourseUser;
  private COUNT = 1000;
  private PAGE = 1;
  private percentages: Percentage[];
  private subjectId: string;

  constructor(private percentagesService: PercentagesService,
    public dialog: MatDialog,
    private snackBar: MatSnackBar,
    private store: Store<IAppState>) {

  }

  ngOnInit() {
    this.store.pipe(select(getSubjectId)).subscribe( next: subjectId => {
      this.subjectId = subjectId;
      this.retrievePercentages();
    });
  }

  retrievePercentages() {
    this.percentagesService.getPercentages( params: {
      count: this.COUNT,
      page: this.PAGE,
      filter: '{"subjectId":"' + this.subjectId + '"}',
    })
    .subscribe( next: res => this.percentages = res.Items);
  }

  addStage() {
    const dialogRef = this.dialog.open(AddStageDialogComponent, config: {
      width: '600px',
      data: { }
    });

    dialogRef.afterClosed().subscribe( next: result => {
      if (result != null) {
        const date = new Date(result.date);
        date.setMinutes( min: date.getMinutes() - date.getTimezoneOffset());
        this.percentagesService.editStage( id: null, date.toISOString(), this.subjectId, result.name, result.percentage)
        .subscribe( next: () => {
          this.ngOnInit();
          this.addFlashMessage( msg: 'График успешно сохранен');
        });
      }
    });
  }

  addFlashMessage(msg: string) {
    this.snackBar.open(msg, action: null, config: {
      duration: 2000
    });
  }
}

```

Рисунок 4.3 – Реализация компонента PercentagesComponent

Перечень всех компонентов приложения приведен на рисунке 4.4

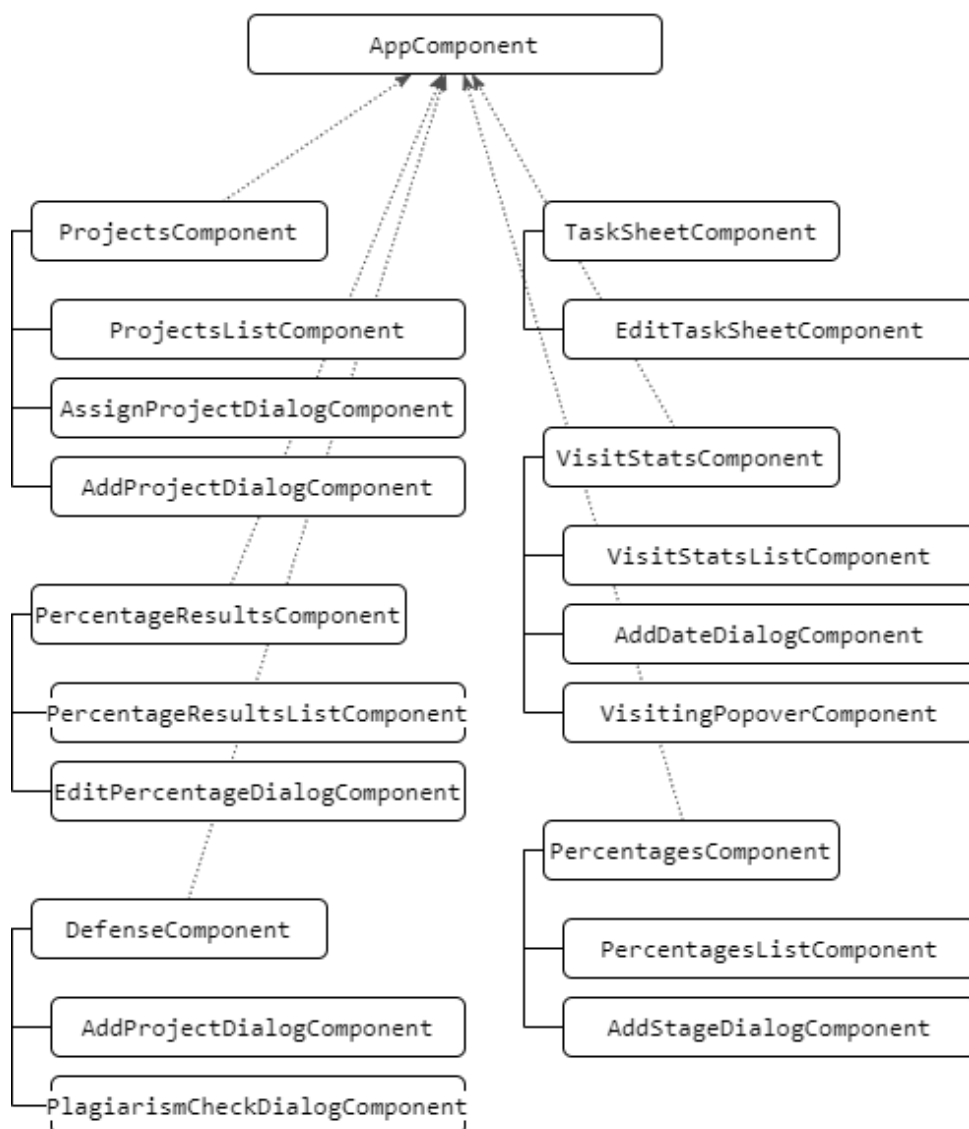


Рисунок 4.4 – Перечень компонентов приложения

Так как данное приложение является одностраничным, обмен данными с сервером не требует обновления страницы в браузере. Поэтому необходимо каким-то образом уведомлять пользователя, когда вызванные им операции выполняются. Для этого будем использовать сервис MatSnackBar из пакета @angular/material. Код метода для вывода сообщения на экран приведен на рисунке 4.5.

```

addFlashMessage(msg: string) {
  this.snackBar.open(msg, action: null, config: {
    duration: 2000
  });
}

```

Рисунок 4.5 – Код метода addFlashMessage()

4.4 Шаблоны

Итак, каждый компонент (component) сопоставлен с одним шаблоном (template). Шаблон по сути является HTML-страницей.

Шаблон компонента PercentagesComponent представлен на рисунке 4.6.

```
<div class="percentages-component" *ngIf="percentages; else Load">
  <div class="header">
    <button mat-raised-button color="primary" *ngIf="courseUser.IsLecturer" (click)="addStage()">
      Добавить этап
    </button>
  </div>
  <div class="percentages-component__body">
    <app-percentages-list [percentages]="percentages"></app-percentages-list>
  </div>
</div>
<ng-template #Load>
  <div class="spinner">
    <mat-spinner></mat-spinner>
  </div>
</ng-template>
```

Рисунок 4.6 – Шаблон компонента PercentagesComponent

Здесь можно увидеть использование директивы `*ngIf` и селекторов `<app-percentage-list>` и `<mat-spinner>`.

4.5 Сервисы

В данном приложении сервисы используются только для обмена данными с сервером. Они также может быть использованы в качестве класса совместного использования данных для обмена данными между компонентами в приложении, а также для написания бизнес-логики. На рисунке 4.7 в качестве примера показана реализация сервиса для работы с данными результатов процентки.

Декоратор `@Injectable()` используется для объявления класса как сервиса. Такие классы автоматически внедряются в компоненты инжектором.

Сервисы неизменно асинхронные. В данном приложении они возвращают данные типа `Observable`. `Observable` – это последовательность событий во времени.

Для взаимодействия с сервером и отправки запросов по протоколу `http` в сервисах применяется класс `HttpClient`. Этот класс определяет ряд методов для отправки различного рода запросов: `GET`, `POST`, `PUT`, `DELETE`. Данный класс построен поверх стандартного объекта в JavaScript - `XMLHttpRequest`.


```

@Injectables({
  providedIn: 'root'
}))

export class PercentagesService {

  constructor(private http: HttpClient) { }

  public getPercentages(params: any): Observable<any> {
    return this.http.get( url: 'api/CpPercentage',
      options: { params: new HttpParams( options: { fromObject: params } ) });
  }

  public editStage(id: string, date: string, subjectId: string, name: string, percentage: number): Observable<any> {
    return this.http.post( url: 'api/CpPercentage',
      body: { Id: id, Date: date, SubjectId: subjectId, Name: name, Percentage: percentage });
  }

