

РЕФЕРАТ

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ КОНТРОЛЯ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА C#: дипломный проект / В. В. Садовский. – Минск : БГУИР, 2020, – п.з. – 103 с., чертежей (плакатов) – 6 л. формата А1.

Объектом исследования является процесс контроля знаний.

Цель дипломного проекта: разработка средств и методов компьютеризированного контроля знаний.

Разработка данного программного средства обеспечит возможности формирования тестового материала, организации и проведения тестирования с использованием динамически формируемых тестовых наборов, учитывающих рейтинг обучающегося и результаты прохождения тестирования.

Проанализированы существующие программные средства для тестирования и контроля знаний и выполнена постановка задачи.

Выполнено моделирование предметной области, разработаны функциональная модель в нотации IDEF0, диаграмма вариантов использования и функциональные требования к программному средству.

В процессе проектирования программного средства тестирования и контроля знаний разработаны схема модели базы данных, а также ряд диаграмм в нотации UML. На основании проектной документации выполнена реализация программного средства. Проведено тестирование и разработана использования программного средства.

Проведенные расчеты технико-экономического обоснования проекта показали эффективность разработки и применения программного средства у пользователя.

Разработанное программное средство предоставляет средства и методы для компьютерного тестирования и контроля знаний. Разработанное программное средство позволяет решать специфические задачи. Его преимуществом является наличие функции автоматического формирования тестовых наборов. Также разрабатываемая система предоставляет доступ по ролям (администратор, преподаватель, обучающийся) и веб-интерфейс для всех категорий пользователей.

СОДЕРЖАНИЕ

Введение	6
1 Анализ предметной области и разработка требований к программному средству	8
1.1 Анализ исходных требований	8
1.2 Сравнительный анализ программных средств для тестирования и контроля знаний	9
1.3 Постановка задачи на разработку автоматизированной системы контроля знаний	13
2 Моделирование предметной области	19
2.1 Функциональная модель	19
2.2 Логическая модель данных	23
2.3 Функциональная спецификация	28
3 Проектирование программного средства	33
3.1 Структура системы	33
3.2 Схема модели базы данных	34
3.3 Логическая структура системы	38
3.4 Взаимодействие классов	44
3.5 Реализация проекта приложения	45
3.6 Схема работы системы	47
3.7 Физическая структура системы	48
4 Тестирование программного средства	50
4.1 Программа испытаний	50
4.2 Методика испытаний	52
4.3 Результаты тестирования	53
5 Методика использования программного средства	56
5.1 Установка и настройка	56
5.2 Авторизация	57
5.3 Функции администратора	57
5.4 Функции преподавателя	61
5.5 Функции обучающегося	68
6 Техничко-экономическое обоснование разработки и реализации программного средства для автоматизации контроля знаний	71
6.1 Описание функций, назначения и потенциальных пользователей ПО	71
6.2 Расчет затрат на разработку ПО	71
6.3 Оценка результата (эффекта) от использования (или продажи) ПО	74
Заключение	78
Список использованных источников	79
ПРИЛОЖЕНИЕ А (обязательное) Фрагменты программного кода	81
ПРИЛОЖЕНИЕ Б (обязательное) Фрагменты cshtml-разметки	97

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения

.NET Framework – программная платформа, выпущенная компанией Microsoft в 2002 году, основанная на общезыковой среде исполнения, которая подходит для разных языков программирования.

CAM (Computer-Aided Manufacturing) – автоматизированная система, либо модуль автоматизированной системы, предназначенный для подготовки управляющих программ.

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

SQL – язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

UML – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

WPF – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework, использующая язык XAML.

Программное средство – это совокупность программных модулей, обеспечивающих автоматизацию выполнения операций с данными, представленными в формате той или иной информационной модели.

Программный модуль – программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

БД – база данных.

СУБД – система управления базой данных.

ВВЕДЕНИЕ

На сегодняшний день контроль успеваемости в рамках образовательного процесса может и должен строиться с использованием современных коммуникационных и информационных технологий, в том числе с применением специализированных программных систем. Проблемой является то, что представленное на рынке программное обеспечение для тестирования и контроля знаний имеет существенные ограничения как по наличию, так и по функциональным возможностям. Следовательно, актуальной является задача разработки программного средства для организации тестирования и контроля знаний, обладающего всем необходимым функционалом.

Целью дипломного проекта является разработка средств и методов компьютерного тестирования и контроля знаний.

Для достижения поставленной цели в дипломном проекте решаются следующие задачи:

- проанализировать существующие программные средства для тестирования и контроля знаний и выполнить постановку задачи;
- выполнить моделирование предметной области и разработать функциональные требования к программному средству;
- осуществить проектирование и реализацию программного средства;
- выполнить тестирование программного средства;
- разработать методику использования программного средства;
- выполнить технико-экономическое обоснование проекта.

Программное средство должно быть разработано в виде веб-приложения, использующего для хранения базу данных.

В первом разделе дипломного проекта проводится анализ исходных требований, проводится сравнительный анализ программных средств для тестирования и контроля знаний и, на основании полученных сведений, выполняет постановка задачи на разработку программного средства тестирования и контроля знаний.

Во втором разделе дипломного проекта выполняется моделирование предметной области, результатом чего являются функциональная модель процесса тестирования и контроля знаний в нотации IDEF0, функциональная модель предметной области в виде диаграммы вариантов использования, логическая модель данных, функциональная спецификация программного средства.

Третий раздел посвящен проектированию программного средства, в процессе которого разрабатываются структура системы, структура базы данных, логическая структура системы в виде диаграммы классов, UML-диаграммы, программный код и схема работы системы.

В четвертом разделе приводится программа и методика испытаний, а также результаты тестирования программного средства, проводимого на основании разработанной программы и методики испытаний.

В пятом разделе описывается методика использования программного средства, включая особенности установки и настройки веб-приложения, авторизацию, подробное описание выполнения функций по ролям пользователей.

В шестом разделе приводятся расчеты технико-экономического обоснования проекта программного средства, включая расчет затрат на разработку и оценку экономического эффекта от реализации программного средства.

Практическая значимость результатов проекта заключается в возможности применения разработанного программного средства для организации тестирования и контроля знаний в различных учебных заведениях.

Дипломный проект выполнен самостоятельно, проверен в системе «Антиплагиат». Процент оригинальности составляет 89,04%, что соответствует норме, установленной кафедрой программного обеспечения информационных технологий.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ

1.1 Анализ исходных требований

Объектом автоматизации настоящего дипломного проекта является процесс контроля знаний.

Целью автоматизации при этом является повышение эффективности процесса контроля знаний. Повышение эффективности достигается за счет автоматизации подготовки тестовых заданий, оперативного получения информации о результатах тестирования, автоматического формирования вопросов следующего теста в зависимости от результатов предыдущего и рейтинга тестирования.

Основными показателями процесса являются следующие данные:

- процент правильных ответов;
- рейтинг обучающегося.

Реализация процесса осуществляется следующими участниками образовательного процесса:

- администратор системы;
- преподаватель;
- обучающийся.

Для автоматизации выбраны следующие задачи:

- управление данными участников тестирования;
- разработка тестового материала;
- автоматическое формирование тестовых наборов;
- ответы на вопросы теста;
- обработка результатов тестирования;
- формирование отчетов по результатам тестирования.

Перечисленные задачи должны решаться в оперативном режиме.

В рамках настоящего проекта необходимо реализовать веб-приложение для персональных компьютеров, ноутбуков и мобильных устройств, размещаемых в рамках одной сети (локальная сеть, сеть Интернет), обеспечивающее решение перечисленных задач.

Суть функционирования программного средства заключается в следующем:

- администратор системы формирует списки пользователей – участников тестирования: преподавателей и обучающихся;
- преподаватели формируют тестовый материал, необходимый для контроля знаний;
- на основании сформированного тестового материала преподаватели назначают тестирование для групп обучающихся;
- система тестирования автоматически формирует тестовые наборы для обучающихся;

- обучающиеся отвечают на вопросы теста;
- система обрабатывает ответы на вопросы теста и сохраняет их;
- преподаватели на основании обработанных результатов формируют отчеты.

В приложении должен быть реализован дифференцированный доступ к функциям на основе пользовательских привилегий. Система должна представлять собой веб-приложение, использующее для хранения данных общую БД.

Также следует выделить следующие особенности разработки системы:

- необходимо реализовать удобный и понятный адаптивный интерфейс, позволяющий использовать приложение на экранах стационарных и мобильных компьютерных устройств;
- веб-приложение должно функционировать в актуальных версиях браузеров Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome и его производных;
- обязательна поддержка русского языка;
- необходимо реализовать автоматическое формирование тестовых наборов для обучающихся на основании заданных правил в зависимости от результатов предыдущего теста и рейтинга тестирования;
- тест должен предусматривать следующие типы ответов на вопросы: выбор одного варианта из нескольких возможных, выбор нескольких вариантов из нескольких возможных, произвольный ответ в текстовой либо числовой форме.

1.2 Сравнительный анализ программных средств для тестирования и контроля знаний

Среди программных продуктов, удовлетворяющих требованиям задания на разработку, можно выделить следующие:

- PikaTest;
- UniTest;
- Indigo.

1.2.1 PikaTest – это бесплатная программа для создания и проведения двухуровневых тестов с неограниченным количеством вопросов [1]. Она интересна в первую очередь тем, что является offline системой тестирования.

С помощью этой программы можно создать полноценный тест, с неограниченным количеством вопросов. Тест может содержать аудио- видео-файлы, а также таблицы и изображения. Имеется возможность добавления вопроса с вариантами ответов и без них, а также указание стоимости каждого отдельного вопроса. Программа позволяет создавать тесты с ограниченным временем прохождения. Файлы тестов сохраняются в формате *.pikatest,

который воспринимается только программой. Подробный отчет о тестировании сохраняется в формате *.txt и доступен для просмотра [4].

Прохождение тестирования осуществляется по следующей схеме:

- регистрация пользователя;
- настройки прохождения тестирования;
- вывод подробной статистики о результатах тестирования с детализацией до каждого вопроса;
- сохранение отчета о результатах тестирования;
- отправка отчета по электронной почте.

Основные преимущества:

- система независима и имеет небольшой размер (всего 1761 Kb);
- система является бесплатной;
- программа легка в обращении, и не требует каких-то специальных знаний;
- совместима со всеми версиями Windows, начиная с XP.

Недостатки:

- система работает в режиме offline, при этом справка по программе доступна только в интернете;
- отсутствие четкого разделения администратор-пользователь;
- незащищенность данных;
- отсутствие ряда типовых функций;
- недоступна для мобильных устройств.

1.2.2 UniTest – это комплексное программное решение для проведения компьютерного тестирования, функционально реализованное по САМ-технологии [2].

Программное средство предназначено для формирования банка тестовых заданий и организации процесса проверки знаний. Позволяет организовать процесс контроля знаний с помощью компьютерного тестирования в сети с использованием транспортного протокола TCP/IP, а также локального тестирования.

Система UniTest программно реализована с применением технологий Microsoft.NET 3.0, уникальных алгоритмов параллельной обработки информации и передовых средств криптографической защиты [3].

Система UniTest состоит из двух программных модулей: UniTest TUTOR и UniTest CLIENT.

Модуль TUTOR поддерживает работу двух категорий пользователей («Администратор», «Тьютор») и предназначен для обеспечения работы с банками тестовых заданий, которые кроме текстовой части могут включать в себя различные мультимедиа-компоненты: статическую и динамическую графику, Shockwave Flash-анимацию, видео- и аудио фрагменты. Модуль TUTOR позволяет организовать хранение тестовых заданий по секциям, а также создавать любое количество сценариев тестирования, дает возможность

ведения электронных журналов с результатами тестирования, обеспечивает структурирование студентов по подразделениям и учебным группам.

Модуль CLIENT предназначен для категории пользователей «Студент», обеспечивает доступ к готовым тестам системы UniTest и виртуальной зачетной книжки с результатами тестирования [3].

Основные преимущества:

- удобный интерфейс;
- поддержка всех основных и ряда дополнительных типов вопросов;
- возможность проведения тестирования как локально, так и по сети;
- поддержка большого числа языков;
- наличие дополнительных функций.

Недостатки:

- высокие требования к техническим средствам;
- недоступна для мобильных устройств.

1.2.3 INDIGO – представляет собой мультифункциональный комплекс программного обеспечения, позволяющий автоматизировать процесс проведения тестирования и обработки результатов [4].

Система «INDIGO» является универсальным инструментом, который можно использовать для решения широкого спектра задач:

- тестирование и контроль знаний учащихся;
- определение профессионального уровня сотрудников;
- проведение психологического тестирования;
- проведение опросов;
- организация олимпиад и конкурсов.

Программное средство предоставляет следующие функциональные возможности:

- система тестирования устанавливается на один компьютер-сервер с помощью инсталляционного пакета;
- все данные хранятся централизованно в базе данных системы;
- администраторы работают через программу клиент;
- функции администратора: создание и редактирование тестов, управление базой пользователей, назначение тестов пользователям, проведение тестирования, просмотр результатов, построение отчетов и анализ статистики;
- пользователи работают через web-браузеры (Google Chrome, Mozilla Firefox, Opera, Internet Explorer, Safari и другие);
- функции пользователя: регистрация и авторизация, выбор теста, прохождение тестирования, просмотр результатов и ошибок [5].

Основные преимущества:

- простая установка системы;
- доступный интерфейс пользователя;
- продукт совместим со всеми ОС семейства Windows;

- веб-интерфейс пользователя с поддержкой всех распространенных браузеров;

- централизованное хранение данных.

Недостатки:

- отсутствие разделения администратор-преподаватель;

- отсутствие веб-интерфейса для администратора;

- высокие требования к оборудованию.

1.2.4 Сравнительный анализ существующих систем и системы, разрабатываемой в настоящем проекте, по функциональным и технологическим критериям приведен в таблице 1.1.

Таблица 1.1 – Сравнительный анализ разрабатываемой системы с существующими аналогами

Критерий	PikaTest	UniTest	INDIGO	Разраба- тываемая система
Функциональные критерии				
Управление данными участников тестирования	–	+	+	+
Разработка тестового материала	+	+	+	+
Автоматическое формирование тестовых наборов	–	–	–	+
Ответы на вопросы теста	+	+	+	+
Обработка результатов тестирования	–	+	+	+
Формирование отчетов по результатам тестирования	+	+	+	+
Технологические критерии				
Технология «Клиент-Сервер»	–	+	+	+
Использование базы данных	–	+	+	+
Веб-интерфейс	–	–	+– (только для обучающихся)	+
Доступ по ролям: администратор, преподаватель, обучающийся	–	+	+– (не выделен преподаватель)	+

Данные таблицы 1.1 показывают, что разрабатываемая система в отличие от конкурентов позволяет решать специфические задачи. Ее преимуществом является наличие функции автоматического формирования тестовых наборов. Также разрабатываемая система предоставляет доступ по ролям (администратор, преподаватель, обучающийся) и веб-интерфейс для всех категорий пользователей.

Следовательно, система, разрабатываемая в настоящем дипломном проекте, актуальна и имеет преимущества в сравнении с конкурентами.

1.3 Постановка задачи на разработку автоматизированной системы контроля знаний

1.3.1 Разрабатываемая автоматизированная система предназначена для организации контроля знаний с помощью тестирования. Область применения программного средства – учебные заведения общего, среднего специального и высшего образования.

Объектом использования автоматизированной системы являются данные преподавателей и обучающихся, данные проверочных тестов, данные ответов обучающихся на проверочные тесты.

1.3.2 Автоматизированная система тестирования и контроля знаний должна обеспечивать корректное выполнение следующих функций:

- 1) авторизация пользователя;
- 2) изменение пароля пользователя;
- 3) доступ к функциям в зависимости от роли пользователя:
 - а) функции роли «Администратор»:
 - управление данными пользователей;
 - управление данными справочника учебных предметов;
 - управление данными справочника учебных групп;
 - изменение пароля;
 - функции роли «Преподаватель»;
 - б) функции роли «Преподаватель»:
 - просмотр данных обучающихся;
 - управление данными проверочных тестов;
 - открытие и закрытие тестирования;
 - просмотр отчетов по результатам тестирования;
 - изменение пароля;
 - в) функции роли «Обучающийся»:
 - просмотр доступных тестов;
 - ответы на вопросы выбранного теста;

Интерфейс клиентского приложения должен соответствовать следующим требованиям:

- 1) понятный и удобный графический интерфейс;
- 2) наличие защиты от ввода недопустимых значений (верификация вводимых данных);
- 3) наличие подсказок;
- 4) поддержка русского языка;
- 5) доступ к данным в оперативном режиме.

На функции приложения налагаются следующие ограничения:

- 1) обучающийся может принадлежать только к одной учебной группе;
- 2) преподаватель может работать с несколькими учебными группами;

- 3) проверочные тесты принадлежат преподавателю и имеют следующие характеристики:
 - а) владелец;
 - б) учебный предмет;
 - в) минимальное количество баллов, при котором тест считается пройденным;
- 4) тесты разделяются на разделы с указанием количества вопросов, попадающих в тестовый набор;
- 5) вопросы теста относятся к определенному разделу, имеют характеристику сложности по десятибалльной шкале и предусматривают следующие типы ответов на вопросы:
 - а) выбор одного варианта из нескольких возможных;
 - б) выбор нескольких вариантов из нескольких возможных;
 - в) произвольный ответ в текстовой либо числовой форме;
- 6) тестирование инициируется преподавателем и имеет следующие характеристики:
 - а) выбранный тест, принадлежащий преподавателю;
 - б) учебная группа, для которой проводится тестирование;
 - в) дата начала и дата окончания доступности теста;
- 7) при инициировании тестирования осуществляется проверка теста на наличие необходимого количества вопросов каждого раздела;
- 8) тестовые наборы формируются системой автоматически из вопросов теста в соответствии с результатами предыдущего тестирования и рейтингом обучающегося;
- 9) повторное тестирование назначается автоматически, если обучающийся не набрал минимальное количество баллов, при котором тест считается пройденным; для повторного тестирования формируется новый тестовый набор.

1.3.3 К входным данным автоматизированной системы относится информация, вводимая пользователями:

- данные пользователей;
- данные обучающихся;
- данные преподавателей;
- данные учебных предметов;
- данные учебных групп;
- данные учебных групп, доступных преподавателю;
- данные тестов;
- данные разделов теста;
- данные вопросов раздела теста;
- данные ответов на вопросы раздела теста;
- данные тестирований;
- данные тестовых наборов, сформированных для обучающихся;
- данные ответов обучающихся на вопросы тестового набора.

1.3.4 К выходным данным автоматизированной системы относится информация о результатах тестирования:

- сводный отчет по определенному тестированию;
- сводный отчет по вопросам определенного теста;
- отчет по рейтингу обучающихся определенной группы;
- сводный отчет по всем тестированиям определенного обучающегося;
- отчет по прохождению определенного тестирования определенным обучающимся.

1.3.5 Программное средство должно удовлетворять следующим требованиям к надежности:

- надежность хранения, использования и поддержания в актуальном состоянии информационного обеспечения после сбоев в электропитании устройств хранения информации;
- надежность функционирования программного обеспечения и его восстановление после сбоев в оборудовании;
- контроль целостности пользовательских данных на уровне значений атрибутов таблиц и связей между таблицами при изменении состояния пользовательских данных;
- настройка возможностей санкционированного доступа пользователей к ресурсам системы, включающие возможности для авторизации пользователей и управление правами их доступа;
- пароль должен передаваться между подсистемами и храниться в БД в зашифрованном виде;
- время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

1.3.6 К программному средству предъявляются следующие требования к информационной и программной совместимости:

- 1) система должна быть выполнена в клиент-серверной архитектуре:
 - а) СУБД – используется для хранения данных;
 - б) веб-приложение – реализует бизнес-логику и формирование интерфейса пользователя;
 - в) веб-браузер, осуществляющий вывод пользовательского интерфейса и взаимодействие с веб-приложением;
- 2) веб-приложение должно корректно функционировать в актуальных версиях браузеров Internet Explorer, Microsoft Edge, Mozilla Firefox, Google Chrome и его производных.

1.3.7 Обоснование выбора технологий разработки. Среди технологий, применяемых при разработке веб-приложений, наиболее распространенными в настоящее время являются:

1 Java Enterprise Edition (Java EE) – серверная платформа для задач средних и крупных предприятий. Сервер приложений Java EE – это реализация системы в соответствии со спецификацией Java EE, обеспечивающая работу модулей с логикой конкретного приложения [6].

2 .NET Framework – программная платформа от компании Microsoft, предназначенная для создания настольных приложений и веб-приложений. Одной из основных идей .NET является совместимость различных служб, написанных на разных языках. Подобно технологии Java, среда разработки .NET создает байт-код, предназначенный для исполнения виртуальной машиной [7].

3 AMP – акроним, обозначающий набор (комплекс) серверного программного обеспечения, широко используемый в сети Интернет. AMP назван по первым буквам входящих в его состав компонентов: Apache – веб-сервер; MySQL – СУБД; PHP – язык программирования, используемый для создания веб-приложений [8].

Сравнение приведенных платформ приведено в таблице 1.2.

Таблица 1.2 – Сравнение платформ разработки веб-приложений

Критерий	Java Platform	Microsoft.NET Framework	AMP
Кроссплатформенность	Да	Нет	Да
Документация на русском языке	Частично	Да	Да
Веб-сервер	Apache Tomcat, GlassFish	IIS, Apache	Apache
Сервер приложений	Java EE	Фреймворк .NET	Среда исполнения PHP
СУБД	Oracle, Sybase, MySQL	MS SQL, MySQL, Oracle, Sybase	MySQL
Язык программирования	Java	C#, VB, C++	PHP

На основании таблицы 1.2 в качестве платформы реализации был выбран .NET Framework по причине более высокой производительности и поддержки языков программирования.

Для настоящего проекта выбрана технология ASP.NET MVC по причине большей гибкости в реализации клиентской составляющей [9].

Взаимодействие веб-приложения и СУБД предполагается организовать с использованием технологии EntityFramework, которая реализует ORM-модель доступа к данным, позволяет вести как прямое, так и обратное проектирование базы данных и классов модели данных, а также использовать средства языка запросов LINQ to Object [10].

При формировании интерфейса пользователя, выводимого в веб-браузер в виде веб-страниц, предполагается использование следующих фреймворков и библиотек:

- bootstrap 3 для адаптивной верстки веб-интерфейса [11];
- jquery – вспомогательная библиотека javascript [12].

Для разработки выбрана среда Microsoft Visual Studio 2017 Community – официальная среда разработки для платформы DOT.NET от Microsoft [13].

Выбранная среда разработки позволяет разрабатывать проекты с использованием выбранных технологий:

- с использованием менеджера пакетов NuGet имеется возможность встраивания в проект актуальных версий дополнительных библиотек и фреймворков, в том числе: EntityFramework, jquery, bootstrap;
- имеется возможность разработки проектов ASP.NET MVC [14].

Сравнение языков программирования для платформы .NET приведено в таблице 1.3.

Таблица 1.3 – Сравнение языков программирования C# и C++

Сходства	Различия
в C# сохранены основные возможности языка C++, которые не были включены в Java. Это, касается, например, перечислений. Более того в C# для перечислений обеспечивается безопасность типов	В C# использован комплексный метод программирования. В C++ необходимо объявлять классы в заголовочном файле и реализовать классы в отдельном файле. Заголовочные файлы (Interface Definition Language - язык описания интерфейсов) и сложные интерфейсы не являются необходимыми
C# заимствовал из C++ понятия операторов и ключевых слов. Обеспечивается перегрузка операторов	В C++ переменные экземпляров объектов должны быть явно инициализированы в конструкторе. В C# это делается заранее
В C# реализована возможность добавления блоков ненадежного кода	C++ поддерживает множественное наследование классов. Это приводит к определенному усложнению коду. C# не поддерживает множественного наследования. Оно может быть реализовано посредством интерфейсов
	В C# все классы имеют единый базовый класс. Если класс не является производным от другого класса, он автоматически наследует свойства и члены из System.Object

Исходя из преимуществ, описанных в таблице 1.3 для реализации приложения был выбран язык C#.

