МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

*КОЛЛЕДЖ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ*

**ДИПЛОМНАЯ РАБОТА (ПРОЕКТ)**

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ МНОГОВАРИАНТНОЙ КОНТРОЛЬНОЙ РАБОТЫ**

выпускник 4 курса группы

специальности 09.02.07 Информационные системы и программирование

ФИО Жилина А.А./\_\_\_\_\_\_\_\_\_\_\_\_

выпускник подпись

ФИО Корлякова Е.В./ \_\_\_\_\_\_\_\_\_\_

руководитель подпись

Пермь 2023

# СОДЕРЖАНИЕ

[Обозначения и сокращения 4](#_Toc137596002)

[Введение 5](#_Toc137596003)

[Раздел 1 Анализ и разработка требований к проекту 7](#_Toc137596004)

[1.1 Анализ и характеристика предприятий 7](#_Toc137596005)

[1.1.1 Характеристика предприятия ООО «СЕУСЛАБ» 7](#_Toc137596006)

[1.1.2 Характеристика предприятия Колледж профессионального образования при ПГНИУ 8](#_Toc137596007)

[1.2 Анализ предметной области 8](#_Toc137596008)

[1.3 Требования к интерфейсу 11](#_Toc137596009)

[1.4 Требования к функционированию информационной системы 12](#_Toc137596010)

[1.5 Анализ существующих информационных систем 13](#_Toc137596011)

[1.6 Описание бизнес процесса «AS-IS» 15](#_Toc137596012)

[Раздел 2 Практическая разработка приложения 22](#_Toc137596013)

[2.1 Разработка функциональной модели «TO-BE» 22](#_Toc137596014)

[2.2. Проектирование БД 27](#_Toc137596015)

[2.2.1 Этапы проектирования БД 27](#_Toc137596016)

[2.2.2 Реляционная БД 28](#_Toc137596017)

[2.2.3 Обоснование выбора СУБД 29](#_Toc137596018)

[2.2.4 Разработка инфологической модели БД 32](#_Toc137596019)

[2.2.5 Разработка даталогической модели БД 34](#_Toc137596020)

[2.2.6 Разработка физической модели БД 39](#_Toc137596021)

[2.2.7 Обеспечение надежности и безопасности 41](#_Toc137596022)

[2.3 Разработка приложения 41](#_Toc137596023)

[2.3.1 Выбор языка программирования 41](#_Toc137596024)

[2.3.2 Выбор платформы разработки 42](#_Toc137596025)

[2.3.3 Построение UML-диаграмм 42](#_Toc137596026)

[2.3.4 Интерфейс приложения 56](#_Toc137596027)

[2.3.5 Тестирование ПП 61](#_Toc137596028)

[2.3.6 Внедрение ПП 62](#_Toc137596029)

[2.4 Экономическая эффективность проекта 62](#_Toc137596030)

[Заключение 68](#_Toc137596031)

[Список использованных источников 69](#_Toc137596032)

[Приложение А 74](#_Toc137596033)

[Приложение Б 80](#_Toc137596034)

[Приложение В 87](#_Toc137596035)

[Приложение Г 96](#_Toc137596036)

# **Обозначения и сокращения**

|  |  |
| --- | --- |
| АИС | Автоматизированная информационная система |
| БД | База данных |
| ИС | Информационная система |
| ООО | Общество с ограниченной ответственностью |
| ПО | Программное обеспечение |
| ПП | Программный продукт |
| СА | СА |
| СУБД | Система управления базами данных |
| ТЗ | Техническое задание |

# ВВЕДЕНИЕ

В век компьютерных технологий и массового перехода на дистанционный формат обучения тесты в электронном варианте приобретают небывалую популярность и актуальность. Проверка и оценка успеваемости студентов играет важную роль в учебном процессе. Для предотвращения проблемы списывания среди студентов, было решено создать многовариантные задания, которые помогают лучше мотивировать студента на обучение. Вопрос с возможностью выбора альтернативных ответов – наиболее распространенный тип вопросов, позволяющий оценить степень усвоения материала. Проверка таких тестов довольно проста, но является времязатратной задачей. В отличии от ручного, а значит долгого составления однотипных вариантов, автоматизация генерации тестов призвана сократить время на их составление, а также на проверку вводимых ответов.

Также, тема является особенно актуальной в настоящее время в связи с тем, что сейчас в мире сложилась ситуация, когда многие зарубежные компании приостанавливают свою деятельность на территории России, в следствии чего появилась необходимость в разработке собственного, Российского ПО.

Целью выпускной квалификационной работы является создание для колледжа профессионального образования системы генерации многовариантных работ и с последующим тестированием студентов.

Для выполнения данной цели требуется выполнить следующие задачи:

1. Провести анализ существующих на данный момент ИС в данной сфере;
2. Провести анализ деятельности колледжа, в котором проводится тестирование, и построить функциональную модель «AS-IS»;
3. Выполнить проектировку функциональной модели ИС и разработать диаграмму «TO-BE»;
4. Выполнить проектировку информационной модели ИС и разработать UML-диаграммы;
5. Выполнить проектировку интерфейса;
6. Разработать базу данных для ИС;
7. Разработать программный модуль для генерации тестов;
8. Провести тестирование ПП;
9. Рассчитать экономическую эффективность.

Объектом исследования является процесс обучения студентов колледжа профессионального образования при ПГНИУ.

Предметом исследования является система составления многовариантных тестов.

В первом разделе содержится информация о характеристике деятельности предприятия, анализе рынка систем генерации вариантов для тестирования и анализ бизнеспроцессов AS-IS.

Во втором разделе содержится информация об анализе бизнес-процессов TO-BE, UML-диаграммы, выборе средств разработки, описание приложения и базы данных, рассчет экономической эффективности проекта.

В результате выпускной квалификационной работы для колледжа профессионального образования будет разработана эффективная система генерации вариантов для контрольных работ, которая будет предоставлять предприятию возможность автоматической генерации вариантов контрольных работ и ведения журнала успеваемости по каждому студенту.

# Раздел 1 Анализ и разработка требований к проекту

## 1.1 Анализ и характеристика предприятий

Колледжем профессионального образования при ПГНИУ компании ООО «СЕУСЛАБ» было поручено разработать программный продукт для генерации многовариантной контрольной работы.

### 1.1.1 Характеристика предприятия ООО «СЕУСЛАБ»

Компания ООО «СЕУСЛАБ» основана в 2012 г. Первоначально компания была зарегистрирована под названием «Интеллектуальные Интернет-Системы», а свое новое название, «СЕУСЛАБ», получила в 2014 г. Сейчас предприятие ведет успешную деятельность, разрабатывая программно-аппаратные решения, которые, как позиционирует сама компания, необходимы для защиты российской государственности и культуры.

«СЕУСЛАБ» создает специальные программные решения, применяемые в государственном секторе. Программное обеспечение компании используется в подразделениях силовых структур, органах государственной власти и на оборонно-промышленных предприятиях России.

Компания ищет простые решения сложнейших задач, не теряя фундаментальность и основательность подхода. Предприятие использует актуальные методы IT-рынка, разделяет тенденцию к объединению технологических мощностей и всегда готово к сотрудничеству.

Миссией компании является создание и внедрение программных продуктов, технологий и решений в информационной сфере для проведения оперативно-розыскных мероприятий, аналитических исследований, цифровых расследований и информационного противоборства в социальных сетях.

Направления деятельности:

* разработка и поставка IT-решений для информационного противоборства;
* проведение аналитических исследований информационного пространства;
* обучение сотрудников правоохранительных органов современным методам и средствам информационного противоборства.

### 1.1.2 Характеристика предприятия Колледж профессионального образования при ПГНИУ

Полное название предприятия: Федеральное государственное автономное образовательное учреждение высшего образования «Пермский государственный национальный исследовательский университет».

Основан в 1916 г.

И.о. ректора: [Игорь Анатольевич Германов](http://www.psu.ru/personalnye-stranitsy-prepodavatelej/g/igor-anatolevich-germanov), кандидат социологических наук, доцент.

Президент: Владимир Владимирович Маланин, доктор технических наук, профессор, заслуженный деятель науки РФ.

В состав вуза входят 12 факультетов и 76 кафедр, колледж профессионального образования и филиал – Соликамский государственный педагогический институт. Для проведения НИОКР, учебных и производственных практик студентов используются Естественнонаучный институт, особое конструкторское бюро «Маяк», Ботанический сад ПГНИУ, учебно-научная база «Предуралье», а также учебно-опытное лесное хозяйство «Троицкое», расположенное в Челябинской области.

На территории кампуса расположены 16 корпусов. В университет входят 21 научная лаборатория, 4 ведущие лаборатории и 156 учебных лабораторий, 8 научно-образовательных центров.

## 1.2 Анализ предметной области

Требуется разработать ИС для автоматизации составления вариантов теста и процесса тестирования студентов.

С ИС должны работать следующие группы пользователей:

* СА;
* Преподаватели;
* Студенты;
* Администрация.

Вследствие разграничения ролей пользователей, ИС должна содержать данные о каждом из них:

* Уникальный код пользователя;
* Право пользователя (студент, преподаватель, СА или администратор);
* Логин;
* Пароль;
* Кодовое слово.

Система также должна предусматривать режим хранения дисциплин, по которым составляются тесты, и преподавателей, которые составляют тесты. О каждом преподавателе заносятся следующие сведения:

* Уникальный код преподавателя;
* Фамилия;
* Имя;
* Отчество;
* Код пользователя, совпадающий с уникальным кодом из описания пользователей;

Дисциплины имеют следующие характеристики:

* Уникальный код дисциплины;
* Наименование дисциплины;
* Код преподавателя, совпадающий с уникальным кодом из описания преподавателей;

Также система должна иметь возможность хранить темы тестов, относящихся к ним вопросов, а также в соответствии с каждым из вопросов должны храниться к ним ответы с отметкой верного(ых).

Так, каждая тема характеризуется следующими параметрами:

* Уникальный код темы;
* Код дисциплины, совпадающий с уникальным кодом из описания дисциплин;
* Наименования темы;
* Количества вопросов в тесте;
* Код преподавателя, совпадающий с уникальным кодом из описания преподавателей;
* Количества баллов, которое дается за тест.

Вопросы имеют такие данные, как:

* Уникальный код вопроса;
* Код темы, совпадающий с уникальным кодом из описания тем;
* Текст вопроса;
* Тип вопроса (один из многих, многие из многих или развернутый);
* Количество баллов, которое дается за данный тип вопроса.

Ответы имеют такие данные, как:

* Уникальный код ответа;
* Код вопроса, совпадающий с уникальным кодом из описания вопросов;
* Текст ответа;
* Отметка, правильный ответ или нет.

Также в системе должны храниться данные о студентах и их группе. Группы должны содержать данные о:

* Уникальном коде группы;
* Наименовании группы;
* Курсе (1-4).

О каждом студенте заносятся следующие сведения:

* Уникальный код студента;
* Фамилия;
* Имя;
* Отчество;
* Код группы, совпадающий с уникальным кодом из описания групп;
* Код пользователя, совпадающий с уникальным кодом из описания пользователей.

Для открытия доступа к тестированию студентов и занесения количества попыток, необходимо хранить такие данные, как:

* Уникальный код;
* Код темы, совпадающий с уникальным кодом из описания тем;
* Код дисциплины, совпадающий с уникальным кодом из описания дисциплин;
* Код преподавателя, совпадающий с уникальным кодом из описания пользователей;
* Код студента, совпадающий с уникальным кодом из описания студентов;
* Количество попыток.

## 1.3 Требования к интерфейсу

Интерфейс – важная составляющая любой программы. Через интерфейс происходит взаимодействие пользователя и ПО, поэтому пользовательский интерфейс – это единственный способ воспринимать пользователем информационную систему. ИС должна обладать понятным, последовательным, привлекательным и приятным для глаз и нервной системы интерфейсом. Можно выделить несколько основных требований к интерфейсу:

* Функциональность;
* Понятность, последовательность, логичность;
* Защита от человеческого фактора;
* Быстрое обучение пользователя;
* Обеспечение высокой скорости работы;
* Расположение, цвет и яркость элементов интерфейса должны соответствовать зрительным и психофизическим возможностям человека. Надписи должны быть хорошо обозримы.

## 1.4 Требования к функционированию информационной системы

Требования к функционалу ИС – это описание ожидаемого поведения системы. Указывается перечень задач и автоматизируемых функций системы. К основным функциональным требованиям системы можно отнести:

* Авторизация в системе;
* Создание тестовых заданий;
* Проведение тестирования;
* Выведение результатов тестирования;
* Возможность просмотра результатов и времени проведения тестирования.

Прикладное приложение должно представлять собой программу, в которой есть следующие роли: студент, преподаватель, администрация и системный администратор. У каждой из этих ролей есть свой функционал:

* студент имеет возможность зарегестрироваться, проходить тестирование, а также имеет доступ к просмотру оценок проведенных тестирований;
* преподаватель может составлять и впоследствии редактировать, и удалять тесты, а также добавлять студентов в определенную группу;
* администрации дан доступ к просмотру результатов теста и времени его проведения;
* системный администратор выполняет задачи по регистрированию преподавателей в системе, созданию, редактированию и удалению новых групп студентов, а также имеет возможность сбросить пароли в случае проблем с их восстановлением.

Система должна поддерживать единую технологию обработки данных и представления информации.

Для информационного обмена между компонентами системы должна быть организована локальная сеть c клиент-серверной архитектурой. Это необходимо для параллельной обработки информации, возможности совместного, одновременного использования данных и устройств пользователями.

## 1.5 Анализ существующих информационных систем

Хранение информации в бумажном виде уступает хранению в электронном, цифровом, виде. Составление многовариантных тестов вручную нерационально и ресурсоемко. Проверка тестов также требует больших затрат времени, чего можно избежать путем автоматизации процесса проверки введенных студентом ответов. Базы данных позволяют хранить и структурировать большие объемы данных, а также извлекать необходимую информацию.

Все ИС по тестированию похожи между собой, так как выполняют идентичные задачи. Однако не так много удобных и понятных систем, позволяющих сгенерировать многовариантные контрольные работы. Поэтому данная система должна помочь в предотвращении тенденции списываний студентами. Кроме того, для контроля проведения преподавателями тестирований данная система будет предоставлять доступ администрации ко времени их прохождения. Внутренняя система всегда удобна тем, что имеет возможность разработки и модернизации функционала под конкретные требования предприятия.

При рассмотрении различных аналогичных систем, которые частично или полностью соответствуют требованиям разрабатываемой ИС, можно отнести: moodle, google формы, indigo. Сравнительная характеристика таких систем представлена в табл. 1.