  public deleteStage(params: any): Observable<any> {
    return this.http.delete( url: 'api/CpPercentage',
      options: { params: new HttpParams( options: { fromObject: params } ) });
  }
}

```

Рисунок 4.7 – Реализация сервиса PercentagesService

4.6 Тестирование программного обеспечения

Функциональное тестирование проводится для определения, насколько компонент или система соответствуют заданным функциональным требованиям, описанным в спецификациях.

Целью тестирования является поиск ошибок. Ошибка – это несоответствие ожидаемых результатов фактически полученным.

В рамках данного дипломного проекта будет проведено критическое и углубленное тестирование системы.

4.6.1 Используемые аппаратные средства

Тестирование проводится на персональном компьютере с параметрами, указанными в таблице 4.1

Таблица 4.1 – Аппаратные средства, используемые при тестировании

Роль	Аппаратная конфигурация	Программная конфигурация
Рабочая станция	Intel Core i7-6700HQ CPU 2.60 GHz, RAM 8 GB, SSD 512 GB	Windows 10 Pro версия 1909

4.6.2 Критическое тестирование

Критическое тестирование – это процесс поиска ошибок в программе при стандартной ее работе (при правильной последовательности действий, при верном заполнении полей и т. д.). Этот вид тестирования проводят всегда над каждой версией продукта, по заранее подготовленным тестовым случаям. Если в процессе тестирования находятся ошибки непредусмотренные тестовыми случаями, то необходимо добавить новый тестовый случай, а также составить отчет об ошибке [8].

В таблице 4.2 приведен перечень граничных и эквивалентных значений для критического тестирования.

Таблица 4.2 – Граничные и эквивалентные значения для критического тестирования

Поле	Формат данных	Граничные значения	Эквивалентные значения
1	2	3	4
Тема курсового проекта	Обязательное поле, от 3 до 255 символов	Строки длиной: 3 символа 255 символов	Строки длиной: 4 символа 128 символов 254 символа
Название шаблона	От 1 до 30 символов	Строки длиной: 1 символ 30 символов	Строки длиной: 2 символа 15 символов 29 символов
Входные данные	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1 символ 500 символов 998 символов
Содержание	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1 символ 500 символов 998 символов
Графические материалы	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1 символ 500 символов 998 символов
Название учебного заведения	От 0 до 255 символов	Строки длиной: 0 символов 255 символов	Строки длиной: 1 символ 128 символов 254 символа
Название кафедры	От 0 до 255 символов	Строки длиной: 0 символов 255 символов	Строки длиной: 1 символ 128 символов 254 символа

Продолжение таблицы 4.2

1	2	3	4
Фамилия, инициалы заведующего кафедры	От 0 до 30 символов	Строки длиной: 0 символов 30 символов	Строки длиной: 1 символ 15 символов 29 символов
Дата выдачи задания	ДД.ММ.ГГГГ	01.01.1970	29.06.2020 23.07.2019 01.02.2021
Сроки сдачи курсового проекта (работы)	ДД.ММ.ГГГГ	01.01.1970	29.06.2020 23.07.2019 01.02.2021
Название этапа	Обязательное поле, от 3 до 100 символов	Строки длиной: 3 символа 100 символов	Строки длиной: 4 символа 52 символа 99 символов
Процент выполнения	Целое число от 0 до 100	0; 100	1; 50; 99
Результат процентки	Целое число от 0 до 100	0; 100	1; 50; 99
Оценка	Целое число от 1 до 10	1; 10	2; 5; 9
Пропущено часов	Целое число больше 0	1	2; 50; 234
Комментарий	Обязательное поле, от 1 до 255 символов	Строки длиной: 1 символ 255 символов	Строки длиной: 2 символа 128 символов 254 символа

В таблице 4.3 приведены примеры тестовых случаев для критического тестирования.

Таблица 4.3 – Пример тестовых случаев критического тестирования

№	Название тестового случая	Описание тестового случая	Ожидаемые результаты	Тест пройден?
1	2	3	4	5
1	Создание темы курсового проекта	1) нажать кнопку «Добавить тему» 2) заполнить поля допустимыми значениями 3) нажать кнопку «Сохранить»	Новая тема отображается в списке	Да

Продолжение таблицы 4.3

1	2	3	4	5
2	Редактирование темы курсового проекта	1) нажать кнопку «Редактировать» 2) заполнить поля допустимыми значениями 3) нажать кнопку «Сохранить»	Измененная тема отображается в списке	Да
3	Удаление темы курсового проекта	1) нажать кнопку «Удалить» 2) подтвердить действие в диалоговом окне	Тема больше не отображается в списке	Да
4	Редактирование листа задания	1) нажать кнопку «Редактировать» 2) заполнить поля допустимыми значениями 3) нажать кнопку «Сохранить»	Лист задания изменен	Да
5	Сохранение шаблона	1) нажать кнопку «Редактировать» 2) заполнить поле «Название шаблона» допустимыми значениями 3) нажать кнопку «Сохранить шаблон»	Новый шаблон отображается в выпадающем списке	Да
6	Создание этапа процентовки	1) нажать кнопку «Добавить этап» 2) заполнить поля допустимыми значениями 3) нажать кнопку «Сохранить»	Новый этап отображается в списке	Да
7	Редактирование этапа процентовки	1) нажать кнопку «Редактировать» 2) заполнить поля допустимыми значениями 3) нажать кнопку «Сохранить»	Измененный этап отображается в списке	Да
8	Удаление этапа процентовки	1) нажать кнопку «Удалить» 2) подтвердить действие в диалоговом окне	Этап больше не отображается в списке	Да
9	Редактирование результата процентовки	1) нажать на ячейку таблицы вкладки «Результаты процентовки» 2) заполнить поля допустимыми значениями 3) нажать кнопку «ОК»	Новое значение результата отображается в таблице	Да

4.6.3 Углубленное тестирование

Углубленное тестирование – это процесс поиска ошибок в программе в нестандартных ситуациях, например, когда пользователь вводит данные, не соответствующие формату. В таких ситуациях система должна адекватно реагировать и выводить пользователю сообщение о том, что он неправильно ввел данные.

В таблице 4.4 приведен перечень граничных и эквивалентных значений для углубленного тестирования.

Таблица 4.4 – Граничные и эквивалентные значения для углубленного тестирования

Поле	Формат данных	Граничные значения	Эквивалентные значения
1	2	3	4
Тема курсового проекта	Обязательное поле, от 3 до 255 символов	Строки длиной: 3 символа 255 символов	Строки длиной: 2 символа 256 символов
Название шаблона	От 1 до 30 символов	Строки длиной: 1 символ 30 символов	Строки длиной: 0 символов 31 символ
Входные данные	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1000 символов
Содержание	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1000 символов
Графические материалы	От 0 до 999 символов	Строки длиной: 0 символов 999 символов	Строки длиной: 1000 символов
Название учебного заведения	От 0 до 255 символов	Строки длиной: 0 символов 255 символов	Строки длиной: 256 символов
Название кафедры	От 0 до 255 символов	Строки длиной: 0 символов 255 символов	Строки длиной: 256 символов
Фамилия, инициалы заведующего кафедры	От 0 до 30 символов	Строки длиной: 0 символов 30 символов	Строки длиной: 31 символ
Дата выдачи задания	ДД.ММ.ГГГГ	01.01.1970	31.12.1969
Сроки сдачи курсового проекта (работы)	ДД.ММ.ГГГГ	01.01.1970	31.12.1969
Название этапа	Обязательное поле, от 3 до 100 символов	Строки длиной: 3 символа 100 символов	Строки длиной: 2 символа 101 символ
Процент выполнения	Целое число от 0 до 100	0; 100	-1; 101
Результат процентовой	Целое число от 0 до 100	0; 100	-1; 101

Продолжение таблицы 4.4

1	2	3	4
Оценка	Целое число от 1 до 10	1; 10	0; 11
Пропущено часов	Целое число больше 0	1	0
Комментарий	Обязательное поле, от 1 до 255 символов	Строки длиной: 1 символ 255 символов	Строки длиной: 0 символов 256 символов

В таблице 4.5 приведены примеры тестовых случаев для углубленного тестирования.

Таблица 4.5 – Пример тестовых случаев критического тестирования

№	Название тестового случая	Описание тестового случая	Ожидаемые результаты	Тест пройден?
1	2	3	4	5
1	Создание темы курсового проекта	1) нажать кнопку «Добавить тему» 2) заполнить поля некорректными значениями	Кнопка «Сохранить» неактивна, появляется сообщение о некорректности данных	Да
2	Редактирование темы курсового проекта	1) нажать кнопку «Редактировать» 2) заполнить поля некорректными значениями	Кнопка «Сохранить» неактивна, появляется сообщение о некорректности данных	Да
3	Удаление темы курсового проекта	1) нажать кнопку «Удалить» 2) отменить действие в диалоговом окне	Тема по-прежнему отображается в списке	Да
4	Редактирование листа задания	1) нажать кнопку «Редактировать» 2) заполнить поля некорректными значениями 3) нажать кнопку «Сохранить»	Кнопка «Сохранить» неактивна, появляется сообщение о некорректности данных	Да
5	Сохранение шаблона	1) нажать кнопку «Редактировать» 2) заполнить поле «Название шаблона» некорректными значениями	Кнопка «Сохранить шаблон» неактивна, появляется сообщение о некорректности данных	Да

Продолжение таблицы 4.5

1	2	3	4	5
6	Создание этапа процентовки	1) нажать кнопку «Добавить этап» 2) заполнить поля некорректными значениями	Кнопка «Сохранить» неактивна, появляется сообщение о некорректности данных	Да
7	Редактирование этапа процентовки	1) нажать кнопку «Редактировать» 2) заполнить поля некорректными значениями	Кнопка «Сохранить» неактивна, появляется сообщение о некорректности данных	Да
8	Удаление этапа процентовки	1) нажать кнопку «Удалить» 2) отменить действие в диалоговом окне	Этап по-прежнему отображается в списке	Да
9	Редактирование результата процентовки	1) нажать на ячейку таблицы вкладки «Результаты процентовки» 2) заполнить поля некорректными значениями	Кнопка «ОК» неактивна, появляется сообщение о некорректности данных	Да

4.6.4 Результаты тестирования

В процессе тестирования были выявлены ошибки. Список ошибок представлен в таблице 4.6.

Таблица 4.6 – Анализ ошибок

Номер ошибки	Описание ошибки	Гипотеза местонахождения	Гипотеза прошла? Да/Нет
1	2	3	4
1	Возможность добавить работу для защиты с некорректными данными в поле «Комментарий»	Отсутствие валидации данных для поля «Комментарий»	Да