Наиболее распространенными средами разработки под C# являются: Microsoft Visual Studio – официальная интегрированная среда разработки приложений на платформе .NET [13] и SharpDevelop – свободная интегрированная среда разработки.

Сравнение перечисленных IDE приведено в таблице 1.4.

Таблица 1.4 – Сравнение IDE для разработки

Критерий	Visual Studio	SharpDevelop
Подсветка синтаксиса	да	да
Визуальный редактор WPF и Windows Forms	да	частично
Интегрированный отладчик	да	да
Расширяемость внешними инструментами	да	нет

По преимуществам, перечисленным в таблице 1.4 выберем интегрированную среду разработки Visual Studio.

При реализации приложений на базе платформы .NET наиболее часто применяются следующие СУБД: MS SQL, MySQL, PostgreSQL [15].

Сравнение приведенных СУБД [16, 17, 18] приведено в таблице 1.5.

Таблица 1.5 – Сравнительный анализ СУБД

Критерий	MS SQL	MySQL	PostgreSQL
Кроссплатформенность (Windows, Linux)	+ / –	+ / +	+ / +
Ограничение на использование RAM, GB	64	не ограничено	не ограничено
Ограничение на использование CPU, количество ядер	4	не ограничено	не ограничено
Лицензия	Проприетарная	GPL	GPL
Скорость работы	Высокая	Высокая	Средняя
Поддержка в Visual Studio	да	нет	нет

Таким образом, в сравнении с конкурентами СУБД MS SQL имеет преимущество, так как поддерживается выбранной средой разработки, что определяет ее выбор для реализации проекта.

2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1 Функциональная модель

Построим функциональную модель процесса тестирования и контроля знаний. Такая модель позволит выявить основные функции процесса, связи между функциями, распределение функций между исполнителями, а также наборы входных и выходных данных функций [19].

Контекстная диаграмма процесса тестирования и контроля знаний приведена на рисунке 2.1.

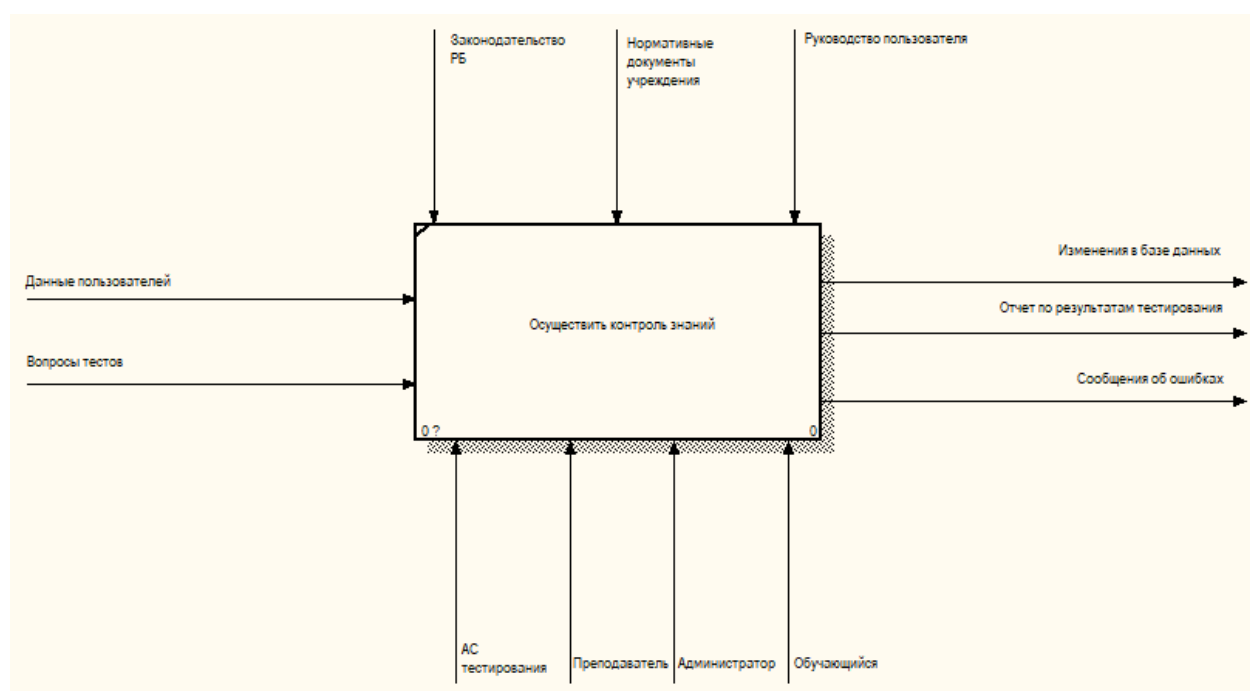


Рисунок 2.1 – Контекстная диаграмма процесса тестирования и контроля знаний

Управление процессом осуществляется на основании руководства пользователя.

Процесс реализуется с использованием АС тестирования следующими категориями пользователей:

- администратор;
- преподаватель;
- обучающийся.

На вход процесса подаются:

- списки пользователей;
- вопросы теста.

На выходе процесс формирует отчеты по результатам тестирования.

Диаграмма декомпозиции процесса тестирования и контроля знаний приведена на рисунке 2.2.

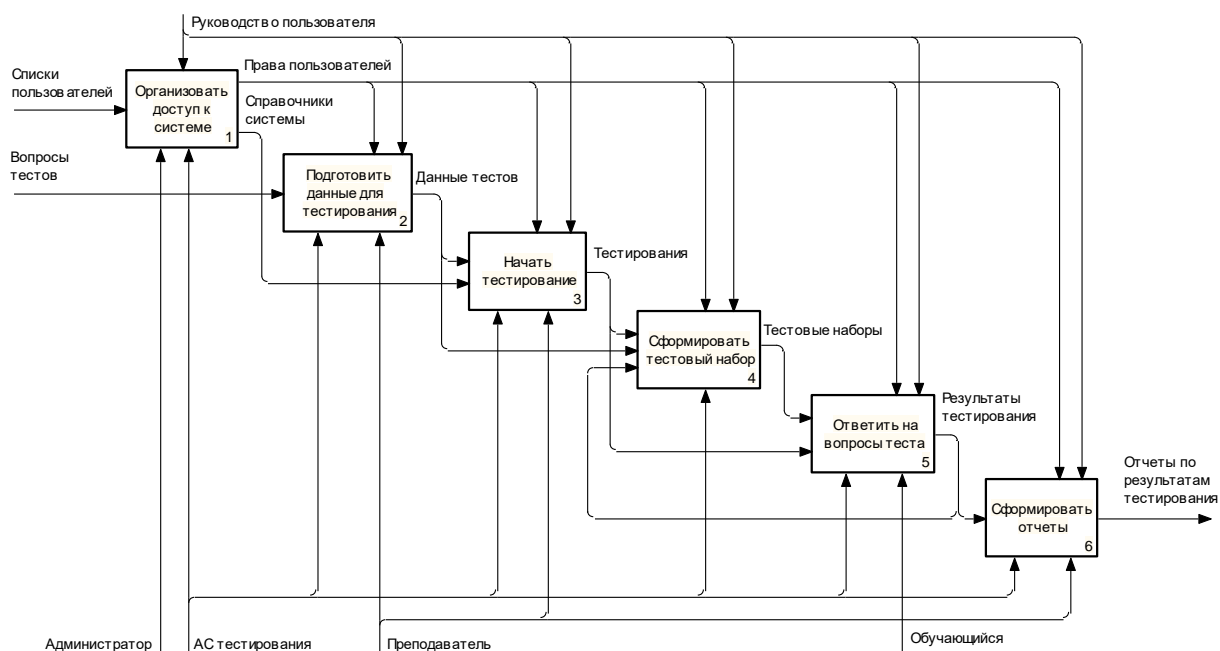


Рисунок 2.2 – Диаграмма декомпозиции процесса тестирования и контроля знаний

В ходе декомпозиции выделено шесть функций:

- организовать доступ к системе;
- подготовить данные для тестирования;
- начать тестирование;
- сформировать тестовый набор;
- ответить на вопросы теста;
- сформировать отчеты.

Функция А1 «Организовать доступ к системе» реализуется администратором с использованием АС тестирования. На вход получает списки пользователей, на выходе формирует справочники системы, подающиеся на вход функции А3, а также права пользователей, подающиеся на управление функций А1-А6.

Функция А2 «Подготовить данные для тестирования» реализуется преподавателем с использованием АС тестирования. На вход функция получает вопросы тестов, на выходе формирует данные тестов, подающиеся на вход функций А3 и А4.

Функция А3 «Начать тестирование» реализуется преподавателем с использованием АС тестирования. На вход функция получает данные тестов (из А2), на выходе формирует тестирования, подающиеся на вход функций А4 и А5.

Функция А4 «Сформировать тестовый набор» реализуется АС тестирования автоматически. На вход функция получает тестирования (из А3) и данные тестов (из А2) и результаты тестирования (из А5), на выходе формирует тестовые наборы, подающиеся на вход функции А5.

Функция А5 «Ответить на вопросы теста» реализуется обучающимся с использованием АС тестирования. На вход функция получает тестовые наборы (из А2) и тестирования (из А4), на выходе формирует результаты тестирования, подающиеся на вход функции А6 и А4 (по обратной связи). При этом обработка ответов и формирование результатов тестирования осуществляется АС тестирования автоматически.

Функция А6 «Сформировать отчеты» реализуется преподавателем с использованием АС тестирования. На вход функция получает результаты тестирования (из А5), на выходе формирует отчеты по результатам тестирования, подающиеся на выход процесса.

Таким образом, в рамках проектируемой системы можно выделить двух основных актеров – преподаватель и обучающийся, а также одного вспомогательного актера – администратора.

Диаграмма вариантов использования системы приведена на рисунке 2.3.

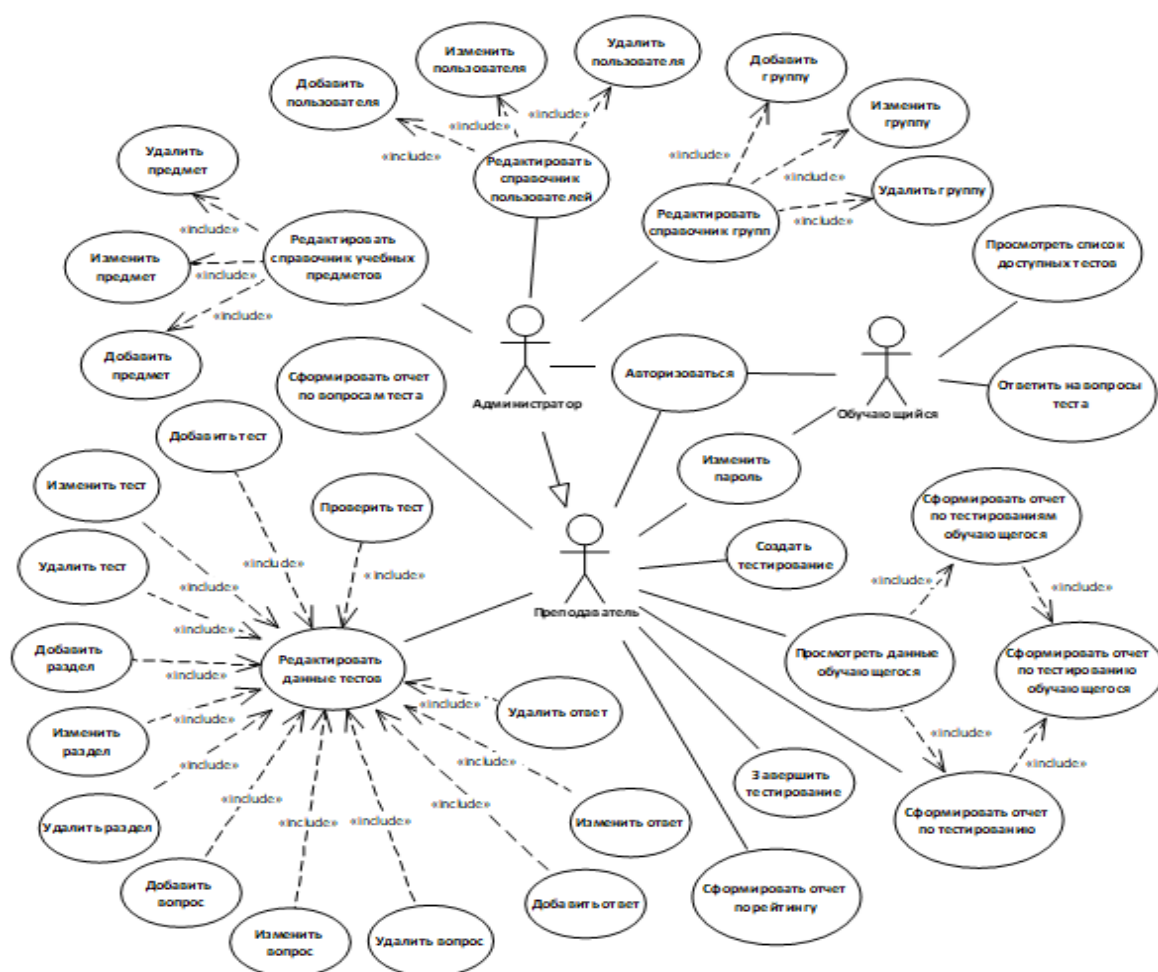


Рисунок 2.3 – Диаграмма вариантов использования системы

Согласно приведенной диаграмме актеру «Преподаватель» доступны следующие варианты использования:

- 1) авторизоваться;
- 2) изменить пароль;
- 3) редактировать данные тестов, включающий следующие варианты использования:
 - а) добавить тест;
 - б) изменить тест;
 - в) удалить тест;
 - г) добавить раздел;
 - д) изменить раздел;
 - е) удалить раздел;
 - ж) добавить вопрос;
 - з) изменить вопрос;
 - и) удалить вопрос;
 - к) добавить ответ;
 - л) изменить ответ;
 - м) удалить ответ;
 - н) проверить тест;
- 4) создать тестирование;
- 5) завершить тестирование;
- 6) просмотреть данные обучающегося, включающий следующие варианты использования:
 - а) сформировать отчет по тестированиям обучающегося;
 - б) сформировать отчет по тестированию;
 - в) сформировать отчет по тестированию обучающегося;
- 7) сформировать отчет по вопросам теста;
- 8) сформировать отчет по рейтингу.

Актер «Администратор» связан с актером «Преподаватель» отношением агрегации, означающим, что администратору доступны все варианты использования преподавателя. Также актеру «Администратор» доступны следующие варианты использования:

- 1) редактировать справочник учебных предметов, включающий следующие варианты использования:
 - а) добавить предмет;
 - б) изменить предмет;
 - в) удалить предмет;
- 2) редактировать справочник пользователей, включающий следующие варианты использования:
 - а) добавить пользователя;
 - б) изменить пользователя;
 - в) удалить пользователя;
- 3) редактировать справочник групп, включающий следующие варианты использования:
 - а) добавить группу;

б) изменить группу;

в) удалить группу;

Актеру «Обучающийся» доступны следующие варианты использования:

1) авторизоваться;

2) изменить пароль;

3) просмотреть список доступных тестов;

4) ответить на вопросы теста.

2.2 Логическая модель данных

В ходе анализа требований к системе и разработки технического задания выявлены следующие категории данных:

- данные о пользователях (пользователи, роли);
- данные справочников (учебные предметы, учебные группы);
- данные тестов (тесты, разделы, типы вопросов, вопросы, ответы);
- данные тестирований (тестирования, тестовые наборы, вопросы тестовых наборов).

Логическая модель данных включает следующие сущности:

- пользователь – содержит информацию о зарегистрированных пользователях;
- роль – содержит перечень ролей пользователей (перечисление – администратор, преподаватель, обучающийся);
- обучающийся – содержит информацию о зарегистрированных обучающихся (группа, рейтинг);
- группа – содержит данные учебных групп;
- предмет – содержит данные учебных предметов;
- тест – содержит данные тестов;
- раздел – содержит данные разделов теста;
- тип вопроса – содержит перечень типов вопросов теста (перечисление – один ответ, несколько ответов, текстовый ответ, ответ-число);
- вопрос – содержит данные вопросов, относящихся к разделам тестов;
- тестирование – содержит данные проводимых тестирований;
- тестовый набор – содержит данные сформированных тестовых наборов;
- вопрос тестового набора – содержит данные вопросов тестовых наборов, включая ответы;
- ответ на вопрос – содержит данные ответов студентов на вопросы тестового набора.

Схема логической модели данных, включающая атрибуты перечисленных сущностей и связи между ними, в нотации IDEF1.X приведена на рисунке 2.4.

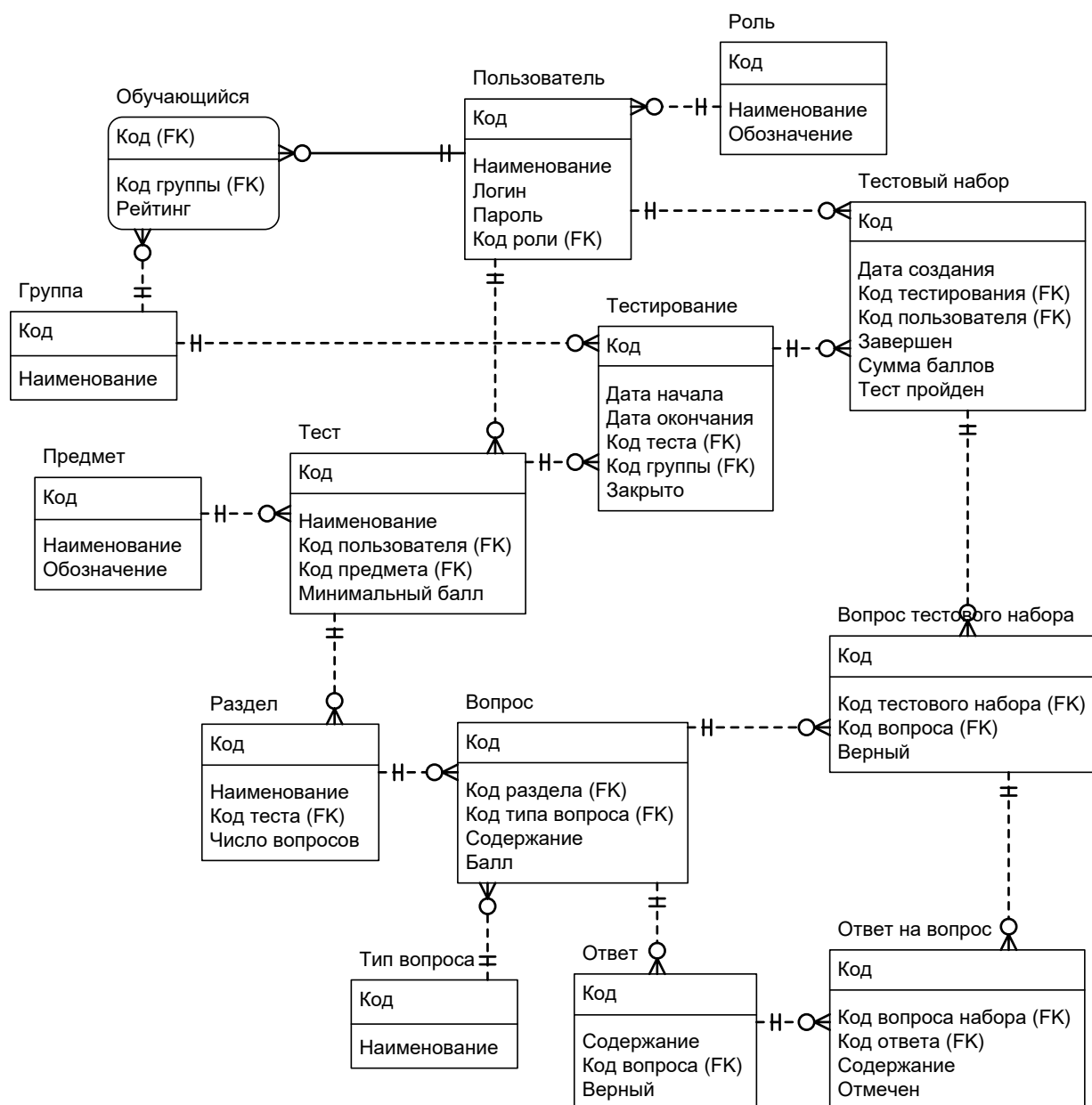


Рисунок 2.4 – Схема логической модели данных

Сущность «Пользователь» описывает пользователей системы и включает следующие атрибуты:

- код – идентификатор пользователя;
- наименование – фамилия, имя, отчество пользователя;
- логин – текстовый идентификатор пользователя для входа в систему;
- пароль – хеш пароля пользователя;
- код роли – ссылка на идентификатор роли пользователя.

Сущность «Пользователь» связана с сущностью «Тестовый набор» связью «один-ко-многим»: одному пользователю может принадлежать много тестовых наборов, тестовый набор может принадлежать только одному пользователю.

Сущность «Пользователь» связана с сущностью «Тест» связью «один-ко-многим»: одному пользователю может принадлежать много тестов, тест может принадлежать только одному пользователю.

Сущность «Пользователь» связана с сущностью «Обучающийся» связью «один-к-одному»: одному пользователю может принадлежать только один обучающийся, обучающийся соответствует только одному пользователю. Данная связь идентифицирует обучающегося.

Сущность «Роль» описывает данные ролей пользователей (администратор, преподаватель, обучающийся) и включает следующие атрибуты:

- код – идентификатор роли;
- наименование – наименование роли;
- обозначение – обозначение роли для использования в программном коде.

Сущность «Роль» связана с сущностью «Пользователь» связью «один-ко-многим»: одной роли может принадлежать много пользователей, пользователь может принадлежать только одной роли.

Сущность «Группа» описывает данные учебных групп и включает следующие атрибуты:

- код – идентификатор группы;
- наименование – наименование группы.

Сущность «Группа» связана с сущностью «Обучающийся» связью «один-ко-многим»: одной группе может принадлежать много обучающихся, обучающийся может принадлежать только одной группе.

Сущность «Группа» связана с сущностью «Тестирование» связью «один-ко-многим»: одной группе может принадлежать много тестирований, тестирование может принадлежать только одной группе.

Сущность «Обучающийся» описывает данные обучающихся, проходящих тестирования, и включает следующие атрибуты:

- код – ссылка на идентификатор пользователя;
- код группы – ссылка на идентификатор группы;
- рейтинг – значение рейтинга обучающегося.

Сущность «Предмет» описывает учебные предметы и включает следующие атрибуты:

- код – идентификатор предмета;
- наименование – наименование предмета.

Сущность «Предмет» связана с сущностью «Тест» связью «один-ко-многим»: одному предмету может принадлежать много тестов, тест может принадлежать только одному предмету.

Сущность «Тест» описывает данные учебных тестов и включает следующие атрибуты:

- код – идентификатор теста;
- наименование – наименование теста;
- код пользователя – ссылка на идентификатор пользователя, которому принадлежит тест;
- код предмета – ссылка на идентификатор предмета;
- минимальный балл – минимальная сумма баллов, при которой тест считается пройденным.

Сущность «Тест» связана с сущностью «Раздел» связью «один-ко-многим»: одному тесту может принадлежать много разделов, раздел может принадлежать только одному тесту.