Таблица 1 - сравнительная характеристика аналогичных ИС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Продукты  Функции | Moodle | Google формы | Indigo | Разрабатываемая  ИС |
| Авторизация, разграничение прав доступа | + | + | + | + |
| Генерация многовариантных тестов | + | - | - | + |
| Выведение отчета по результатам тестирования | + | + | + | + |
| Понятный интерфейс, легкость обучения | - | + | - | + |
| Возможность просмотра проведения даты и времени тестирований администрацией | + | - | - | + |
| Ведение учета успеваемости студентов | + | - | + | + |
| Стоимость | от 110$ в год | Бесплатно | От 66 тыс. за 100 пользователей одновременно | Бесплатно |

Рассмотренные системы имеют как преимущества, так и недостатки. Несмотря на большинство преимуществ Moodle и Indigo, главный недостаток – переизбыток возможностей и сложный для изучения интерфейс. Также во избежание возможных санкций, из-за приостановления работы большинства иностранных компаний на территории РФ, лучшим решением станет изготовление отечественного ПО. Поэтому создание собственного ПП является наиболее оптимальным решением для выполнения поставленных задач и удовлетворения всех пожеланий заказчика.

Таким образом, собственная разработка позволяет создать систему с нужным функционалом и удобным интерфейсом под конкретное предприятие, который при тестировании всегда можно изменить и улучшить, т.к. ИС создается под конкретные запросы пользователей, которым работать с этой ИС. Помимо этого, сопровождение собственного ПО обойдется предприятию дешевле, чем купленное «коробочное» ПО.

## 1.6 Описание бизнес процесса «AS-IS»

Для анализа возможных улучшений системы тестирования колледжа за счет внедрения нового ПП выполним моделированите бизнес-процессов. Для моделирования бизнесс-процессов выберем методологию IDEF0.

Методологию IDEF0 можно считать следующим этапом развития очень известного графического языка описания функциональных систем SADT (Structured Analysis and Design Tcqniquc). IDEF0 – это методология функционального моделирования. С помощью графического языка IDEF0, изучаемая система предстает перед аналитиками и разработчиками в виде набора взаимосвязанных функций (функциональных блоков - в терминах IDEF0). Как правило, моделирование средствами IDEF0 является одним из важных этапов разработки и внедрения ИС. С помощью данного метода создадим контекстную диаграмму (IDEF0) генерации тестирования (AS-IS – «как есть») (рисунок 1).

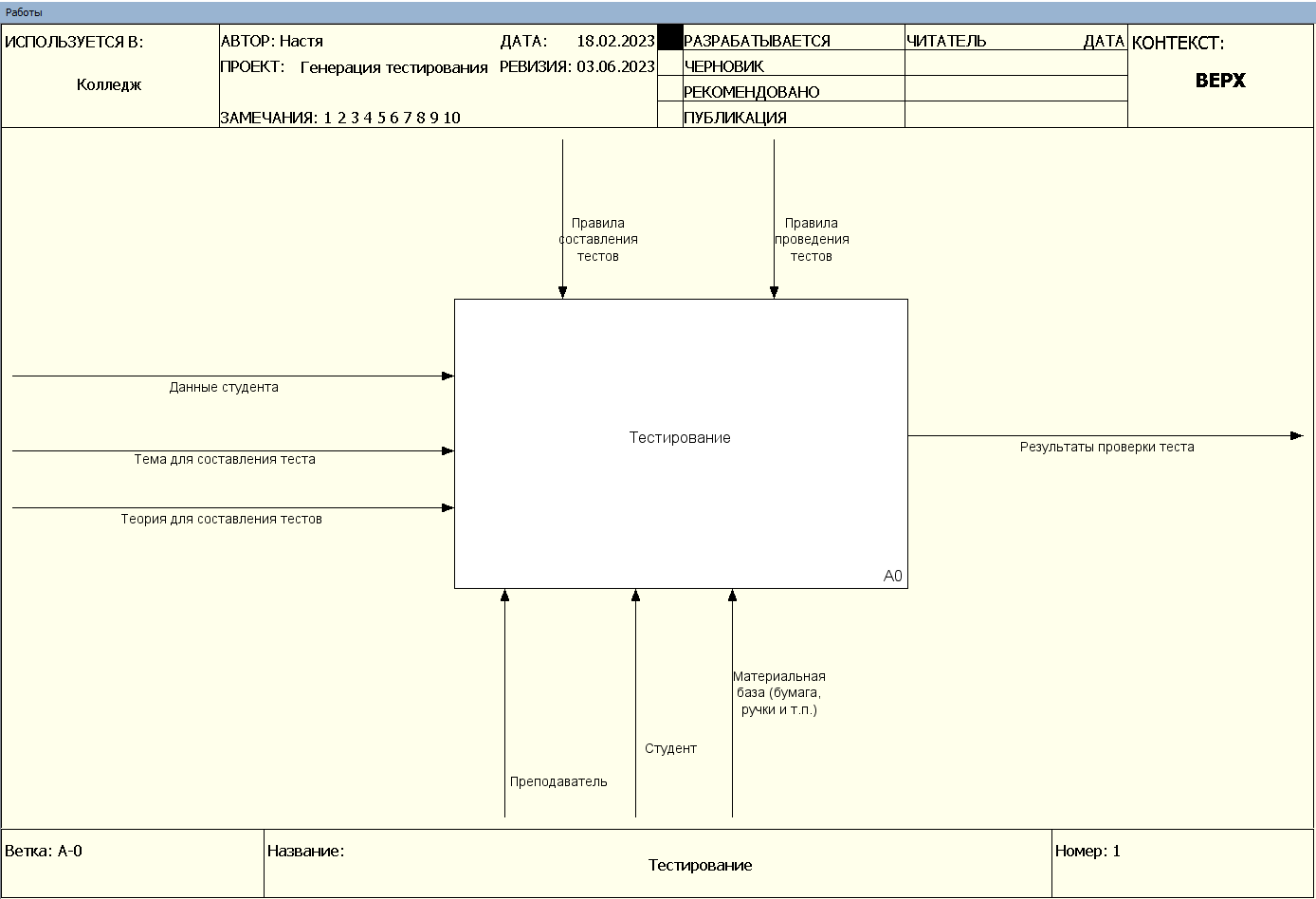


Рисунок 1 – Контекстная диаграмма модели «AS-IS»

Входными данными для следующей модели являются:

* Данные студента;
* Темы для составления тестов.

Опишем процесс тестирования.

При ознакомлении с учебным материалом, преподаватель выделяет темы, по которым будут составляться тесты. После чего для каждой темы теста преподаватель создает пул тестовых вопросов, которые в последующем вручную группируются им в различные варианты.

Следующим этапом проводится само тестирование студента. Раздаются бумажные варианты тестов, куда студент записывает свои ответы. По завершению тестирования, студент сдает свой вариант, который в дальнейшем вручную проверяется преподавателем. По истечению времени, отведенного преподавателю на проверку, студенту оглашаются результаты тестирования.

Детальное описание данной диаграммы представлено в таблице 2.

Таблица 2 - Описание контекстной диаграммы «AS-IS» (как есть)

|  |  |  |
| --- | --- | --- |
| Тип | Название | Описание |
| Вход | Данные студента | Фамилия, имя, отчество, номер курса, название группы |
| Вход | Тема для составления теста | Одна из тем, предоставляемых учебным материалом |
| Вход | Теория для составления тестов | Учебный материал, сторонние учебные ресурсы |
| Управление | Правила составления тестов | Положение о БРС оценки образовательных результатов студентов ФГАОУ ВО «ПГНИУ»  (в ред. приказов от 25.03.2015, 11.08.2016, 26.02.2021) |
| Управление | Правила проведения тестов | Положение о форме, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по программам СПО в ФГАОУ ВО «ПГНИУ»  (с изменениями, внесенными Ученым советом ПГНИУ 27.06.2018. 24.02.2021,26.05.2021, 29.06.2022) |
| Механизм | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Студент | Субъект, проходящий тестирование |
| Механизм | Материальная база | Бумага, письменные принадлежности |
| Выход | Результаты проверки теста | Количество баллов, оценка студента |

Декомпозиция контекстной диаграммы – это процесс представления деятельности более детально. Каждая из функций, представленная на диаграмме отдельным блоком, может в свою очередь быть декомпозирована на уровень ниже в иерархии. В декомпозиции контекстной диаграммы декомпозируемый модуль всегда делится не более, чем на семь модулей. Каждый из этих модулей может быть так же декомпозирован для более детального представления. Декомпозиция контекстной диаграммы модели «AS-IS» представлена на рисунке 2.

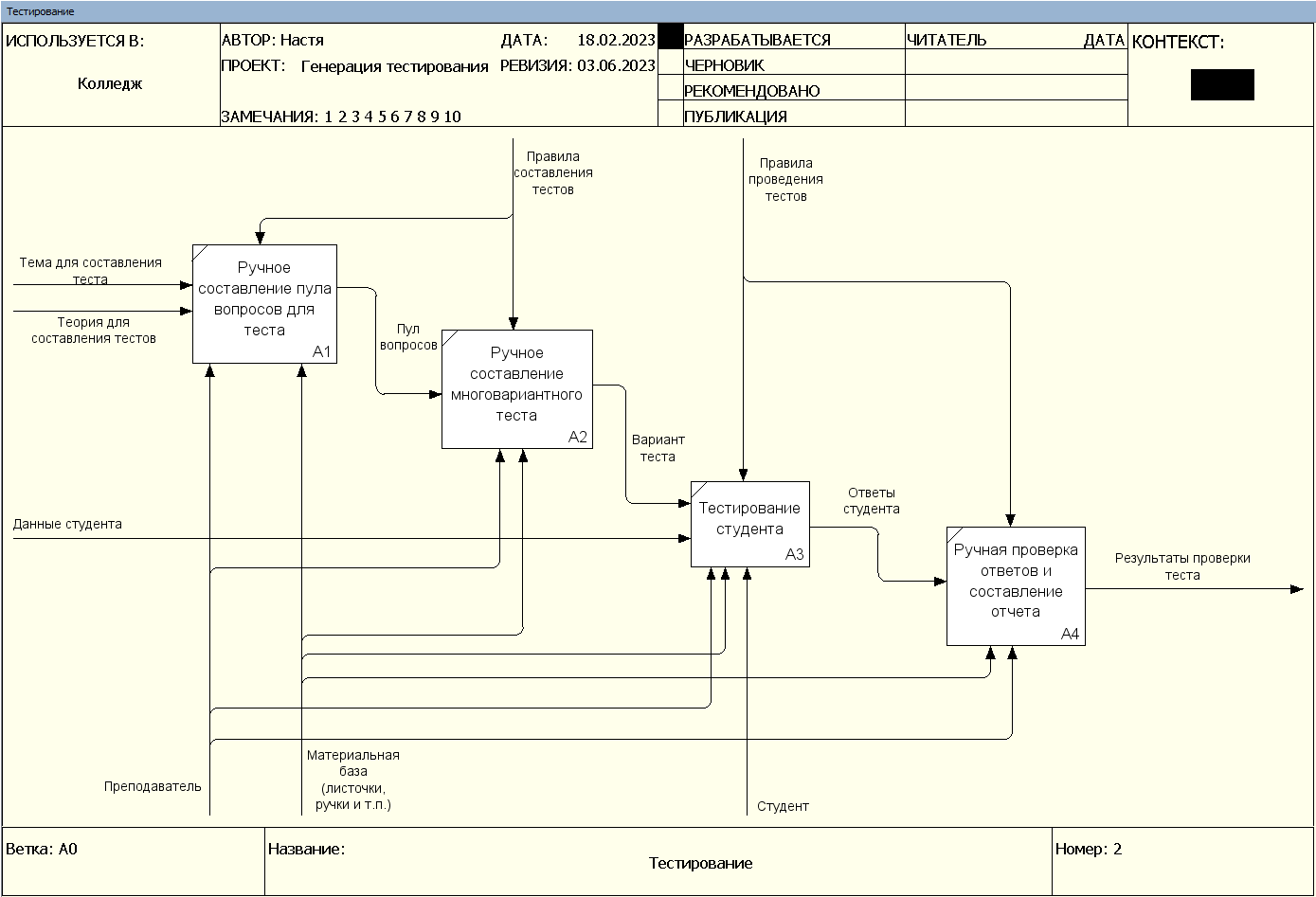


Рисунок 2 – Декомпозиция контекстной диаграммы «AS-IS»

Далее более детально рассмотрим процессы движения входящих и исходящих документов.

Таблица 3 - Описание декомпозиции первого уровня контекстной диаграммы «AS-IS»

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Процесс | Название | Описание |
| Вход | Ручное составление пула вопросов для теста | Тема для составления теста | Одна из тем, предоставляемых учебным материалом |
| Вход | Ручное составление пула вопросов для теста | Теория для составления тестов | Учебный материал, сторонние учебные ресурсы |
| Механизм | Ручное составление пула вопросов для теста | Преподаватель | Субъект, составляющий вручную тестовые варианты |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| Механизм | Ручное составление пула вопросов для теста | Материальная база | Бумага, письменные принадлежности |
| Выход | Ручное составление пула вопросов для теста | Пул вопросов | Список вопросов, из которых будет составляться каждый из вариантов теста |
| Вход | Ручное составление многовариантного теста | Пул вопросов | Список вопросов, из которых будет составляться каждый из вариантов теста |
| Управление | Ручное составление многовариантного теста | Правила составления тестов | Положение о БРС оценки образовательных результатов студентов ФГАОУ ВО «ПГНИУ»  (в ред. приказов от 25.03.2015, 11.08.2016, 26.02.2021) |
| Механизм | Ручное составление многовариантного теста | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Ручное составление многовариантного теста | Материальная база | Бумага, письменные принадлежности |
| Выход | Ручное составление многовариантного теста | Вариант теста | Готовый бумажный вариант теста |
| Вход | Тестирование студента | Вариант теста | Готовый бумажный вариант теста |
| Вход | Тестирование студента | Данные студента | Фамилия, имя, отчество, номер курса, название группы |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| Управление | Тестирование студента | Правила проведения тестов | Положение о форме, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по программам СПО в ФГАОУ ВО «ПГНИУ»  (с изменениями, внесенными Ученым советом ПГНИУ 27.06.2018. 24.02.2021,26.05.2021, 29.06.2022) |
| Механихм | Тестирование студента | Студент | Субъект, проходящий тестирование |
| Механихм | Тестирование студента | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Тестирование студента | Материальная база | Бумага, письменные принадлежности |
| Выход | Тестирование студента | Ответы студента | Готовый бланк с ответами и данными студента |
| Вход | Ручная проверка ответов и составление отчета | Ответы студента | Готовый бланк с ответами и данными студента |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| Управление | Ручная проверка ответов и составление отчета | Правила проведения тестов | Положение о форме, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по программам СПО в ФГАОУ ВО «ПГНИУ»  (с изменениями, внесенными Ученым советом ПГНИУ 27.06.2018. 24.02.2021,26.05.2021, 29.06.2022) |
| Механихм | Ручная проверка ответов и составление отчета | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Ручная проверка ответов и составление отчета | Материальная база | Бумага, письменные принадлежности |
| Выход | Ручная проверка ответов и составление отчета | Результаты проверки теста | Количество баллов, оценка студента |

# Раздел 2 Практическая разработка приложения

Техническое задание на разработку ПП представлено в ПРИЛОЖЕНИИ А.