2	Этап процентовки не создается, если в поле «Дата» оставить предустановленное значение.	Ссылка на объект предустановленной даты отсутствует в модели данных	Да
3	Кнопка «Сохранить» является активной при неправильном заполнении поля «Преподаватель»	Отсутствие валидации данных для поля «Преподаватель»	Да

Каждой ошибке присваивался свой уровень важности, в зависимости от влияния на работу приложения. Ошибки с более высоким приоритетом исправлялись в первую очередь.

Результаты тестирования приложения, статистика найденных и исправленных ошибок представлена в таблице 4.7.

Таблица 4.7 – Результаты тестирования приложения

	Критических	Важных	Средних	Незначительных	Всего
Найдено ошибок	0	3	0	0	3
Исправлено ошибок	0	3	0	0	3

Тестирование приложения позволило выявить ряд ошибок, допущенных во время реализации. Критических ошибок, влияющих на работоспособность системы в целом, обнаружено не было. Все описанные выше ошибки были устранены.

5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для доступа к приложению необходимо ввести в адресную строку веб-браузера адрес сайта и войти в систему. Чтобы перейти к курсовым проектам, необходимо выбрать соответствующий пункт меню.

5.1 Руководство пользователя «Студент»

В верхней части страницы находятся вкладки «Темы проектов», «Лист задания», «График процентов», «Результаты процентов», «Посещение консультаций» и «Защита проектов» (рисунок 5.1). Нажимая на нужную вкладку, пользователь может переключать содержимое страницы.

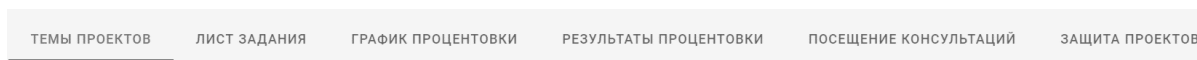


Рисунок 5.1 – Вкладки в верхней части страницы

При переходе на страницу курсового проекта пользователь видит содержимое вкладки «Темы проектов» (рисунок 5.2). Здесь находится список всех тем проектов для данного предмета. Чтобы произвести поиск, достаточно ввести символы в поле «Поиск темы, студента, группы». Список можно отсортировать, нажав на заголовок столбца таблицы.

ТЕМЫ ПРОЕКТОВ ЛИСТ ЗАДАНИЯ ГРАФИК ПРОЦЕНТОК РЕЗУЛЬТАТЫ ПРОЦЕНТОК ПОСЕЩЕНИЕ КОНСУЛЬТАЦИЙ ЗАЩИТА ПРОЕКТОВ					
Поиск темы, студента, группы					
№	Тема курсового проекта	Студент	Группа	Дата	Действие
1	Онлайн-игра "Морской бой"	Захарко Евгений Николаевич	10701116	28/05/2020	
2	Информационная система "Отдел кадров"	Иванов Иван Иванович	10701216	04/06/2020	
3	Интернет-магазин спортивного питания	Багиров Руслан Аликберович	10701116	27/05/2020	
4	Система статистических опросов пользователей	Жолох Виктор Геннадьевич	10701216	01	Скачать лист задания
5	Электронный магазин бытовой техники	Жибурт Максим Геннадьевич	10701216	17/05/2020	
6	Информационная система "Автовокзал"	Жук Иван Вячеславович	10701216	17/05/2020	

Рисунок 5.2 – Содержимое вкладки «Темы проектов» для роли «Студент»

Если студент еще не имеет назначенной ему темы, то он может выбрать любую свободную тему, нажав в соответствующей строке кнопку «Выбрать тему». При

нажатии на эту кнопку появляется диалоговое окно (рисунок 5.3). Подобные окна появляются для подтверждения большинства действий в системе.

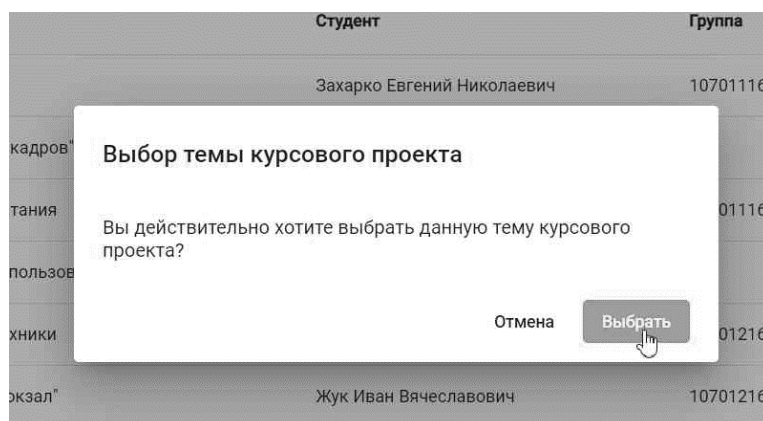


Рисунок 5.3 – Диалоговое окно для подтверждения выбора темы

При нажатии на кнопку «Отмена» диалоговое окно закрывается, и никаких изменений не происходит. При нажатии на кнопку «Выбрать» диалоговое окно закрывается, и студенту назначается выбранная им тема, однако его выбор должен быть подтвержден преподавателем, прежде чем студент получит возможность добавить готовый проект для защиты.

Когда выбор темы будет подтвержден преподавателем, студент также сможет скачать сгенерированный лист задания по своей теме. Сделать это можно нажав кнопку «Скачать лист задания» в списке тем курсовых проектов (рис. 5.4).

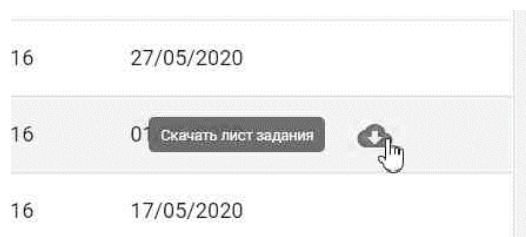


Рисунок 5.4 – Кнопка для скачивания листа задания в списке тем

Если в браузере не задан путь для сохранения файлов, то откроется системное диалоговое окно для указания этого пути.

Скачать лист задания можно и во вкладке «Лист задания» (рисунок 5.5), соответствующая кнопка здесь расположена в правом верхнем углу страницы.

ТЕМЫ ПРОЕКТОВ ЛИСТ ЗАДАНИЯ ГРАФИК ПРОЦЕНТОВКИ РЕЗУЛЬТАТЫ ПРОЦЕНТОВКИ ПОСЕЩЕНИЕ КОНСУЛЬТАЦИЙ ЗАЩИТА ПРОЕКТОВ

Выберите тему курсового проекта
Система статистических опросов пользователей

Белорусский национальный технический университет
Факультет информационных технологий и робототехники

Утверждаю
Заведующий кафедрой _____ Полосов Ю.В.
« _____ » _____ 2020г. (подпись) (фамилия, инициалы)

Задание на курсовой проект (курсовую работу)

Обучающемуся _____ Жалоз Виктор Геннадьевич
(фамилия, имя собственное, отчество)

_____ группа 10701216

1. Тема Система статистических опросов пользователей
(указать: курсового проекта или курсовой работы)

2. Сроки сдачи студентом законченного проекта (работы) 8 июня 2018г.

3. Исходные данные _____
(указать: курсового проекта или курсовой работы)

Рисунок 5.5 – Вкладка «Лист задания»

Увидеть и скачать лист задания можно по любой теме, для этого в верхней части страницы расположен выпадающий список тем всех курсовых проектов студента.

На вкладке «График процентовки» (рисунок 5.6) студент может посмотреть требуемый процент выполнения работы по датам.

ТЕМЫ ПРОЕКТОВ	ЛИСТ ЗАДАНИЯ	ГРАФИК ПРОЦЕНТОВКИ	РЕЗУЛЬТАТЫ ПРОЦЕНТОВКИ	ПОСЕЩЕНИЕ КОНСУЛЬТАЦИЙ	ЗАЩИТА ПРОЕКТОВ
№	Название этапа	Процент выполнения	Дата		
1	Этап 1	10	22/03/2020		
2	Этап 2	60	20/04/2020		
3	Этап 3	90	30/05/2020		
4	Этап 4	100	01/06/2020		

Рисунок 5.6 – Вкладка «График процентовки»

На вкладке «Результаты процентовки» (рисунок 5.7) студент может увидеть результаты в процентах, выставяемые преподавателем, а также итоговые оценки по курсовым проектам. Вверху слева находится выпадающий список для выбора группы, а сверху справа находится поисковая строка.

ТЕМЫ ПРОЕКТОВ		ЛИСТ ЗАДАНИЯ	ГРАФИК ПРОЦЕНТОК	РЕЗУЛЬТАТЫ ПРОЦЕНТОК	ПОСЕЩЕНИЕ КОНСУЛЬТАЦИЙ	ЗАЩИТА ПРОЕКТОВ	
10701216						Поиск по теме, фамилии	
№	Студент	Тема курсового проекта	22.03.2020	20.04.2020	30.05.2020	01.06.2020	Оценка
1	Жибурт Максим Геннадьевич	Электронный магазин бытовой техники	0	20	-	-	9
2	Жолох Виктор Геннадьевич	Система статистических опросов пользователей	10	-	-	-	-
3	Жук Иван Вячеславович	Информационная система "Автовокзал"	10	-	-	-	-
4	Иванов Иван Иванович	Информационная система "Отдел кадров"	-	-	-	-	-

Рисунок 5.7 – Вкладка «Результаты процентовки»

На вкладке «Посещение консультаций» (рисунок 5.8) студент может увидеть количество пропущенных часов консультаций по датам. Здесь тоже есть выпадающий список для выбора группы и поисковая строка.

10701216						Поиск по теме, фамилии			
№	Студент	Тема курсового проекта	05.04.2020	14.04.2020	14.04.2020	27.05.2020	31.05.2020	02.06.2020	05.06.2020
1	Жибурт Максим Геннадьевич	Электронный магазин бытовой техники	3	1	2				
2	Жолох Виктор Геннадьевич	Система статистических опросов пользователей	1	4	3				
3	Жук Иван Вячеславович	Информационная система "Автовокзал"	2	1					
4	Иванов Иван Иванович	Информационная система "Отдел кадров"							

Рисунок 5.8 – Вкладка «Посещение консультаций»

На вкладке «Защита проектов» находится таблица с добавленными работами и кнопка «Добавить работу». При нажатии на кнопку открывается диалоговое окно (рисунок 5.9).

На защиту курсового проекта

Комментарий

Прикрепленный материал (Можно добавить только один файл)

Добавить файл

Отмена Отправить работу

Рисунок 5.9 – Диалоговое окно для добавления работы на защиту

Для добавления работы необходимо заполнить поле «Комментарий» и прикрепить файл. Чтобы прикрепить файл, необходимо нажать кнопку «Добавить файл», после чего в открывшемся системном диалоговом окне выбрать нужный файл. Прикрепленный файл сразу же отобразится вместо кнопки «Добавить файл» (рисунок 5.10). Файл можно удалить, нажав на «крестик».

The screenshot shows a web form titled 'На защиту курсового проекта'. It has two tabs: 'Комментарий' (selected) and 'Действие'. The 'Комментарий' tab contains a text input field with the placeholder 'Готовая работа'. Below the input field is a section for 'Прикрепленный материал (Можно добавить только один файл)'. It displays a list of files with a preview of a document icon and the filename 'ТП.doc'. To the right of the filename, it shows 'Document', '6248.00 КБ', and 'Загруженный файл'. A close button (X) is next to the file entry. At the bottom right, there are two buttons: 'Отмена' and 'Отправить работу'.

Рисунок 5.10 – Прикрепленный файл

При правильно заполненной форме кнопка «Отправить работу» становится активной, и после нажатия на нее работа отправляется на защиту и отображается в списке работ (рисунок 5.11).

ТЕМЫ КУРСОВЫХ ПРОЕКТОВ			
ЛИСТ ЗАДАНИЯ			
ГРАФИК ПРОЦЕНТОВКИ			
РЕЗУЛЬТАТЫ ПРОЦЕНТОВКИ			
СТАТИСТИКА ПОСЕЩЕНИЯ КОНСУЛЬТАЦИЙ			
ЗАЩИТА			
Файл	Комментарий	Действие	
ТП.doc	Готовая работа	✎ 🗑	

Рисунок 5.11 – Список добавленных работ

Прикрепленный файл можно скачать, нажав на его название. Работу можно редактировать и удалить, нажимая соответствующие кнопки в столбце «Действие».

5.2 Руководство пользователя «Преподаватель»

У пользователя в роли «Преподаватель» содержимое всех вкладок, кроме вкладки «Защита проектов», повторяет содержимое этих же вкладок у пользователя в роли «Студент», с тем лишь отличием, что преподавателю доступно большее количество действий и соответствующих им кнопок.

На вкладке «Темы проектов» (рисунок 5.12) преподаватель может с помощью соответствующих кнопок назначить тему студенту (рисунок 5.13), отменить назначение, а также подтвердить выбор темы студентом.

ТЕМЫ КУРСОВЫХ ПРОЕКТОВ

ЛИСТ ЗАДАНИЯ

ГРАФИК ПРОЦЕНТОVKИ

РЕЗУЛЬТАТЫ ПРОЦЕНТОVKИ

СТАТИСТИКА ПОСЕЩЕНИЯ КОНСУЛЬТАЦИЙ

ЗАЩИТА

Добавить тему

Поиск темы, студента, группы

№	Тема курсового проекта	Студент	Группа	Дата	Действие
1	Онлайн-игра "Морской бой"	Захарко Евгений Николаевич	10701116	28/05/2020	<div></div> <div></div> <div></div> <div></div>
2	Информационная система "Отдел кадров"				<div></div> <div></div> <div></div> <div></div>