Сущность «Тест» связана с сущностью «Тестирование» связью «один-ко-многим»: одному тесту может принадлежать много тестирований, тестирование может принадлежать только одному тесту.

Сущность «Раздел» описывает данные разделов теста и включает следующие атрибуты:

- код – идентификатор раздела;
- наименование – наименование раздела;
- код теста – ссылка на идентификатор теста;
- число вопросов – число вопросов раздела, попадающих в тестовый набор при его автоматическом формировании.

Сущность «Раздел» связана с сущностью «Вопрос» связью «один-ко-многим»: одному разделу может принадлежать много вопросов, вопрос может принадлежать только одному разделу.

Сущность «Тип вопроса» описывает типы вопросов теста и включает следующие атрибуты:

- код – идентификатор типа вопроса;
- наименование – наименование типа вопроса.

Сущность «Тип вопроса» связана с сущностью «Вопрос» связью «один-ко-многим»: одному типу может принадлежать много вопросов, вопрос может принадлежать только одному типу.

Сущность «Вопрос» описывает данные вопросов теста и включает следующие атрибуты:

- код – идентификатор вопроса;
- код раздела – ссылка на идентификатор раздела;
- код типа вопроса – ссылка на идентификатор типа вопроса;
- содержание – содержание вопроса;
- балл – число баллов, засчитываемых тестируемому при правильном ответе.

Сущность «Вопрос» связана с сущностью «Ответ» связью «один-ко-многим»: одному вопросу может принадлежать много ответов, ответ может принадлежать только одному вопросу.

Сущность «Вопрос» связана с сущностью «Вопрос тестового набора» связью «один-ко-многим»: одному вопросу может принадлежать много вопросов набора, вопрос набора может принадлежать только одному вопросу.

Сущность «Ответ» описывает данные вариантов ответа на вопросы теста и включает следующие атрибуты:

- код – идентификатор ответа;
- содержание – содержание ответа;
- код вопроса – ссылка на идентификатор вопроса;
- верный – признак верного ответа.

Сущность «Ответ» связана с сущностью «Ответ на вопрос» связью «один-ко-многим»: одному ответу может принадлежать много ответов на вопрос, ответ на вопрос может принадлежать только одному ответу.

Сущность «Тестирование» описывает данные тестирований, которые проходят обучающиеся, и включает следующие атрибуты:

- код – идентификатор тестирования;
- дата начала – дата, начиная с которой тестирование доступно для прохождения;
- дата окончания – дата, до которой доступно тестирование;
- код теста – ссылка на идентификатор теста, по которому проводится тестирование;
- код группы – ссылка на идентификатор группы, для которой доступно тестирование;
- закрыто – признак того, что тестирование завершено (не доступно).

Сущность «Тестирование» связана с сущностью «Тестовый набор» связью «один-ко-многим»: одному тестированию может принадлежать много тестовых наборов, тестовый набор может принадлежать только одному тестированию.

Сущность «Тестовый набор» описывает данные тестовых наборов, сформированных автоматически, и включает следующие атрибуты:

- код – идентификатор тестового набора;
- дата создания – дата создания тестового набора;
- код тестирования – ссылка на идентификатор тестирования, для которого создан тестовый набор;
- код пользователя – ссылка на идентификатор пользователя, для которого создан тестовый набор;
- завершен – признак того, что тестирование по данному тестовому набору завершено;
- сумма баллов – сумма баллов, набранных по тестовому набору;
- тест пройден – признак того, что тест, по которому собирается тестовый набор, пройден.

Сущность «Тестовый набор» связана с сущностью «Вопрос тестового набора» связью «один-ко-многим»: одному тестовому набору может принадлежать много вопросов, вопрос может принадлежать только одному тестовому набору.

Сущность «Вопрос тестового набора» описывает данные вопросов тестовых наборов и включает следующие атрибуты:

- код – идентификатор вопроса тестового набора;
- код тестового набора – ссылка на идентификатор тестового набора, к которому относится вопрос;
- код вопроса – ссылка на идентификатор вопроса теста;
- верный – признак того, что ответ верный.

Сущность «Вопрос тестового набора» связана с сущностью «Ответ на вопрос» связью «один-ко-многим»: одному тестовому набору может принадлежать много ответов на вопрос, ответ на вопрос может принадлежать только одному вопросу тестового набора.

Сущность «Ответ на вопрос» описывает данные вопросов тестовых наборов и включает следующие атрибуты:

- код – идентификатор ответа на вопрос;
- код вопроса набора – ссылка на идентификатор вопроса тестового набора, к которому относится ответ;
- код ответа – ссылка на идентификатор ответа;
- содержание ответа – данные, введенные обучающимся при ответе на вопрос;
- отмечен – признак того, что ответ отмечен.

2.3 Функциональная спецификация

Требования к функции «Авторизация»:

- авторизация должна осуществляться перед доступом к основным функциям системы;
- при авторизации пользователь должен ввести логин и пароль;
- при неудачной авторизации доступ к функциям программного средства не предоставляется, а пользователю выводится сообщение об ошибке авторизации.

Требования к функции «Изменить пароль»:

- функция доступна только авторизованным пользователям;
- для изменения пароля пользователь должен указать текущий пароль, новый пароль и подтверждение нового пароля.

Требования к функции «Редактировать данные тестов»:

- редактирование данных тестов осуществляется преподавателем или администратором;
- функция редактирования данных тестов включает вывод списка, добавление, изменение, удаление тестов, разделов, вопросов, ответов;
- редактирование данных тестов осуществляется с использованием форм списков, добавления, изменения, подтверждения удаления тестов, разделов, вопросов, ответов;

- необходимо реализовать иерархический доступ к редактированию связанных данных тестов: от данных теста – переход к редактированию разделов, от данных раздела – переход к редактированию вопросов, от данных вопроса – переход к редактированию ответов;

- пользователь, которому принадлежит тест, определяется автоматически по данным авторизованного пользователя при сохранении нового теста и изменению не подлежит;

- для изменения и удаления доступны только тесты, которые принадлежат авторизованному пользователю;

- для добавления, изменения, удаления доступны разделы, вопросы, ответы, относящиеся к тестам, принадлежащим авторизованному пользователю;

- не допускается изменять и удалять тесты, по которым существуют тестирования;

- не допускается добавлять, изменять, удалять разделы, вопросы, ответы тестов, по которым существуют тестирования;

- при удалении тестов, разделов, вопросов необходимо обеспечить каскадное удаление подчиненных данных;

- необходимо реализовать проверку теста «по требованию» на корректность данных: число вопросов раздела, достижимость минимального балла теста.

Требования к функции «Создать тестирование»:

- тестирование создается преподавателем или администратором на основании теста, принадлежащего пользователю, для определенной группы обучающихся;

- при создании тестирования указывается дата, начиная с которой доступно тестирование, а также дата завершения тестирования;

- перед сохранением тестирования необходимо реализовать автоматическую проверку теста на корректность данных аналогично функции редактирования тестов.

Требования к функции «Завершить тестирование»:

- тестирование завершается преподавателем или администратором, которому принадлежит тест, «по требованию»;

- если тестирование завершается ранее даты окончания, указанной в тестировании, необходимо вывести соответствующее предупреждение при подтверждении завершения;

- при подтверждении завершения тестирования необходимо вывести сводную информацию по тестированию: общее число обучающихся в группе, общее число обучающихся приступивших к тестированию (сформированы тестовые набор), общее число обучающихся завершивших тестирование с положительным результатом;

- перед сохранением состояния тестирования необходимо автоматически рассчитать результаты тестирования;

- после сохранения состояния тестирования необходимо автоматически сформировать отчет по тестированию;

- для завершеного тестирования необходимо реализовать возможность автоматического формирования отчета по тестированию с переходом из списка тестирований;

- отчет по тестированию включает список учащихся группы и результаты тестирования, а также итоговые показатели.

Требования к функции «Просмотреть данные обучающегося»:

- данные обучающегося выводятся в списке по выбранной группе и на странице просмотра данных обучающихся;

- страница просмотра данных обучающегося включает отчет по тестированиям обучающегося, который формируется автоматически;

- отчет по тестированиям обучающегося включает список всех тестирований, в которых принимал участие обучающийся, с указанием результата тестирования;

- необходимо реализовать автоматическое формирование отчета по тестированию обучающегося с переходом из отчета по тестированиям обучающегося;

- необходимо реализовать автоматическое формирование отчета по тестированию обучающегося с переходом из отчета по тестированию;

- отчет по тестированию обучающегося включает результаты ответа на вопросы тестового набора, итоговые показатели.

Требования к функции «Сформировать отчет по вопросам теста»:

- отчет по вопросам теста формируется «по требованию» с переходом из списка тестов;

- отчет по вопросам теста включает список вопросов теста, число тестовых наборов, в которые вошел вопрос, число правильных и неправильных ответов.

Требования к функции «Сформировать отчет по рейтингу»:

- отчет по рейтингу формируется автоматически при выводе списка обучающихся группы;

- отчет по рейтингу включает список обучающихся группы и текущий рейтинг каждого обучающегося.

Требования к функции «Редактировать справочник учебных предметов»:

- редактирование справочника учебных предметов осуществляется администратором;

- редактирование данных учебных предметов осуществляется с использованием списка предметов, формы предмета, формы изменения данных предмета, формы подтверждения удаления предмета;

- функция редактирования данных учебных предметов включает вывод списка предметов, а также добавление, редактирование, удаление данных предмета;

- при удалении учебного предмета необходимо обеспечить целостность данных: нельзя удалить предмет, для записи которого существуют ссылки.

Требования к функции «Редактировать справочник пользователей»:

- редактирование справочника пользователей осуществляется администратором;

- редактирование данных пользователей осуществляется с использованием списка пользователей с отбором по наименованию и роли, формы добавления пользователя, формы изменения пользователя, формы подтверждения удаления пользователя, формы добавления данных обучающегося, формы изменения данных обучающегося;

- функция редактирования данных пользователей включает вывод списка пользователей, а также добавление, редактирование, удаление данных пользователя;

- при удалении пользователя необходимо обеспечить целостность данных: нельзя удалить пользователя, для записи которого существуют ссылки.

Требования к функции «Редактировать справочник групп»:

- редактирование справочника групп осуществляется администратором;

- редактирование данных групп осуществляется с использованием списка групп, формы группы, формы изменения данных группы, формы подтверждения удаления группы;

- функция редактирования данных групп включает вывод списка групп, а также добавление, редактирование, удаление данных группы;

- при удалении группы необходимо обеспечить целостность данных: нельзя удалить группу, для записи которой существуют ссылки.

Требования к функции «Просмотреть список доступных тестов»:

- просмотр списка доступных тестов осуществляется обучающимся;

- список доступных тестов представлен в виде наименования, даты начала, даты окончания, а также статусом прохождения данного теста авторизованным пользователем;

- статус прохождения теста авторизованным пользователем формируется автоматически.

Требования к функции «Ответить на вопросы теста»:

- функция ответа на вопросы теста доступна обучающимся;

- функция доступна только для тех тестирований, которые не завершены, соответствуют текущей дате, относятся к группе, к которой принадлежит текущий пользователь;

- если пользователь завершил тест со статусом «пройден», то функция недоступна;

- при первом переходе к функции формируется новый тестовый набор;

- для первой попытки тестовый набор формируется исходя из рейтинга обучающегося;

- для каждой из последующих попыток формируется новый тестовый набор исходя из среднего балла предыдущего тестового набора;
- новый тестовый набор может быть сформирован, если не имеется тестовых наборов со статусом «открыт»;
- при ответе на вопросы теста необходимо реализовать верификацию вводимых данных;
- завершение ответа на вопросы тестового набора осуществляется «по требованию», при этом автоматически рассчитываются результаты.

Требования к формированию тестовых наборов:

- тестовый набор формируется из вопросов разделов теста по количеству вопросов, указанному в разделе;
- тестовый набор формируется исходя из базового значения, который равен для первого тестового набора тестирования рейтингу обучающегося, для последующих тестовых наборов тестирования – среднему баллу за предыдущий тестовый набор;
- при формировании тестового набора в первую очередь добавляются вопросы, стоимость которых меньше базового значения, берущиеся из списка вопросов раздела теста отсортированному в порядке убывания стоимости;
- в случае, если меньших по стоимости вопросов недостаточно, то тестовый набор дополняется вопросами, стоимость которых больше базового значения, берущимися из списка вопросов раздела теста, отсортированному в порядке возрастания стоимости;
- стоимость вопроса определяется баллом.

Требования к расчету рейтинга обучающегося:

- начальное значение рейтинга обучающегося равно пяти;
- рейтинг обучающегося рассчитывается исходя из результатов прохождения тестирования;
- рейтинг рассчитывается при сохранении результатов тестирования по каждому тестовому набору;
- рейтинг определяется как среднее арифметическое стоимости вопросов завершенных тестовых наборов.

Общие требования:

- программное средство должно быть реализовано в виде веб-приложения;
- доступ к функциям должен осуществляться через главное меню, включающее пункты, соответствующие функциям, которые доступны авторизованному пользователю;
- меню должно быть доступно на всех страницах клиентского приложения.

В результате выполнения моделирования предметной области были выявлены основные функции процесса контроля знаний. Построена контекстная диаграмма осуществления процесса тестирования и контроля знаний, диаграмма вариантов использования системы, а также схема логической модели данных.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

3.1 Структура системы

Разрабатываемая система представлена в трехуровневой архитектуре:

- веб-браузер, обеспечивающий интерфейс пользователя;
- веб-приложение, обеспечивающее обработку запросов от веб-браузера, формирование пользовательского интерфейса и бизнес-логику;
- сервер баз данных, обрабатывающий запросы от веб-приложения к базе данных, реализующей хранение данных.

Укрупненная структурная схема системы приведена на рисунке 3.1.

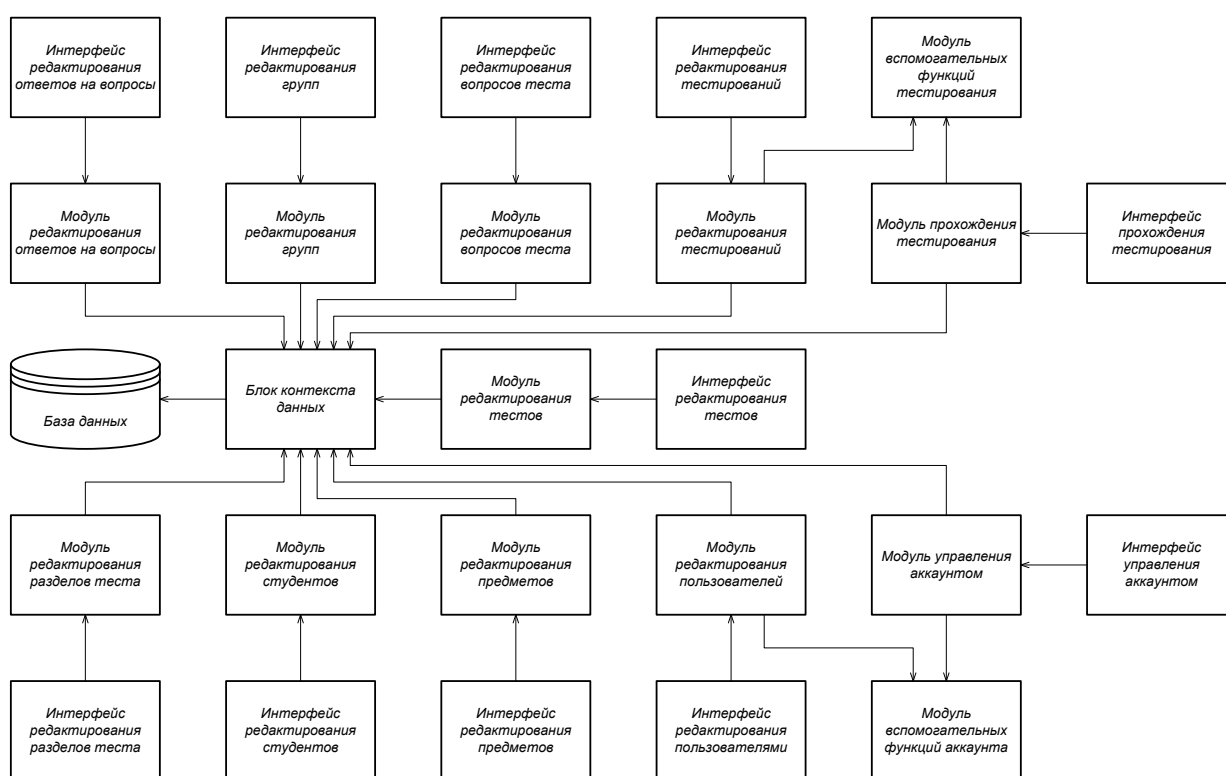


Рисунок 3.1 – Структурная схема системы

Согласно приведенной схеме система включает следующие структурные компоненты:

- база данных, обеспечивающая хранение данных;
- блок контекста данных – совокупность модулей, обеспечивающих модель данных;
- модуль и интерфейс редактирования ответов на вопросы;
- модуль и интерфейс редактирования групп;
- модуль и интерфейс редактирования вопросов теста;
- модуль и интерфейс редактирования тестирований;
- модуль и интерфейс прохождения тестирования;

- модуль и интерфейс редактирования разделов теста;
- модуль и интерфейс редактирования студентов;
- модуль и интерфейс редактирования предметов;
- модуль и интерфейс редактирования пользователей;
- модуль и интерфейс управления аккаунтом;
- модуль вспомогательных функций аккаунта;
- модуль вспомогательных функций тестирования.

3.2 Схема модели базы данных

Схема физической модели базы данных приведена на рисунке 3.2.

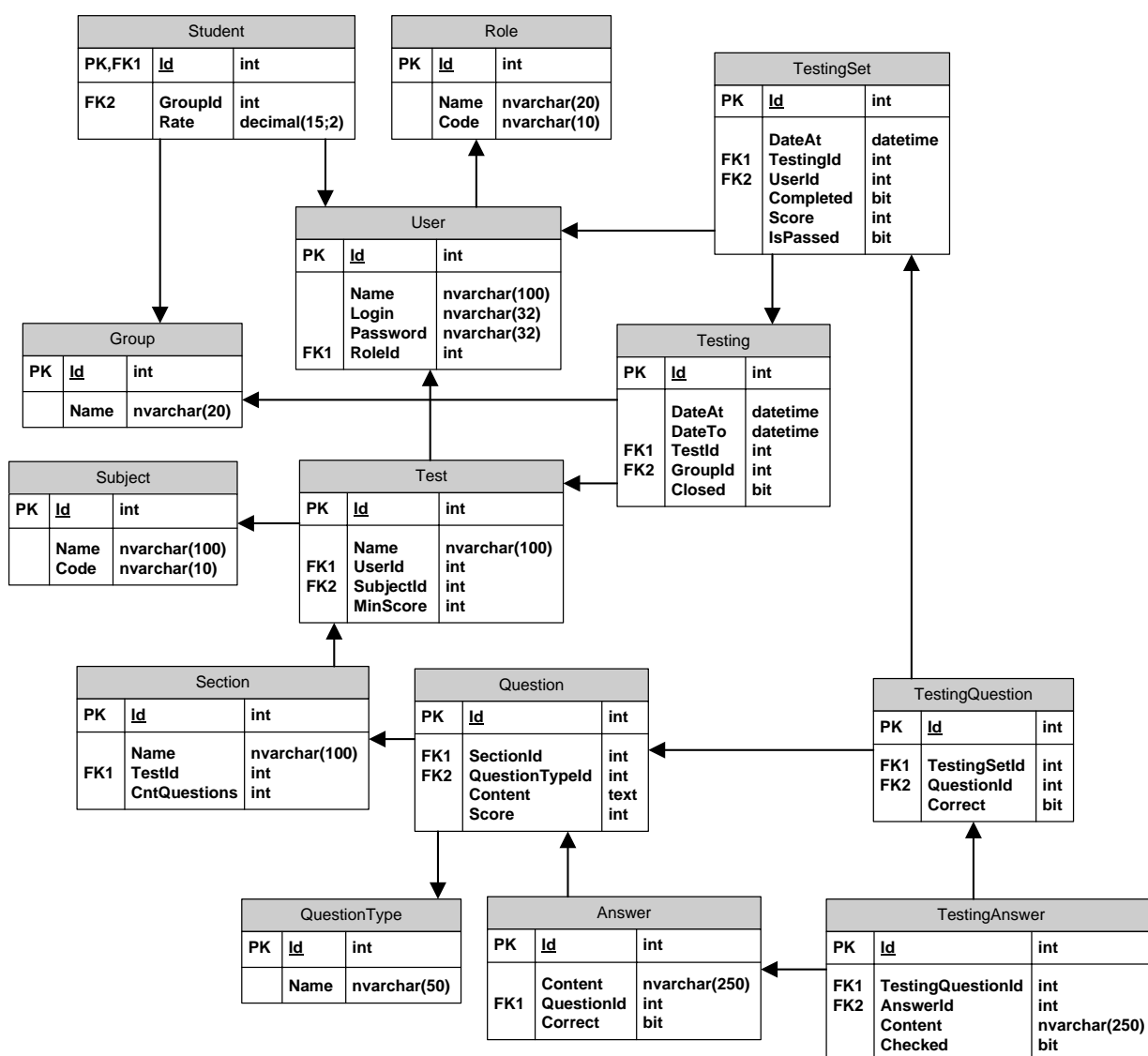


Рисунок 3.2 – Схема физической модели базы данных

Физическая модель построена на основе логической с учетом ограничений, накладываемых возможностями СУБД Microsoft SQL Server.

Описание таблиц базы данных приведено в таблицах 3.1 – 3.14.

Таблица 3.1 – Таблица Student (Студент)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK, FK1
GroupId	int	10	FK2
Rate	decimal	15; 2	

Таблица 3.2 – Таблица Role (Роль)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	20	
Code	nvarchar	10	

Таблица 3.3 – Таблица User (Пользователь)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	100	
Login	nvarchar	32	
Password	nvarchar	32	
RoleId	int	10	FK1

Таблица 3.4 – Таблица Group (Группа)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	20	

Таблица 3.5 – Таблица Subject (Предмет)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	100	
Code	nvarchar	10	

Таблица 3.6 – Таблица Test (Тест)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	100	
UserId	int	10	FK1
SubjectId	int	10	FK2
MinScore	int	10	

Таблица 3.7 – Таблица QuestionType (Тип вопроса)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	50	

Таблица 3.8 – Таблица Section (Раздел)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Name	nvarchar	100	
TestId	int	10	FK1
CntQuestion	int	10	

Таблица 3.9 – Таблица Question (Вопрос)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
SectionId	int	10	FK1
QuestionTypeId	int	10	FK2
Content	text	10	
Score	int	10	

Таблица 3.10 – Таблица Answer (Ответ)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
Content	nvarchar	250	
QuestionId	int	10	FK1
Correct	bit	1	

Таблица 3.11 – Таблица Testing (Тестирование)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
DateAt	datetime		
DateTo	datetime		
TestId	int	10	FK1
GroupId	int	10	FK2
Closed	bit	1	

Таблица 3.12 – Таблица TestingQuestion (Вопрос тестового набора)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
TestingSetId	int	10	FK1
QuestionId	int	10	FK2
Correct	bit	1	

Таблица 3.13 – Таблица TestingAnswer (Ответ на вопрос тестового набора)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
TestingQuestionId	int	10	FK1
AnswerId	int	10	FK2
Content	nvarchar	250	
Correct	bit	1	

Таблица 3.14 – Таблица TestingSet (Тестовый набор)

Имя поля	Тип поля	Размер поля	Признак ключа
Id	int	10	PK
DateAt	datetime		
TestingId	int	10	FK1
UserId	int	10	FK2
Completed	bit	1	
Score	int	10	
IsPassed	bit	1	

Связи таблиц базы данных приведены в таблице 3.15.