## 2.1 Разработка функциональной модели «TO-BE»

Для разработки функциональной модели ИС применим технологию IDEF0. Данная модель отражает функциональную структуру объекта, то есть производимые им действия и связи между ними.

Контекстная диаграмма проектируемой для организации модели тестирования представлена на рисунке 3.

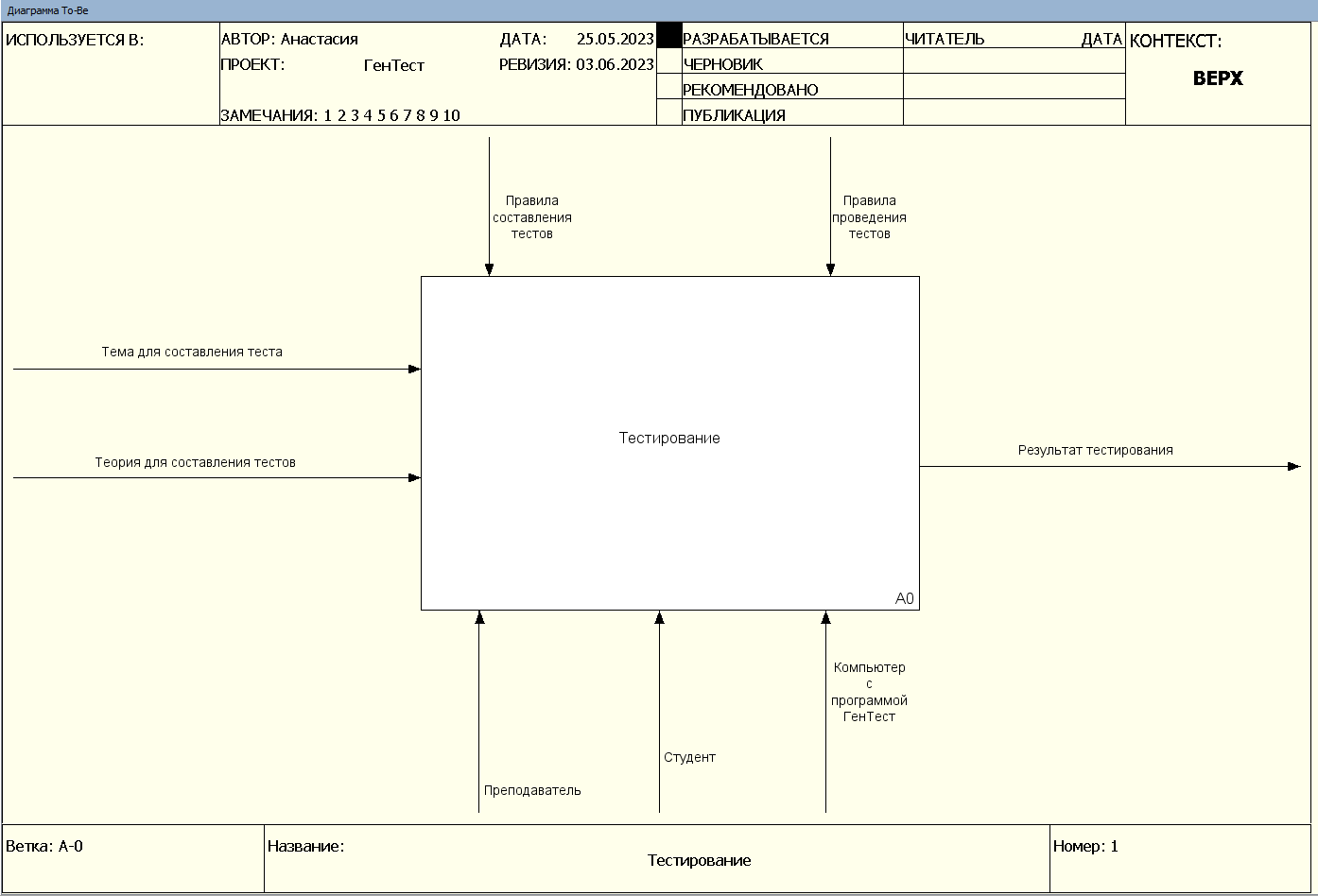


Рисунок 3 - Контекстная диаграмма модели «TO-BE»

Входными данными для модели являются темы для составления теста.

Опишем универсальный процесс тестирования с помощью программы ГенТест.

После ознакомления с учебным материалом, преподаватель так же, как и в ручном варианте, выделяет темы, по которым будут составляться тесты. Далее он заносит в программу все возможные варианты вопросов и в соответствии с ними ответы, отмечая при этом верные. Когда пул минимального размера вопросов будет создан, преподаватель в праве сохранить тест и открыть видимость для прохождения теста, а также установить количество попыток группе студентов или какому-либо из студентов выборочно. Вопросы в тесте генерируются автоматически, при этом не попадаются повторно. Вернуться к предыдущему вопросу нельзя, что так же уменьшает попытки списывания.

После завершения прохождения студентом теста, ему моментально выплывет уведомление о завершении и количестве набранных им баллов и в соответствии с этим оценки.

Детальное описание данной диаграммы представлено в таблице 4.

Таблица 4 - Описание контекстной диаграммы «TO-BE» (как должно быть)

|  |  |  |
| --- | --- | --- |
| Тип | Название | Описание |
| Вход | Тема для составления теста | Одна из тем, предоставляемых учебным материалом |
| Вход | Теория для составления тестов | Учебный материал, сторонние учебные ресурсы |
| Управление | Правила составления тестов | Положение о БРС оценки образовательных результатов студентов ФГАОУ ВО «ПГНИУ»  (в ред. приказов от 25.03.2015, 11.08.2016, 26.02.2021) |
| Управление | Правила проведения тестов | Положение о форме, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по программам СПО в ФГАОУ ВО «ПГНИУ»  (с изменениями, внесенными Ученым советом ПГНИУ 27.06.2018. 24.02.2021,26.05.2021, 29.06.2022) |
| Механизм | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Студент | Субъект, проходящий тестирование |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| Механизм | Компьютер с программой ГенТест | Компьютер, имеющий установленную программу ГенТест, а также подключенный к базе, на которой размещена база данных, необходимая для работы приложения ГенТест |
| Выход | Результаты проверки теста | Количество баллов, оценка студента |

Далее более детально рассмотрим процесс тестирования с использованием программы «ГенТест».

Декомпозиция контекстной диаграммы модели «TO-BE» представлена на рисунке 4.

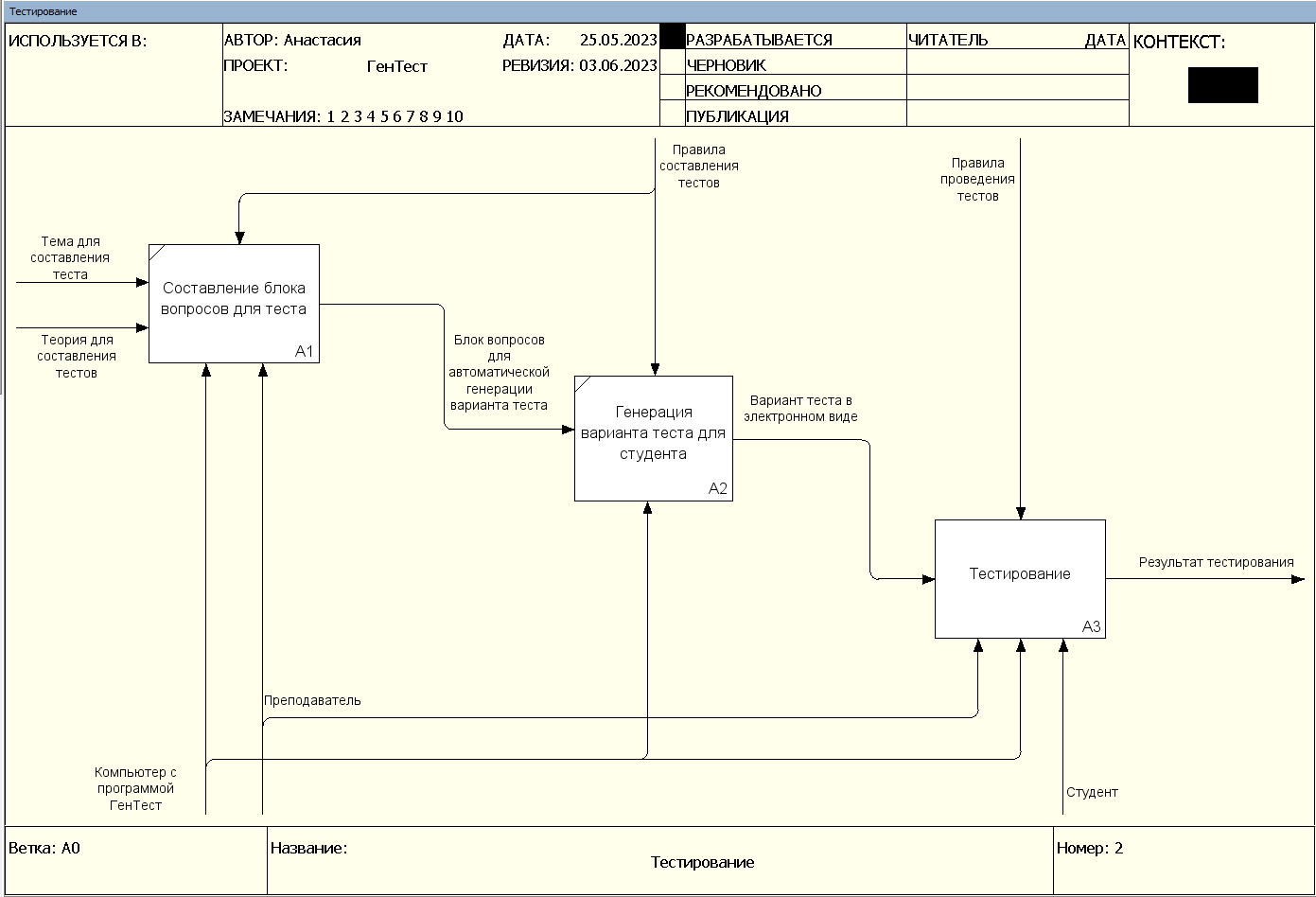


Рисунок 4 - Декомпозиция контекстной диаграммы модели «TO-BE»

Описание представленной декомпозиции представлено в таблице 5.

Таблица 5 - Описание декомпозиции контекстной диаграммы «TO-BE» (как должно быть)

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Процесс | Название | Описание |
| Вход | Составление блока вопросов дл теста | Тема для составления теста | Одна из тем, предоставляемых учебным материалом |
| Вход | Составление блока вопросов дл теста | Теория для составления тестов | Учебный материал, сторонние учебные ресурсы |
| Механизм | Составление блока вопросов дл теста | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Составление блока вопросов дл теста | Компьютер с программой ГенТест | Компьютер, имеющий установленную программу ГенТест, а также подключенный к базе, на которой размещена база данных, необходимая для работы приложения ГенТест |
| Выход | Составление блока вопросов дл теста | Блок вопросов для автоматической генерации варианта теста | Список вопросов, из которых будет составляться каждый из вариантов теста |
| Вход | Автоматическая генерация теста для студента | Блок вопросов для автоматической генерации варианта теста | Список вопросов, из которых будет составляться каждый из вариантов теста |

Продолжение таблицы 5

|  |  |  |  |
| --- | --- | --- | --- |
| Управление | Автоматическая генерация теста для студента | Правила составления тестов | Положение о БРС оценки образовательных результатов студентов ФГАОУ ВО «ПГНИУ»  (в ред. приказов от 25.03.2015, 11.08.2016, 26.02.2021) |
| Механизм | Автоматическая генерация теста для студента | Компьютер с программой ГенТест | Компьютер, имеющий установленную программу ГенТест, а также подключенный к базе, на которой размещена база данных, в которой находится блок вопросов для данного теста |
| Выход | Автоматическая генерация теста для студента | Вариант теста в электронном виде | Готовый электронный вариант теста |
| Вход | Тестирование студента | Вариант теста в электронном виде | Готовый электронный вариант теста |
| Управление | Тестирование студента | Правила проведения тестов | Положение о форме, периодичности и порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по программам СПО в ФГАОУ ВО «ПГНИУ»  (с изменениями, внесенными Ученым советом ПГНИУ 27.06.2018. 24.02.2021,26.05.2021, 29.06.2022) |

Продолжение таблицы 5

|  |  |  |  |
| --- | --- | --- | --- |
| Механизм | Тестирование студента | Преподаватель | Субъект, составляющий вручную тестовые варианты |
| Механизм | Тестирование студента | Компьютер с программой ГенТест | Компьютер, имеющий установленную программу ГенТест, а также подключенный к базе, на которой размещена база данных, в которой находится блок вопросов для данного теста |
| Механизм | Тестирование студента | Студент | Субъект, проходящий тестирование |
| Выход | Тестирование студента | Результат тестирования | Количество баллов, оценка студента |

## 2.2. Проектирование БД

### 2.2.1 Этапы проектирования БД

Проектирование БД происходит в несколько этапов.

Первый этап предполагает системный анализ предметной области. Была проведена аналитическая деятельность, в результате которой получилась ER — модель. Результаты первого этапа – анализа предметной области – это модули данных на семантическом уровне, то есть исходные данные для решения задач проектирования БД.

На данном этапе анализируются все потребности для будущих пользователей БД. Рассматриваются входные и выходные данные, их формы. Уточняются алгоритмы и процедуры обработки данных. Формируются требования к будущей БД, список объектов предметной области, характеристики которых будут использоваться при разработке БД.

На втором этапе ведется моделирование данных – создание логической структуры данных. Уточняются типы данных, которые хранятся в системе и будут использоваться атрибутами, типы отношений между объектами, способы группировки и организации данных.