3	Интернет-магазин спортивного питания	Багиров Руслан Аликберович	10701116	27/05/2020	<div></div> <div></div> <div></div> <div></div>
4	Система статистических опросов пользователей	Жолох Виктор Геннадьевич	10701216		<div>Подтвердить выбранную тему</div> <div></div>
5	Электронный магазин бытовой техники	Жибурут Максим Геннадьевич	10701216	17/05/2020	<div></div> <div></div> <div></div> <div></div>
6	Информационная система "Автовокзал"	Жук Иван Вячеславович	10701216	17/05/2020	<div></div> <div></div> <div></div> <div></div>

Рисунок 5.12 – Вкладка «Темы проектов» для роли «Преподаватель»










Назначение темы курсового проекта			
№	Студент	Группа	Действие
1	Баран Дмитрий Александрович	10701116	
2	Безкровный Кирилл Игоревич	10701116	
3	Богданчук Павел Андреевич	10701116	
4	Гайдук Илья Олегович	10701116	
5	Еремейко Анастасия Александровна	10701116	
6	Ершов Никита Вадимович	10701116	
7	Захаров Егор Викторович	10701116	
8	Карастелёва Наталья Николаевна	10701116	
9	Киселев Александр Владимирович	10701116	
			Отмена

Рисунок 5.13 – Диалоговое окно для назначения темы студенту

При нажатии на кнопку «Добавить тему» или «Редактировать» открывается соответствующее диалоговое окно (рисунок 5.14). В нем необходимо заполнить поле «Тема курсового проекта» и выбрать группы, студентам которых можно будет назначить данную тему.

Добавление темы курсового проекта

Тема курсового проекта

Группы:

>>	<<
	10701116
	10701216

Отмена Сохранить

Рисунок 5.14 – Диалоговое окно для добавления или редактирования темы

На вкладке «Лист задания» преподаватель может выбрать в выпадающем списке тему курсового проекта и изменить лист задания для выбранной темы, нажав на кнопку «Редактировать» в правом верхнем углу, при этом откроется диалоговое окно (рисунок 5.15)

Редактирование листа задания

Шаблон

Название шаблона Сохранить шаблон

< Входные данные Содержание Графические материалы >

1. Среда разработки: на выбор.
2. Инструменты автоматизации модульного тестирования: MSTest, NUnit, JUnit, cppUnit, TestComplete и др.
3. Графические конструкции МакКейба.
4. Литература по предметной области.
5. Литература по модульному тестированию.

Отмена Сохранить

Рисунок 5.15 – Диалоговое окно для редактирования листа задания

Диалоговое окно для редактирования листа задания содержит несколько вкладок. Вкладки «Входные данные», «Содержание» и «Графические данные» содержат по одному текстовому полю. Вкладка «Сведения о ВУЗе» содержит текстовые поля «Название учебного заведения», «Название факультета» и «Фамилия, инициалы заведующего кафедры». Вкладка «Даты» содержит поля «Дата выдачи задания» и «Сроки сдачи курсового проекта».

Поля, предназначенные для ввода даты можно заполнять в виде текста, а можно нажать соответствующий значок справа, чтобы открылся виджет календаря (рисунок 5.16), в котором можно выбрать нужный день щелчком мыши.

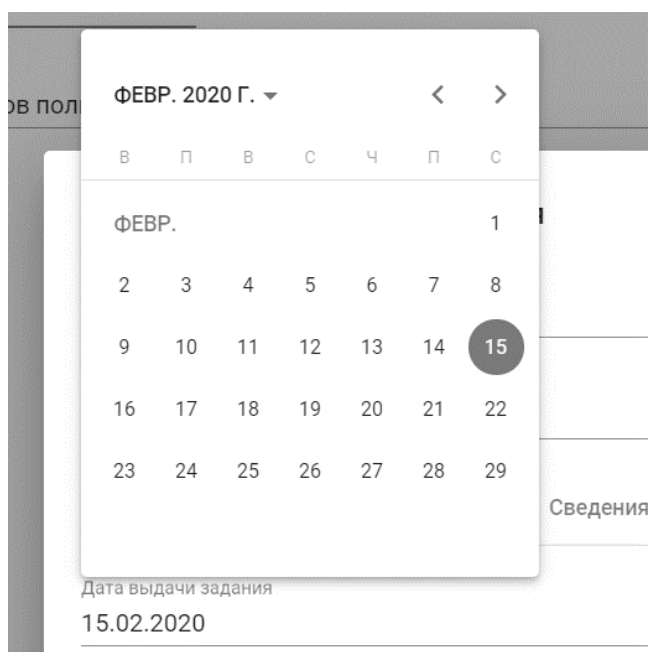


Рисунок 5.16 – Виджет для выбора даты

Все поля в диалоговом окне редактирования листа задания можно заполнить значениями из шаблона, для этого нужно выбрать шаблон в выпадающем списке. Чтобы сохранить новый шаблон, нужно заполнить поле «Название шаблона» и нажать на кнопку «Сохранить шаблон».

На вкладке «График процентовки» преподаватель может добавлять, редактировать и удалять этапы процентовки. При нажатии на кнопку «Добавить этап» или «Редактировать» открывается диалоговое окно с полями «Название этапа», «Процент выполнения» и «Дата» (рисунок 5.17).

Редактирование графика процентовой

Название этапа
Этап 1

Процент выполнения
10 %

Дата
22.03.2020

Отмена Сохранить

Рисунок 5.17 – Диалоговое окно редактирования графика процентовой

На вкладке «Результаты процентовой» преподаватель может выставить проценты, характеризующие фактически выполненный студентами объем работы. Для этого необходимо нажать на соответствующую ячейку таблицы, после чего откроется диалоговое окно (рисунок 5.18).

Результат

%

Ожидаемый процент 90

Комментарий

Отмена Сохранить

Рисунок 5.18 – Диалоговое окно выставления результата процентовой

На этой вкладке преподаватель также может выставить оценку по курсовому проекту для определенного студента. Для этого надо нажать на ячейку в столбце «Оценка», после чего откроется диалоговое окно (рисунок 5.19).

Оценка

Преподаватель

Оценка (от 1 до 10)

Дата выставления
06.06.2020

Комментарий

Отмена Сохранить

Рисунок 5.19 – Диалоговое окно выставления оценки

Чтобы загрузить данные в виде таблицы Excel, необходимо нажать на кнопку «Загрузить документ Excel» в верхней части страницы.

Во вкладке «Статистика посещения консультаций» преподаватель может добавить дату консультации, нажав на кнопку «Добавить дату консультации», при этом откроется диалоговое окно (рисунок 5.20). Чтобы удалить дату консультации, необходимо нажать на заголовок соответствующего столбца таблицы.

Добавление даты консультации

Дата консультации
02.06.2020

Отмена Сохранить

Рисунок 5.20 – Диалоговое окно добавления даты консультации

Чтобы выставить количество пропущенных студентами часов, необходимо сделать двойной клик по соответствующей ячейке таблицы, при этом откроется диалоговое окно со списком всех студентов (рисунок 5.21). Кроме поля для ввода количества часов есть поле «Комментарий», заполнять которое необязательно.

Посещаемость студентов

Редактируемая дата: **14.04.2020**




№	ФИО	Пропущено часов	Комментарий
1	Жибурт Максим Геннадьевич	2	
2	Жолох Виктор Геннадьевич	3	
3	Жук Иван Вячеславович		
4	Иванов Иван Иванович		

Отмена Сохранить

Рисунок 5.21 – Диалоговое окно для выставления пропущенных часов

На вкладке «Защита» (рисунок 5.22) отображается список студентов выбранной в выпадающем списке группы. При нажатии на определенного студента открывается таблица со списком его работ.

ТЕМЫ КУРСОВЫХ ПРОЕКТОВ ЛИСТ ЗАДАНИЯ ГРАФИК ПРОЦЕНТОВКИ РЕЗУЛЬТАТЫ ПРОЦЕНТОВКИ СТАТИСТИКА ПОСЕЩЕНИЯ КОНСУЛЬТАЦИЙ **ЗАЩИТА**

10701216 Открепленные группы   







Ассанович Кирилл Маратович	▼								
Бертош Павел Николаевич	▼								
Демидчик Юрий Дмитриевич	▼								
Деревянкина Диана Денисовна	▼								
Жибурт Максим Геннадьевич	▼								
Жолох Виктор Геннадьевич	▲								
<table> <thead> <tr> <th>Файл</th> <th>Комментарий</th> <th>Дата</th> <th>Действие</th> </tr> </thead> <tbody> <tr> <td>ТП.doc</td> <td>Готовая работа</td> <td>01.06.2020 22:40</td> <td> </td> </tr> </tbody> </table>		Файл	Комментарий	Дата	Действие	ТП.doc	Готовая работа	01.06.2020 22:40	 
Файл	Комментарий	Дата	Действие						
ТП.doc	Готовая работа	01.06.2020 22:40	 						
Жолудев Евгений Александрович	▼								
Жук Иван Вячеславович	▼								

Рисунок 5.22 – Вкладка «Защита» для роли «Преподаватель»

Чтобы принять работу, надо нажать на кнопку «Принять работу» в столбце «Действие». Чтобы отклонить принятую работу, необходимо нажать на кнопку «Отклонить работу». Чтобы вернуть работу студенту для исправления, надо нажать на кнопку «Вернуть работу», при этом откроется диалоговое окно, такое же, как и при добавлении работы в роли студента (рисунок 5.9).

В верхней части расположены кнопки «Скачать архивом», «Проверка на плагиат всех работ» и «Обновить данные». Также здесь расположен переключатель, позволяющий выбрать открепленные группы в списке.

6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

6.1 Определение единовременных затрат на создание программного продукта

При расчете экономической эффективности разработки программного обеспечения необходимо сопоставить затраты на решение задачи при ручном методе ее решения с затратами, связанными с ее автоматизацией. В том случае, если разрабатываемая задача внедряется взамен уже функционирующей или она представляет собой модификацию существующей задачи, необходимо осуществить сравнение затрат на создание и функционирование старой и новой задачи.

Единовременные капитальные затраты представляют собой цену программного продукта (ПП) или модели. Различают оптовую и отпускную цены. Все расчеты между покупателем и продавцом продукции, к числу которой относят и программные продукты (модели), производятся на основе отпускных цен. В настоящее время в соответствии с законодательством РБ в отпускную цену наряду с оптовой ценой включается налог на добавленную стоимость [9].

Определяющим фактором оптовой цены разработки является трудоёмкость создания ПП.

6.1.1 Определение трудоемкости разработки программного продукта

Предусмотрено четыре степени новизны разрабатываемых задач.

А – разработка задач, предусматривающая применение принципиально новых методов разработки, проведение научно-исследовательских работ.

Б – разработка типовых проектных решений, оригинальных задач и систем, не имеющих аналогов.

В – разработка проекта с использованием типовых проектных решений при условии их изменения; разработка проектов, имеющих аналогичные решения.

Г – привязка типовых проектных решений.

Сложность алгоритма представлена тремя группами.

1 – алгоритмы оптимизации и моделирования систем и объектов.

2 – алгоритмы учета, отчетности, статистики и поиска.

3 – алгоритмы, реализующие стандартные методы решения, а также не предусматривающие применения сложных численных и логических методов.

В данном дипломном проекте программный продукт имеет степень новизны В и сложность алгоритма 1.

Трудоёмкость разработки ПП может быть определена на основе типовых норм времени для программирования задач на ЭВМ. Она включает время на постановку

задачи и время на программирование задачи и определяется по формуле:

$$T_{pz} = T_{oa} + T_{bc} + T_{п} + T_{отл} + T_{др} + T_{до}, \quad (6.1)$$

где T_{oa} - трудоемкость подготовки описания задачи и исследования алгоритма решения;

T_{bc} - трудоемкость разработки блок-схемы алгоритма;

$T_{п}$ - трудоемкость программирования по готовой блок-схеме;