Таблица 3.15 – Связи таблиц базы данных

Таблица первичного ключа	Таблица внешнего ключа	Первичный ключ	Внешний ключ	Связь
Group	Student	Id	GroupId	1:M
User	Student	Id	UserId	1:M
Role	User	Id	RoleId	1:M
User	TestingSet	Id	UserId	1:M
Testing	TestingSet	Id	TestingId	1:M
TestingSet	TestingAnswer	Id	TestingSetId	1:M
Group	Testing	Id	GroupId	1:M
User	Test	Id	UserId	1:M
Subject	Test	Id	SubjectId	1:M
Test	Testing	Id	TestId	1:M
Test	Section	Id	TestId	1:M
Section	Question	Id	SectionId	1:M
QuestionType	Question	Id	QuestionTypeId	1:M
Question	Answer	Id	QuestionId	1:M
Question	TestingQuestion	Id	QuestionId	1:M
Answer	TestingAnswer	Id	AnswerId	1:M
TestingQuestion	TestingAnswer	Id	TestingQuestion Id	1:M

Связь между таблицами Group и Student определяется как «один ко многим», так как в одной группе может быть много студентов.

Связь между таблицами User и Student определяется как «один к одному», так как одному пользователю может соответствовать только один студент.

Связь между таблицами Role и User определяется как «один ко многим», так как одной роли может принадлежать много пользователей.

Связь между таблицами User и TestingSet определяется как «один ко многим», так как для одного пользователя может быть создано много тестовых наборов.

Связь между таблицами Testing и TestingSet определяется как «один ко многим», так как для одного тестирования может быть создано много тестовых наборов.

Связь между таблицами TestingSet и TestingQuestion определяется как «один ко многим», так как к одному тестовому набору может относиться много вопросов тестового набора.

Связь между таблицами Group и Testing определяется как «один ко многим», так как для одной группы может быть создано много тестирований.

Связь между таблицами User и Test определяется как «один ко многим», так как один пользователь может быть автором многих тестов.

Связь между таблицами Subject и Test определяется как «один ко многим», так как для одного предмета может быть создано много тестов.

Связь между таблицами Test и Testing определяется как «один ко многим», так как для одного теста может быть создано много тестирований.

Связь между таблицами Test и Section определяется как «один ко многим», так как одному тесту может соответствовать много разделов.

Связь между таблицами Section и Question определяется как «один ко многим», так как одному разделу может соответствовать много вопросов.

Связь между таблицами QuestionType и Question определяется как «один ко многим», так как одному типу может соответствовать много вопросов.

Связь между таблицами Question и Answer определяется как «один ко многим», так как для одного вопроса может быть много ответов.

Связь между таблицами Question и TestingQuestion определяется как «один ко многим», так как одному вопросу может соответствовать много вопросов тестового набора.

Связь между таблицами Answer и TestingAnswer определяется как «один ко многим», так как одному ответу может соответствовать много ответов на вопрос тестового набора.

Связь между таблицами TestingQuestion и TestingAnswer определяется как «один ко многим», так как одному вопросу тестового набора может соответствовать много ответов на вопрос тестового набора.

3.3 Логическая структура системы

В рамках проектируемой системы можно выделить классы предметной области, описывающие модель данных, а также классы модели реализации, используемые непосредственно в программном коде [12, 13].

Базовая диаграмма классов приведена на рисунке 3.3.

На диаграмме представлены следующие классы и пакеты классов:

- Models – классы, реализующие модель данных ASP.NET MVC, а также классы контекста данных EntityFramework;
- Controllers – классы контроллеров ASP.NET MVC;
- Helpers – классы вспомогательных функций;
- MvcApplication – класс приложения ASP.NET MVC;
- RouteConfig, BundleConfig, FilterConfig – классы настройки приложения ASP.NET MVC.

Основные классы модели данных и класс контекста данных (пакет Models) приведены на рисунке 3.4.

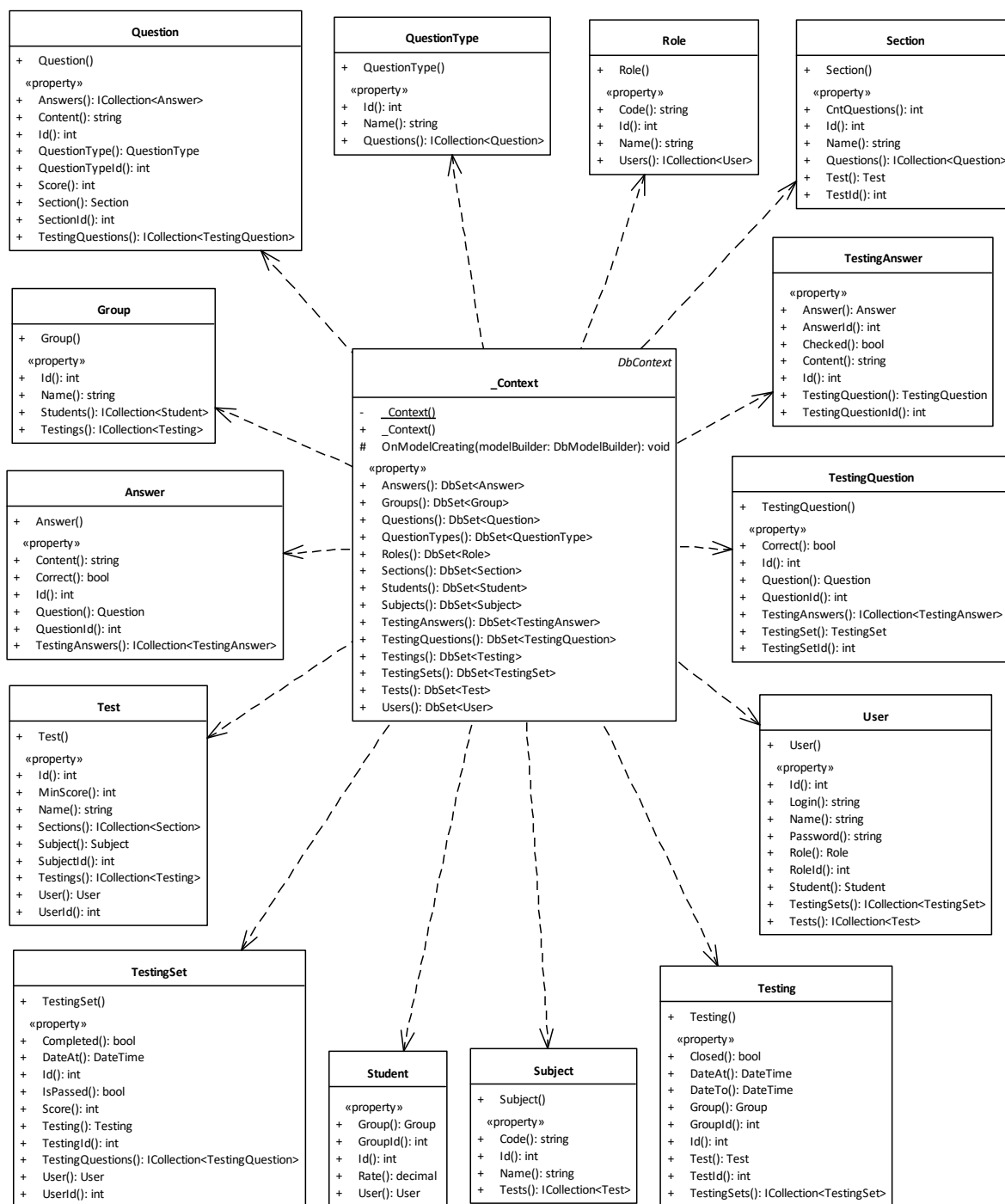


Рисунок 3.4 – Диаграмма основных классов пакета Models

Пакет Models включает следующие основные классы, реализующие модель данных и соответствующие таблицам базы данных:

- Answer – класс модели данных ответа;
- Group – класс модели данных группы;
- Question – класс модели данных вопроса;

- QuestionType – класс модели данных типа вопроса;
- Role – класс модели данных роли пользователя;
- Section – класс модели данных раздела теста;
- TestingAnswer – класс модели данных ответа на вопрос тестового набора;
- TestingQuestion – класс модели данных вопроса тестового набора;
- User – класс модели данных пользователя;
- Testing – класс модели данных тестирования;
- Subject – класс модели данных предмета;
- Student – класс модели данных обучающегося;
- TestingSet – класс модели данных тестового набора;
- Test – класс модели данных теста.

Также пакет Models включает следующие вспомогательные классы, не предназначенные для хранения в БД:

- LoginData – вспомогательный класс данных авторизации;
- PasswordData – вспомогательный класс данных изменения пароля;
- UsersFilter – вспомогательный класс данных фильтра пользователей;
- UsersList – вспомогательный класс данных списка пользователей;
- TestsFilter – вспомогательный класс данных фильтра тестов;
- TestsList – вспомогательный класс данных списка тестов;
- TestingsFilter – вспомогательный класс данных фильтра тестирований;
- TestingsList – вспомогательный класс данных списка тестирований;
- Result – вспомогательный класс данных результатов проверки теста;
- ReportStudentRow – вспомогательный класс данных строки отчета по тестированиям студента;
- ReportTestQuestionRow – вспомогательный класс данных строки отчета по вопросам теста;
- AnswerData – вспомогательный класс данных ответа на вопрос тестового набора.

Диаграмма вспомогательных классов (пакет Helpers) приведена на рисунке 3.5. На диаграмме представлены следующие вспомогательные классы: Authorization, TestWorks.

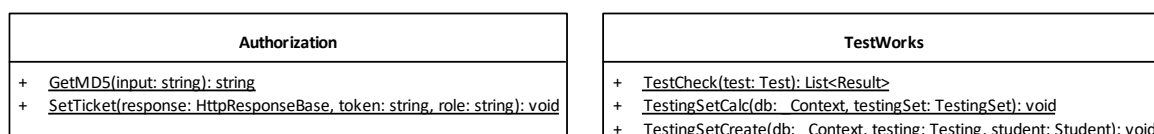


Рисунок 3.5 – Диаграмма классов пакета Helpers

Класс Authorization пакета Helpers включает вспомогательные функции, связанные с аккаунтом пользователя, и содержит следующие методы:

- GetMD5() – возвращает MD5-хеш строки;

– SetTicket() – сохраняет билет авторизованного пользователя.

Класс TestWorks пакета Helpers включает вспомогательные функции, связанные с прохождением тестирования, и содержит следующие методы:

– TestCheck() – возвращает результат проверки теста на корректность данных – список ошибок;

– TestingSetCalc() – реализует расчет заданного тестового набора по окончании тестирования и сохранение данных в базе данных;

– TestingSetCreate() – реализует создание тестового набора для заданных студента и тестирования и сохранение его в базе данных.

Диаграмма классов контроллеров (пакет Controllers) приведена на рисунке 3.6.

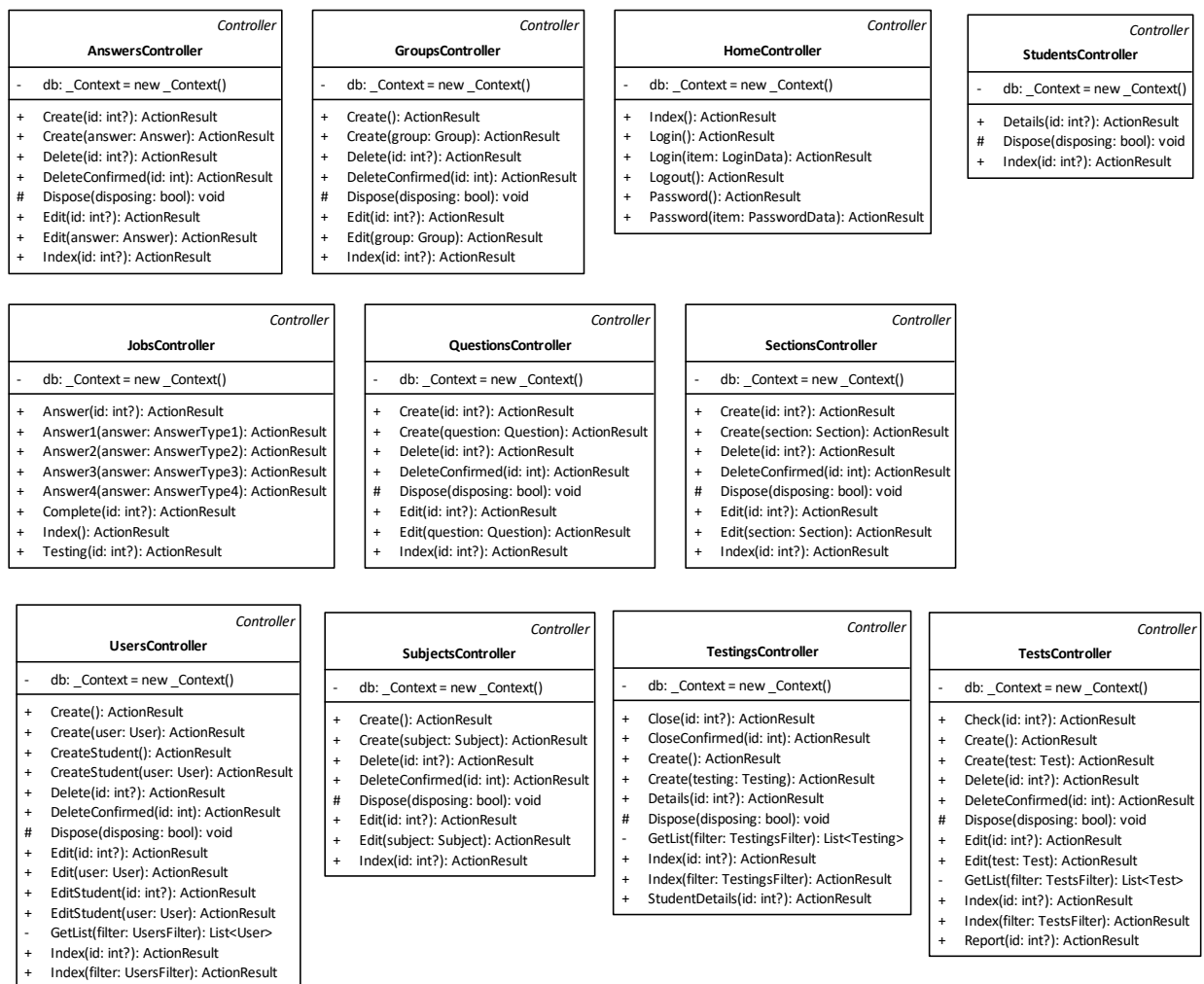


Рисунок 3.6 – Диаграмма классов пакета Controllers

Каждый класс соответствует модулю редактирования, выделенному в ходе структурного проектирования.

На диаграмме приведены следующие классы контроллеров:

– AnswersController – контроллер редактирования ответов на вопросы;

– GroupsController – контроллер редактирования групп;

- StudentsController – контроллер редактирования студентов;
- JobsController – контроллер прохождения тестирования;
- QuestionsController – контроллер редактирования вопросов теста;
- SectionsController – контроллер редактирования разделов теста;
- SubjectsController – контроллер редактирования предметов;
- TestingsController – контроллер редактирования тестирований;
- TestsController – контроллер редактирования тестов;
- UsersController – контроллер редактирования пользователей;
- HomeController – контроллер управления аккаунтом.

Контроллер управления аккаунтами пользователей HomeController включает следующие методы:

- Index() – вывод главной страницы;
- Login() – вывод и обработка формы авторизации;
- Logout() – обработка запроса на выход авторизованного пользователя;
- Password() – вывод и обработка формы изменения пароля.

Все контроллеры редактирования данных имеют типичные методы:

- Index() – вывод списка элементов;
- Create() – вывод и обработка формы нового элемента;
- Edit() – вывод и обработка формы редактирования элемента;
- Delete() и DeleteConfirmed() – вывод и обработка формы удаления элемента справочника.

Если список предусматривает отбор, то контроллер также включает метод GetList(), возвращающий список элементов согласно условиям отбора.

Контроллер редактирования тестов TestsController дополнительно включает следующие методы:

- Check() – проверка корректности теста и вывод результатов проверки;
- Report() – вывод отчета по вопросам теста.

Контроллер редактирования тестирований TestingsController дополнительно включает следующие методы:

- Details() – вывод отчета по тестированию;
- StudentDetails() – вывод отчета по тестированию студента.

Контроллер редактирования пользователей UsersController дополнительно включает следующие методы, позволяющие редактировать данные обучающихся:

- CreateStudent() – вывод и обработка формы нового обучающегося;
- EditStudent() – вывод и обработка формы редактирования данные обучающегося.

Контроллер StudentsController ограничивается следующими методами:

- Index() – вывод списка обучающихся группы;
- Details() – вывод подробных данных обучающегося.

Контроллер прохождения тестирования JobsController включает следующие методы:

- Index() – вывод списка доступных тестирований;

- Testing() – выводит вопросы теста;
- Answer() – вывод формы ответа на вопрос;
- Answer1(), Answer2(), Answer3(), Answer4() – обработка формы ответа на вопрос соответствующего типа;
- Complete() – обработка завершения теста.

Все перечисленные методы выводят представление, соответствующее наименованию метода.

Фрагменты программного кода, реализующего перечисленные классы, приведены в Приложении А.

3.4 Взаимодействие классов

Диаграмма последовательности для варианта использования «Авторизоваться» приведена на рисунке 3.7.

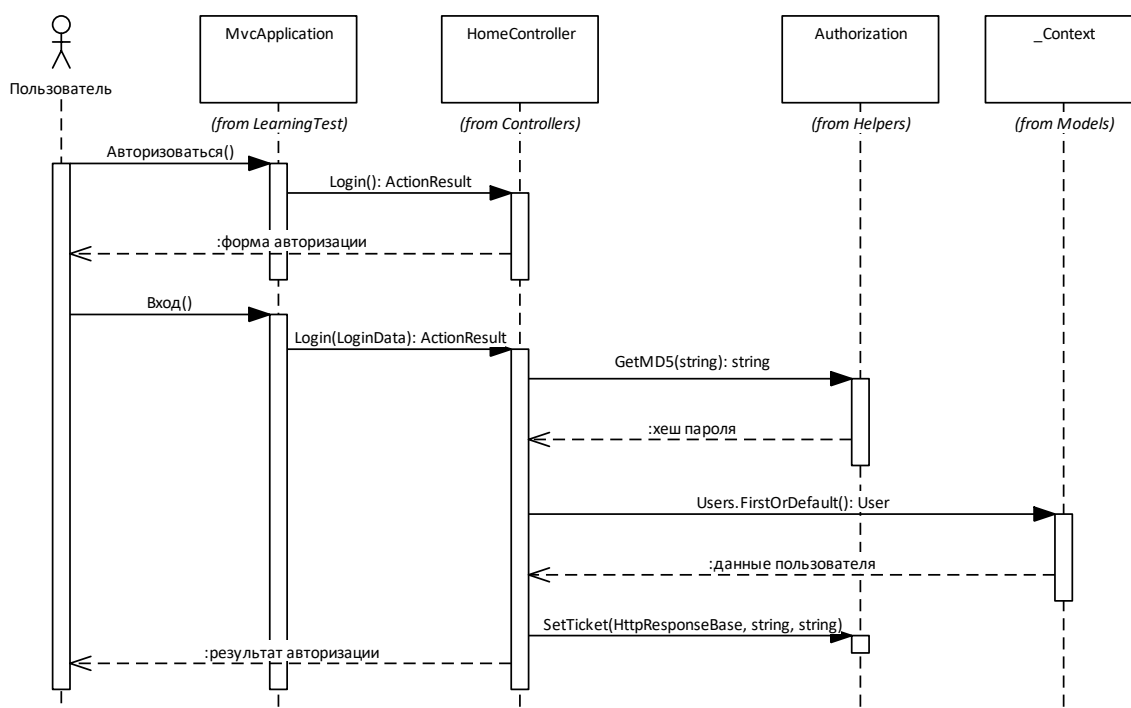


Рисунок 3.7 – Диаграмма последовательности «Авторизоваться»

Приведенная диаграмма последовательности описывает типичное взаимодействие классов приложения. Во взаимодействии принимают участие объекты следующих классов:

- MvcApplication – класс веб-приложения ASP.NET MVC;
- HomeController – класс контроллера управления аккаунтом;
- Authorization – класс дополнительных функций авторизации;
- Context – класс контекста модели данных.

Вариант использования предполагает следующую последовательность взаимодействия классов:

- пользователь нажимает кнопку «Авторизоваться», в результате чего идет обращение к классу MvcApplication;
- класс MvcApplication инициирует обращение к методу Login() контроллера HomeController;
- класс HomeController на основании соответствующего представления формирует форму авторизации, которая возвращается пользователю;
- пользователь вводит данные в форму авторизации и нажимает кнопку «Вход», в результате чего идет обращение к классу MvcApplication;
- класс MvcApplication инициирует обращение к методу Login(LoginData) контроллера HomeController;
- обрабатывая запрос класс HomeController последовательно обращается к методам GetMD5() класса Authorization, Users.FirstOrDefault() класса _Context, SetTicket() класса Authorization;
- пользователь возвращается результат авторизации.

3.5 Реализация проекта приложения

В среде разработки Visual Studio 2017 Community был создан проект веб-приложения ASP.NET MVC LearningTest (рисунок 3.8).