На третьем этапе выбирается система, в которой будет осуществляться, храниться БД. Теоретически необходимо учесть множество факторов, однако на практике при выборе СУБД программисты руководствуются интуицией и некоторыми факторами, такими как:

* Характеристики производительности СУБД;
* Большой функционал для дальнейшего развития системы;
* Тип модели данных, который поддерживается СУБД, пригодность потребностям будущей БД;
* Удобство использования СУБД персоналом администрирования БД, наличие понятной документации по работе с СУБД;
* Надежность СУБД в эксплуатации;
* Стоимость СУБД и ПО, необходимого для ее работы.

После выбора СУБД , на четвертом этапе проектируется даталогическая модель БД. Учитываются особенности выбранной СУБД. При формировании схемы, каждая из сущностей, определенных в концептуальной схеме, формируется в таблицу, которая является отношением.

Пятым и заключительным этапом БД реализуется в выбранной раннее СУБД по той информации, которую собрали, обработали и подготовили на предыдущем этапе проектирования БД. Результат данного этапа – готовая, реализованная БД в конкретной СУБД.

### 2.2.2 Реляционная БД

Существует три типа моделей баз данных – иерархическая, сетевая и реляционная.

Первая, иерархическая модель, имеет древовидную структуру, которая поддерживает только отношения типа «один к одному» или «один ко многим». Благодаря данной модели можно быстро получить необходимые данные, однако она не отличается своей гибкостью. Не всегда роль элемента ясна и подходит для иерархической модели.

Вторая модель данных, сетевая, имеет более гибкую структуру, чем иерархическая, а также поддерживает отношения «многие ко многим». Однако в модели может произойти путаница, от чего сетевая модель быстро становится слишком сложной и неудобной для управления.

Третья модель, реляционная, более гибкая, чем иерархическая и проще для управления, чем сетевая. Такая модель сегодня используется чаще всего.

Реляционная база данных – это двумерный массив, который представляет из себя совокупность двумерных таблиц, называемых сущностями, имеющими между собой отношения, называемые связями. Строка такой модели называется записью, а столбец – полем.

Каждая реляционная модель должна обладать такими свойствами, как:

1. Каждый элемент таблицы соответствует одному элементу данных, то есть в одной ячейке таблицы не может указываться более одного значения параметра;
2. Все столбцы в таблице однородные, то есть имеют одинаковый тип и длину;
3. У каждого столбца уникальное имя;
4. Каждая строка уникальна, строки не должны быть одинаковыми, повторяться;
5. Порядок строк – произвольный.

Таким образом, это наиболее удобная, распространенная и привычная модель представления данных для пользователя.

### 2.2.3 Обоснование выбора СУБД

Наша база будет содержать миллионы записей. Поэтому СУБД необходимо подобрать надежную, но в том же время не дорогостоящую и понятную для пользователя.

Рассмотрим возможные варианты СУБД.

1. Oracle Database

Это почти первая система, которая стала успешной и обрела востребованность на рынке. Одно время она занимала половину рынка. СУБД позволяет соединить несколько баз данных в одну большую, что крайне удобно для крупных компаний. Диапазон использования системы огромен.

Главный недостаток данной СУБД – высокая стоимость. Поэтому данная СУБД доступна только клиентам государственных структур, банкам и другим крупным компаниям. В среднем покупка обходится в 300 тысяч рублей. В связи с этим, данная СУБД не подходит под разработку нашей программы.

1. Microsoft SQL-сервер

Ещё одной из популярных СУБД является Microsoft SQL-сервер. Это СУБД, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно.

Достоинства

* СУБД проста в использовании;
* Работает быстро и стабильно.

Недостатки

* Цена оказывается слишком высокой для большей части организаций;
* Даже при тщательной настройке производительности способен задействовать все доступные ресурсы;
* Есть проблемы с интеграцией импорта файлов.

1. PostgreSQL

Является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это весьма старая система, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными.

Недостатки

* Непонятная документация, могут возникать непонимания;
* Конфигурация может смутить неподготовленного пользователя;
* Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

1. MySQL

Это бесплатный пакет программ. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала. Для нашей системы вполне подойдет бесплатная версия СУБД.

Достоинства

* Бесплатный пакет;
* Понятная документация;
* Большой функционал.

Недостатки

* Придётся потратить много времени и усилий, чтобы заставить MySQL выполнять несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии;
* Отсутствует встроенная поддержка XML или OLAP;
* Поддержка только платная.

Наш выбор пал именно на последнюю представленную СУБД – MySql. Данная система идеально подходит для организаций, требующих надежную СУБД с полным основным функционалом, при этом бесплатную. Недостатки минимальны и несущественны для нашей системы.

### 2.2.4 Разработка инфологической модели БД

Разработка инфологической модели необходима для обеспечения естественных для человека способов представления и сбора информации, которая будет храниться в разрабатываемой БД. Строится данная модель в соответствии с понятным для человека языком, но не используется в таком виде в связи со сложностями обработки текстов на естественном языке компьютером.

Инфологическая модель по своей сути – это концептуальный уровень информации, это потоки информации, где выделяются сущности и связи между ними данной предметной области.

Сущность – это различимый конкретный или абстрактный объект предметной области, имеющий набор определенных характеристик. Каждая таблица в базе должна отражать одну сущность.

Связь – это то, с помощью чего соединяются сущности с целью обеспечения целостности данных, чтобы в базе не было потерянных записей, то есть ссылок на не существующие записи. Бывает три типа связи, а именно: один к одному (1:1), когда каждая запись соответствует только одной записи в другой таблице, один ко многим (1:М), когда одна запись таблицы соответствует нескольким записям в другой, и многие ко многим (М:М), то есть множество записей в одной таблице соответствует множеству записей в другой.

БД может содержать безграничное количество сущностей, а также связей между ними.

Атрибут – это одна из характеристик сущности. Может быть как классификацией, так и идентификационной или числовой характеристикой. Внутри одной сущности наименование атрибута должно быть уникальным, однако может совпадать с атрибутом другой сущности. Атрибуты необходимы для описания сущности и определения информации, которая должна храниться в БД.

Ключ – минимальный набор атрибутов, который необходим для однозначной идентификации экземпляра, то есть конкретного объекта сущности.

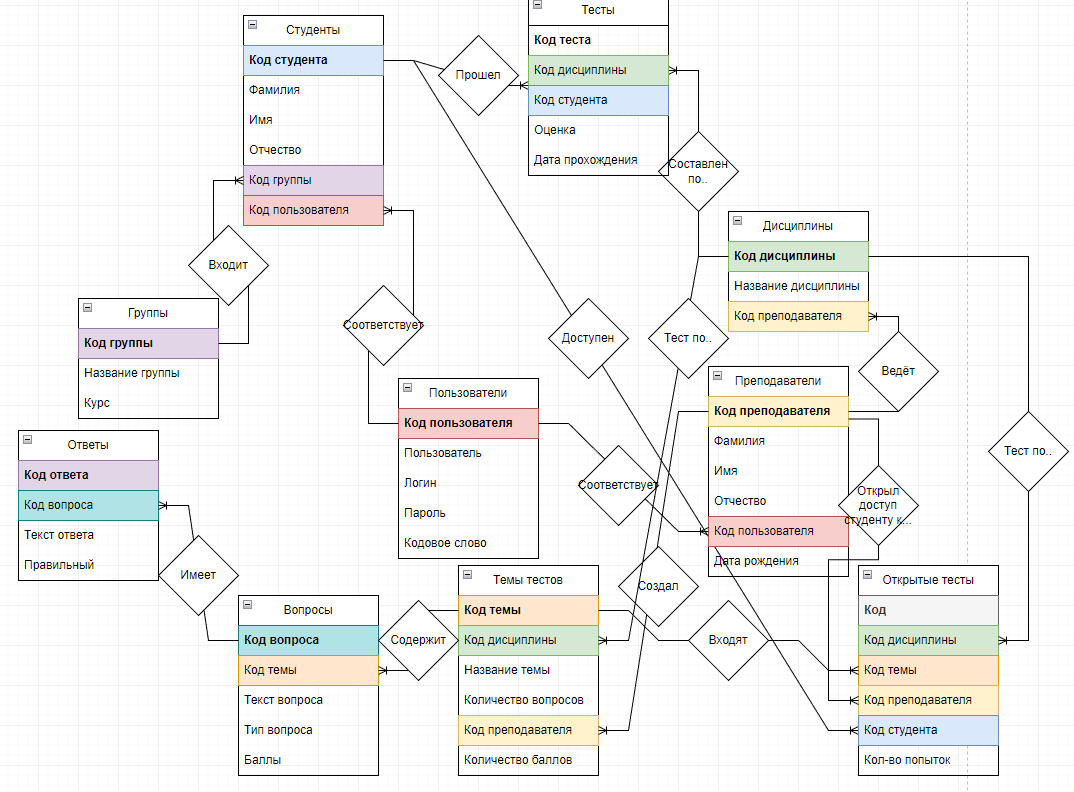
Цель инфологического моделирования – обеспечения оптимального способа сбора и представления хранимой в базе информации.

Выделим следующие сущности:

* Пользователи;
* Студенты;
* Группы;
* Преподаватели;
* Дисциплины;
* Тесты;
* Темы тестов;
* Вопросы;
* Ответы;
* Открытые тесты.

Наглядно инфологическая или ER-модель БД ГенТест представлена на рисунке 5.

Сущность представляется в концептуальной модели прямоугольником, в котором указано ее имя. Атрибуты представляются в прямоугольнике построчно с указанием их имен. Связи представляются ромбами.

Рисунок 5 - ER-модель БД

### 2.2.5 Разработка даталогической модели БД

Даталогическая модель – это логический уровень системы, который представляет собой отображение логических связей между элементами базы данных в среде конкретной СУБД.

Исходные данные – инфологическая модель БД в конкретной предметной области.

Конечный результат даталогического проектирования – описание логической структуры БД на языке описания данных, который поддерживает конкретная СУБД.

Физический уровень определяет то, как определения сущностей из логического уровня будут отражены в реальной базе данных, с помощью какой СУБД она будет реализована, какие типы данных будут использоваться по умолчанию и так далее.

Структура сущностей указана в таблицах 6 – 15.

В таблице в примечании указаны сокращения: PK – первичный ключ, FK – внешний ключ.

Таблица 6 - Структура сущности «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код пользователя | ID\_user | int(11) | PK |
| Пользователь | User | varchar(40) |  |
| Логин | Login | varchar(40) |  |
| Пароль | Password | varchar(50) |  |
| Кодовое слово | Code\_word | varchar(25) |  |

Сущность «Пользователи» связана с сущностями «Студенты» и «Преподаватели» посредством первичного ключа «Код пользователя».

Таблица 7 - Структура сущности «Студенты»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код студента | ID\_student | int(6) | PK |
| Фамилия | Surname | varchar(40) |  |
| Имя | Name | varchar(40) |  |
| Отчество | Middle\_name | varchar(40) |  |
| Код группы | ID\_group | int(6) | FK |
| Код пользователя | ID\_user | int(6) | FK |

Сущность «Студенты» связана с сущностями «Открытые тесты» и «Тесты» посредством первичного ключа «Код студента».

Таблица 8 - Структура сущности «Группы»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код группы | ID\_group | int(6) | PK |
| Название группы | Name\_group | varchar(30) |  |
| Курс | Kurs | int(1) |  |

Сущность «Группы» связана с сущностью «Студенты» посредством первичного ключа «Код группы».

Таблица 9 - Структура сущности «Преподаватели»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код преподавателя | ID\_teacher | int(6) | PK |
| Фамилия | Surname | varchar(40) |  |
| Имя | Name | varchar(40) |  |
| Отчество | Middle\_name | varchar(40) |  |
| Код пользователя | ID\_user | int(6) | FK |
| Дата рождения | Year\_of\_birth | date |  |

Сущность «Преподаватели» связана с сущностями «Дисциплины», «Темы тестов» и «Открытые тесты» посредством первичного ключа «Код преподавателя».

Таблица 10 - Структура сущности «Дисциплины»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код дисциплины | ID\_discipline | int(6) | PK |
| Название дисциплины | Name\_discipline | varchar(200) |  |
| Код преподавателя | ID\_teacher | int(6) | FK |

Сущность «Дисциплины» связана с сущностями «Темы тестов» и «Открытые тесты» посредством первичного ключа «Код дисциплины».

Таблица 11 - Структура сущности «Тесты»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код теста | ID\_test | int(6) | PK |
| Код темы | ID\_topic | int(6) | FK |
| Код студента | ID\_student | int(6) | FK |

Продолжение таблицы 11

|  |  |  |  |
| --- | --- | --- | --- |
| Оценка | Аssessment | int(3) |  |
| Дата прохождения | Date | date |  |

Сущность «Тесты» связана с другими сущностями только посредством внешних ключей.

Таблица 12 - Структура сущности «Темы тестов»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код темы | ID\_topic | int(6) | PK |
| Код дисциплины | ID\_discipline | int(6) | FK |
| Название темы | Name\_topic | varchar(200) | FK |
| Количество вопросов | Number\_of\_questions | int(2) |  |
| Код преподавателя | ID\_teacher | int(6) | FK |
| Количество баллов | Kol\_balls | int(3) |  |

Сущность «Темы тестов» связана с сущностями «Вопросы» и «Открытые тесты» посредством первичного ключа «Код темы».

Таблица 13 - Структура сущности «Вопросы»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код вопроса | ID\_question | int(6) | PK |
| Код темы | ID\_topic | int(6) | FK |
| Текст вопроса | Text\_question | varchar(500) |  |
| Тип вопроса | Type | varchar(25) |  |
| Баллы | Score | int(2) |  |

Сущность «Вопросы» связана с сущностью «Ответы» посредством первичного ключа «Код вопроса».

Таблица 14 - Структура сущности «Ответы»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код ответа | ID\_answer | int(6) | PK |
| Код вопроса | ID\_question | int(6) | FK |
| Текст ответа | Text\_answer | varchar(200) |  |
| Правильный | Correctly | tinyint(1) |  |

Сущность «Ответы» связана с другими сущностями только посредством внешних ключей.

Таблица 15 - Структура сущности «Открытые тесты»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | | Тип данных | Примечание |
| Логический уровень | Физический уровень |
| Код | ID | int(11) | PK |
| Код темы | ID\_topic | int(6) | FK |
| Код дисциплины | ID\_discipline | int(6) | FK |
| Код преподавателя | ID\_teacher | int(6) | FK |
| Код студента | ID\_student | int(6) | FK |
| Количество попыток | Try | int(2) |  |

Сущность «Открытые тесты» связана с другими сущностями только посредством внешних ключей.