$T_{отл}$ - трудоемкость отладки программы на ЭВМ;

$T_{др}$ - трудоемкость подготовки документации по задаче в рукописи;

$T_{до}$ - трудоемкость редактирования, печати и оформления документации по установленной задаче.

Составляющие приведенной формулы определяются, в свою очередь, через условное число операторов Q в разрабатываемом ПП по формуле:

$$Q = q \cdot C \cdot (1 + p), \quad (6.2)$$

где q – число операторов в программе (принят 1000);

C – коэффициент сложности программы (принят 1,2);

p – коэффициент коррекции программы в ходе её разработки (принят 0,4).

Коэффициент сложности программы C характеризует относительную сложность программ задачи по отношению к так называемой типовой задаче, сложность которой принята за единицу. Значение коэффициента определяется на базе экспертных оценок.

Коэффициент коррекции программ p характеризует увеличение объема работ за счет внесения изменений в алгоритм и программу, изменения состава и структуры информации, а также уточнений, вносимых разработчиком программы для улучшения её качества без изменения постановки задачи. Значение p может быть принято равным 0,15...0,5 (принято 0,25).

$$Q = 1000 \cdot 1,2 \cdot (1 + 0,4) = 1680 \text{ (операторов)}.$$

Составляющие трудоемкости разработки программы определяются по формулам:

$$T_{oa} = \frac{Q \cdot W \cdot K}{80}; \quad (6.3)$$

$$T_{bc} = \frac{Q \cdot K}{20}; \quad (6.4)$$

$$T_{п} = \frac{Q \cdot K}{25}; \quad (6.5)$$

$$T_{отл} = \frac{Q \cdot K}{5}; \quad (6.6)$$

$$T_{др} = \frac{Q \cdot K}{18}; \quad (6.7)$$

$$T_{до} = 0,75 \cdot T_{др}, \quad (6.8)$$

где W – коэффициент увеличения затрат труда вследствие недостаточного или некачественного описания задачи ($W = 1,2 \dots 1,5$);

K – коэффициент квалификации разработчика алгоритмов и программ (при стаже работы до двух лет $K = 0,8$, при стаже от двух до трех лет $K = 1,0$, при стаже от трех до пяти лет $K = 1,1 \dots 1,2$, при стаже от пяти до семи лет $K = 1,3 \dots 1,4$, при стаже свыше семи лет $K = 1,5 \dots 1,6$).

$$T_{oa} = \frac{1680 \cdot 1,2 \cdot 0,8}{80} = 20,16(\text{ч}).$$

$$T_{6c} = \frac{1680 \cdot 0,8}{20} = 67,2(\text{ч}).$$

$$T_{п} = \frac{1680 \cdot 0,8}{25} = 53,76(\text{ч}).$$

$$T_{отл} = \frac{1680 \cdot 0,8}{5} = 268,8(\text{ч}).$$

$$T_{др} = \frac{1680 \cdot 0,8}{18} = 74,67(\text{ч}).$$

$$T_{до} = 0,75 \cdot 375,5 = 56(\text{ч}).$$

Исходя из полученных данных можно найти трудоёмкость разработки программного продукта, используя формулу (6.1):

$$T_{pz} = 20,16 + 67,2 + 53,76 + 268,8 + 74,67 + 56 = 540,59(\text{ч}).$$

6.1.2 Определение себестоимости создания программного продукта

Для определения себестоимости создания программного продукта необходимо определить затраты на заработную плату разработчика по формуле:

$$З_{pz} = T_{pz} \cdot t_{чp} \cdot (1 + k_{пр}) \cdot (1 + a) \cdot (1 + b), \quad (6.9)$$

где T_{pz} – трудоёмкость разработки программного продукта, чел-ч.;

$t_{чp}$ – среднечасовая ставка работника, осуществлявшего разработку программного продукта;

$k_{пр}$ – коэффициент, учитывающий процент премий в организации-разработчике (принят 0,4);

a – коэффициент, учитывающий доп. заработную плату (принят 0,15);

b – коэффициент, учитывающий отчисления от фонда заработной платы (отчисления в фонд социальной защиты населения и отчисления на обязательное медицинское страхование от несчастных случаев) ($b = 0,346$).

Среднечасовая ставка работника определяется исходя из Единой тарифной системы оплаты труда в Республике Беларусь по следующей формуле:

$$t_{\text{др}} = \frac{ЗП_{1р} \cdot k_{\text{т}}}{170}, \quad (6.10)$$

где $ЗП_{1р}$ – среднемесячная заработная плата программиста в Минске за 2020 год (принята 700 руб.) [10];

170 – среднее нормативное количество рабочих часов в месяце для 2020 года.

$$t_{\text{др}} = \frac{700}{170} = 4,117 \text{ (руб.)}.$$

Теперь можно определить затраты на заработную плату разработчиков программы:

$$З_{\text{рз}} = 540,59 \cdot 4,117 \cdot (1 + 0,4) \cdot (1 + 0,15) \cdot (1 + 0,346) = 4823,03 \text{ (руб.)}.$$

В себестоимость разработки программного продукта включаются также затраты на отладку ПП в процессе его создания. Для определения их величины необходимо рассчитать стоимость машино-часа работы ЭВМ, на которой осуществлялась отладка. Даная величина соответствует величине арендной платы за один час работы ЭВМ.

Затраты на отладку программы определяются по формуле:

$$З_{\text{от}} = T_{\text{отл}} \cdot S_{\text{мч}}, \quad (6.11)$$

где $T_{\text{отл}}$ – трудоемкость отладки программы (340,8 час.);

$S_{\text{мч}}$ – стоимость машино-часа работы ЭВМ (0,318 руб./час.).

$$З_{\text{от}} = 340,8 \cdot 0,318 = 108,37 \text{ (руб.)}.$$

Себестоимость разработки ПП определяется по формуле:

$$C_{\text{пр}} = З_{\text{рз}} \cdot F + З_{\text{от}}, \quad (6.12)$$

где F – коэффициент накладных расходов проектной организации без учета эксплуатации ЭВМ (принят 1,15).

Таким образом

$$C_{\text{пр}} = 4823,03 \cdot 1,15 + 108,37 = 5654,85(\text{руб.}).$$

6.1.3 Определение оптовой и отпускной цены программного продукта

Оптовая цена складывается из себестоимости создания программного продукта и плановой прибыли на программу.

Оптовая цена ПП определяется по формуле:

$$Ц_o = C_{\text{пр}} + П_p, \quad (6.13)$$

где $П_p$ – плановая прибыль на программу, руб.

Плановая прибыль на программу определяется по формуле:

$$П_{\text{пр}} = C_{\text{пр}} \cdot Н_{\text{п}}, \quad (6.14)$$

где $C_{\text{пр}}$ – себестоимость программы;

$Н_{\text{п}}$ – норма прибыли проектной организации (принята $Н_{\text{п}} = 0,3$).

Таким образом

$$П_{\text{пр}} = 5654,85 \cdot 0,3 = 1696,46(\text{руб.}).$$

Тогда

$$Ц_o = 5654,85 + 1696,46 = 7351,3(\text{руб.}).$$

Отпускная цена программы определяется по формуле:

$$Ц_{\text{пр}} = Ц_o + Ц_o \cdot K_{\text{ндс}}, \quad (6.15)$$

где $Ц_o$ – оптовая цена программы, руб;

$K_{\text{ндс}}$ – ставка налога на добавленную стоимость (принято 20%).

Из этого следует, что

$$Ц_{\text{пр}} = 7351,3 + 7351,3 \cdot 0,2 = 8821,57(\text{руб.}).$$

6.1.4 Определение стоимости машино-часа работы ЭВМ

Стоимость машино-часа работы ЭВМ определяется по формуле:

$$S_{\text{мч}} = C_{\text{э}} + \frac{(A_{\text{эвм}} + P_{\text{эвм}} + A_{\text{пл}} + P_{\text{пл}} + P_{\text{ар}})}{\Phi_{\text{эвм}}}, \quad (6.16)$$

где $C_{\text{э}}$ – расходы на электроэнергию за час работы ЭВМ, руб;

$A_{\text{эвм}}$ – годовая величина амортизационных отчислений на реновацию ЭВМ;

$P_{\text{эвм}}$ – годовые затраты на ремонт и техническое обслуживание ЭВМ, руб;

$A_{\text{пл}}$ – годовая величина амортизационных отчислений на реновацию производственных площадей, занимаемых ЭВМ, руб;

$P_{\text{пл}}$ – годовые затраты на ремонт и содержание производственных площадей, руб;

$P_{\text{ар}}$ – годовая величина арендных платежей за помещение, занимаемое электронно-вычислительными машинами, руб;

$\Phi_{\text{эвм}}$ – годовой фонд времени работы ЭВМ, час.

Расходы на электроэнергию за час работы ЭВМ определяются по формуле:

$$C_{\text{э}} = N_{\text{э}} \cdot k_{\text{ис}} \cdot \Pi_{\text{э}}, \quad (6.17)$$

где $N_{\text{э}}$ – установленная мощность электродвигателя ЭВМ (принято 0,09 кВт);

$k_{\text{ис}}$ – коэффициент использования энергоустановок по мощности ($k_{\text{ис}} = 0,9$);

$\Pi_{\text{э}}$ – стоимость 1 кВт-часа электроэнергии (принято 0,19 руб.) [11].

$$C_{\text{э}} = 0,09 \cdot 0,9 \cdot 0,19 = 0,01539 (\text{руб.}).$$

Годовая величина амортизационных отчислений на реновацию ЭВМ определяется по формуле:

$$A_{\text{эвм}} = \Pi_{\text{эвм}} \cdot k_{\text{у}} \cdot k_{\text{м}} \cdot \frac{H_{\text{эвм}}^{\text{а}}}{100} = \Pi_{\text{эвм}}^{\text{б}} \cdot \frac{H_{\text{эвм}}^{\text{а}}}{100}, \quad (6.18)$$

где $\Pi_{\text{эвм}}$ – цена ЭВМ на момент ее выпуска (750 руб.);

$k_{\text{у}}$ – коэффициент удорожания ЭВМ (принят $k_{\text{у}} = 1$);

$k_{\text{м}}$ – коэффициент, учитывающий затраты на монтаж и транспортировку электронно-вычислительных машин ($k_{\text{м}} = 1,05$);

$H_{\text{эвм}}^{\text{а}}$ – норма амортизационных отчислений на ЭВМ, % ($H_{\text{эвм}}^{\text{а}} = 10\%$);

$\Pi_{\text{эвм}}^{\text{б}}$ – балансовая стоимость ЭВМ, руб.

Из формулы 6.18 получим:

$$A_{\text{ЭВМ}} = 750 \cdot 1 \cdot 1,05 \cdot \frac{10}{100} = 78,75(\text{руб.}).$$

$$\Pi_{\text{ЭВМ}}^6 = 750 \cdot 1 \cdot 1,05 = 787,5 (\text{руб.}).$$

Годовые затраты на ремонт и техническое обслуживание ЭВМ укрупненно могут быть определены по формуле:

$$P_{\text{ЭВМ}} = \Pi_{\text{ЭВМ}}^6 \cdot k_{\text{ро}}, \quad (6.19)$$

где $k_{\text{ро}}$ – коэффициент, учитывающий затраты на ремонт и техническое обслуживание ЭВМ, в том числе затраты на запчасти, зарплату ремонтного персонала и др. ($k_{\text{ро}} = 0,13$).

Тогда

$$P_{\text{ЭВМ}} = 787,5 \cdot 0,13 = 102,38(\text{руб.}).$$

Годовая величина амортизационных отчислений на реновацию производственных площадей, занятых ЭВМ определяется по формуле:

$$A_{\text{пл}} = \Pi_{\text{пл}}^6 \cdot \frac{H_{\text{пл}}^a}{100} = S_{\text{ЭВМ}} \cdot k_d \cdot \Pi_{\text{пл}} \cdot \frac{H_{\text{пл}}^a}{100}, \quad (6.20)$$

где $\Pi_{\text{пл}}^6$ – балансовая стоимость площадей, руб;

$H_{\text{пл}}^a$ – норма амортизационных отчислений на производственные площади, % ($H_{\text{пл}}^a = 1,2\%$);

$S_{\text{ЭВМ}}$ – площадь, занимаемая ЭВМ (1 кв.м.);

k_d – коэффициент, учитывающий дополнительную площадь ($k_d = 3$);

$\Pi_{\text{пл}}$ – цена 1 квадратного метра производственной площади (принято 300 руб.).

$$A_{\text{пл}} = 1 \cdot 3 \cdot 300 \cdot \frac{1,2}{100} = 10,8(\text{руб.}).$$

Балансовая стоимость площади

$$\Pi_{\text{пл}}^6 = 1 \cdot 3 \cdot 300 = 900(\text{руб.}).$$

Годовые затраты на ремонт и содержание производственных площадей