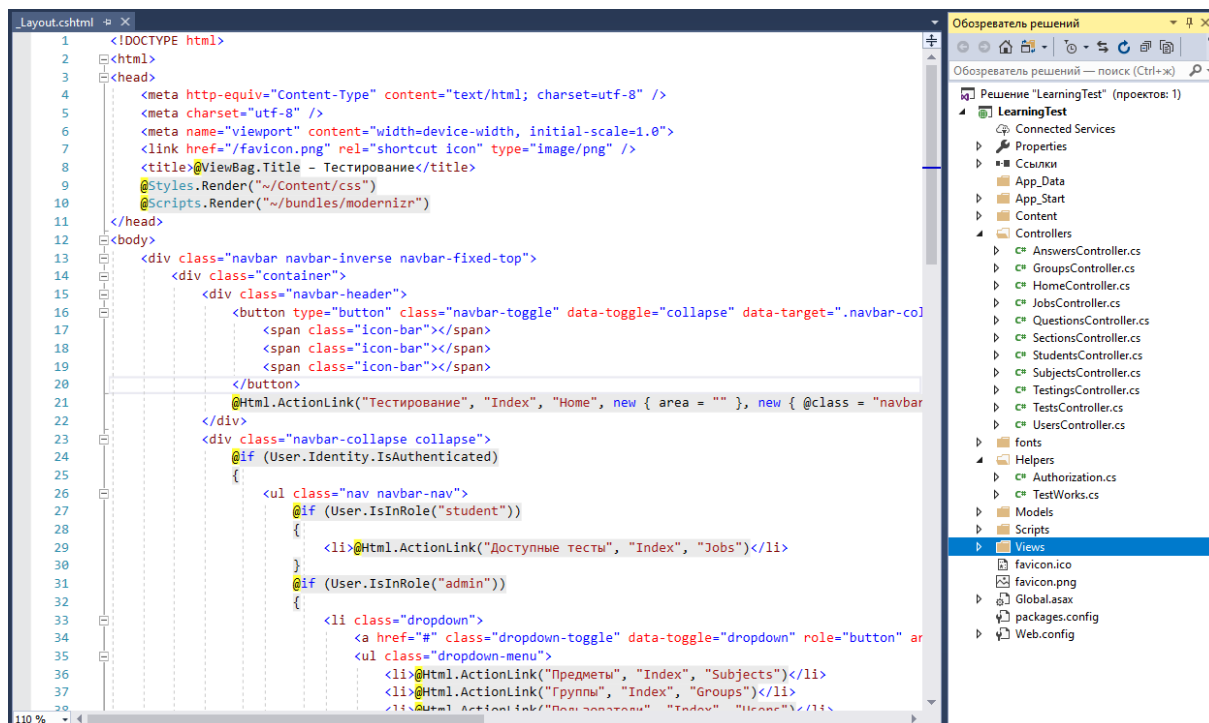


Рисунок 3.8 – Веб-приложение в среде разработки

Проект имеет следующую структуру файлов и каталогов, соответствующую классам и пакетам, выделенным на этапе проектирования, а также требованиям проекта ASP.NET MVC:

- 1) App_Start – каталог классов настройки веб-приложения:
 - а) BundleConfig.cs – класс, описывающий ссылки на внешние компоненты (файлы css и javascript);
 - б) FilterConfig.cs – класс, описывающий фильтры веб-приложения;
 - в) RouteConfig – класс, описывающий маршруты паттерна MVC, в соответствии с которыми интерпретируются запросы к веб-приложению;
- 2) Content – каталог файлов каскадных таблиц стилей (css);
- 3) Controllers – каталог классов контроллеров ASP.NET MVC:
 - а) AnswersController.cs – контроллер редактирования ответов на вопросы;
 - б) GroupsController.cs – контроллер редактирования групп;
 - в) StudentsController.cs – контроллер редактирования студентов;
 - г) JobsController.cs – контроллер прохождения тестирования;
 - д) QuestionsController.cs – контроллер редактирования вопросов теста;
 - е) SectionsController.cs – контроллер редактирования разделов теста;
 - ж) SubjectsController.cs – контроллер редактирования предметов;
 - з) TestingsController.cs – контроллер редактирования тестирований;
 - и) TestsController.cs – контроллер редактирования тестов;
 - к) UsersController.cs – контроллер редактирования пользователей;
 - л) HomeController.cs – контроллер управления аккаунтом;
- 4) fonts – каталог шрифтов;
- 5) Helpers – каталог вспомогательных классов:
 - а) Authorization.cs – помощник авторизации;
 - б) TestWorks.cs – помощник прохождения тестирования;
- 6) Models – каталог классов модели данных:
 - а) _Context.cs – класс контекста данных;
 - б) _Local.cs – файл вспомогательных классов данных;
 - в) Answer – класс модели данных ответа;
 - г) Group – класс модели данных группы;
 - д) Question – класс модели данных вопроса;
 - е) QuestionType – класс модели данных типа вопроса;
 - ж) Role – класс модели данных роли пользователя;
 - з) Section – класс модели данных раздела теста;
 - и) TestingAnswer – класс модели данных ответа на вопрос тестового набора;
 - к) TestingQuestion – класс модели данных вопроса тестового набора;
 - л) User – класс модели данных пользователя;
 - м) Testing – класс модели данных тестирования;
 - н) Subject – класс модели данных предмета;
 - о) Student – класс модели данных обучающегося;
 - п) TestingSet – класс модели данных тестового набора;

- p) Test – класс модели данных теста.
- 7) Scripts – каталог файлов javascript;
- 8) Views – каталог представлений ASP.NET MVC;
- 9) Global.asax – файл класса приложения ASP.NET MVC;
- 10) Web.config – файл с настройками, включая настройки подключения к базе данных.

Фрагменты cshtml-представлений, включающих код C# и разметку html, приведены в Приложении Б.

3.6 Схема работы системы

Обобщенная схема алгоритма работы системы приведена на рисунке 3.9.

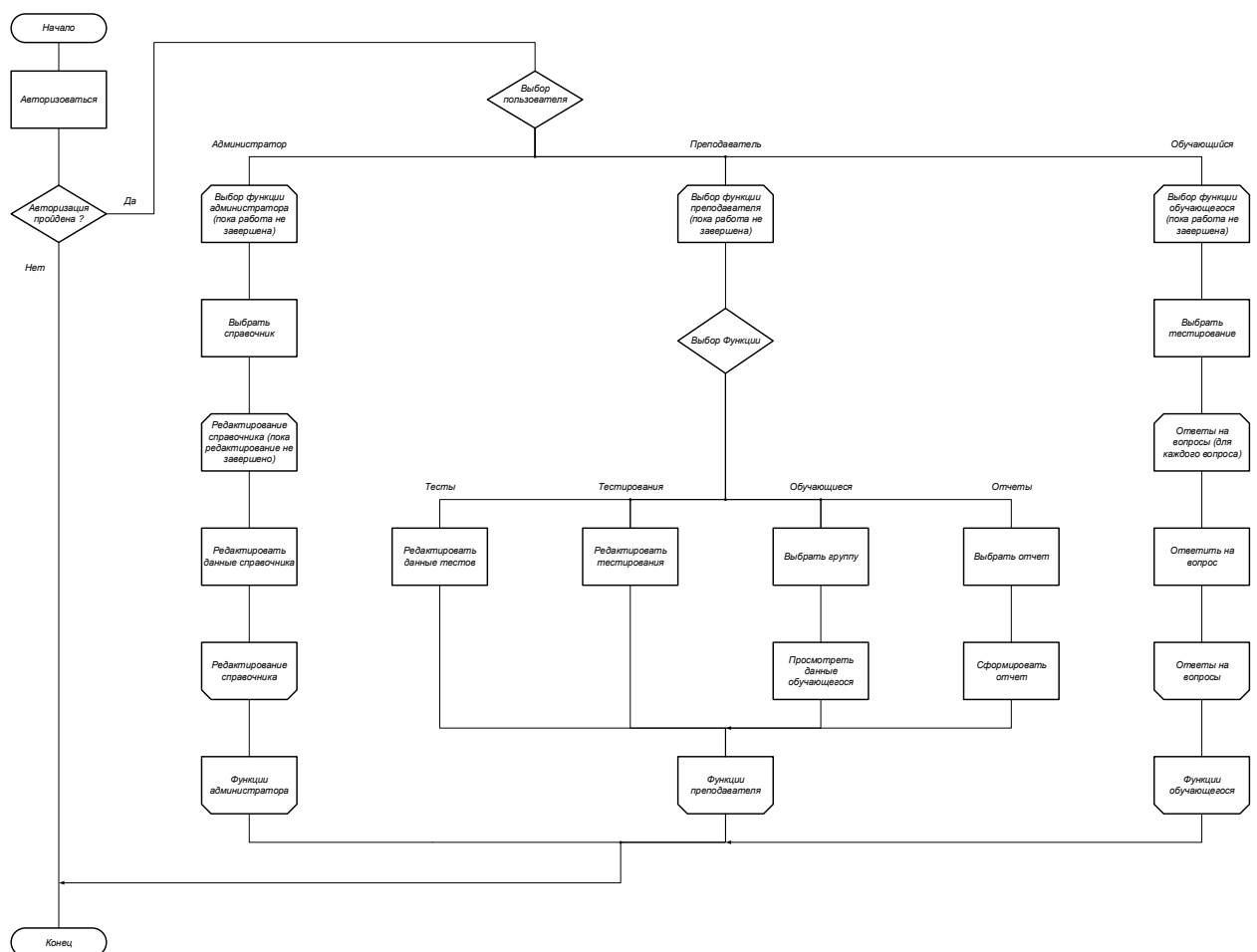


Рисунок 3.9 – Блок-схема алгоритма работы системы

Сценарий взаимодействия пользователя с системой предусматривает предварительную авторизацию. Доступ к функциям предоставляется только авторизованному пользователю.

В зависимости от роли, пользователю доступны определенные функциональные направления. Функции направления выполняются в соответствующих циклах:

- цикл функций администратора;
- цикл функций преподавателя;
- цикл функций обучающегося.

Администратор выполняет следующие функции:

- выбирает справочник;
- далее – в цикле редактирования справочника редактирует элементы.

Преподаватель выполняет следующие функции:

- редактирует данные тестов;
- редактирует тестирования;
- просматривает данные обучающихся;
- формирует отчеты.

Функции преподавателя также доступны администратору.

Обучающийся выполняет следующие функции:

- выбирает тестирование;
- далее – в цикле отвечает на вопросы теста.

Доступ к функциям осуществляется с использованием главного меню.

Доступ к операциям, относящихся к функциям, осуществляется с использованием экранных кнопок пользовательского интерфейса.

3.7 Физическая структура системы

Физическая структура системы в виде диаграммы развертывания в нотации UML приведена на рисунке 3.10.

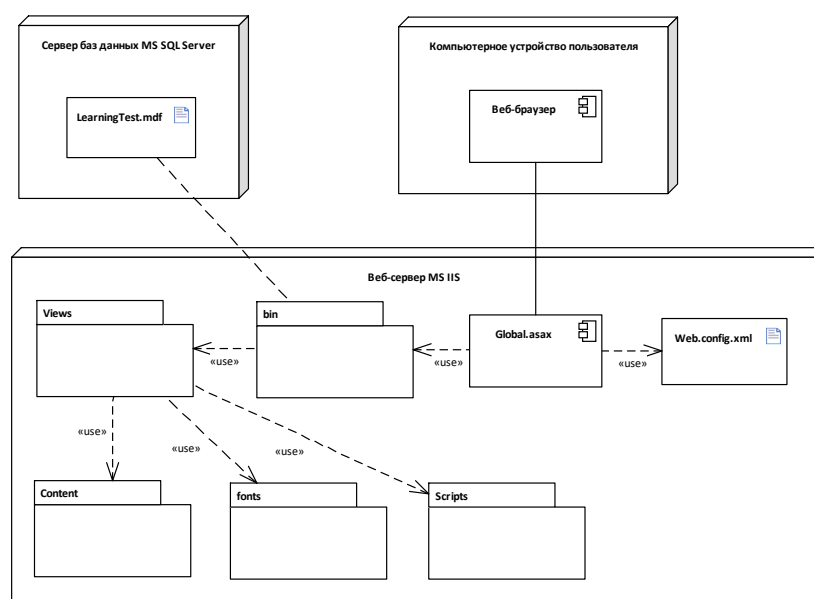


Рисунок 3.10 – Диаграмма развертывания приложения

Согласно приведенной диаграмме на этапе развертывания система включает следующие узлы:

- сервер баз данных MS SQL Server;
- компьютерное устройство пользователя;
- веб-сервер MS IIS.

Сервер баз данных MS SQL Server содержит базу данных LearningTest.mdf

Компьютерное устройство пользователя содержит веб-браузер.

Веб-сервер MS IIS содержит следующие компоненты веб-приложения:

- Global.asax – точка входа в приложение;
- Web.config.xml – файл конфигурации приложения;
- bin – пакет dll-библиотек, включая LearningTest.dll;
- Views – пакт представлений cshtml;
- Content – пакет файлов css;
- fonts – пакет файлов шрифтов;
- Scripts – пакет файлов javascript.

По результатам проектирования программного средства была разработана структурная схема, схема физической модели базы данных построенная на основе логической модели, построена диаграмма последовательности, которая описывает типичное взаимодействие классов приложения, а также обобщенная схема алгоритма работы системы.

4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

4.1 Программа испытаний

Объект испытаний: программное средство автоматизированной системы тестирования и контроля знаний.

Рабочее название: LearningTest.

Цель проведения испытаний: проверка соответствия характеристик разработанного программного средства функциональным требованиям, изложенным в задании на разработку.

Испытания проводятся разработчиком программного средства по окончании процесса его разработки согласно настоящей программе и методике испытаний. Ход проведения приемо-сдаточных испытаний документируется в протоколе испытаний.

Перечень проверок, проводимых при испытаниях:

- проверка комплектности программной документации;
- проверка комплектности состава программных средств;
- проверка соответствия технических характеристик программного средства;
- проверка степени выполнения требований функционального назначения программного средства.

Методика проведения проверок изложена в разделе 4.2 пояснительной записки к дипломному проекту.

Количественные характеристики, подлежащие оценке:

- комплектность программной документации;
- комплектность состава программных средств.

Качественные характеристики, подлежащие оценке: в ходе проведения приемосдаточных испытаний оценке подлежат качественные (функциональные) характеристики программного средства.

Проверке подлежит возможность выполнения программным средством перечисленных ниже функций:

- 1) авторизация пользователя;
- 2) изменение пароля пользователя;
- 3) доступ к функциям в зависимости от роли пользователя:
 - а) функции роли «Администратор»:
 - управление данными пользователей;
 - управление данными справочника учебных предметов;
 - управление данными справочника учебных групп;
 - изменение пароля;
 - б) функции роли «Преподаватель»:
 - просмотр данных обучающихся;
 - управление данными проверочных тестов;
 - открытие и закрытие тестирования;

- просмотр отчетов по результатам тестирования;
- изменение пароля;
- в) функции роли «Обучающийся»:
 - просмотр доступных тестов;
 - ответы на вопросы выбранного теста;

Последовательность проведения и режимы испытаний: испытания проводятся в последовательности, указанной в перечне проверок, проводимых при испытании.

Состав программной документации должен включать в себя:

- пояснительная записка к дипломному проекту;
- методика использования программного средства;
- программа и методика испытаний.

При проверке испытаний функциональные характеристики (возможности) программного средства подлежат проверке на соответствие требованиям, изложенным в техническом задании.

Технические средства: в состав технических средств должен входить IBM-совместимый персональный компьютер со следующими характеристиками:

- процессор Pentium 4 и старше с тактовой частотой, ГГц – 2, не менее;
- оперативная память объемом, ГБ – 2, не менее;
- свободное место на жестком диске, ГБ – 40, не менее;
- монитор с разрешением, пиксель – 1024x768, не менее;
- устройства ввода: клавиатура, мышь.

Программные средства: системные программные средства, используемые программного средства, должны быть представлены следующим программным обеспечением:

- локализованная версия операционной системы Windows 7 и старше;
- СУБД Microsoft SQL 2016 Local;
- локализованная версия платформы DOT.NET версии 4.6.2.

Для проведения испытаний предоставляется разработанное приложение с тестовыми данными, настроенное на подключение к базе данных Microsoft SQL 2016 Local, размещаемой в каталоге приложения.

Таблица 4.1 – Состав программных средств

Наименование	Размещение
Веб-приложение	Publish\Global.asax, Publish\bin, Publish\Views
Файл конфигурации приложения	Publish\Web.config
Вспомогательные файлы	Publish\Content, Publish\fonts, Publish\Scripts
Файл базы данных MS SQL 2016	Publish\App_Data\LearningTest.mdf

Необходимым и достаточным условием завершения текущего этапа испытаний и начала следующего является успешное завершение проверок, проводимых на текущем этапе согласно перечню.

4.2 Методика испытаний

Проверка комплектности программной документации на программное средство производится визуально. В ходе проверки сопоставляется состав и комплектность программной документации, представленной исполнителем, с перечнем программной документации. Результаты проверки заносятся в протокол. Проверка считается завершенной в случае соответствия состава и комплектности программной документации, представленной исполнителем, перечню программной документации, приведенному в указанном выше пункте. По результатам проведения проверки вносится запись в Протокол испытаний – «Комплектность программной документации соответствует (не соответствует) требованиям».

Проверка комплектности и состава программных средств производится визуально. В ходе проверки сопоставляется состав и комплектность программных средств, представленных исполнителем, с перечнем программных средств, приведенным в таблице 4.1. Результаты проверки заносятся в протокол. Проверка считается завершенной в случае соответствия состава и комплектности программных средств, представленных исполнителем, с перечнем программных средств. По результатам проведения проверки вносится запись в Протокол испытаний – «Комплектность программных средств соответствует (не соответствует) требованиям».

Проверка соответствия технических характеристик программного средства производится визуально. В ходе проверки осуществляется оценка (соответствует / не соответствует) по техническим характеристикам, приведенным в таблице 4.2. Результаты проверки заносятся в протокол.

Таблица 4.2 – Состав технических характеристик программного средства

Характеристика	Требование
Тип программного средства	Веб-приложение ASP.NET MVC
Язык программирования	C#
Платформа	DOT.NET 4.6.2
СУБД	Microsoft SQL 2016

Проверка считается завершенной в случае соответствия требованиям всех технических характеристик. По результатам проведения проверки вносится запись в Протокол испытаний – «Технические характеристики программного средства соответствуют (не соответствуют) требованиям».

Проверка выполнения функциональных требований выполняется согласно соответствующим пунктам методики использования программного средства. Проверка отдельной функции считается завершенной в случае

соответствия состава и последовательности действий оператора при выполнении данной функции указанному в методике использования программного средства. Результаты испытания (соответствует / не соответствуют) заносятся в Протокол испытаний по каждой функции. По результатам проведения проверки вносится запись в Протокол испытаний – «Функциональное назначение программного средства соответствует (не соответствует) требованиям».

По итогам всех этапов проверки в Протокол испытаний вносится запись – «Программное средство соответствует (не соответствует) требованиям задания на разработку».

4.3 Результаты тестирования

Работоспособность программного средства проверена путем его тестирования в соответствии с программой и методикой испытаний (разделы 4.1 и 4.2 пояснительной записки к дипломному проекту).

Настоящий протокол составлен по результатам тестирования программного средства автоматизированной системы тестирования и контроля знаний, проводимого на базе программно-технических средств разработчика.

Условия, в которых проводилось тестирование, соответствуют требованиям к техническим и программным средствам, приведенным в программе испытаний.

Для разработанного программного средства было проведено 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.).

В таблице 4.3 приведен результат критического тестирования проверки функциональности работы с данными.

В ходе испытаний получены следующие результаты.

Таблица 4.3 – Результат критического тестирования

Действие	Ожидаемый результат	Тест пройден?
Запустить АИС	Приложение запущено	Да
Авторизоваться в системе	Выполнен вход в систему	Да
Дождаться загрузки данных	Загрузка данных прошла успешно	
Открыть главную форму	Открылась главная форма	

Продолжение таблицы 4.3

Действие	Ожидаемый результат	Тест пройден?
Нажать на кнопку «Создать тест»	Отображение соответствующих форм	Да
Нажать на кнопку «Пройти тест»		
Нажать на кнопку «Учащиеся»		
Нажать на кнопку «Пройти тест»		
Нажать на кнопку «Пройти тест заново»	Перезагрузка формы	Да
Изменить тест	Данные в БД изменены	Да
Удалить тест	Удалена запись из БД	
Отчет по тестированию	Отчет успешно сформирован	Да
Выполнить команду «Выход»	Приложение закрыто	

Критическое тестирование пройдено успешно.

Таблица 4.4 – Проверка комплектности программной документации

Документ	Требуемое количество	Фактическое количество	Результат испытания
Пояснительная записка к дипломному проекту	1	1	соответствует
Методика использования программного средства	1	1	соответствует
Программа и методика испытаний	1	1	соответствует

Комплектность программной документации соответствует требованиям.

Таблица 4.5 – Проверка комплектности состава программных средств

Наименование	Требуемое количество	Фактическое количество	Результат испытания
Веб-приложение	1	1	соответствует
Файл конфигурации приложения	1	1	соответствует
Вспомогательные файлы	1	1	соответствует
Файл базы данных MS SQL 2016	1	1	соответствует

Комплектность программных средств соответствует требованиям.

Таблица 4.6 – Проверка технических характеристик программной системы

Характеристика	Требование	Результат испытания
Тип программного средства	Веб-приложение ASP.NET MVC	соответствует
Язык программирования	C#	соответствует
Платформа	DOT.NET 4.6.2	соответствует
СУБД	Microsoft SQL 2016	соответствует

Технические характеристики программной системы соответствуют требованиям.

Таблица 4.7 – Проверка выполнения функциональных требований

Функциональное требование	Результат испытания
Авторизация пользователя	соответствует
Изменение пароля пользователя	соответствует
Доступ к функциям в зависимости от роли пользователя:	
Функции роли «Администратор»:	
Управление данными пользователей	соответствует
Управление данными справочника учебных предметов	соответствует
Управление данными справочника учебных групп	соответствует
Изменение пароля	соответствует
Функции роли «Преподаватель»:	
Просмотр данных обучающихся	соответствует
Управление данными проверочных тестов	соответствует
Открытие и закрытие тестирования	соответствует
Просмотр отчетов по результатам тестирования	соответствует
Изменение пароля	соответствует
Функции роли «Обучающийся»:	
Просмотр доступных тестов	соответствует
Ответы на вопросы выбранного теста	соответствует

Функциональное назначение программного средства соответствует требованиям.

Выводы по результатам тестирования:

- испытания проведены в соответствии с планом и методикой испытаний;
- программное средство соответствует требованиям задания на разработку;
- критическое тестирование пройдено успешно;
- замечания и ошибки отсутствуют.

5 МЕТОДИКА ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО СРЕДСТВА

5.1 Установка и настройка

Программное обеспечение LearningTest выполнено в виде веб-приложения и предназначено для проверки знаний, обучающихся в учреждении образования.

Доступ к функциям системы осуществляется по результатам авторизации в соответствии с ролями: администратор, преподаватель, обучающийся (администратору доступны функции преподавателя)

Для функционирования веб-приложения обеспечения требуется:

- платформа DOT.NET 4.6.2;
- веб-сервер IIS 10;
- MS SQL Server 2016.

Доступ к веб-приложению осуществляется с использованием одного из перечисленных веб-браузеров:

- актуальная версия браузера на базе Chrome (Chrome, Opera и т.п.);
- актуальная версия Firefox;
- актуальная версия Internet Explorer.

Настройки подключения к базе данных производятся в файле Web.config, который находится в корне веб-приложения. Пользователь СУБД, указываемый в строке подключения, должен иметь полные права на базу данных.

Настройки доступности веб-приложения осуществляются в соответствии с инструкцией к веб-серверу IIS 10.

Для запуска веб-приложения необходимо в адресной строке браузера указать адрес, по которому доступно веб-приложение.

После запуска приложения откроется стартовая страница (рисунок 5.1).