В нашей модели использованы только связи один ко многим.

Наглядно даталогическая модель БД ГенТест представлена на рисунке 6.

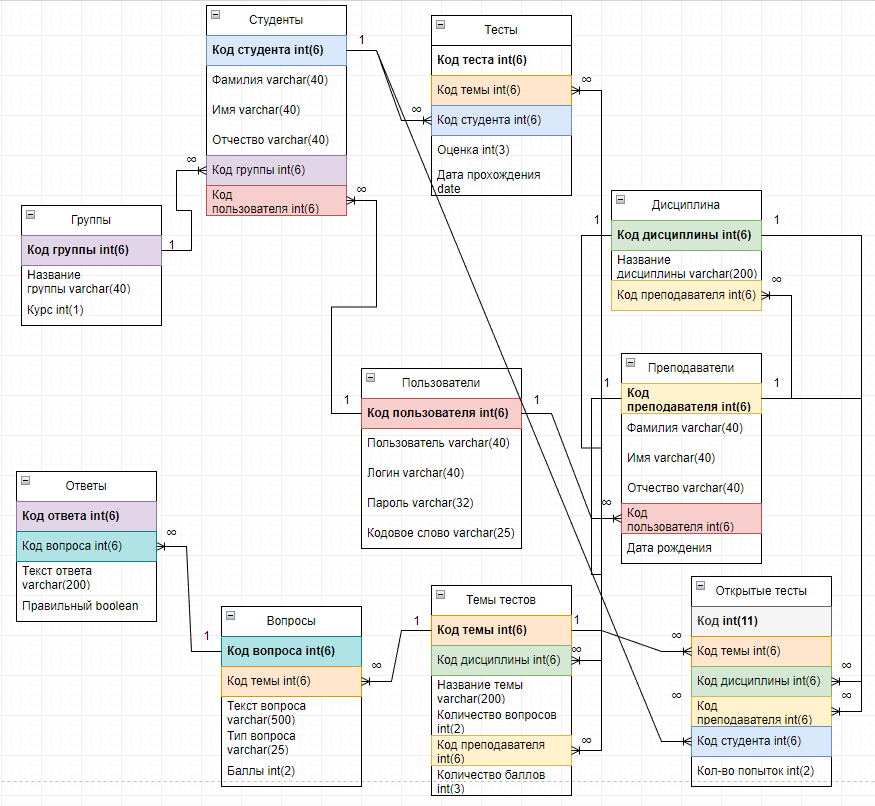


Рисунок 6 - Даталогическая модель БД

### 2.2.6 Разработка физической модели БД

Физическая модель БД – это конечная модель в построении БД, это фактическая реализация данных. Физическая модель выполняется разработчиками и системным администратором. Модель разрабатывается для конкретной СУБД.

В таблице 16 представлены имена сущностей логического уровня и в соответствии им имена таблиц в физической модели.

Таблица 16 - Соответствие наименований сущностей и таблиц моделей БД

|  |  |
| --- | --- |
| Логический уровень | Физический уровень |
| Пользователи | users |
| Студенты | [students](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=students) |
| Группы | [groups](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=groups) |
| Преподаватели | [teachers](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=teachers) |
| Дисциплины | [disciplines](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=disciplines) |
| Тесты | [tests](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=tests) |
| Темы тестов | [test\_topics](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=test_topics) |
| Вопросы | [questions](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=questions) |
| Ответы | answers |
| Открытые тесты | [open\_tests](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=gentest&table=open_tests) |

Физическая модель БД ГенТест представлена на рисунке 7.

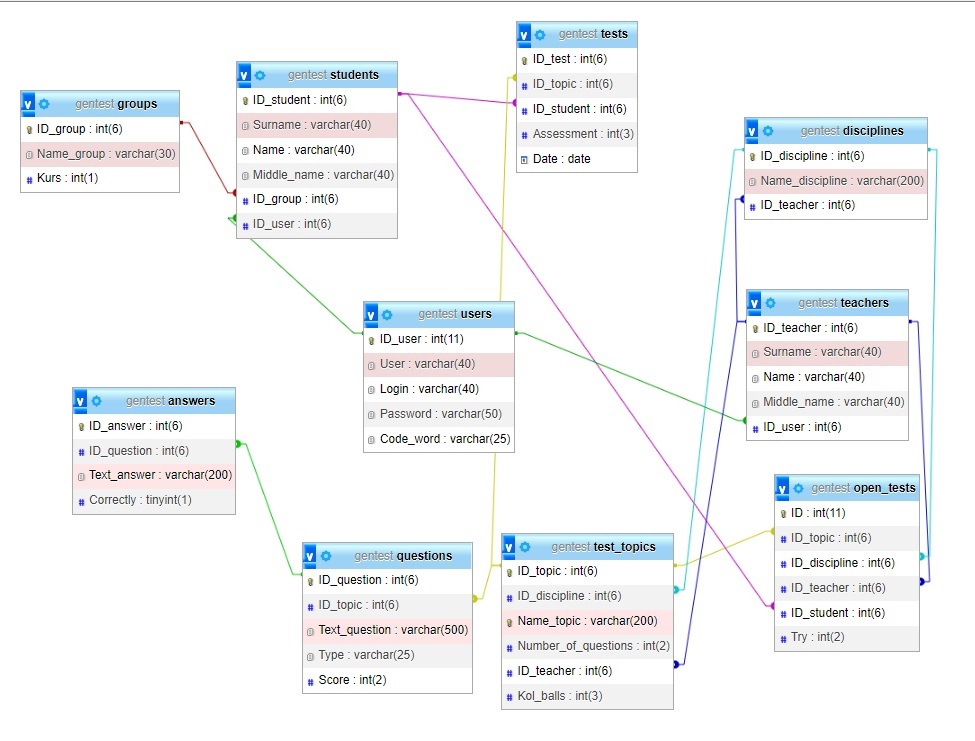


Рисунок 7- Физическая модель БД

### 2.2.7 Обеспечение надежности и безопасности

Надежность системы обеспечивается работой дискового массива RAID 10. При такой организации массива два массива RAID 1 объединяются в массив RAID 0, а это означает, что при выходе из строя любого из накопителей данные не будут потеряны.

Помимо этого, на всех серверах настроено еженедельное резервное копирование критически важной информации. Резервное копирование выполняется ночью, когда нагрузка на систему минимальна. Еженедельные резервные копии также хранятся на отказоустойчивых дисковых массивах.

Для защиты информации и оборудования от сбоев в сети электропитания, используются автономные источники резервного питания. И хотя их мощности хватает на несколько минут, этого вполне достаточно, чтобы запустить резервные дизельные генераторы и переложить всю нагрузку на них.

Безопасность персональных данных, аутентификация и разграничение доступа возложена на приложение по работе с БД.

## 2.3 Разработка приложения

### 2.3.1 Выбор языка программирования

Для разработки модуля информационной системы был выбран язык программирования С#. Это объектно-ориентированный язык программирования, который быстро развивается и достаточно востребован среди других языков программирования. Понятный и простой синтаксис являются его преимуществами. Благодаря нетипичным конструкциям языка и характерному синтаксису в короткие сроки можно создать полноценное функциональное приложение. Понятный синтаксис кроме разработки также упрощает работу при совместном программировании – читать чужой код становится проще. От чего процесс рефакторинга и исправления багов, особенно в кроманде, становится проще. Также значимым примеуществом является популярность языка – большое количество поклонников шарпа помогают его развитию. Кроме того, это благотворно сказывается на росте числа вакансий на этом языке. Программисты, которые знают си шарп, востребованы всегда, даже несмотря на их постоянное увеличение.

### 2.3.2 Выбор платформы разработки

Для разработки модуля информационной системы была выбрана среда разработки Visual Studio на платформе Windows Forms .NET. Данная среда разработки была выбрана исходя из того, что Visual Studio предлагает ряд высокоуровневых функциональных возможностей, что является большим преимуществом перед другими средами разработки, а именно:

– Платформа поддерживает написание кода на нескольких языках и позволяет писать код на любом предпочитаемом программистом языке, используя один и тот же привычный интерфейс;

– Для написания большинства приложений используется меньше кода;

– Более высокая скорость разработки благодаря функции IntelliSense, которая позволяет найти и заменить ключевые слова как в одном файле, так и во всем проекте, также функции автоматического добавления и удаления комментариев, что позволяет скрыть участки кода;

– Возможности отладки. Благодаря предложенным инструментам отладки, можно отследить любые ошибки до этого непонятного происхождения. Разработчик может выполнять свой код построчно, устанавливать точки останова, при желании просматривать в любое время текущую информацию из памяти, например текущие значения переменных.

### 2.3.3 Построение UML-диаграмм

#### 2.3.3.1 Язык UML

UML в переводе – унифицированный язык моделирования. Это язык, в котором каждая фигура, стрелочка, человечек, любой другой объект и их сочетания однозначно индентифицируются, то есть им присвоены конкретные значения. Язык позволяет визуально представить явление или процесс так, чтобы схема была понятна всем тем, кто знает язык UML.

Иначе говоря, UML – это набор правил, по которым рисуются схемы. Зная обозначения, можно легко и быстро создавать универсальные чертежи сложных структур, процессов и явлений. Именно благодаря этому, специалисты в области разработки ПО часто прибегают к моделиролванию и проектированию процессов на языке UML.

Язык необходим для описания и визуализирования какой-либо абстрактной модели, например, модели процесса выполнения запросов ПО для получения клиентом ожидаемого результата.

Достоинства UML:

* Стандартизация, то есть схема будет понятна любом, кто знаком с UML;
* Полнота. В языке предусмотрено все, что необходимо, то есть не придется заново изобретать велосипед и что-то выдумывать в процессе, чем можно вызвать непонимание коллег;
* Распространенность. Язык известен во многих странах мира, его используют в разных сферах: менеджменте, инженерии, а не только в сфере информационных технологий;
* Инструменты для автоматизации. На основе кода они помогают автоматически генерировать UML схемы. Например, программа Umbrello позволяет быстро создавать наглядные представления кода.  
  Напротив, есть инструменты, которые меняют код при изменениях в UML-схеме, например, Microsoft Visual Studio 2010 Feature Pack 2. Так, язык UML помогает разработчикам в автоматизации процессов.

#### 2.3.3.2 UML-диаграмма

UML-диаграмма – это схема, которая нарисована с помощью символов языка UML. Она может содержать множество элементов, объектов и связей, соединений между ними. Полное описание проекта может состоять из нескольких таких диаграмм, которые могут быть связаны или не связаны между собой.

Элемента диаграммы UML – классы программного кода, так же могут быть страницы веб-сайта или какие-либо части механизма, то есть вся зависит от того, какую часть, процесс или сущность описывает разработчик.

**2.3.3.3 Диаграмма прецедентов**

Диаграмма прецедентов описывает, какой функционал разрабатываемого ПО доступен каждой группе пользователей.

Диаграмма прецедентов – первая, исходная диаграмма или концептуальная модель, которая строится в процессе проектирования и разработки.

Цели диаграммы прецедентов:

* Определить границы и контекст предметной области, подлежащей моделированию на ранних стадиях проектирования системы;
* Определить общие требования к функционалу проектируемой системы;
* Разработать начальную концептуальную модель системы, которая будет детализирована в виде логической и физической моделей;
* Разработать начальный документ для диалога между разработчиками системы и их заказчиками и пользователями.

Таким образом, суть диаграммы прецедентов такова, что создаваемая система представляется в виде набора сущностей или актеров, взаимодействующих с этой системой посредством прецедентов.

Актер – действующее лицо, которым называется любая сущность, которая внешне взаимодействует с системой. Им может быть как человек, так и неодушевленный предмет, то есть технической устройство, ПО или любая другая система, которая может воздействовать на создаваемую систему.

Прецедент необходим для описания возможностей, которые предоставляет система актеру. Иначе говоря, каждый прецедент – некоторый набор действий, совершаемый системой при диалоге с актером. Однако каким образом будет реализовано взаимодействие актеров с системой – не будет описано.

При построении диаграммы прецедентов актеров и прецеденты необходимо связывать. При этом один актер может быть связан с одним прецедентом и наоборот – у одного прецедента может быть больше, чем один актер. Нельзя строить связи между двумя актерами. При построении диаграммы актер всегда определяет какой-либо прецедент, то есть обязательно должна идти стрелка от актера к прецеденту.

Кроме связей, на диаграмме могут быть представлены так же включения, которые используются, когда в системе есть фрагмент, который повторяется в системе несколько раз в разных прецедентах, чтобы не нагромождать схему.

Также, в схеме может использоваться отношение обобщения, когда некоторый актер может быть обобщен до другого актера, при этом стрелка будет направлена от частного случая к общему.

В нашей схеме представлено четыре основных актера, которые обобщаются до пятого, объединяющего их, а именно:

Основные, частные актеры:

* СА;
* Преподаватель;
* Студент;
* Администрация.

Обобщающий актер:

* Любой пользователь.

Диаграмма прецедентов разрабатываемого ПО представлена на рисунке 8.

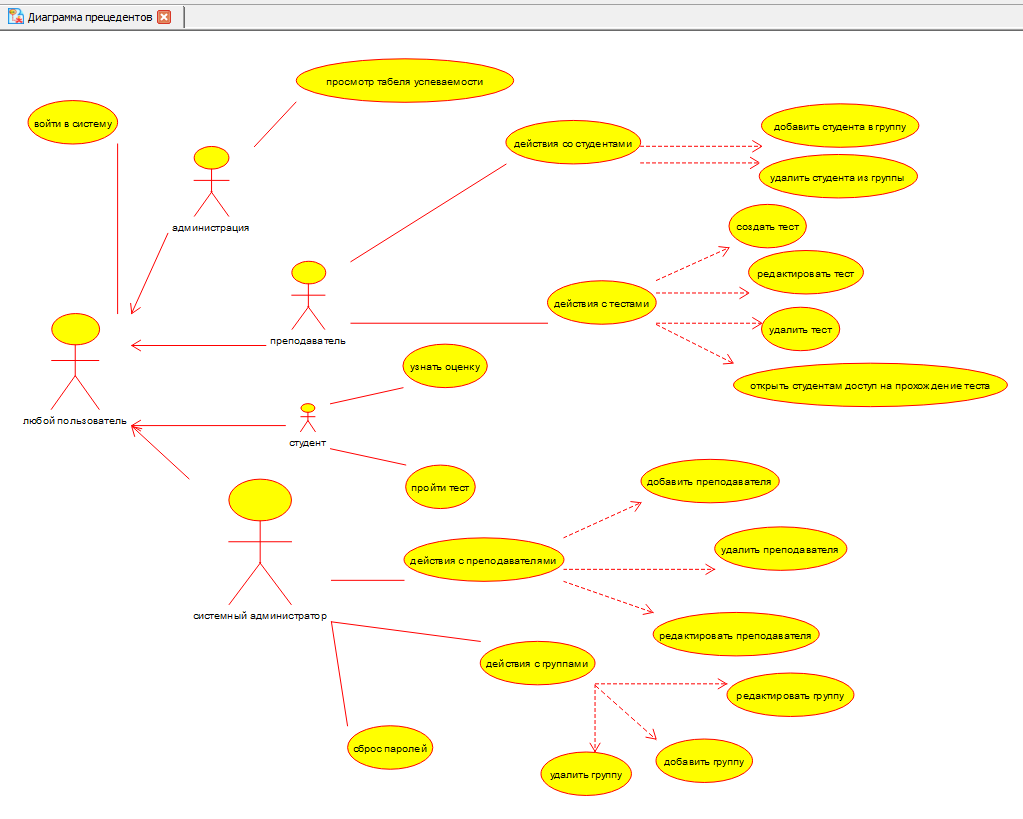


Рисунок 8 - Диаграмма прецедентов

Актеры и связанные с ними прецеденты, а также возможные включения помещены в таблицу 17.