укрупненно могут быть определены по формуле:

$$P_{пл} = P_{пл} = \Pi_{пл}^6 \cdot k_{рэ}, \quad (6.21)$$

где $k_{рэ}$ – коэффициент, учитывающий затраты на ремонт и эксплуатацию производственных площадей ($k_{рэ} = 0,05$).

$$P_{пл} = 900 \cdot 0,05 = 45(\text{руб.}).$$

Годовая величина арендных платежей за помещение, занимаемое ЭВМ, рассчитывается по формуле:

$$P_{ар} = S_{эвм} \cdot k_d \cdot k_{ар} \cdot k_{комф} \cdot k_{пов} \cdot 12, \quad (6.22)$$

где $S_{эвм}$ – площадь, занимаемая ЭВМ (1 кв.м);

k_d – коэффициент, учитывающий дополнительную площадь ($k_d = 3$);

$k_{ар}$ – ставка арендных платежей за помещение (базовая ставка арендной платы – 12 руб. за 1 кв. метр);

$k_{комф}$ – коэффициент комфортности помещения (0,8);

$k_{пов}$ – повышающий коэффициент, учитывающий географическое размещение площади (0,85).

Тогда

$$P_{ар} = 1 \cdot 3 \cdot 12 \cdot 0,8 \cdot 0,85 \cdot 12 = 293,76(\text{руб.}).$$

Годовой фонд времени работы ЭВМ определяется исходя из режима ее работы и может быть рассчитан по формуле:

$$\Phi_{эвм} = t_{сс} \cdot T_{ст}, \quad (6.23)$$

где $t_{сс}$ – среднесуточная фактическая загрузка ЭВМ (принято 8 час.);

$T_{ст}$ – среднее количество дней работы ЭВМ в год (принято 250 дней).

$$\Phi_{эвм} = 8 \cdot 250 = 2000(\text{час}).$$

Таким образом, получим стоимость машино-часа работы ЭВМ:

$$S_{\text{мч}} = 0,01026 + \frac{(115,5 + 150,15 + 10,8 + 45 + 293,76)}{2000} = 0,318(\text{руб.}).$$

6.1.5 Определение ожидаемого прироста прибыли в результате внедрения программного продукта

Внедрение ПП может обеспечить пользователю ожидаемый прирост прибыли за счет сокращения трудоемкости решения задачи, являющейся предметом автоматизации и, как результат, снижения текущих затрат, связанных с решением данной задачи.

В том случае, если внедряемый ПП заменяет ручной труд, то производится сопоставление текущих затрат, связанных с решением задачи в ручном режиме и автоматизированном. В том случае, если разрабатываемая задача внедряется взамен уже функционирующей или она представляет собой модификацию существующей задачи, необходимо осуществить сравнение затрат на создание и функционирование старой и новой задачи.

6.2.1 Определение годовых эксплуатационных расходов при ручном решении поставленной задачи

Годовые эксплуатационные расходы при ручной обработке информации (ручном решении задачи) определяются по формуле:

$$З_{\text{п}} = T_{\text{р}} \cdot k \cdot t_{\text{чр}} \cdot (1 + q) \cdot (1 + a) \cdot (1 + b), \quad (6.24)$$

где $T_{\text{р}}$ – трудоемкость разового решения задачи вручную;

k – периодичность решения задачи в течение года;

$t_{\text{чр}}$ – среднечасовая ставка работника, осуществляющего ручной расчет задачи (0,594 руб.);

q – коэффициент, учитывающий процент премий (принят 0,3);

a – коэффициент, учитывающий доп. заработную плату (принят 0,2);

b – коэффициент, учит. отчисления от фонда заработной платы (принят 0,346).

Трудоемкость разового ручного решения задачи определяется по формуле:

$$T_{\text{р}} = \sum_{i=1}^n T_{\text{pi}}, \quad (6.25)$$

При расчете трудоемкости подразумевается, что человек обладает всей доступной информацией, которая может быть сохранена в разработанном приложении. В данном

расчете не учитывается время на поиск информации.

Для начала выполнения задачи необходимо включить компьютер: $T_{p1} = 0,1$ ч. Затем создать электронный документ: $T_{p2} = 0,01$ ч. После добавить темы курсовых проектов: $T_{p3} = 0,25$ ч. Далее необходимо разработать график процентовки: $T_{p4} = 0,3$ ч. Затем добавить даты консультаций: $T_{p5} = 0,2$ ч. После этого необходимо принять курсовые проекты $T_{p5} = 6$ ч.

Периодичность решения задачи в течение года зависит непосредственно от популярности технологии. Периодичность решения задачи в год составляет порядка 500 раз.

$$T_p = 0,1 + 0,01 + 0,25 + 0,3 + 0,2 + 6 = 6,86(\text{чел.} - \text{ч}).$$

$$Z_{\pi} = 6,86 \cdot 500 \cdot 0,594 \cdot (1 + 0,3) \cdot (1 + 0,2) \cdot (1 + 0,346) = 4278,09(\text{руб.}).$$

В состав затрат, связанных с решением задачи, включаются также затраты, связанные с эксплуатацией ЭВМ.

Затраты на оплату аренды ЭВМ для решения задачи определяются по следующей формуле:

$$Z_a = T_p \cdot k \cdot S_{\text{мч}}, \quad (6.26)$$

где $S_{\text{мч}}$ - стоимость одного машино-часа работы ЭВМ.

Отсюда

$$Z_a = 6,86 \cdot 500 \cdot 0,318 = 1090,74(\text{руб.}).$$

Годовые текущие затраты, связанные с эксплуатацией задачи, определяются по формуле:

$$Z_{\text{тр}} = Z_{\pi} + Z_a, \quad (6.27)$$

где Z_{π} – затраты на заработную плату пользователя программы;

Z_a – затраты на оплату аренды ЭВМ при решении задачи.

$$Z_{\text{тр}} = 4278,09 + 1090,74 = 5368,83(\text{руб.}).$$

6.2.2 Определение годовых затрат, связанных с эксплуатацией задачи

Для расчета годовых текущих затрат, связанных с эксплуатацией ПП, необходимо определить время решения данной задачи на ЭВМ.

Время решения задачи на ЭВМ определяется по формуле:

$$T_3 = (T_{\text{зап}} + T_{\text{ввт}} + T_{\text{ввэ}} + T_{\text{ввд}}) \cdot \frac{1+d_{\text{пз}}}{60}, \quad (6.28)$$

где $T_{\text{зап}}$ – время запуска компьютера, мин (принято 6 мин);

$T_{\text{ввт}}$ – время ввода тем курсовых проектов, мин (принято 5 мин);

$T_{\text{ввэ}}$ – время ввода этапов процентовки, мин (принято 8 мин);

$T_{\text{ввд}}$ – время ввода дат консультаций, мин (принято 8 мин);

$d_{\text{пз}}$ - коэффициент, учитывающий подготовительно-заключительное время (принято $d_{\text{пз}}=0,1$).

$$T_3 = (6 + 5 + 8 + 8) \cdot \frac{1 + 0,1}{60} = 0,495(\text{чел.} - \text{ч}).$$

На основе рассчитанного времени решения задачи может быть определена заработная плата пользователя данного ПП. Затраты на заработную плату пользователя ПП определяются по формуле:

$$З_{\text{п}} = T_3 \cdot k \cdot t_{\text{чп}} \cdot (1 + q) \cdot (1 + a) \cdot (1 + b), \quad (6.29)$$

где T_3 – время решения задачи на ЭВМ, час;

$t_{\text{чп}}$ – среднечасовая ставка пользователя программы, руб. (определяется аналогично ставке работника, осуществляющего ручной расчет).

$$З_{\text{п}} = 0,495 \cdot 500 \cdot 0,594 \cdot (1 + 0,3) \cdot (1 + 0,2) \cdot (1 + 0,346) = 308,7(\text{руб.}).$$

В состав затрат, связанных с решением задачи, включаются также затраты, связанные с эксплуатацией ЭВМ.

Затраты на оплату аренды ЭВМ для решения задачи определяются по следующей формуле:

$$З_{\text{а}} = T_3 \cdot k \cdot S_{\text{мч}}, \quad (6.30)$$

где $S_{\text{мч}}$ - стоимость одного машино-часа работы ЭВМ.

Отсюда

$$З_{\text{а}} = 0,495 \cdot 500 \cdot 0,318 = 78,7(\text{руб.}).$$

Годовые текущие затраты, связанные с эксплуатацией задачи, определяются по формуле:

$$З_{\text{тп}} = З_{\text{п}} + З_{\text{а}}, \quad (6.31)$$

где $З_{\text{п}}$ – затраты на заработную плату пользователя программы;

$З_{\text{а}}$ – затраты на оплату аренды ЭВМ при решении задачи.

$$З_{\text{тп}} = 308,7 + 78,7 = 387,4(\text{руб.}).$$

6.2.3 Определение ожидаемого прироста прибыли в результате внедрения программного продукта

Ожидаемый прирост прибыли в результате внедрения задачи взамен ручного ее расчета укрупнено может быть определен по формуле:

$$П_y = (З_{\text{тр}} - З_{\text{тп}}) \cdot (1 - С_{\text{нп}}), \quad (6.32)$$

где $С_{\text{нп}}$ – ставка налога на прибыль (18%) [12].

$$П_y = (5368,83 - 387,4) \cdot (1 - 0,18) = 4084,77(\text{руб.}).$$

6.3 Расчёт показателей эффективности использования программного продукта

Для определения годового экономического эффекта от разработанной программы необходимо определить суммарные капитальные затраты на разработку и внедрения программы по формуле:

$$K_o = K_z + Ц_{\text{пр}}, \quad (6.33)$$

где K_z - капитальные и приравненные к ним затраты;

$Ц_{\text{пр}}$ - отпускная цена программы.

Капитальные и приравненные к ним затраты определяются, если необходимо приобретение новой ЭВМ для решения комплекса задач, в который входит рассматриваемая, по формуле:

$$K_3 = \frac{\Pi_{\text{ЭВМ}}^{\text{б}} \cdot T_3 \cdot K}{\Phi_{\text{ЭВМ}}} \quad (6.34)$$

где $\Pi_{\text{ЭВМ}}^{\text{б}}$ – балансовая стоимость комплекта вычислительной техники, необходимого для решения задачи (принято за 750 руб.).

$$K_3 = \frac{750 \cdot 0,495 \cdot 500}{2000} = 92,81 \text{ (руб.)}.$$

$$K_0 = 92,81 + 8821,57 = 8914,38 \text{ (руб.)}.$$

Годовой экономический эффект от сокращения ручного труда при обработке информации определяется по формуле:

$$\text{ЭФ} = \Pi_y - E \cdot K_0 = \Pi_y - E \cdot (K_3 + \Pi_{\text{пр}}), \quad (6.35)$$

где E – коэффициент эффективности, равный ставке за кредиты на рынке долгосрочных кредитов ($E = 0,2$).

$$\text{ЭФ} = 4084,77 - 0,2 \cdot 8914,38 = 2301,9 \text{ (руб.)}.$$

Срок возврата инвестиций определяется по формуле:

$$T_{\text{в}} = \frac{K_0}{\Pi_y}; \quad (6.36)$$

$$T_{\text{в}} = \frac{8914,38}{4084,77} = 2,18 \text{ (г.)}.$$

В таблице 6.1 приведены технико-экономические показатели разработки программного продукта.

Таким образом, разрабатываемый в дипломном проекте «Клиентская часть для работы с курсовым проектом обучающей системы CATS» модуль с отпускной ценой в 8821,57 руб. будет иметь ожидаемый прирост прибыли на сумму 4084,77 руб. Годовой экономический эффект составит 2301,9 руб., а срок возврата инвестиции составит 2,18 лет, т.е. примерно 2 года и 2 месяца.

Таблица 6.1 – Техничко-экономические показатели проекта

Наименование задачи	Варианты	
	базовый	проектный
Трудоемкость решения задачи, час	6,86	0,495
Периодичность решения задачи, раз/год	500	500
Годовые текущие затраты, связанные с решением задачи, руб.	5368,83	387,4
Отпускная цена программы, руб.		8821,57
Степень новизны программы		В
Группа сложности алгоритма		2
Прирост условной прибыли, руб.		4084,77
Годовой экономический эффект, руб.		2301,9
Срок возврата инвестиций, лет		2,18

7 ОХРАНА ТРУДА