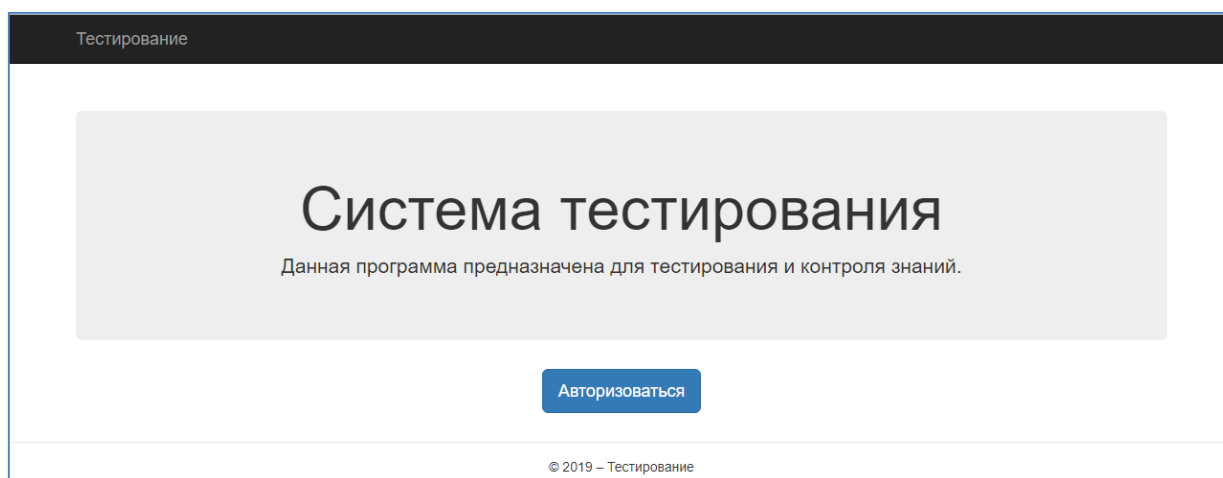
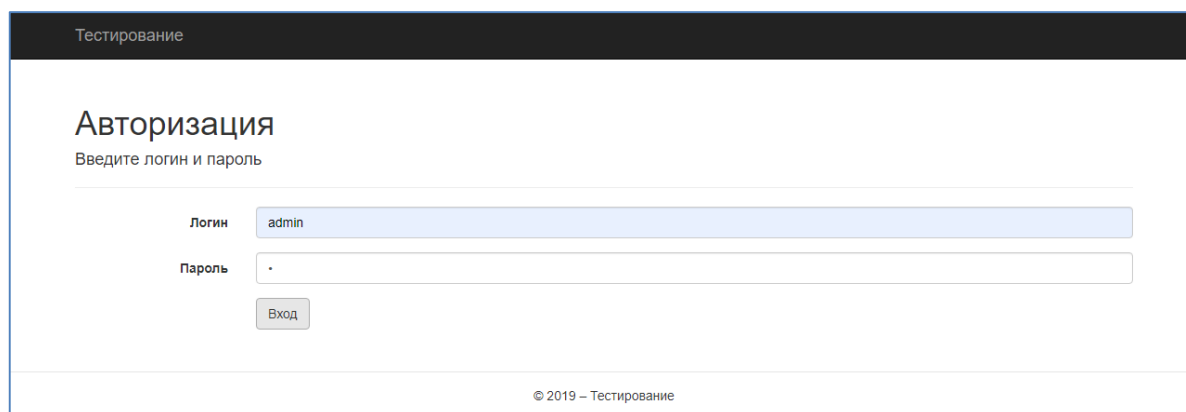


Рисунок 5.1 – Стартовая страница

5.2 Авторизация

Для доступа к функциям приложения требуется авторизация. На стартовой странице необходимо нажать кнопку «Авторизоваться». Для авторизации (рисунок 5.2) необходимо указать логин, пароль и нажать кнопку «Вход» (в тестовой базе данных логин и пароль администратора – admin и 1).



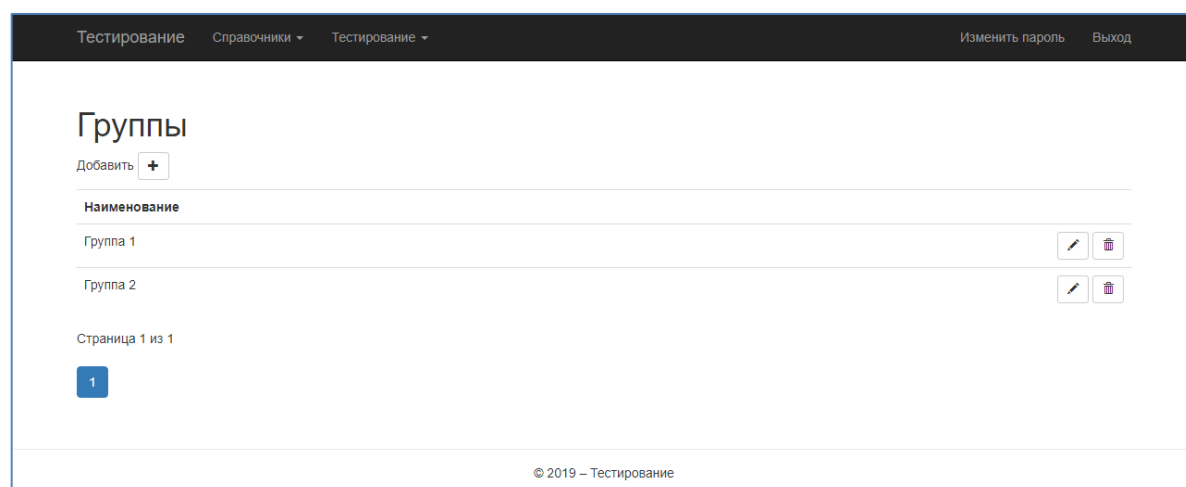
The screenshot shows a web interface for authorization. At the top, there is a dark header with the text "Тестирование". Below the header, the main content area has the title "Авторизация" and the instruction "Введите логин и пароль". There are two input fields: "Логин" (Login) with the value "admin" and "Пароль" (Password) with a single dot. Below the password field is a button labeled "Вход" (Login). At the bottom of the page, there is a footer with the text "© 2019 – Тестирование".

Рисунок 5.2 – Форма авторизации

5.3 Функции администратора

Программное средство включает три справочника, доступных для редактирования пользователю с ролью «Администратор»: группы, учебные предметы, пользователи.

Для редактирования справочника групп необходимо выбрать пункт меню «Справочники» и подпункт «Группы». В результате откроется страница списка групп (рисунок 5.3).



The screenshot shows the "Группы" (Groups) page. The top header has "Тестирование" on the left and "Изменить пароль" and "Выход" on the right. Below the header, there is a navigation bar with "Справочники" and "Тестирование". The main content area has the title "Группы" and a button "Добавить +" (Add +). Below this is a table with the header "Наименование" (Name). The table has two rows: "Группа 1" and "Группа 2". Each row has two icons: a pencil (edit) and a trash can (delete). Below the table, there is a footer with the text "Страница 1 из 1" (Page 1 of 1) and a button "1". At the bottom of the page, there is a footer with the text "© 2019 – Тестирование".

Рисунок 5.3 – Список групп

Для добавления группы необходимо в списке нажать кнопку «Добавить». В результате откроется форма добавления (рисунок 5.4), в которую необходимо внести данные и нажать кнопку «Сохранить».

Тестирование Справочники Тестирование Изменить пароль Выход

Новая группа

Введите данные

Наименование Группа 324

Сохранить К списку

© 2019 – Тестирование

Рисунок 5.4 – Форма добавления группы

Для изменения группы необходимо в списке нажать кнопку «Изменить». В результате откроется форма изменения (рисунок 5.5), в которую необходимо внести данные и нажать кнопку «Сохранить».

Тестирование Справочники Тестирование Изменить пароль Выход

Изменение группы

Введите данные

Наименование Группа 324-4

Сохранить К списку

© 2019 – Тестирование

Рисунок 5.5 – Форма изменения группы

Для удаления группы необходимо нажать кнопку «Удалить». В результате откроется форма подтверждения удаления (рисунок 5.6), в которой необходимо нажать кнопку «Удалить».

Тестирование Справочники Тестирование Изменить пароль Выход

Удаление группы

Подвердите удаление

Наименование Группа 324-4

Удалить К списку

© 2019 – Тестирование

Рисунок 5.6 – Форма подтверждения удаления группы

Для редактирования справочника предметов необходимо выбрать пункт меню «Справочники» и подпункт «Предметы». В результате откроется страница списка предметов (рисунок 5.7).

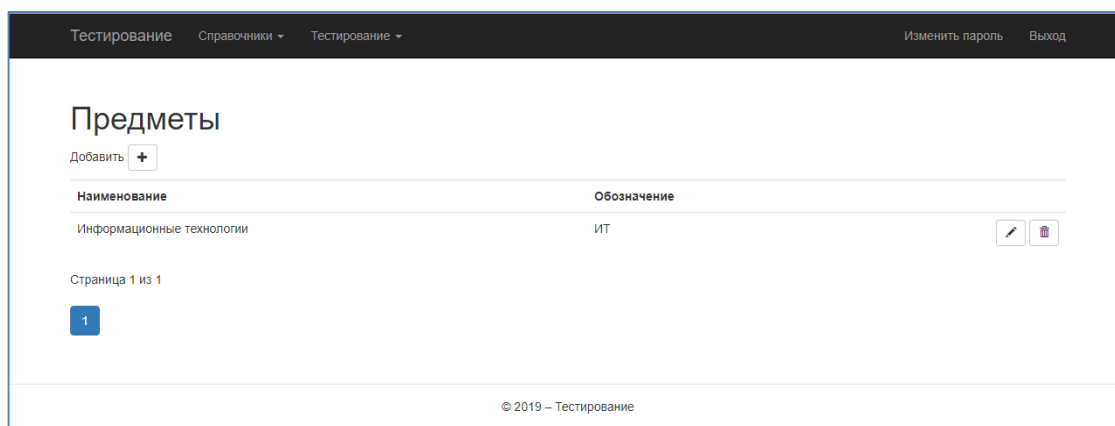


Рисунок 5.7 – Список предметов

Добавление, изменение и удаление предметов выполняется аналогично справочнику групп.

Для редактирования справочника пользователей необходимо выбрать пункт меню «Справочники» и подпункт «Пользователи». В результате откроется страница списка пользователей (рисунок 5.8).

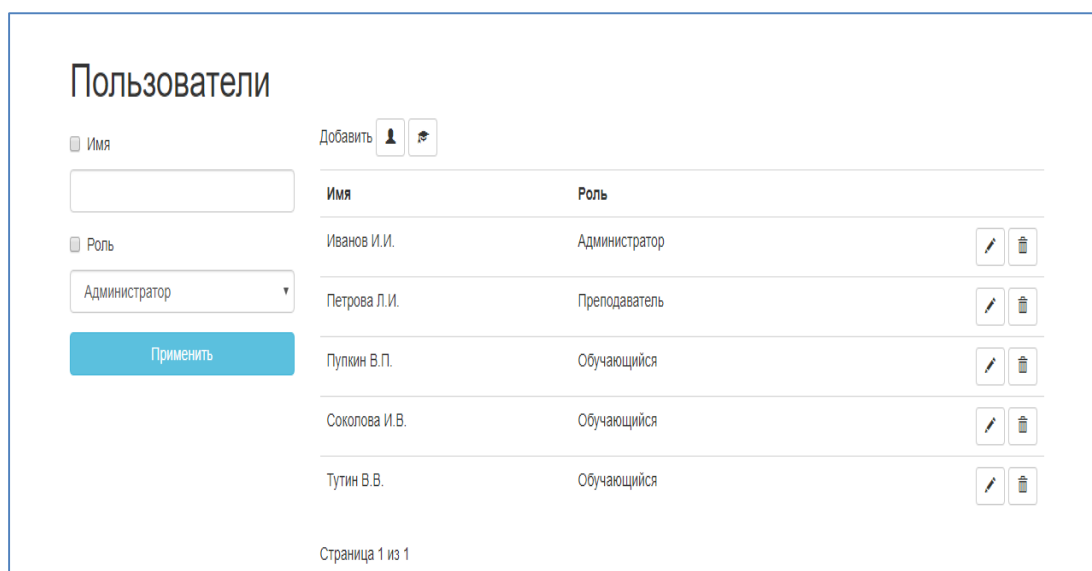


Рисунок 5.8 – Список пользователей

Пользователей в списке можно отобрать по фрагменту имени и роли. Для этого необходимо отметить соответствующие флажки, ввести фрагмент имени, выбрать роль и нажать кнопку «Применить».

Для добавления пользователя с ролями «Преподаватель» или «Администратор» необходимо в списке нажать кнопку «Добавить преподавателя или администратора». В результате откроется форма добавления (рисунок 5.9), в которую необходимо внести данные и нажать кнопку «Сохранить».

The screenshot shows a web application interface with a dark header. The header contains the text 'Тестирование' on the left and 'Изменить пароль' and 'Выход' on the right. Below the header, the main content area has the title 'Новый пользователь' and the instruction 'Укажите данные'. The form consists of four input fields: 'Наименование', 'Логин', 'Пароль', and 'Роль'. The 'Роль' field is a dropdown menu currently showing 'Преподаватель'. At the bottom of the form are two buttons: 'Сохранить' and 'К списку'. The footer of the page contains the text '© 2019 – Тестирование'.

Рисунок 5.9 – Форма добавления преподавателя или администратора

Для добавления пользователя с ролью «Обучающийся» необходимо в списке нажать кнопку «Добавить студента». В результате откроется форма добавления (рисунок 5.10), в которую необходимо внести данные и нажать кнопку «Сохранить».

The screenshot shows a web application interface with a dark header. The header contains the text 'Тестирование' on the left and 'Изменить пароль' and 'Выход' on the right. Below the header, the main content area has the title 'Новый студент' and the instruction 'Укажите данные'. The form consists of four input fields: 'Наименование', 'Логин', 'Пароль', and 'Группа'. The 'Группа' field is a dropdown menu currently showing 'Группа 1'. At the bottom of the form are two buttons: 'Сохранить' and 'К списку'. The footer of the page contains the text '© 2019 – Тестирование'.

Рисунок 5.10 – Форма добавления преподавателя или администратора

Для изменения пользователя необходимо в списке нажать кнопку «Изменить». В результате откроется форма изменения (рисунок 5.11), в которую необходимо внести данные и нажать кнопку «Сохранить» (для изменения пароля необходимо отметить соответствующий флажок).

Тестирование Справочники Тестирование Изменить пароль Выход

Изменение студента

Укажите данные

Наименование

Логин

☐ Изменить пароль

Пароль

Группа

© 2019 – Тестирование

Рисунок 5.11 – Форма изменения пользователя

Удаление пользователя осуществляется аналогично справочнику групп.

5.4 Функции преподавателя

Преподавателю доступны функции редактирования тестов, управления тестированиями, а также просмотра данных обучающихся.

Для редактирования тестов необходимо выбрать пункт меню «Тестирование» и подпункт «Тесты». В результате откроется страница списка тестов (рисунок 5.12).

Тестирование Тестирование Изменить пароль Выход

Тесты

☐ Имя

☐ Предмет

Добавить +

Наименование	Предмет	Минимальный балл
Тест по ИТ №1	Информационные технологии	50

Страница 1 из 1

© 2019 – Тестирование

Рисунок 5.12 – Список тестов

Тесты в списке можно отобразить по фрагменту наименования и предмету. Для этого необходимо отметить соответствующие флажки, ввести фрагмент наименования, выбрать предмет и нажать кнопку «Применить».

Для добавления теста необходимо в списке нажать кнопку «Добавить». В результате откроется форма добавления (рисунок 5.13), в которую необходимо внести данные и нажать кнопку «Сохранить».

Тестирование Тестирование ▾ Изменить пароль Выход

Новый тест

Введите данные

Наименование

Предмет Информационные технологии ▾

Минимальный балл 1

Сохранить К списку

© 2019 – Тестирование

Рисунок 5.13 – Форма добавления теста

Для созданного теста можно выполнить следующие операции: изменить, удалить, перейти к списку разделов, проверить, сформировать отчет по вопросам теста.

Для изменения теста необходимо в списке нажать кнопку «Изменить». В результате откроется форма изменения (рисунок 5.14), в которую необходимо внести данные и нажать кнопку «Сохранить».

Тестирование Справочники ▾ Тестирование ▾ Изменить пароль Выход

Изменение теста

Введите данные

Наименование Информационные технологии - тест №1

Предмет Информационные технологии ▾

Минимальный балл 35

Сохранить К списку

© 2019 – Тестирование

Рисунок 5.14 – Форма изменения теста

Для удаления тестов необходимо в списке нажать кнопку «Удалить». В результате откроется форма подтверждения удаления (рисунок 5.15), в которой необходимо нажать кнопку «Удалить».

Тест, на который имеются ссылки (созданы разделы, открыты тестирования), удалить нельзя.

Тестирование Справочники ▾ Тестирование ▾ Изменить пароль Выход

Удаление теста

Подвердите удаление

Наименование	Информационные технологии - тест №1
Предмет	
Минимальный балл	35

• Нельзя удалить тест, на который имеются ссылки!

Рисунок 5.15 – Форма подтверждения удаления теста

Для редактирования разделов теста необходимо в списке тестов нажать кнопку «Список разделов». В результате откроется список разделов теста (рисунок 5.16). Добавление, изменение и удаление разделов выполняется аналогично редактированию тестов.

Тестирование Справочники ▾ Тестирование ▾ Изменить пароль Выход

Разделы теста

Данные теста

Наименование	Тест по ИТ №2
Предмет	Информационные технологии
Минимальный балл	50

Добавить

Наименование	Число вопросов
Раздел первый	2

Рисунок 5.16 – Список разделов теста

Для редактирования вопросов раздела теста необходимо в списке разделов теста нажать кнопку «Список вопросов». В результате откроется список вопросов раздела теста (рисунок 5.17). Добавление, изменение и удаление вопросов выполняется аналогично редактированию тестов.

Тип	Содержание	Балл
один ответ	Новый вопрос	10

Рисунок 5.17 – Список вопросов раздела

Для редактирования ответов на вопрос необходимо в списке вопросов раздела нажать кнопку «Список ответов». В результате откроется список ответов на вопрос (рисунок 5.18). Добавление, изменение и удаление ответов выполняется аналогично редактированию тестов.

Содержание	Правильный
Ответ 1	✓

Рисунок 5.18 – Список ответов на вопрос

Для проверки теста на корректность необходимо в списке нажать кнопку «Проверить тест». В результате будет выведена страница с результатами проверки теста (рисунок 5.19).

Проверка теста

Данные теста

Наименование Тест по ИТ №2

Предмет Информационные технологии

Минимальный балл 50

Тест не прошел проверку!

Рисунок 5.19 – Результаты проверки теста

Для редактирования назначенных тестирований необходимо выбрать пункт меню «Тестирование» и подпункт «Тесты». В результате откроется страница списка тестирований (рисунок 5.20).

Тестирования в списке можно отобрать по периоду, группе и тесту. Для этого необходимо отметить соответствующие флажки, указать даты, выбрать группу, тест и нажать кнопку «Применить».

Тестирования

☒ Отбор по периоду

Начало периода

01.01.2019

Окончание периода

23.05.2019

☐ Группа

Группа 1

☐ Тест

Добавить +

Группа	Тест	Начало	Окончание	Закрывать
Группа 1	Информационные технологии - тест №1	28.04.2019	09.05.2019	<input type="checkbox"/>

Страница 1 из 1

Рисунок 5.20 – Список тестирований

Для начала нового тестирования необходимо нажать кнопку «Добавить». В результате откроется форма нового тестирования (рисунок 5.21), в которой необходимо указать данные и нажать кнопку «Сохранить».

Тестирование Справочники Тестирование Изменить пароль Выход

Новое тестирование

Укажите данные

Дата начала 23.05.2019

Дата окончания 30.05.2019

Тест Информационные технологии - тест №1

Группа (класс) Группа 1

Сохранить К списку

© 2019 – Тестирование

Рисунок 5.21 – Форма нового тестирования

Для просмотра состояния открытого тестирования необходимо в списке тестирований нажать кнопку «Просмотреть». В результате откроется форма отчета по тестированию в краткой форме, позволяющая завершить тестирование (рисунок 5.22).

Для завершения открытого тестирования необходимо в форме отчета по тестированию нажать кнопку «Завершить тестирование». В результате тестирование будет завершено, будут рассчитаны результаты тестирования и откроется полная форма отчета по тестированию (рисунок 5.23).

Отчет по тестированию

Данные тестирования

Тест Информационные технологии - тест №1

Группа (класс) Группа 1

Дата начала 28.04.2019

Дата окончания 09.05.2019

Информация по прохождению тестирования

Всего обучающихся в группе 2

Приступили к тестированию 1

Завершили 1

Рисунок 5.22 – Краткая форма отчета по тестированию

Отчет по тестированию

Данные тестирования

Тест
Минимальный балл
Группа (класс)
Дата начала
Дата окончания

Информационные технологии - тест №1
35
Группа 1
28.04.2019
09.05.2019

Информация по прохождению тестирования

Обучающийся	Участвовал в тестировании	Балл	Тестирование пройдено
Соколова И.В.	да	39	да
Пупкин В.П.	нет	0	нет

Всего обучающихся в группе
Поступили к тестированию
Прошли тестирование

2
1
1

К списку

Рисунок 5.23 – Полная форма отчета по тестированию

Для просмотра отчета по тестированию обучающегося необходимо в отчете по тестированию нажать кнопку «Отчет по тестированию обучающегося». В результате откроется форма отчета (рисунок 5.24) с подробными данными прохождения тестирования обучающимся. Из отчета можно перейти к данным обучающегося.

Для просмотра данных обучающихся необходимо выбрать пункт меню «Тестирование» и подпункт «Обучающиеся». В результате откроется форма списка обучающихся (рисунок 5.25).

Для просмотра данных обучающегося необходимо в списке обучающихся нажать на кнопку «Тестирования обучающегося». В результате откроется список тестирований, которые проходил обучающийся (рисунок 5.26).

Отчет по тестированию обучающегося

Данные тестирования: Соколова И.В.

Тестирование
Дата тестового набора
Результат

Информационные технологии - тест №1
29.04.2019
39
Тест пройден

Информация по прохождению тестирования

Вопрос	Балл	Ответ
Укажите основные принципы работы новой информационной технологии	8	Верный
Классификация информационных технологий (ИТ) по способу применения средств и методов обработки данных включает	8	Верный
Текстовый процессор – это программа, предназначенная для	5	Верный
Текстовый процессор входит в состав	6	Верный
Основную структуру текстового документа определяет	7	Верный
Заражение компьютерными вирусами может произойти в процессе	5	Верный

Верных ответов
Средний балл

6
6.5

К данным обучающегося

К данным тестирования

Рисунок 5.24 – Отчет по тестированию обучающегося

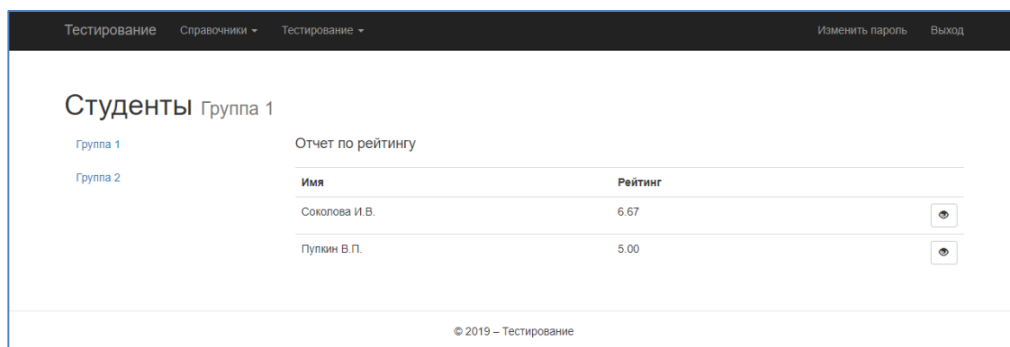


Рисунок 5.25 – Список обучающихся

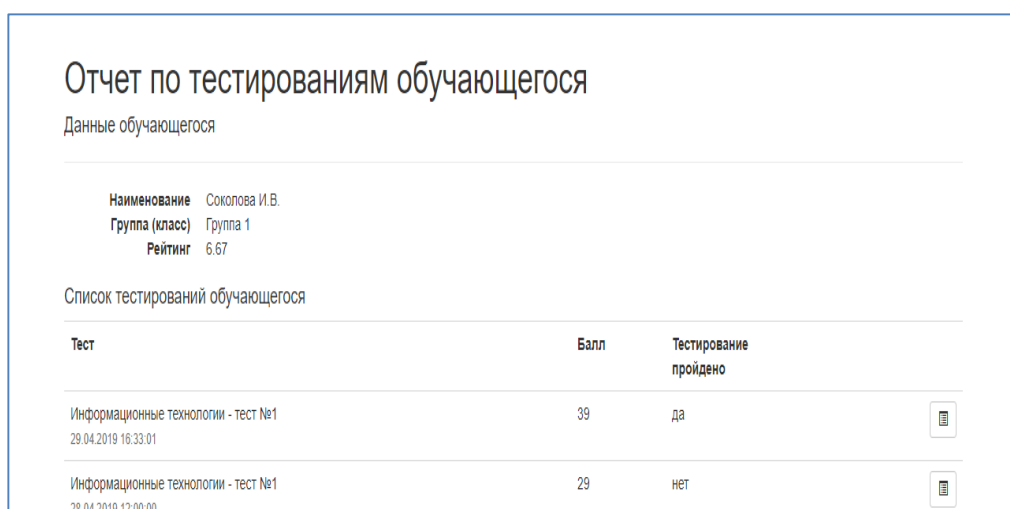


Рисунок 5.26 – Отчет по тестированиям обучающегося

5.5 Функции обучающегося

Обучающийся может просматривать список доступных тестов и проходить тестирования.