Таблица 17 - Элементы диаграммы прецедентов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Актер | Прецедент | Включения | | |
| Любой пользователь | Войти в систему |  | | |
| Администрация | Просмотр табеля успеваемости |  | | |
| Преподаватель | Действия со студентами | Добавить студента в группу | | |
| Удалить студента из группы | | |
| Действия с тестами | Создать тест | | |
| Редактировать тест | | |
|  |  | | Удалить тест |  |
| Открыть студентам доступ на прохождение теста |  |
| Студент | Узнать оцену | |  |  |

Продолжение таблицы 17

|  |  |  |
| --- | --- | --- |
|  | Пройти тест |  |
| СА | Действия с преподавателями | Добавить преподавателя |
| Редактировать преподавателя |
| Удалить преподавателя |
| Действия с группами | Добавить группу |
| Редактировать группу |
| Удалить группу |
| Сброс паролей |  |

#### 2.3.3.4 Диаграмма деятельности

Диаграмма деятельности – это схематичное представление последовательности выполнения процессов и задач, которые выполняются в рамках какого-либо программного проекта или деятельности. Данная диаграмма необходима для планирования бизнес-процессов, их оптимизации и анализа. Диаграмма помогает в принятии решений и управлении проектами. Она, как и любая диаграмма языка UML, состоит из набора символов и линий, имеющих каждый свое значение.

У каждой задачи указывается, какие ресурсы необходимы для ее выполнения, описываются результаты на каждом этапе процесса, а также последовательность задач и операций, необходимых для успешного завершения.

Опишем основные элементы, которые будут использоваться в нашей диаграмме:

1. Начальный узел активности – первый элемент диаграммы, который представляет точку старта процесса. Обозначается в виде круга, имеющей только исходящие связи к другим элементам диаграммы.
2. **Конечный узел активности – последний элемент диаграммы, который обозначает завершение процесса. Обозначается в виде круга и имеет только входящие связи, указывающие на предыдущий элемент диаграммы.**
3. Действие – это основной элемент диаграммы, который может представлять определенное действие или задачу, необходимую выполнить. Обозначается В виде прямоугольника с закругленными углами, внутри которой прописано название выполняемой задачи.
4. **Поток управления – элемент, который определяет порядок выполнения действий и направление передачи управления от элемента к элементу. На диаграмме изображается в виде стрелки, которая связывает элементы диаграммы.**
5. **Узел принятия решений – элемент, использующийся для принятия различных решений и предоставляющий выбор между этими решениями. Обозначается в виде ромба. Имеет несколько выходов, каждый из которых соответствует определенному решению. Выходы обозначаются стрелками.**
6. **Вилка – элемент, который показывает на одновременное начало нескольких параллельно выполняющихся процессов. Обозначается в виде жирной горизонтальной линии, из которой исходит несколько стрелок.**

**Во избежание награможденности, было решено спроектировать четыре разные диаграммы по деятельности каждой из группы пользователей.**

**Диаграмма деятельности преподавателя расположена на рисунке 9.**

Опишем диаграмму. Первым действием преподавателя будет авторизация на портале ГенТест. После чего он попадает в главное меню, где ему предложено выбрать пункт меню «Действия со студентами» или «Действия с тестами». В зависимости о выбора преподаватель попадает в соответствующие пункты меню. На протяжении всей работы с приложением пользователь может выйти из него.

При открытии меню «Действия со студентами» дальнейшим действием будет ввод данных студента и нужную группу, после чего преподаватель может добавить или удалить студента из группы. Далее преподаватель может продолжить или завершить работу в приложении. При продолжении работы преподаватель может выйти как в главное меню, так и остаться в меню «Действия со студентами».

При открытии меню «Действия с тестами» для дальнейшей работы, преподавателю необходимо ввести название предмета, с тестами которого будут производиться манипуляции. После преподавателю дается на выбор: создать блок вопросов для теста, редактировать, удалить блок вопросов или открыть тест для прохождения.

В первом случае преподавателю в дальнейшем необходимо будет ввести название теста, количество вопросов и общий балл, далее преподаватель попадает в мини-цикл по созданию пула вопросов: сначала вводится текст вопроса, далее тексты ответов, затем выбирается(ются) правильный(ые) ответ(ы). При завершении работы с созданием вопроса, преподавателю вновь встают выбор: продолжить ввод вопросов или завершить создание блока вопросов. В очередной раз, при решении о завершении создания теста, преподаватель может продолжить работу с тестами, либо перейти в главное меню.

При выборе второго, редактирования теста, преподавателю на выбор дается изменение название темы теста, или добавление в пул новых вопросов. Принцип действия добавления вопросов схож с принципом при полном создании теста. Далее преподаватель может как завершить работу с приложением, так и продолжить работу с тестами, либо перейти в главное меню.

В случае удаления блока вопросов, необходимо выбрать нужную тему, а затем нажать кнопку «Удалить». Далее преподаватель может как завершить работу с приложением, так и продолжить работу с тестами, либо перейти в главное меню. Для открытия теста необходимо выбрать группу или отдельного студента(ов) и нажать на кнопку «Открыть». Далее преподаватель может как завершить работу с приложением, так и продолжить работу с тестами, либо перейти в главное меню.

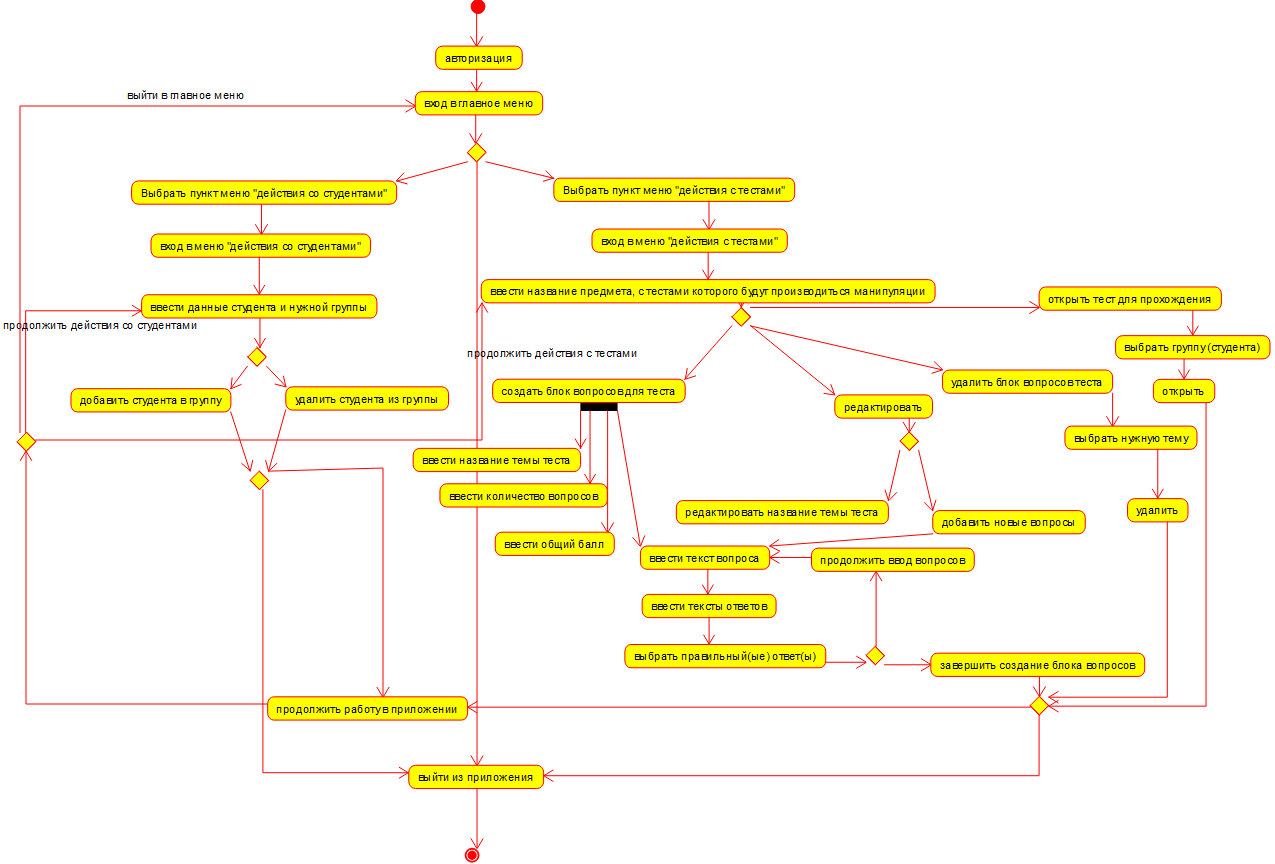


Рисунок 9 - Диаграмма деятельности преподавателя

Диаграмма деятельности СА **расположена на рисунке 10.**

Опишем диаграмму. Первым дествием СА будет авторизация на портале ГенТест. После чего он попадает в главное меню, где ему предложено выбрать одну из следующих вкладок: «Преподаватели», «Группы», «Сброс паролей». Рассмотрим каждую из них. На протяжении всей работы с приложением пользователь может выйти из него.

Войдя во вкладку «Преподаватели» СА предложено так же выбрать одну из трех вкладок: «Добавление», «Редактирование», «Удаление». При выборе первого, СА необходимо ввести ФИО преподавателя, логин, пароль и кодовое слово, а также дисицплину(ы). Далее СА может продолжить работу с добавлением преподавателей, либо перейти в главное меню, либо выйти из приложения полностью. При выборе вкладки «Редактирование» предлагается на выбор редактировать ФИО или нет, а так же любые действия с дисциплинами: удаление, добавление или оставить все, как есть. В любой момент СА может выйти из приложения или зайти в любое из меню. При выборе вкладки «Удаление» необходимо выбрать нужного преподавателя, а затем нажать на соответствующую кнопку.

Войдя во вкладку «Группы» дается возможность добавить, редактировать или удалить группу. Для этого необходимо ввести соответствующие данные.

Войдя во вкладку «Сброс паролей» необходимо ввести тип пользователя, по выбору ввести ФИО или логин, либо и то, и то, пользователя, а затем вести новый пароль, после чего нажать на кнопку «Сохранить».

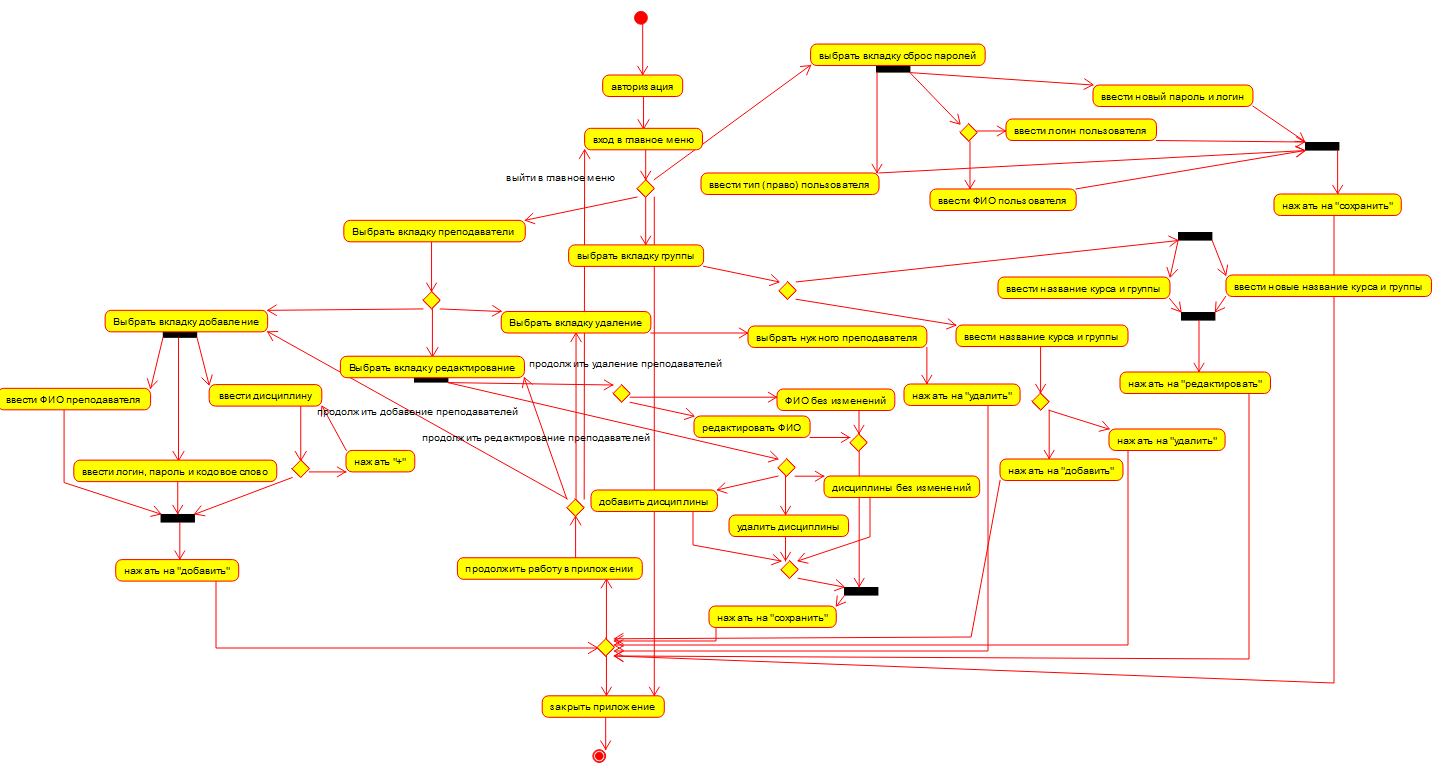


Рисунок 10 - Диаграмма деятельности СА

Диаграмма деятельности **студента расположена на рисунке 11.**

Опишем диаграмму. Первым дествием будет авторизация на портале ГенТест. После чего он попадает в главное меню, где ему предложено выбрать одну из следующих кнопок: «Тестирование» или «Просмотр баллов». На протяжении всей работы с приложением пользователь может выйти из него.