7.1 Производственная санитария, техника безопасности и пожарная профилактика

Работающие с ПЭВМ могут подвергаться воздействию различных опасных и вредных производственных факторов, основными из которых являются повышенные уровни: электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; статического электричества; запыленности воздуха рабочей зоны; повышенное или пониженное содержание аэроионов в воздухе рабочей зоны; повышенный или пониженный уровень освещенности рабочей зоны, содержание в воздухе рабочей зоны оксида углерода, озона, аммиака, фенола, формальдегида и полихлорированных фенилов; напряжение зрения, памяти, внимания; длительное статическое напряжение; большой объем информации, обрабатываемой в единицу времени; монотонность труда; нерациональная организация рабочего места; эмоциональные перегрузки.

Работа с ПЭВМ проводится в соответствии с Санитарными нормами и правилами «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиеническим нормативом «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами» утвержденными постановлением Министерства здравоохранения от 28.06.2013 г. № 59 и Типовой инструкцией по охране труда при работе с персональными ЭВМ, утвержденной постановлением Министерства труда и социальной защиты от 24.12.2013 № 130 [13, 14].

Площадь помещения на одного пользователя ПЭВМ на базе плоских дискретных экранов (жидкокристаллические, плазменные) составляет не менее 4,5 м².

7.1.1 Метеоусловия

В производственных помещениях, в которых работа с использованием ВДТ, ЭВМ или ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники) или связана с нервно-эмоциональным напряжением, обеспечиваются оптимальные параметры микроклимата для категории работ 1а и 1б, представленные в таблице 7.1 [13, 14].

Таблица 7.1 - Оптимальные параметры микроклимата для помещений с ВДТ, ЭВМ и ПЭВМ

Период года	Категория работ	Температура воздуха, °С, не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	легкая-1а	22-24	40-60	0,1
	легкая-1б	21-23	40-60	0,1
Теплый	легкая-1а	23-25	40-60	0,1
	легкая-1б	22-24	40-60	0,2

Оптимальные микроклиматические условия - это сочетание показателей микроклимата (температура воздуха, относительная влажность воздуха, скорость движения воздуха, интенсивность инфракрасных излучений), которое обеспечивает человеку ощущение теплового комфорта в течение рабочей смены без нарушения механизмов терморегуляции и не вызывает отклонений в здоровье. При этом создаются предпосылки для высокого уровня работоспособности.

Работа с компьютером относится к категории 1а (работы, производимые сидя и сопровождающиеся незначительным физическим напряжением, при которых расход энергии составляет до 120 ккал/ч, т.е. до 139 Вт).

Интенсивность теплового излучения работающих от нагретых поверхностей технологического оборудования, осветительных приборов, инсоляции на постоянных местах не превышает значений, указанных в таблице 7.2 [13].

Таблица 7.2 - Предельно допустимые уровни интенсивности излучения в инфракрасном и видимом диапазоне излучения на расстоянии 0,5 м со стороны экрана ВДТ, ЭВМ и ПЭВМ

Диапазоны длин волн	400-760 нм	760-1050 нм	свыше 1050 нм
Предельно допустимые уровни	0,1 Вт/м ²	0,05 Вт/м ²	4,0 Вт/м ²

Для создания нормальных метеорологических условий наиболее целесообразно уменьшить тепловыделения от самого источника — монитора, что предусматривается при разработке его конструкции.

В производственных помещениях для обеспечения необходимых показателей микроклимата предусмотрены системы отопления, вентиляции и кондиционирования воздуха.

7.1.2 Вентиляция и отопление

Воздух рабочей зоны помещения соответствует санитарно-гигиеническим требованиям по содержанию вредных веществ и частиц пыли, приведенным в Санитарных нормах и правилах «Требованию к контролю воздуха рабочей зоны», Гигиеническом нормативе «Предельно допустимые концентрации вредных веществ в воздухе рабочей зоны», утв. пост. Министерства здравоохранения от 10.10.2017 г. № 92.

В помещениях, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы.

Уровни положительных и отрицательных аэроионов, а также коэффициент униполярности в воздухе всех помещений, где расположены ПЭВМ, соответствуют значениям, указанным в таблице 7.3.

Таблица 7.3 - Уровни ионизации и коэффициент униполярности воздуха помещений при работе с ВДТ, ЭВМ и ПЭВМ

Уровни	Число ионов в 1 см ³ воздуха		Коэффициент униполярности (У)
	<i>n</i> +	<i>n</i> -	
Минимально допустимые	400	600	0,4 ≤ У < 1,0
Оптимальные	1500-3000	3000-5000	
Максимально допустимые	50000	50000	

Одним из мероприятий по оздоровлению воздушной среды является устройство вентиляции и отопления. Задачей вентиляции является обеспечение чистоты воздуха и заданных метеорологических условий на рабочих местах. Чистота воздушной среды достигается удалением загрязненного или нагретого воздуха из помещения и подачей в него свежего воздуха. Для поддержания нормального микроклимата необходим достаточный объем вентиляции, для чего в вычислительном центре предусматривается кондиционирование воздуха, осуществляющее поддержание постоянных параметров микроклимата в помещении независимо от наружных условий.

Параметры микроклимата поддерживаются в холодное время за счет системы водяного отопления с нагревом воды до 100°С, а в теплый - за счет кондиционирования, с параметрами, отвечающими требованиям СНБ 4.02.01-03 [15].

7.1.3 Освещение

Помещения для эксплуатации ВДТ, ЭВМ и ПЭВМ имеют естественное и искусственное освещение. Естественное освещение на рабочих местах осуществляется через световые проемы, ориентированные преимущественно на север, северо-восток, восток, запад или северо-запад и обеспечивает коэффициент естественной освещенности не ниже 1,5 %. Оконные проемы оборудованы регулирующими устройствами типа жалюзи, занавесей.

Для внутренней отделки интерьера помещений используются материалы с коэффициентом отражения для потолка – 0,7 - 0,8; для стен – 0,5- 0,6; для пола – 0,3-0,5.

Искусственное освещение в помещениях осуществляется системой общего равномерного освещения. В случаях преимущественной работы с документами применяют системы комбинированного освещения. Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 люкс. Освещенность поверхности экрана не более 300 люкс.

В качестве источников света при искусственном освещении применяем люминесцентные лампы типа ЛБ. Коэффициент запаса для осветительных установок общего освещения принимается равным 1,4. Коэффициент пульсации не превышает 5 %.

7.1.4 Шум

Основными источниками шума в помещениях, оборудованных ПЭВМ, являются принтеры, множительная техника и оборудование для кондиционирования воздуха, в самих ПЭВМ — вентиляторы систем охлаждения и трансформаторы.

Нормированные уровни шума согласно Санитарных норм и правил «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиенических нормативов «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами» приведены в таблице 7.4 и обеспечиваются путем использования малошумного оборудования, применением звукопоглощающих материалов для облицовки помещений, а также различных звукопоглощающих устройств (перегородки и т. д.).

Таблица 7.4 – Предельно-допустимые уровни звука, эквивалентные уровни звука и уровни звукового давления в октавных полосах частот при работе с ВДТ, ЭВМ и ПЭВМ и периферийными устройствами

Категория нормы шума	Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалент ные уровни звука, дБА
	31.5	63	125	250	500	1000	2000	4000	8000	
I	86	71	61	54	49	45	42	40	38	50
II	93	79	70	63	58	55	52	50	49	60
III	96	83	74	68	63	60	57	55	54	65
IV	103	91	83	77	73	70	68	66	64	75

7.1.5 Электробезопасность

Помещение вычислительного центра по степени опасности поражения электрическим током относится к помещениям без повышенной опасности.

Основные меры защиты от поражения током:

- изоляция и недоступность токоведущих частей;
- защитное заземление ($R_3 = 4 \text{ Ом}$ ГОСТ 12.1.030 - 81).

Первая помощь при поражениях электрическим током состоит из двух этапов: освобождение пострадавшего от действия тока и оказание ему доврачебной медицинской помощи. После освобождения пострадавшего от действия электрического тока необходимо оценить его состояние. Во всех случаях поражения электрическим током необходимо вызвать врача независимо от состояния пострадавшего.

7.1.6 Излучение

При работе с дисплеем могут возникнуть следующие опасные факторы: электромагнитные поля, электростатические поля, ультрафиолетовое и инфракрасное излучение.

Уровни физических факторов, создаваемые ВДТ, ЭВМ, ПЭВМ и периферийными устройствами, не превышают предельно-допустимые уровни: электромагнитных и электростатических полей, указанных в таблицах 7.5 и 7.6, ультрафиолетового, указанного в таблице 7.7, установленных Гигиеническим нормативом «Предельно-

допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами».

Таблица 7.5 - Предельно допустимые уровни электромагнитных полей от экранов ВДТ, ЭВМ и ПЭВМ

Наименование параметра	Предельно-допустимые уровни
Напряженность электрического поля в диапазоне частот: 5 Гц-2 кГц 2-400 кГц	не более 25,0 В/м не более 2,5 В/м
Плотность магнитного потока магнитного поля в диапазоне частот: 5 Гц-2 кГц 2-400 кГц	не более 250 нТл не более 25 нТл
Напряженность электростатического поля	не более 15 кВ/м

Таблица 7.6 - Предельно допустимые уровни электромагнитных полей при работе с ВДТ, ЭВМ, ПЭВМ от клавиатуры, системного блока, манипулятора «мышь», беспроводных системам передачи информации и иных периферийных устройств

Диапазоны частот	0,3-300 кГц	0,3-3 МГц	3-30 МГц	30-300 МГц	0,3-300 ГГц
Предельно допустимые уровни	25 В/м	15 В/м	10 В/м	3 В/м	10 мкВт/см ²

Таблица 7.7 - Предельно допустимые уровни интенсивности излучения в ультрафиолетовом диапазоне на расстоянии 0,5 м со стороны экрана ВДТ, ЭВМ и ПЭВМ

Диапазоны длин волн	200-280 нм	280-315 нм	315-400 нм