Для перехода к списку доступных тестов необходимо в главном меню выбрать пункт «Доступные тесты». В результате откроется список доступных тестирований (рисунок 5.27).

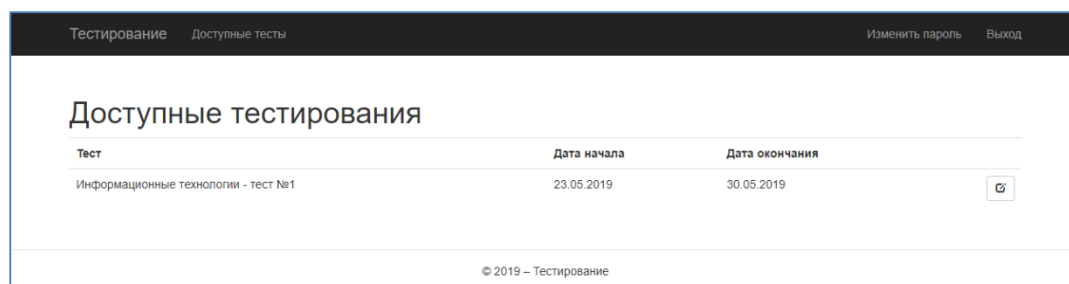


Рисунок 5.27 – Список доступных тестирований

Для ответа на вопросы теста необходимо в списке доступных тестирований нажать кнопку «Перейти к тестированию». В результате откроется список вопросов теста (рисунок 5.28).

Информационные технологии - тест №1

Завершить тестирование ☐

Вопрос	
Укажите основные принципы работы новой информационной технологии	<input type="checkbox"/>
Классификация информационных технологий (ИТ) по способу применения средств и методов обработки данных включает	<input type="checkbox"/>
Текстовый процессор – это программа, предназначенная для	<input type="checkbox"/>
Текстовый процессор входит в состав	<input type="checkbox"/>
Основную структуру текстового документа определяет	<input type="checkbox"/>

Рисунок 5.28 – Список вопросов теста

Для ответа на вопрос необходимо в списке вопросов теста нажать кнопку «Ответ на вопрос». В результате откроется форма ответа на вопрос (рисунки 5.29-5.31), в которую необходимо ввести ответ и нажать кнопку «Сохранить».

Для навигации по вопросам можно использовать кнопки «Предыдущий вопрос», «Следующий вопрос», «К списку вопросов».

Ответьте на вопрос

Выберите один или более вариантов ответа

Укажите основные принципы работы новой информационной технологии

- ☐ интерактивный режим работы с пользователем
- ☒ интегрированность с другими программами
- ☒ взаимосвязь пользователя с компьютером

Рисунок 5.29 – Ответ на вопрос с выбором нескольких вариантов

Рисунок 5.30 – Ответ на вопрос с выбором одного варианта

Рисунок 5.31 – Ответ на вопрос в свободной форме

Для завершения тестирования необходимо в списке вопросов теста нажать кнопку «Завершить тестирование». В результате будет показана страница с результатами прохождения теста (рисунок 5.32).

Тест	Информационные технологии - тест №1
Дата начала	23.05.2019
Сумма баллов	39
Тест сдан?	да/нет

Рисунок 5.32 – Результаты прохождения теста

Если тест не сдан, то до завершения тестирования обучающийся может попытаться сдать тест еще раз.

Набор вопросов для прохождения теста формируется динамически на основании рейтинга обучающегося и результатов выбранного теста.

6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И РЕАЛИЗАЦИИ ПРОГРАММНОГО СРЕДСТВА ДЛЯ АВТОМАТИЗАЦИИ КОНТРОЛЯ ЗНАНИЙ

6.1 Описание функций, назначения и потенциальных пользователей ПО

Автоматизированная система предназначена для организации контроля знаний с помощью тестирования. Область применения программного средства – учебные заведения общего, среднего специального и высшего образования. Объектом использования автоматизированной системы являются данные преподавателей и обучающихся, данные проверочных тестов, данные ответов обучающихся на проверочные тесты.

Автоматизированная система тестирования и контроля знаний обеспечивает выполнение следующих функций:

- авторизация пользователей;
- ведение нормативно-справочной информации;
- ввод данных проверочных тестов;
- открытие и закрытие тестирования;
- ответы на вопросы теста;
- формирование отчетов по результатам тестирования.

Автоматизированная система в отличие от конкурентов позволяет решать специфические задачи. Ее преимуществом является наличие функции автоматического формирования тестовых наборов. Также разрабатываемая система предоставляет доступ по ролям (администратор, преподаватель, обучающийся) и веб-интерфейс для всех категорий пользователей. Следовательно, система, разрабатываемая в настоящем дипломном проекте, актуальна и имеет преимущества в сравнении с конкурентами.

Автоматизированная система разработана в виде веб-приложения на базе платформы ASP.NET MVC. Для хранения данных используется СУБД Microsoft SQL Server.

Поскольку автоматизированная система разработана для свободной реализации на рынке компаний «Soft.NET», экономический эффект достигается за счет получения прибыли от реализации копий.

6.2 Расчет затрат на разработку ПО

Упрощенный расчет затрат на разработку ПО состоит из:

- затраты на основную заработную плату разработчиков;
- затраты на дополнительную заработную плату разработчиков;
- отчисления на социальные нужды;

– прочие затраты (амортизационные отчисления, расходы на электроэнергию, командировочные расходы, арендная плата за офисные помещения и оборудование, расходы на управление и реализацию и т.п.).

Затраты на основную заработную плату команды разработчиков определяются исходя из состава и численности команды, размеров месячной заработной платы каждого из участников команды, а также общей трудоемкости разработки программного обеспечения.

Расчет основной заработной платы участников команды осуществляется по формуле:

$$Z_0 = K_{\text{пр}} \sum_{i=0}^n Z_{\text{чи}} * t_i \quad (6.1)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$K_{\text{пр}}$ – коэффициент премий (1,5-2,0);

$Z_{\text{чи}}$ – часовая заработная плата i -го исполнителя (руб.);

t_i – трудоемкость работ, выполняемых i -м исполнителем (ч).

Месячная заработная плата программиста на предприятии, на котором проходила преддипломная практика, равна 1100. А тестировщика 800. Размер премии равен 50% от основной заработной платы сотрудников.

Часовая заработная плата определяется путем деления месячной заработной платы (оклад + надбавки) на количество рабочих часов в месяце (принять равным 168 часам).

$$Z_{\text{ч0}} = \frac{1100}{168} = 6,55 \text{ руб.}$$

$$Z_{\text{ч1}} = \frac{800}{168} = 4,76 \text{ руб.}$$

Трудоемкость определяется исходя из сложности разработки программного обеспечения и объема выполняемых им функций и равна 440 для программиста, и 110 для тестировщика.

Используя полученные данные, мы можем рассчитать затраты на основную заработную плату разработчиков.

$$Z_0 = 1,5 * 6,55 * 440 + 1,5 * 4,76 * 110 = 5108,4 \text{ руб.}$$

Расчет затрат на основную заработную плату рекомендуется осуществлять в форме таблицы 6.1.

Таблица 6.1 – Расчет затрат на основную заработную плату команды разработчиков

№	Участник команды	Вид выполняемой работы	Месячная заработная плата, руб.	Часовая заработная плата, руб.	Трудовое количество работ, ч	Зарплата по тарифу, руб.
1	2	3	4	5	6	7
1	Программист	Написание программы	1100	6,55	440	2882
2	Тестировщик	Тестирование программы	800	4,76	110	523,6
Премия						50%
Итого затраты на основную заработную плату разработчиков						5108,4

Затраты на дополнительную заработную плату команды разработчиков включает выплаты, предусмотренные законодательством о труде (оплата трудовых отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по формуле:

$$З_д = \frac{З_0 * Н_д}{100} \quad (6.2)$$

где $З_0$ – затраты на основную заработную плату, (руб.);

$Н_д$ – норматив дополнительной заработной платы, рекомендуется брать в пределах 10-20% (или по согласованию с консультантом по экономическому разделу).

$$З_д = \frac{5108,4 * 10}{100} = 510,84 \text{ руб.}$$

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле:

$$Р_{соц} = \frac{(З_0 + З_д) * Н_{соц}}{100} \quad (6.3)$$

где $Н_{соц}$ – норматив отчислений на социальные нужды, % (согласно действующему законодательству).

$$Р_{соц} = \frac{(5108,4 + 1021,68) * 35}{100} = 2145,52$$

Прочие затраты включаются в себестоимость разработки ПО в процентах от затрат на основную заработную плату команды разработчиков (табл.6.1) по формуле:

$$З_{пз} = \frac{З_0 * Н_{пз}}{100} \quad (6.4)$$

где $H_{пз}$ – норматив прочих затрат, рекомендуется брать в пределах 100-150% (или по согласованию с консультантом по экономическому разделу).

$$З_{пз} = \frac{5108,4 * 150}{100} = 7662,6$$

Полная сумма затрат на разработку программного обеспечения находится путем суммирования всех рассчитанных статей затрат (см. таблицу 6.2).

Общая сумма затрат на разработку.

$$З_p = 5108,4 + 1021,68 + 2353,42 + 7662,6 = 15938,2 \text{ руб.}$$

Таблица 6.2 – Затраты на разработку программного обеспечения

Статья затрат	Сумма, руб.
Основная заработная плата команды разработчиков	5108,4
Дополнительная заработная плата команды разработчиков	1021,68
Отчисления на социальные нужды	2145,52
Прочие затраты	7662,6
Общая сумма затрат на разработку	15938,2

6.3 Оценка результата (эффекта) от использования (или продажи) ПО

Любое программное обеспечение разрабатывается для удовлетворения какой-либо потребности, получения эффекта. В общем виде эффект может быть 2 видов:

Экономический эффект. Разработка и использование программного обеспечения напрямую влияет на экономические показатели деятельности пользователя (например, в результате разработки разработчик получает прирост прибыли от продажи ПО, автоматизированная система контроля знаний снижает временные затраты на расчет и оценку знаний учащихся, а также упрощает выполнение трудоемких операций). Данный эффект легко поддается стоимостной оценке и должен быть обязательно рассчитан при экономическом обосновании. В этом случае можно использовать один из типовых подходов к его расчету (п. п. 6.1 – 6.3).

Неэкономический эффект. Это эффект, напрямую не связанный с экономическими результатами деятельности компании: социальный, экологический, политический, технический. В данном случае использование ПО оказывает косвенное (опосредованное) влияние на экономические показатели деятельности пользователя, либо предоставляет ему дополнительные выгоды иного характера, которые зачастую невозможно оценить в стоимостном выражении, либо процесс оценки является сложным, трудоемким и неточным (например, облегчается ведение бухгалтерского учета, увеличивается оперативность получения управленческой информации и т.п.).

Экономический эффект от разработки программного обеспечения для собственных нужд может быть получен организацией от его использования, за счет:

- снижения трудоемкости выполнения «ручных» операций и бизнес процессов при внедрении программного обеспечения;
- ускорения скорости обслуживания клиентов и роста возможности обслуживания большего их количества в единицу времени, т.е. роста производительности труда;
- сокращения времени на подготовку разнообразной финансовой отчетности: увеличения числа выводимых в единицу времени документов, уменьшение времени подготовки отчета;
- повышение уровня сервиса (сокращение времени на устранение инцидентов).

При расчете экономического эффекта необходимо учесть прирост текущих затрат, связанных с использованием (эксплуатацией) ПС: прирост затрат на заработную плату обслуживающего персонала, прирост затрат на интернет-трафик и т.п.

В общем виде экономический эффект при использовании ПО будет рассчитываться по формуле:

$$\Delta\P_{\text{ч}} = (\mathcal{E}_3 - \Delta Z_{\text{тек}}) * (1 - H_{\text{п}}) \quad (6.5)$$

где \mathcal{E}_3 - экономия текущих затрат, полученная в результате применения ПО, руб.;

$\Delta Z_{\text{тек}}$ - прирост текущих затрат, связанных с использованием ПО, руб.;

$H_{\text{п}}$ - ставка налога на прибыль, в соответствии с действующим законодательством, %.

Отчетный год для наших расчетов мы берем 2019 год. И \mathcal{E}_3 - экономия текущих затрат, полученная в результате применения ПО за год составит 26880 рублей, так как по данным из учреждений образования столько средств в год затрачивается на приобретение и обслуживание программных средств для проверки знаний обучающихся. А $\Delta Z_{\text{тек}}$ - прирост текущих затрат, связанных с использованием ПО за год составит 1480, которые будут

использованы для доработки ПО. Учитывая эти данные, мы рассчитали экономический эффект в общем виде при использовании ПО в 2019 году.

$$\Delta\P_{\text{ч}} = (26880 - 1480) * (1 - 0,18) = 20828 \text{ руб.}$$

При разработке и модернизации ПО экономическое обоснование эффективности затрат на разработку (модернизацию) ПО, которые представляют собой инвестиции.

При совершенствовании ПО обоснование экономической эффективности затрат на совершенствование, которые являются текущими затратами предприятия, заключается в определении экономического эффекта, получаемого в результате применения, усовершенствованного ПО, по формуле:

$$\Delta\P_{\text{ч}} = (\text{Э}_3 - \text{З}_p - \Delta\text{З}_{\text{тек}}) * (1 - \text{Н}_{\text{п}}) \quad (6.6)$$

где З_p – затраты на разработку программного обеспечения, руб.

$$\Delta\P_{\text{ч}} = (26880 - 15938,2 - 1480) * (1 - 0,18) = 7758,68 \text{ руб.}$$

Также необходимо рассчитать рентабельность затрат на совершенствование ПО по формуле:

$$Y_p = \frac{\Delta\P_{\text{ч}}}{\text{З}_p} * 100\% \quad (6.7)$$

$$Y_p = \frac{7758,68}{15938,2} * 100\% \approx 48,7\%$$

Также данное программное средство можно реализовывать другим учреждениям, которые занимаются средним образованием. Это значительно увеличит экономический эффект.

В Республике Беларусь более трех тысяч школ, следовательно, минимально ожидаемое количество копий (лицензий) программного обеспечения, которое будет приобретено пользователями (N) равно 50.

Цена ПО определяется на основании анализа цен на аналоги и составляет 1000 руб.

Прибыль, полученная разработчиком от реализации ПО на рынке, в случае, если организация освобождена от уплаты налога на прибыль, рассчитывается по формуле:

$$\Pi = \text{Ц} * N - \text{НДС} - \text{З}_p \quad (6.8)$$

$$\Pi = 1000 * 50 - 8333,3 - 15938,2 = 25728,5 \text{ руб.}$$

Налог на добавленную стоимость определяется по формуле:

$$\text{НДС} = \frac{\text{Ц} * \text{N} * \text{Н}_{\text{дс}}}{100\% + \text{Н}_{\text{дс}}} \quad (6.9)$$

$$\text{НДС} = \frac{1000 * 50 * 20\%}{100\% + 20\%} \approx 8333,3 \text{ руб.}$$

Так как предприятие является плательщиком налога на прибыль, рассчитывается чистая прибыль по формуле (3.6).

$$\text{П}_\text{ч} = \text{П} - \frac{\text{П} * \text{Н}_\text{п}}{100} \quad (6.10)$$

$$\text{П}_\text{ч} = 25728,5 - \frac{25728,5 * 18\%}{100} = 21097,37 \text{ руб.}$$

Для оценки эффективности затрат в разработку ПО необходимо рассчитать уровень рентабельности затрат по формуле (3.7).

$$\text{У}_\text{р} = \frac{\text{П}(\text{П}_\text{ч})}{\text{З}_\text{р}} * 100\% \quad (6.11)$$

$$\text{У}_\text{р} = \frac{21097,37}{15938,2} * 100\% = 132,37\%$$

Проект будет экономически эффективным при его реализации другим учреждениям, так как рентабельность затрат на разработку программного средства будет значительно выше средней процентной ставки по банковским депозитным вкладам.

ЗАКЛЮЧЕНИЕ

В ходе дипломного проектирования были решены и выполнены следующие задачи и цели:

- проанализированы существующие программные средства для автоматизации тестирования и контроля знаний;
- разработаны требования к проектированию программного средства для автоматизации контроля знаний;
- проведено тестирование программного средства для автоматизации контроля знаний;
- разработано руководство по использованию программного средства для автоматизации контроля знаний;
- рассчитана экономическая эффективность разработки и реализации программного средства для автоматизации контроля знаний.

В процессе разработки программного средства для автоматизации контроля знаний выполнено моделирование предметной области, разработаны функциональная спецификация и проектная документация программного средства. Программное средство разработано на базе фреймворка .NET средствами языка программирования C#. Для хранения данных использована СУБД Microsoft SQL Server. Программное средство разработано в виде веб-приложения с использованием технологии ASP.NET MVC.

В разработанном программном средстве реализованы возможности формирования тестового материала, организации и проведения тестирования с использованием динамически формируемых тестовых наборов, учитывающих рейтинг обучающегося и результаты прохождения тестирования. Разработанное программное средство в отличие от конкурента позволяет решать специфические задачи. Его преимуществом является наличие функции автоматического формирования тестовых наборов. Также разрабатываемая система предоставляет доступ по ролям (администратор, преподаватель, обучающийся) и веб-интерфейс для всех категорий пользователей.

Изложенное позволяет сделать вывод о том, что цель, поставленная в дипломном проекте, достигнута. Разработанное программное средство предоставляет средства и методы организации контроля знаний.

Разработанное в рамках дипломного проекта программное средство для автоматизации контроля знаний с использованием языка C#, внедрено в Государственном учреждении образования "Средняя школа №3 г.Ошмяны".

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Программа для проведения тестирований PikaTest [Электронный ресурс]. – Режим доступа: <http://pedsovet.su/load/9-1-0-26215>. – Дата доступа:
- [2] Янаева, У.Н. Система компьютерной проверки знаний тестированием UniTest: руководство пользователя / У.Н. Янаева. – Красноярск: ИПК СФУ, 2016. – 76 с.
- S
у
s
t
- [4] Руководство по системе тестирования INDIGO [Электронный ресурс]. – Режим доступа: <https://indigotech.ru/downloads/files/> – Дата доступа: 12.11.2019.
- [5] Программа обеспечения для автоматизации компьютерного тестирования [Электронный ресурс]. – Режим доступа: <http://right2k.com/ds/unitest/> – Дата доступа: 15.11.2019.
- [6] Сиерра, К. Изучаем Java / К. Сиерра, Б. Бейтс. – М.: Эксмо, 2016. – 720 с.
- [7] Фримен, А. ASP.NET 4.5 с примерами для профессионалов / А. Фримен. – М.: Вильямс, 2017. – 1120 с.
- [8] Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – М.: Вильямс, 2016. – 848 с.
- [9] Дайкстра, Т. Создание веб-проектов ASP.NET в Visual Studio [Электронный ресурс] / Т. Дайкстра. – Microsoft, 2018. – Режим доступа: – Дата доступа: 10.11.2019.
- [10] Entity Framework [Электронный ресурс] / Д. Вега. – Microsoft, 2018. – Режим доступа: <https://docs.microsoft.com/ru-ru/ef/ef6/>. – Дата доступа:
- [11] Bootstrap [Электронный ресурс] / Официальный сайт. – Bootstrap, 2018. – Режим доступа: <https://getbootstrap.com/docs/3.3/>. – Дата доступа:
- [12] JQuery [Электронный ресурс] / Официальный сайт. – The jQuery Foundation, 2018. – Режим доступа: <https://jquery.com>. – Дата доступа:
- [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru->
- [14] Рихтер, Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Д. Рихтер. – СПб.: Питер, 2017. – 896 с.
- [15] Новиков, Б.А. Настройка приложений баз данных / Б.А. Новиков, Г.Р. Домбровская. – СПб.: БХВ-Петербург, 2016. – 240 с.