При выборе кнопки тестирования, пользователь попадает в меню «Тестирование», далее, если нужно пройти тест, необходимо выбрать преподавателя, предмет, тему, а затем нажать на кнопку «Начать». Далее мы попадаем в цикл по решению теста. По завершению теста студент может как продолжить работу в приложении, так и закрыть его.

При выборе кнопки «Просмотр баллов» мы попадаем в соответствующее меню. Если цель верна, далее необходимо выбрать преподавателя и предмет. При желании можно как закрыть приложение, так и перейти в главное меню.

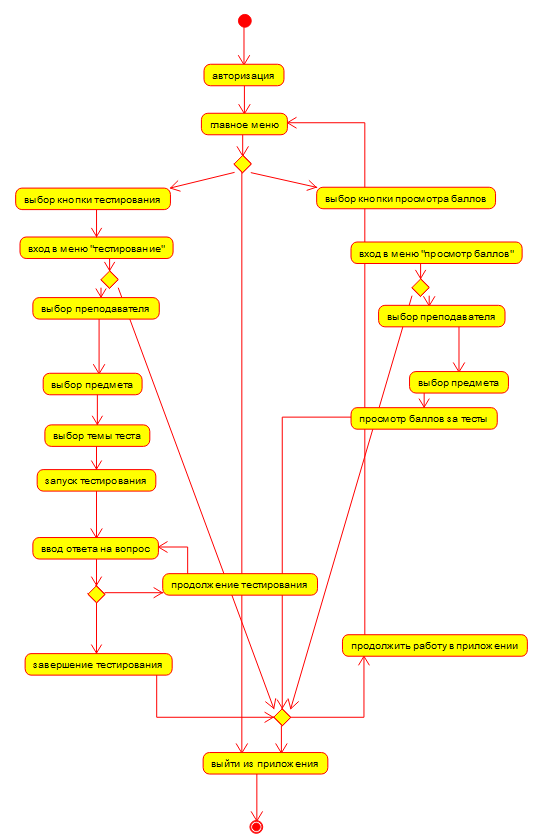


Рисунок 11 - Диаграмма деятельности студента

Диаграмма деятельности администратора **расположена на рисунке 12.**

Опишем диаграмму. Первым дествием будет авторизация на портале ГенТест. После чего он попадает в главное меню, где у него есть выбор выйти из приложения, либо открыть табель успеваемости. При открытии табеля, администратору будет нужно ввести ФИО преподавателя, после чего ввеси дисциплину, ввести курс, группу, а также ввести тему. После ввода всей информации ему откроется табель по успеваемости соответствующей группы по соответствующему предмету.

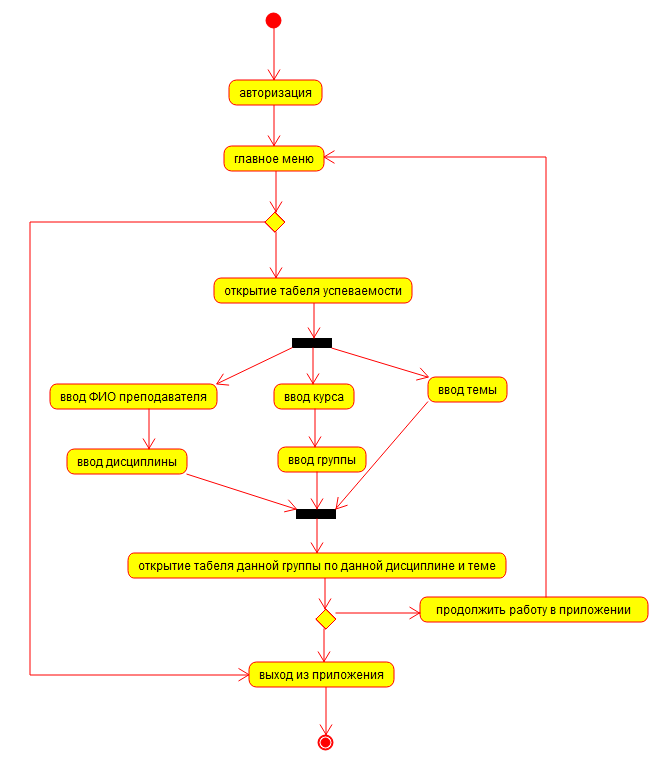


Рисунок 12 - Диаграмма деятельности администратора

#### 2.3.3.5 Диаграмма состояний

Диаграмма состояний – диаграмма UML, которая, после совершении определенного действия, отображает переход объекта из одного состояния в другое.

Диаграмма состояний показывает:

1. Набор состояний объекта;
2. События, которые вызывают переход из одного состояния в другое;
3. Действия, которые происходят в результате перемены состояния.

Состояние, на языке UML – период в жизни объекта, когда он удовлетворяет какому-либоусловию, выполняет определенную деятельность или ожидает какого-то события. Обозначается как прямоугольник с закругленными углами, внутри которого прописано его имя и, при наличии, подстостояния. С состоянием можно связывать входное или выходное действие, деятельность, событие и историю состояния.

Существует начальное состояние и конечное состояние, которые не содержат никаких действий. Графически обозначаются, как кружочки.

Между состояниями возможны переходы – отношение между двумя состояниями, идущими друг за другом, которое указывает факт их смены. Переходы обозначаются стрелками.

Событие – некоторый факт, имеющий место в пространстве и во времени.

Диаграмма состояний главной функции приложения изображена на рисунке 13.

Начальное положение, положение по умолчанию, после действий по созданию бокса, сменяется наличием готового бокса вопросов по теме теста. Далее происходит генерация варианта теста, после чего создается вариант теста. После происходит тестирование, что приводит к решенному тесту и конечному положению диаграммы.

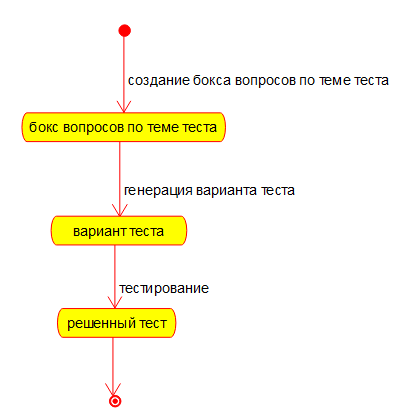


Рисунок 13 - Диаграмма состояний

#### 2.3.3.6 Диаграмма последовательности

Диаграмма последовательности – диаграмма, моделирующая взаимодействие между объектами, упорядоченными по времени их проявления. Они показывают, как различные части системы взаимодествуют друг с другом, выполняя какие-либо функции, а также порядок, в котором происходит взаимодействие.

Схема построена так, что она отражает собой временную шкалу, которая постепенно спускается. Каждый объект имеет свою колонку, а сообщения между ними изображены стрелками.

Основные элементы данной диаграммы – объекты, которые обозначаются прямоугольниками, вертикальные линии, отходящие от них, которые отображают течение времени при деятельности объекта, а также стрелки, которые показывают выполнение действий объектами. Объекты на диаграмме располагаются слева направо.

Диаграмма последовательности помогает определиить обязанности каждого исполнителя в виде набора операций.

Диаграмма последовательности изображена на рисунке 14.

На диаграмме изображены четыре объекта: СА, ГенТест, БД, преподаватель, студент и администрация. Между этими объектами по очереди отходят стрелки. Например, от СА к ГенТесту исходит стрелка: 1. Регистрация преподавателя в системе, что означает выполнение регистрации преподавателя СА.

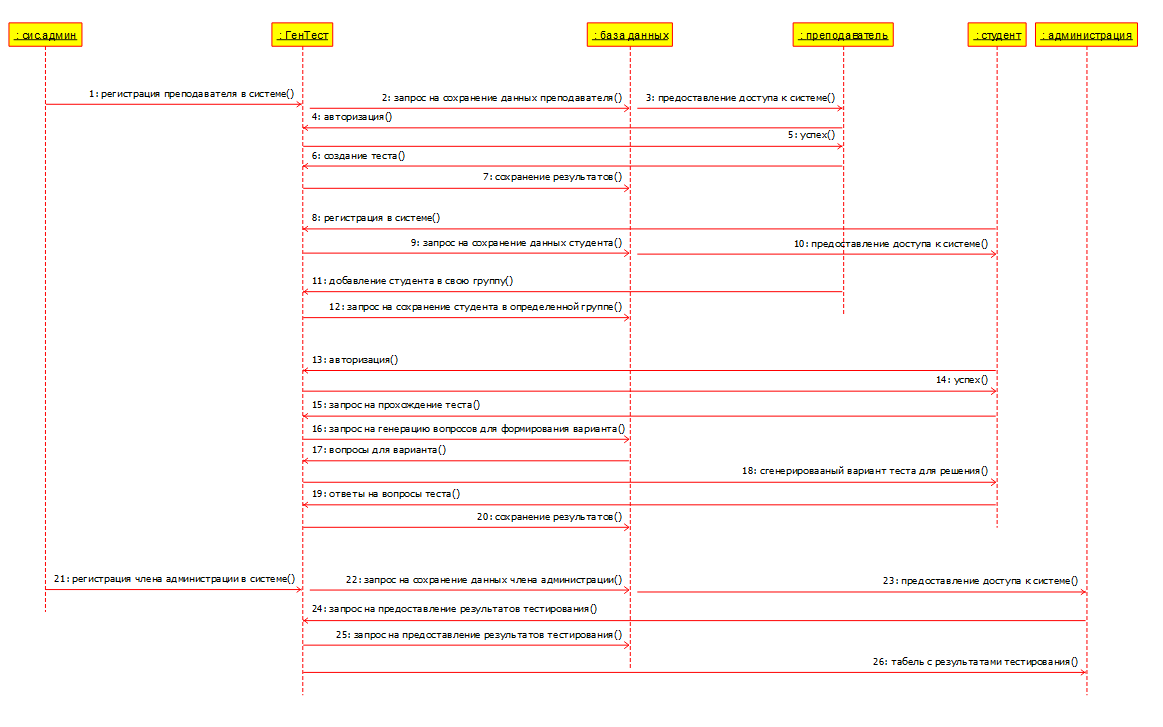


Рисунок 14 - Диаграмма последовательности

### 2.3.4 Интерфейс приложения

Приложение было разработано на Windows Forms. Весь функционал, все, что описано в UML диаграммах, было реализовано в приложении.

Иконка приложения выглядит, как показано на рисунке 15.



Рисунок 15 - Иконка приложения

Меню авторизации расположено на рисунке 16. Авторизоваться может любой пользователь, прошедший аутентификацию. Есть как возможность просмотра введенного пароля, так и его обратная шифрация.

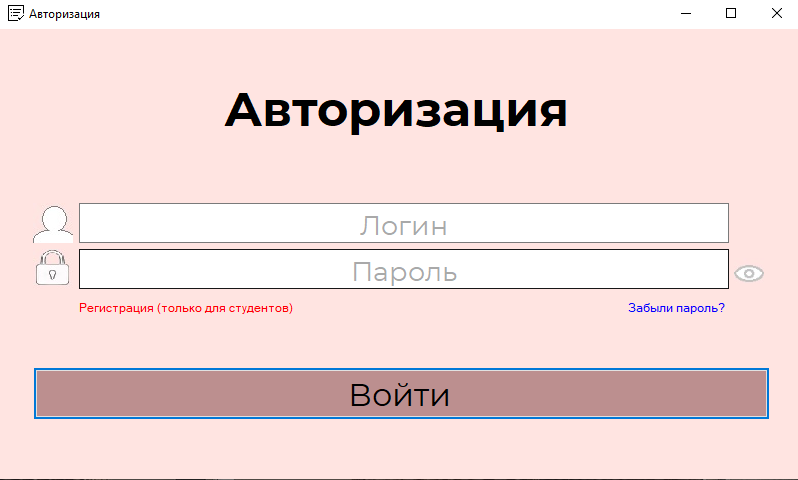


Рисунок 16 - Авторизация

На форме авторизации расположены две кнопки: регистрация, которая возможна только для студентов, так как они имеют минимальные права доступа к системе и не смогут навредить. Других пользователей регистрирует и раздает права СА. Вторая кнопка нужна для восстановления пароля. Формы регистрации для студентов и восстановления пароля представлены на рисунке 17 и 18 соответственно.

На форме регистрации есть возможность просмотра введенного пароля, а также подсказка для лучшего запоминания кодового слова. При отмене регистрации или напротив – ее сохранения, есть соответствующие кнопки.

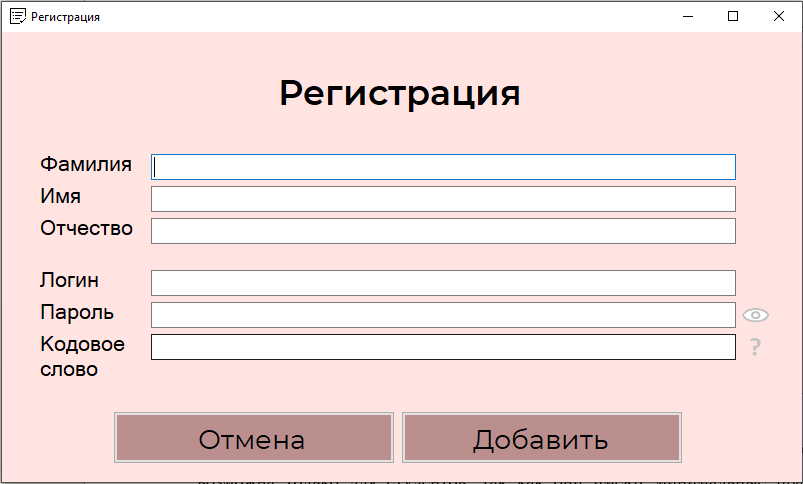


Рисунок 17 - Регистрация для студентов

Форма восстановления доступа содержит все необходимые данные по восстановлению пароля. Главное – нужно помнить кодовое слово, иначе пароль придется восстанавливать через СА.