Предельно допустимые уровни	не допускается	0,0001 Вт/м ²	0,1 Вт/м ²

Наиболее эффективным и часто применяемым методом защиты от электромагнитных излучений является установка экранов. Экранируют либо источник

излучения, либо рабочее место. Часто экран устанавливают непосредственно на монитор.

При работе монитора на экране кинескопа накапливается электростатический заряд, создающий электростатическое поле. При этом персонал, работающий с монитором, приобретают электростатический потенциал. Заметный вклад в общее электростатическое поле вносят электризующиеся от трения поверхности клавиатуры и мыши.

7.1.7 Пожарная безопасность

По взрывопожарной и пожарной опасности помещения и здания относятся по ТКП 474-2013 к категории Д. Здания для ВЦ и части зданий другого назначения, в которых предусмотрено размещение ЭВМ, относятся к 2 степени огнестойкости согласно ТКП 45-2.02-315-2018 [16].

Для предотвращения распространения огня во время пожара с одной части здания на другую устраивают противопожарные преграды в виде стен, перегородок, дверей, окон. Особое требование предъявляется к устройству и размещению кабельных коммуникаций.

Нормы первичных средств пожаротушения для вычислительных центров приведены в таблице 7.8.

Таблица 7.8 - Примерные нормы первичных средств пожаротушения для вычислительного центра

Помещение	Площадь, м ²	Углекислотные огнетушители ручные	Порошковые огнетушители
Вычислительный центр	100	1	1

Для ликвидации пожаров в начальной стадии применяются первичные средства пожаротушения: внутренние пожарные водопроводы, огнетушители типа ОВП-10, ОУ-2, асбестовые одеяла и др.

Эвакуация персонала вычислительного центра осуществляется через эвакуационные выходы. Количество и общая ширина эвакуационных выходов определяются в зависимости от максимального возможного числа эвакуирующихся через них людей и предельно допустимого расстояния от наиболее удаленного места возможного пребывания людей до ближайшего эвакуационного выхода согласно ТКП 45-2.02-315-2018 [17, 18].

Расчетное время эвакуации устанавливается по реальному расчету времени движения одного или нескольких потоков людей через эвакуационные выходы из наиболее удаленных мест размещения людей. Необходимое время эвакуации устанавливается на основе данных о критической продолжительности пожара с учетом степени огнестойкости здания, категории производства по взрывной и пожарной опасности. Для успешной эвакуации необходимо, чтобы расчетное время было меньше необходимого.

7.2 Организация рабочего места пользователя ПЭВМ

Рабочее место - это часть пространства, в котором персонал осуществляет трудовую деятельность, и проводит большую часть рабочего времени. Рабочее место, хорошо приспособленное к трудовой деятельности, правильно и целесообразно организованное, в отношении пространства, формы, размера обеспечивает ему удобное положение при работе и высокую производительность труда при наименьшем физическом и психическом напряжении.

При правильной организации рабочего места производительность труда возрастает с 8 до 20 %. Конструкция рабочего места и взаимное расположение всех его элементов соответствует антропометрическим, физическим и психологическим требованиям ГОСТ 12.2.032-78.

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) составляет не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных производственных факторов должны размещаться в изолированных кабинах с организованным воздухообменом.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5-2,0 м.

Экран видеомонитора должен находиться на расстоянии 0,6-0,7 м от глаз пользователя, но не ближе 0,5 м с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола обеспечивает оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола имеет коэффициент отражения 0,5-0,7.

Конструкция рабочего стула (кресла) обеспечивает поддержание рациональной

рабочей позы при работе с ПЭВМ, позволяет изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) выбирают с учетом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочий стул (кресло) подъемно-поворотный, регулируемый по высоте и углам наклона сиденья и спинки, а расстояние спинки от переднего края сиденья, при этом регулировка каждого параметра независимая, легко осуществляемая и имеет надежную фиксацию. Поверхность сиденья, спинки и других элементов рабочего стула (кресла) полумягкая, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Поверхности периферийных устройств (клавиатура, манипулятор «мышь», принтер, сканер и другое) необходимо протирать мягкой ветошью с применением специальных или бытовых чистящих средств, не содержащих кислот и отбеливателей, не реже 1 раза в неделю, а при необходимости и чаще. Протирка периферийных устройств производится при выключенном оборудовании.

ЗАКЛЮЧЕНИЕ

В рамках дипломного проекта было спроектировано веб-приложение для управления курсовыми проектами для системы управления учебным процессом. Для этого был произведен анализ существующего веб-интерфейса, построена логическая и физическая модель данных.

В новой версии интерфейса решена проблема его адаптивности, а также обновлен дизайн пользовательского интерфейса.

Для реализации приложения был выбран современный веб-фреймворк Angular, т.к. он реализует современные паттерны программирования, имеет подробную и понятную документацию, а также мощную поддержку и большое сообщество разработчиков.

Приложение позволит преподавателям:

- создавать темы курсовых проектов для определенных групп и назначать их студентам;
- редактировать листы задания, создавать этапы процентов проекта и выставлять студентам оценки по каждому этапу;
- отслеживать посещаемость консультаций;
- принимать готовые работы от студентов.

Студенты будут иметь возможность:

- выбирать тему курсового проекта;
- скачивать лист задания;
- получать информацию о датах процентов и консультаций;
- загружать готовые работы для защиты.

Приложение является частью распределённой системы и выступает в роли веб-клиента. Оно будет взаимодействовать с сервером через REST-интерфейс. Таким образом, внешний интерфейс приложения-сервера можно будет использовать и в других приложениях, например в приложениях для смартфонов на базе Android или iOS.

Было проведено критическое и углубленное тестирование разработанного приложения. Это позволило выявить некоторые ошибки, которые затем были исправлены.

Разработка данного приложения оправдана тем, что с его помощью можно значительно упростить организацию учебного процесса. Кроме того, результаты расчета экономических показателей говорят об экономической выгоде от данной разработки.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Попова, Ю.Б. Классификация автоматизированных систем управления обучением / Ю.Б. Попова // Системный анализ и прикладная информатика. – 2016. – №2. – С. 51–58.
2. Попова, Ю.Б. От LMS к адаптивным обучающим системам / Ю.Б. Попова // Системный анализ и прикладная информатика. – 2019. – №2. – С. 58-64.
3. Попова, Ю.Б. Автоматизированная система управления обучением CATS (Care About The Students) / Ю.Б. Попова // Наука и техника. – 2019. – №4 (18). – С. 339-349.
4. Material Design Introduction [Электронный ресурс] – Режим доступа: <https://material.io/design/introduction#goals>, свободный. – Загл. с экрана. Яз. Англ. Дата доступа: 12.04.2020.
5. Angular Docs [Электронный ресурс] – Режим доступа: <https://angular.io/docs>, свободный. – Загл. с экрана. Яз. Англ. Дата доступа: 12.04.2020.
6. Angular Material UI component library [Электронный ресурс] – Режим доступа: <https://material.angular.io/components/categories>, свободный. – Загл. с экрана. Яз. Англ. Дата доступа: 12.04.2020.
7. Angular Architecture [Электронный ресурс] – Режим доступа: <https://www.ngdevelop.tech/angular/architecture>, свободный. – Загл. с экрана. Яз. Англ. Дата доступа: 12.04.2020.
8. Критическое тестирование [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/149903/>, свободный. – Загл. с экрана. Дата доступа: 12.04.2020.
9. Налог на добавленную стоимость [Электронный ресурс] - Режим доступа: <https://myfin.by/wiki/term/nalog-na-dobavlennuyu-stoimost>, - Загл. с экрана. Яз. рус Дата доступа: 12.04.2020.
10. Тарифная ставка первого разряда [Электронный ресурс]. Режим доступа: myfin.by/info/tarifnaya-stavka-pervogo-razryada, свободный. – Загл. с экрана – Яз. Рус. (дата доступа: 30.04.2018)
11. Тарифы на электроэнергию для населения в Беларуси / [Электронный ресурс]. Режим доступа: myfin.by/wiki/term/tarify-na-elektroenergiyu-dlya-naseleniya-v-belarusi, свободный. – Загл. с экрана – Яз. Рус. (дата доступа: 30.04.2018)
12. Налог на прибыль [Электронный ресурс]. Режим доступа: myfin.by/wiki/term/nalog-na-pribyl, свободный. – Загл. с экрана – Яз. Рус. (дата доступа: 30.04.2018)
13. Санитарные нормы и правила «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиенический норматив «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами», утвержденные постановлением МЗ РБ от 28.06.2013 г. №

14. Типовая инструкция по охране труда при работе с персональными ЭВМ утв. Постановлением Министерства труда и социальной защиты от 24.12.2013 № 130.
15. СНБ 4.02.01-03 Отопление, вентиляция и кондиционирование воздуха. Утверждены Приказом Министерства архитектуры и строительства Республики Беларусь от 30 декабря 2003 г. № 259.
16. ТКП 45-2.02-142-2011 Здания, строительные конструкции, материалы и изделия. Правила пожарно-технической классификации.
17. ТКП 45-2.02-22-2006 «Здания и сооружения. Эвакуация людей при пожаре» Строительные нормы проектирования. Утвержден и введен в действие приказом Министерства архитектуры и строительства Республики Беларусь от 2 апреля 2013 г. № 101.
18. ТКП 45-2.02-279-2013. Здания и сооружения. Эвакуационные пути и выходы. Строительные нормы проектирования от 2 апреля 2013 г.