- [16] Станек, У.Р. Microsoft SQL Server 2012. Справочник администратора / У.Р. Станек. – СПб.: БХВ-Петербург, 2018. – 576 с.
- [17] Кузнецов, С.Д. Базы данных / С.Д. Кузнецов. – М.: Academia, 2016. – 496 с.
- [18] Илюшечкин, В.М. Основы использования и проектирования баз данных / В.М. Илюшечкин. – М.: Юрайт, 2016. – 214 с.
- [19] Дубейковский, В.И. Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM / В.И. Дубейковский. – М.: Нолидж, 2015. – 284 с.
- [20] Хаф, Л. Проектирование информационных систем / Л. Хаф. – М.: Бином, 2013. – 728 с.
- [21] Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Джекобсон. – Санкт-Петербург: Питер, 2016. – 432 с.
- [22] Ларман, К. Применение UML и шаблонов проектирования / Крэг Ларман. – М.: Издательский дом «Вильямс», 2017. – 624 с.
- [23] Палицын, В.А. Техничко-экономическое обоснование дипломных проектов: Метод. пособие для студ. всех спец. / В.А. Палицын. – Минск: БГУИР, 2015. – 51 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Фрагменты программного кода

Helpers\TestWorks.cs

```
using LearningTest.Models;
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;

namespace LearningTest.Helpers
{
    public class TestWorks
    {
        // Проверка теста
        public static List<Result> TestCheck(Test test)
        {
            List<Result> results = new List<Result>();
            if (test.Sections.Count > 0)
            {
                int score = 0;
                foreach (Section section in test.Sections)
                {
                    if (section.Questions.Count > 0)
                    {
                        if (section.Questions.Count <
section.CntQuestions)
                        {
                            results.Add(new Result(false, "Раздел \""
+ section.Name + "\": не достаточно вопросов!", "/Questions/Index/" +
section.Id));
                        }
                        int cnt = 0;
                        foreach (Question question in
section.Questions.OrderBy(p => p.Score))
                        {
                            if (cnt < section.CntQuestions)
                            {
                                score += question.Score;
                            }
                            cnt++;
                            if (question.Answers.Count > 0)
                            {
                                if (question.Answers.Where(p =>
p.Correct).Count() == 0)
                                {
                                    results.Add(new Result(false,
"Вопрос \"" + question.Content + "\" (" + question.QuestionType.Name
```

```

+ "): не указано ни одного правильного ответа!", "/Answers/Index/" +
question.Id));
    }
    else if (question.QuestionTypeId != 2
&& question.Answers.Where(p => p.Correct).Count() > 1)
    {
        results.Add(new Result(false,
"Вопрос \"\" + question.Content + "\" (" + question.QuestionType.Name
+ "): более одного правильного ответа!", "/Answers/Index/" +
question.Id));
    }
    }
    else
    {
        results.Add(new Result(false, "Вопрос
\"\" + question.Content + "\": отсутствуют ответы!", "/Answers/Index/"
+ question.Id));
    }
    }
    }
    else
    {
        results.Add(new Result(false, "Раздел \"\" +
section.Name + "\": отсутствуют вопросы!", "/Questions/Index/" +
section.Id));
    }
    }
    if (score < test.MinScore)
    {
        results.Add(new Result(false, "Тест: минимальный
балл недостижим!", "/Tests/Edit/" + test.Id));
    }
    }
    else
    {
        results.Add(new Result(false, "Тест: отсутствуют
разделы!", "/Sections/Index/" + test.Id));
    }
    //
    return results;
}

// Расчет тестового набора по окончании тестирования
public static void TestingSetCalc(_Context db, TestingSet
testingSet)
{
    // Расчет результатов тестирования
    int score = 0;
    foreach (TestingQuestion testingQuestion in
db.TestingQuestions.Where(p => p.TestingSetId == testingSet.Id))
    {
        bool flag = true;
        switch (testingQuestion.Question.QuestionTypeId)

```



```

        {
            case 1:
                int cnt1 = 0;
                foreach (TestingAnswer testingAnswer in
db.TestingAnswers.Where(p => p.TestingQuestionId ==
testingQuestion.Id))
                {
                    if (testingAnswer.Checked ==
testingAnswer.Answer.Correct)
                    {
                        cnt1++;
                    }
                }
                if (cnt1 == 1)
                {
                    flag = true;
                }
                break;
            case 2:
                int cnt2 = 0;
                int cntCorrect = db.Answers.Where(p =>
p.QuestionId == testingQuestion.QuestionId && p.Correct).Count();
                foreach (TestingAnswer testingAnswer in
db.TestingAnswers.Where(p => p.TestingQuestionId ==
testingQuestion.Id))
                {
                    if (testingAnswer.Checked ==
testingAnswer.Answer.Correct)
                    {
                        cnt2++;
                    }
                }
                if (cnt2 == cntCorrect)
                {
                    flag = true;
                }
                break;
            case 3:
                TestingAnswer testingAnswer3 =
db.TestingAnswers.FirstOrDefault(p => p.TestingQuestionId ==
testingQuestion.Id);
                if (testingAnswer3 != null)
                {
                    string str =
testingAnswer3.Content.Trim().ToLower();
                    string strCorrect =
testingAnswer3.Answer.Content.Trim().ToLower();
                    if (str.Equals(strCorrect))
                    {
                        flag = true;
                    }
                }
                break;
        }
    }
}

```

```

        case 4:
            TestingAnswer testingAnswer4 =
db.TestingAnswers.FirstOrDefault(p => p.TestingQuestionId ==
testingQuestion.Id);
            if (testingAnswer4 != null)
            {
                int number =
Convert.ToInt32(testingAnswer4.Content.Trim());
                int numberCorrect =
Convert.ToInt32(testingAnswer4.Answer.Content.Trim());
                if (number == numberCorrect)
                {
                    flag = true;
                }
            }
            break;
        default:
            break;
    }
    if (flag)
    {
        testingQuestion.Correct = true;
        score += testingQuestion.Question.Score;
        db.Entry(testingQuestion).State =
EntityState.Modified;
    }
}
testingSet.Score = score;
testingSet.Completed = true;
if (score >= testingSet.Testing.Test.MinScore)
{
    testingSet.IsPassed = true;
}
db.Entry(testingSet).State = EntityState.Modified;
// Расчет рейтинга
int cntTestingQuestion = 0;
int cntScore = 0;
foreach (TestingQuestion testingQuestion in
db.TestingQuestions.Where(p => p.TestingSet.UserId ==
testingSet.UserId && p.TestingSet.Completed))
{
    cntTestingQuestion++;
    cntScore += testingQuestion.Question.Score;
}
Student student = db.Students.Find(testingSet.UserId);
if (student != null && cntTestingQuestion > 0)
{
    student.Rate = Math.Round((decimal)cntScore /
cntTestingQuestion, 2);
    db.Entry(student).State = EntityState.Modified;
}
// Сохраняем изменения
db.SaveChanges();

```

```

    }

    // Создание тестового набора
    public static void TestingSetCreate(_Context db, Testing
testing, Student student)
    {
        // Определяем базовое значение по рейтингу обучающегося
        decimal baseScore = student.Rate;
        // Находим предыдущий тестовый набор, если таковой
имеется
        TestingSet testingSetOld =
db.TestingSets.OrderByDescending(p => p.DateAt).FirstOrDefault(p =>
p.UserId == student.Id && p.TestingId == testing.Id && p.Completed ==
true);
        // Если найден, определяем базовое значение по среднему
баллу набора
        if (testingSetOld != null)
        {
            baseScore = testingSetOld.Score /
testingSetOld.TestingQuestions.Count();
        }
        // Создаем тестовый набор
        TestingSet testingSet = new TestingSet()
        {
            DateAt = DateTime.Now,
            Testing = testing,
            User = student.User,
            Completed = false,
            Score = 0,
            IsPassed = false
        };
        db.TestingSets.Add(testingSet);
        // Обход разделов теста
        foreach (Section section in db.Sections.Where(p =>
p.TestId == testing.TestId))
        {
            List<Question> questions = new List<Question>();
            // Отбираем все вопросы раздела, стоимость которых
меньше либо равна базовой
            foreach (Question question in db.Questions.Where(p =>
p.SectionId == section.Id && p.Score <=
baseScore).OrderByDescending(p => p.Score))
            {
                if (questions.Count() < section.CntQuestions)
                {
                    questions.Add(question);
                }
                else
                {
                    break;
                }
            }
        }
    }

```

```

        // Если вопросов недостаточно, дополняем вопросами,
стоимость которых выше базовой
        if (questions.Count() < section.CntQuestions)
        {
            foreach (Question question in
db.Questions.Where(p => p.SectionId == section.Id && p.Score >
baseScore).OrderBy(p => p.Score))
            {
                if (questions.Count() < section.CntQuestions)
                {
                    questions.Add(question);
                }
                else
                {
                    break;
                }
            }
        }
        // Создаем вопросы тестового набора
        foreach (Question question in questions)
        {
            TestingQuestion testingQuestion = new
TestingQuestion()
            {
                Question = question,
                TestingSet = testingSet,
                Correct = false
            };
            db.TestingQuestions.Add(testingQuestion);
            // Создаем ответы
            if (question.QuestionTypeId < 3)
            {
                foreach (Answer answer in db.Answers.Where(p
=> p.QuestionId == question.Id))
                {
                    db.TestingAnswers.Add(new TestingAnswer()
                    {
                        TestingQuestion = testingQuestion,
                        Answer = answer,
                        Content = "",
                        Checked = false
                    });
                }
            }
            else
            {
                Answer answer = db.Answers.First(p =>
p.QuestionId == question.Id);
                db.TestingAnswers.Add(new TestingAnswer()
                {
                    TestingQuestion = testingQuestion,
                    Answer = answer,
                    Content = "",

```

```

                Checked = false
            });
        }
    }
    // Сохраняем изменения
    db.SaveChanges();
}
}
}

```

Controllers\HomeController.cs

```

using System;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web.Mvc;
using System.Web.Security;
using LearningTest.Models;

namespace LearningTest.Controllers
{
    public class HomeController : Controller
    {
        private _Context db = new _Context();

        // Домашняя страница
        public ActionResult Index()
        {
            return View();
        }

        // Форма авторизации
        public ActionResult Login()
        {
            // Если пользователь авторизован - редирект на рабочий
            стол

            if (HttpContext.User.Identity.IsAuthenticated)
            {
                return RedirectToAction("Index");
            }
            return View();
        }

        // Обработка формы авторизации
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Login(LoginData item)
        {
            // Если пользователь авторизован - редирект на рабочий
            стол

            if (HttpContext.User.Identity.IsAuthenticated)

```

```

        {
            return RedirectToAction("Index");
        }
        // Проверка валидации модели
        if (ModelState.IsValid)
        {
            // Проверка результата авторизации
            string passwordMD5 =
Helpers.Authorization.GetMD5(item.Password);
            User user = db.Users.FirstOrDefault(p => p.Login ==
item.Login && p.Password == passwordMD5);
            if (user == null)
            {
                // Если авторизация не пройдена
                ModelState.AddModelError("", "Не удалось
авторизоваться!");
            }
            else
            {
                // Если авторизация успешна
                // Запись билета авторизации
                Helpers.Authorization.SetTicket(Response,
user.Id.ToString(), user.Role.Code);
                // Переадресация
                return RedirectToAction("Index");
            }
        }
        return View(item);
    }

    // Форма изменения пароля
    [Authorize]
    public ActionResult Password()
    {
        int authUserId = Convert.ToInt32(User.Identity.Name);
        // Поиск записи пользователя по Id
        User user = db.Users.FirstOrDefault(p => p.Id ==
authUserId);
        if (user == null)
        {
            return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        return View();
    }

    // Обработка формы изменения пароля
    [Authorize]
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Password(PasswordData item)
    {
        if (ModelState.IsValid)

```

```

        {
            int authUserId = Convert.ToInt32(User.Identity.Name);
            // Поиск записи пользователя по Id
            User user = db.Users.FirstOrDefault(p => p.Id ==
authUserId);
            if (user == null)
            {
                return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            else
            {
                user.Password =
Helpers.Authorization.GetMD5(item.Password);
                try
                {
                    db.Entry(user).State = EntityState.Modified;
                    db.SaveChanges();
                    return RedirectToAction("Index");
                }
                catch
                {
                    ModelState.AddModelError("", "Не удалось
изменить пароль...");
                }
            }
            return View(item);
        }

// Обработка выхода пользователя
[Authorize]
public ActionResult Logout()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Index");
}
}
}

```

Controllers\TestingsController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using LearningTest.Models;
using PagedList;

```

```

namespace LearningTest.Controllers
{
    [Authorize(Roles = "admin, teacher")]
    public class TestingsController : Controller
    {
        private _Context db = new _Context();

        // Формирует список по фильтру
        private List<Testing> GetList(TestingsFilter filter)
        {
            int userId = Convert.ToInt32(User.Identity.Name);
            List<Testing> list = db.Testings.OrderByDescending(p =>
p.DateAt).Where(p => p.Test.UserId == userId).ToList();
            if (filter.UseGroup)
            {
                list = list.Where(p => p.GroupId ==
filter.GroupId).ToList();
            }
            if (filter.UseTest)
            {
                list = list.Where(p => p.TestId ==
filter.TestId).ToList();
            }
            if (filter.UsePeriod)
            {
                DateTime dateAt = filter.DateAt.Date;
                DateTime dateTo = filter.DateTo.Date;
                dateAt = new DateTime(dateAt.Year, dateAt.Month,
dateAt.Day, 0, 0, 0);
                dateTo = new DateTime(dateTo.Year, dateTo.Month,
dateTo.Day, 23, 59, 59);
                if (dateTo <= dateAt)
                {
                    dateTo = new DateTime(dateAt.Year, dateAt.Month,
dateAt.Day, 23, 59, 59);
                }
                list = list.Where(p => p.DateAt >= dateAt && p.DateAt
<= dateTo).ToList();
            }
            return list;
        }

        // Список с фильтром
        public ActionResult Index(int? id)
        {
            int userId = Convert.ToInt32(User.Identity.Name);
            if (db.Tests.Where(p => p.UserId == userId).Count() == 0)
            {
                return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            TestingsFilter filter = Session["TestingsFilter"] as
TestingsFilter;

```



```

        if (filter == null)
        {
            DateTime dateTo = DateTime.Now.Date;
            DateTime dateAt = new DateTime(dateTo.Year, 1, 1);
            filter = new TestingsFilter()
            {
                UsePeriod = true,
                DateAt = dateAt,
                DateTo = dateTo,
                UseGroup = false,
                GroupId = db.Groups.First().Id,
                UseTest = false,
                TestId = db.Tests.First(p => p.UserId ==
userId).Id
            };
        }
        List<Testing> items = GetList(filter);
        int pageSize = 10;
        int pageNumber = (id ?? 1);
        IPagedList<Testing> pagedList = items.ToPagedList(1,
pageSize);
        try
        {
            pagedList = items.ToPagedList(pageNumber, pageSize);
        }
        catch { }
        TestingsList list = new TestingsList()
        {
            Filter = filter,
            Items = pagedList
        };
        ViewBag.GroupId = new SelectList(db.Groups, "Id", "Name",
filter.GroupId);
        ViewBag.TestId = new SelectList(db.Tests.Where(p =>
p.UserId == userId), "Id", "Name", filter.TestId);
        return View(list);
    }

    // Обработка формы фильтра
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Index(TestingsFilter filter)
    {
        Session["TestingsFilter"] = filter;
        return RedirectToAction("Index");
    }

    // Форма нового
    public ActionResult Create()
    {
        int userId = Convert.ToInt32(User.Identity.Name);
        if (db.Tests.Where(p => p.UserId == userId).Count() == 0)
        {

```

```

        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    ViewBag.GroupId = new SelectList(db.Groups, "Id",
    "Name");
    ViewBag.TestId = new SelectList(db.Tests.Where(p =>
    p.UserId == userId), "Id", "Name");
    Testing testing = new Testing()
    {
        DateAt = DateTime.Now,
        DateTo = DateTime.Now.AddDays(7),
        GroupId = db.Groups.First().Id,
        TestId = db.Tests.Where(p => p.UserId ==
    userId).First().Id,
        Closed = false
    };
    return View(testing);
}

// Обработка формы нового
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
    "Id,DateAt,DateTo,TestId,GroupId,Closed")] Testing testing)
{
    if (ModelState.IsValid)
    {
        bool flag = true;
        Test test = db.Tests.Find(testing.TestId);
        if (Helpers.TestWorks.TestCheck(test).Count() > 0)
        {
            ModelState.AddModelError("TestId", "Тест не
    прошел проверку!");
            flag = false;
        }
        if (db.Students.Where(p => p.GroupId ==
    testing.GroupId).Count() < 1)
        {
            ModelState.AddModelError("GroupId", "В группе нет
    обучающихся!");
            flag = false;
        }
        DateTime yestarday = DateTime.Now.AddDays(-1);
        yestarday = new DateTime(yestarday.Year,
    yestarday.Month, yestarday.Day, 23, 59, 59);
        if (testing.DateTo <= yestarday)
        {
            ModelState.AddModelError("DateTo", "Дата
    окончания не может быть меньше текущей даты!");
            flag = false;
        }
        if (testing.DateTo < testing.DateAt)
        {

```

```

        ModelState.AddModelError("DateAt", "Дата начала
не может быть больше даты окончания!");
        flag = false;
    }
    if (flag)
    {
        testing.DateAt = new
DateTime(testing.DateAt.Year, testing.DateAt.Month,
testing.DateAt.Day, 0, 0, 0);
        testing.DateTo = new
DateTime(testing.DateTo.Year, testing.DateTo.Month,
testing.DateTo.Day, 23, 59, 59);
        testing.Closed = false;
        db.Testings.Add(testing);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}
ViewBag.GroupId = new SelectList(db.Groups, "Id", "Name",
testing.GroupId);
ViewBag.TestId = new SelectList(db.Tests, "Id", "Name",
testing.TestId);
return View(testing);
}

// Отчет по тестированию
public ActionResult Details(int? id)
{
    int userId = Convert.ToInt32(User.Identity.Name);
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Testing testing = db.Testings.Find(id);
    if (testing == null)
    {
        return HttpNotFound();
    }
    if (testing.Test.UserId != userId)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    ViewBag.CntGroupStudents = db.Students.Where(p =>
p.GroupId == testing.GroupId).Count();
    int cntStartTest = 0;
    int cntTestingPassed = 0;
    List<ReportStudentRow> list = new
List<ReportStudentRow>();
    foreach (Student student in db.Students.Where(p =>
p.GroupId == testing.GroupId))
    {

```

```

        TestingSet testingSet =
db.TestingSets.OrderByDescending(p => p.DateAt).FirstOrDefault(p =>
p.TestingId == testing.Id && p.UserId == student.Id);
        if (testingSet != null)
        {
            cntStartTest++;
            if (testingSet.IsPassed)
            {
                cntTestingPassed++;
            }
            list.Add(new ReportStudentRow() {
                User = student.User,
                TestingSetId = testingSet.Id,
                Score = testingSet.Score,
                IsPassed = testingSet.IsPassed
            });
        }
        else
        {
            list.Add(new ReportStudentRow()
            {
                User = student.User,
                TestingSetId = 0,
                Score = 0,
                IsPassed = false
            });
        }
        ViewBag.Items = list;
        ViewBag.CntStartTest = cntStartTest;
        ViewBag.CntTestingPassed = cntTestingPassed;
        return View(testing);
    }

    // Отчет по тестированию студента
    public ActionResult StudentDetails(int? id)
    {
        if (id == null)
        {
            return new
HttpStatusCodResult(HttpStatusCode.BadRequest);
        }
        TestingSet testingSet = db.TestingSets.Find(id);
        if (testingSet == null)
        {
            return HttpNotFound();
        }
        return View(testingSet);
    }

    // Форма завершения тестирования
    public ActionResult Close(int? id)
    {

```

```

        int userId = Convert.ToInt32(User.Identity.Name);
        if (id == null)
        {
            return new
HttpStatusResult(HttpStatusCode.BadRequest);
        }
        Testing testing = db.Testings.Find(id);
        if (testing == null)
        {
            return HttpNotFound();
        }
        if (testing.Test.UserId != userId)
        {
            return new
HttpStatusResult(HttpStatusCode.BadRequest);
        }
        ViewBag.CntGroupStudents = db.Students.Where(p =>
p.GroupId == testing.GroupId).Count();
        int cntStartTest = 0;
        int cntTestingPassed = 0;
        foreach (Student student in db.Students.Where(p =>
p.GroupId == testing.GroupId))
        {
            if (db.TestingSets.Where(p => p.UserId == student.Id
&& p.TestingId == id).Count() > 0)
            {
                cntStartTest++;
            }
            if (db.TestingSets.Where(p => p.UserId == student.Id
&& p.TestingId == id && p.IsPassed).Count() > 0)
            {
                cntTestingPassed++;
            }
        }
        ViewBag.CntStartTest = cntStartTest;
        ViewBag.CntTestingPassed = cntTestingPassed;
        return View(testing);
    }

    // Обработка формы завершения
    [HttpPost, ActionName("Close")]
    [ValidateAntiForgeryToken]
    public ActionResult CloseConfirmed(int id)
    {
        Testing testing = db.Testings.Find(id);
        testing.Closed = true;
        db.Entry(testing).State = EntityState.Modified;
        db.SaveChanges();
        // Расчет результатов тестирования
        foreach (TestingSet testingSet in db.TestingSets.Where(p
=> p.TestingId == id && p.Completed == false))
        {
            Helpers.TestWorks.TestingSetCalc(db, testingSet);
        }
    }

```

```

        }
        //
        return RedirectToAction("Details", new { id = testing.Id
    });
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

ПРИЛОЖЕНИЕ Б (обязательное)

Фрагменты cshtml-разметки

Views\Shared_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link href="/favicon.png" rel="shortcut icon" type="image/png" />
    <title>@ViewBag.Title - Тестирование</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Тестирование", "Index", "Home", new
{ area = "" }, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                @if (User.Identity.IsAuthenticated)
                {
                    <ul class="nav navbar-nav">
                        @if (User.IsInRole("student"))
                        {
                            <li>@Html.ActionLink("Доступные тесты",
"Index", "Jobs")</li>
                        }
                        @if (User.IsInRole("admin"))
                        {
                            <li class="dropdown">
                                <a href="#" class="dropdown-toggle"
data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">Справочники <span class="caret"></span></a>
                                <ul class="dropdown-menu">
                                    <li>@Html.ActionLink("Предметы",
"Index", "Subjects")</li>
```

```

<li>@Html.ActionLink("Группы",
"Index", "Groups")</li>

<li>@Html.ActionLink("Пользователи", "Index", "Users")</li>
</ul>
</li>
}
@if (User.IsInRole("admin") ||
User.IsInRole("teacher"))
{
<li class="dropdown">
<a href="#" class="dropdown-toggle"
data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">Тестирование <span class="caret"></span></a>
<ul class="dropdown-menu">

<li>@Html.ActionLink("Тестирования", "Index", "Testings")</li>

<li>@Html.ActionLink("Обучающиеся", "Index", "Students")</li>
<li class="divider"
role="separator"></li>
<li>@Html.ActionLink("Тесты",
"Index", "Tests")</li>
</ul>
</li>

}
</ul>
<ul class="nav navbar-nav navbar-right">
<li>@Html.ActionLink("Изменить пароль",
"Password", "Home")</li>
<li>@Html.ActionLink("Выход", "Logout",
"Home")</li>
</ul>
}
</div>
</div>
<main>
<div class="container body-content">
@RenderBody()
</div>
</main>
<footer>
<div class="container text-center">
<p>&copy; @DateTime.Now.Year - Тестирование</p>
</div>
</footer>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>

```


</html>

Views\Home\Logins.cs

```
@model LearningTest.Models.LoginData
```

```
@{
    ViewBag.Title = "Авторизация";
}
```

```
<h1>@ViewBag.Title</h1>
```

```
@using (Html.BeginForm())
{
```

```
    @Html.AntiForgeryToken()
```

```
    <div class="form-horizontal">
```

```
        <h4>Введите логин и пароль</h4>
```

```
        <hr />
```

```
        @Html.ValidationSummary(true, "", new { @class = "text-
danger" })
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.Login, htmlAttributes: new
{ @class = "control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.Login, new {
htmlAttributes = new { @class = "form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.Login, "",
new { @class = "text-danger" })
```

```
            </div>
```

```
        </div>
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.Password, htmlAttributes:
new { @class = "control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.Password, new {
htmlAttributes = new { @class = "form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.Password,
"", new { @class = "text-danger" })
```

```
            </div>
```

```
        </div>
```

```
        <div class="form-group">
```

```
            <div class="col-md-offset-2 col-md-10">
```

```
                <input type="submit" value="Вход" class="btn btn-
default" />
```

```
            </div>
```

```
        </div>
```

```
    </div>
```

```
}
```

```
@section Scripts {
```

```
@Scripts.Render("~/bundles/jqueryval")
}
```

Views\Testings\Details.cshtml

```
@model LearningTest.Models.Testing

@{
    ViewBag.Title = "Отчет по тестированию";
}

<h1>@ViewBag.Title</h1>

<div>
    <h4>Данные тестирования</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Test)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Test.Name)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Test.MinScore)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Test.MinScore)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Group)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Group.Name)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.DateAt)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.DateAt)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.DateTo)
        </dt>
    </dl>
</div>
```

```

        <dd>
            @Html.DisplayFor(model => model.DateTo)
        </dd>
    </dl>
</div>

<h4>Информация по прохождению тестирования</h4>

<table class="table">
    <tr>
        <th>
            Обучающийся
        </th>
        <th>
            Участвовал в<br />
            тестировании
        </th>
        <th>
            Балл
        </th>
        <th>
            Тестирование<br />
            пройдено
        </th>
    </tr>

    @foreach (LearningTest.Models.ReportStudentRow item in
ViewBag.Items)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.User.Name)
            </td>
            <td>
                @if (item.TestingSetId > 0)
                {
                    <span>да</span>
                }
                else
                {
                    <span>нет</span>
                }
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Score)
            </td>
            <td>
                @if (item.IsPassed)
                {
                    <span>да</span>
                }
                else

```

```

        {
            <span>нет</span>
        }
    </td>
    <td class="text-right">
        @if (item.TestingSetId > 0)
        {
            <a
href="~/Testings/StudentDetails/@item.TestingSetId" class="btn btn-
default btn-sm" title="Отчет по тестированию обучающегося"><i
class="glyphicon glyphicon-list-alt"></i></a>
        }
    </td>
</tr>
}
</table>

<div>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            Всего обучающихся в группе
        </dt>

        <dd>
            @ViewBag.CntGroupStudents
        </dd>

        <dt>
            Приступили к тестированию
        </dt>

        <dd>
            @ViewBag.CntStartTest
        </dd>

        <dt>
            Прошли тестирование
        </dt>

        <dd>
            @ViewBag.CntTestingPassed
        </dd>
    </dl>
</div>

<p>
    <a href="~/Testings" class="btn btn-default" role="button">К
списку</a>
</p>

```

[illegible]