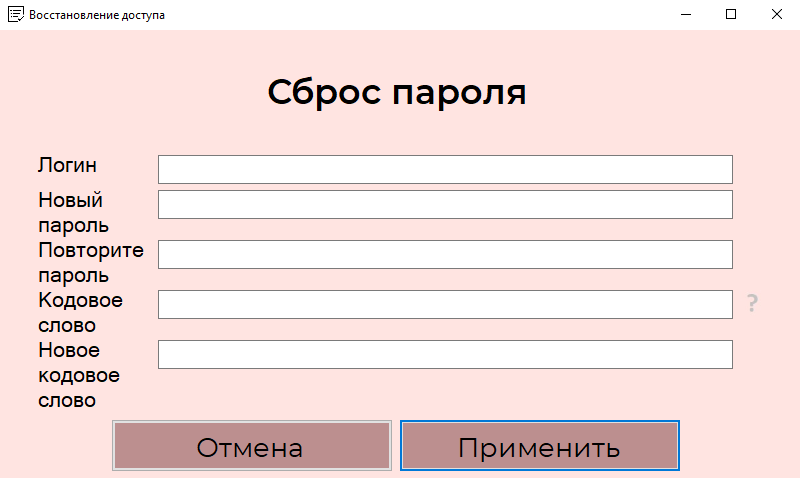


Рисунок 18 - Сброс пароля

Форма меню СА содержит три вкладки: по работе с преподавателями, группами, а также сброс паролей. Вкладка преподаватели содержит дополнительно вкладки добавление, редактирование и удаление. Скрины с полным меню СА представлены в ПРИЛОЖЕНИИ Б. Скрин с ознакомительным меню СА расположен на рисунке 19.

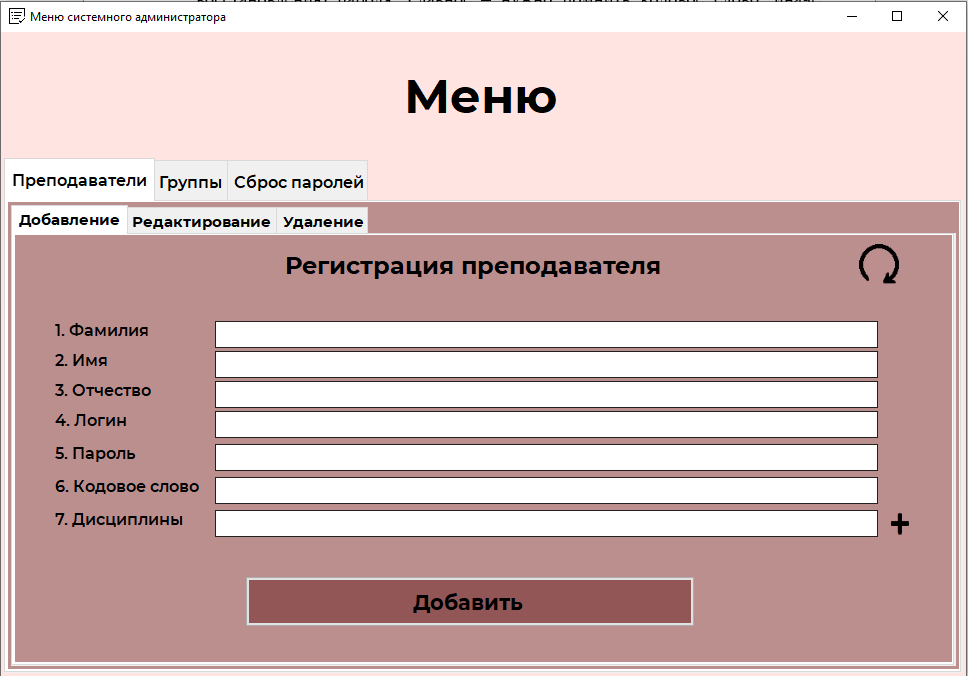


Рисунок 19 - Меню СА

Форма меню преподавателя содержит две кнопки – «Тесты», «Студенты», под которыми кратко изложен функционал при нажатии на них. Скрины с полным меню преподавателя представлены в ПРИЛОЖЕНИИ Б. Скрин с ознакомительным меню преподавателя расположен на рисунке 20.

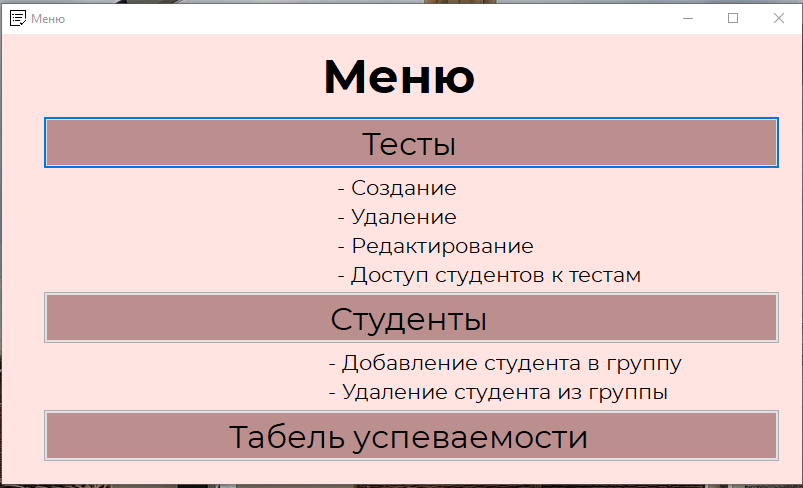


Рисунок 20 - Меню преподавателя

Форма меню студента содержит две кнопки – «Тестирование», «Просмотр баллов». При переходе на них, студент попадает на соответствующие формы. Скрины с полным меню студента представлены в ПРИЛОЖЕНИИ Б. Скрин с ознакомительным меню студента расположен на рисунке 21.

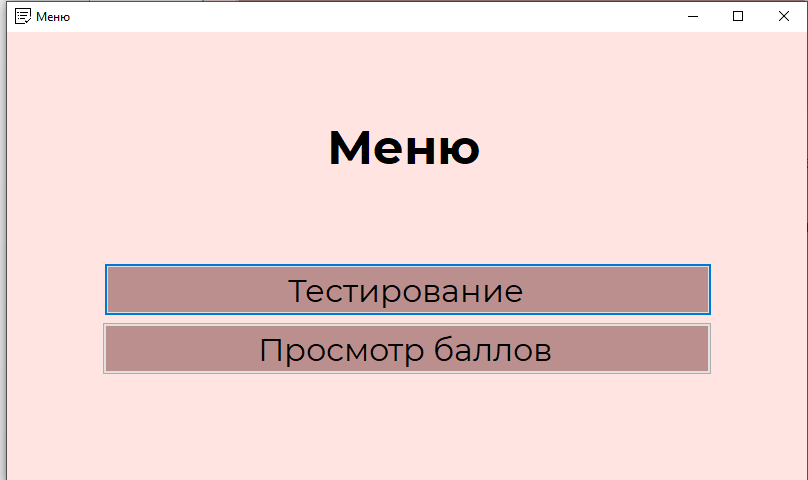


Рисунок 21 - Меню студента

Форма меню администратора содержит одну кнопку «Табель успеваемости», при нажатии на которую студент попадает на форму, расположенную на рисунке 22. Это весь функционал администратора.

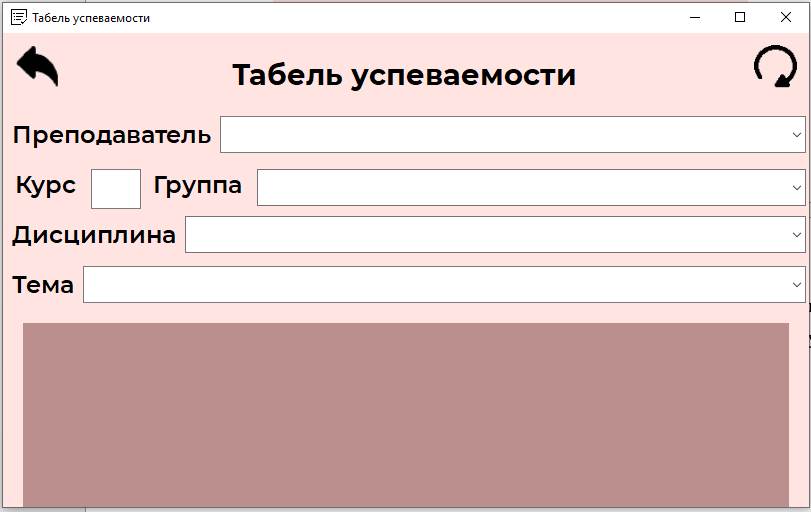


Рисунок 22 - Меню администратора

### 2.3.5 Тестирование ПП

Тестирование ПО – проверка на соответствие ожидаемых и полученных результатов от поведения программы.

Цель тестирования – проверка ПО на соответствие предъявляемым требованиям, поиск возможных ошибок в ПО, которые необходимо выявить до тестирования ПП конечным пользователем, также получение гарантии уверенности в качестве ПО.

Причины необходимости тестирования:

* Для проверки соответствия требованиям;
* Для обнаружения проблем на более ранних стадиях разработки и недопущения повышения стоимости ПП;
* Для обнаружения вариантов использования, которые не были предусмотрены при разработке;
* Для взгляда на ПП глазами пользователя;
* Для повышения доверия к компании и продукту.

Существует несколько методов тестирования.

Тестирование белого ящика – метод, который предполагает, что тестировщику известна внутренняя структура системы. Это взгляд с точки зрения разработчика, рассматриваются рабочие алгоритмы, агализируется внутренняя структура компонента или всей системы.

Тестирование серого ящика – метод, когда комбинируются два крайних подхода – метод белого и черного ящиков. Внутреннее устройство программы известно частично.

Тестирование черного ящика – тестирование, основанное исключительно с интерфейсом тестируемой системы. Оценивают только вводимые данные и полученные результаты. Это тестирование с точки зрения пользователя.

В нашем случае будет использоваться метод серого ящика, так как знания структуры будут частично использованы в тестировании, однако тест кейсы написаны с точки зрения конечного пользователя.

Тестирование ПП полностью представлено в ПРИЛОЖЕНИИ В.

### 2.3.6 Внедрение ПП

После прохождения ПП процесса отладки, возможно внедрение на предприятие и сдача в опытно-промышленную эксплуатацию. Инструкция по использованию программы состоит из двух частей, а именно из инструкции для оператора и инструкции для программиста.

Инструкция по использованию программы представлена в ПРИЛОЖЕНИИ Г.

## 2.4 Экономическая эффективность проекта

Главный экономический эффект от внедрения разработки заключается в улучшении эффективности рабочего процесса за счет снижения трудозатрат на составление вариантов и проверки тестов, а также в экономии на расходных материалах (бумага, картриджи).

Критерием эффективности создания и внедрения новых средств автоматизации является ожидаемый экономический эффект. Он определяется по формуле:

Э = Эр - Ен \* Кп,

где Эр – годовая экономия;

Ен – нормативный коэффициент (Ен = 0.15);

Кп – капитальные затраты на проектирование и внедрение, включая первоначальную стоимость программы.

Годовая экономия Эр складывается из экономии эксплуатационных расходов и экономии в связи с повышением производительности труда пользователя. Таким образом, получаем:

Эр = (Р1 - Р2) + ΔРп,            (1)

где Р1 и Р2 – соответственно эксплуатационные расходы до и после внедрения разрабатываемой программы;

ΔРп – экономия от повышения производительности труда дополнительных пользователей.

Расчитаем капитальные затраты на проектирование и внедрение ИС.

Капитальные затраты расчитываются исходя из длительности работ на этапе проектирования и внедрения ИС.

Проектирование – это совокупность работ, необходимых для проектирования системы. Внедрением называется комплекс работ по вводу системы в эксплуатацию с ее возможными доработками.

Продолжительность работ рассчитывают на основании экспертных оценок по формуле:

То=(3\*Тmin+2\*Тmax)/5                  (2),

где Т0 – ожидаемая продолжительность работ;

Тmin и Тmax – соответственно наименьшая и наибольшая по мнению эксперта длительность работы.

Данные расчетов ожидаемой длительности работ приведены в таблицу 1.

Таблица – Продолжительность работ на этапе проектирования и внедрения ИС.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ | Длительность работ, дней | | |
| Минимум | Максимум | Ожидание |
| Разработка технического задания | 1 | 2 | 2 |
| Анализ технического задания | 3 | 6 | 4 |
| Изучение литературы | 5 | 10 | 7 |
| Работа в библиотеке с источниками | 2 | 5 | 3 |
| Знакомство с основными этапами дипломной работы | 1 | 3 | 2 |
| Оформление ТЗ | 1 | 2 | 2 |
| Разработка алгоритма | 7 | 10 | 8 |

Продолжение таблицы 18

|  |  |  |  |
| --- | --- | --- | --- |
| Доработки программы | 25 | 30 | 27 |
| Отладка программы | 20 | 30 | 24 |
| Экономическое обоснование | 2 | 5 | 3 |
| Оформление пояснительной записки | 12 | 15 | 13 |

Капитальные затраты на этапе проектирования Кк рассчитываются по формуле:

Kк=С + Zп+Mп+H                   (3),

где:

С – первоначальная стоимость программного продукта, которая принимается за 0;

Zп – заработная плата специалистов на всех этапах проектирования и внедрения;

Мп – затраты на использование ЭВМ на этапе проектирования и внедрения;

Н – накладные расходы на этапе проектирования и внедрения.

Zп=Zп\*Тп\* (l+Aс/100)\*(l+Aп/100)                  (4),

где:

Zп – заработная плата разработчика на этапе проектирования;

Zд – дневная заработная плата разработчика на этапе проектирования;

Ас – процент отчислений на социальное страхование;

Ап – процент премий.

По формуле 4 рассчитаем заработную плату разработчику включая все отчисления:

Примем заработную плату разработчику 50000 руб/месяц

1) пенсионные отчисления - 22%;

2) взносы на обязательное социальное страхование на случай временной нетрудоспособности - 2,9%;

3) взносы на обязательное медицинское страхование в ФФОМС - 5,1%;

4) среднегодовой процент премий - 12%.

Zп = 2500 \* 59 \* (1+ (22+2,9+5,1) /100) \* (1+12/100) = 214 760‬ руб

В общем случае, расходы на машинное время состоят из расходов на процессорное время и расходов на дисплейное время. Формула для расчетов имеет вид:

Мп=tд\*Сд+ tп\*Cп                   (5),

где:

Сп и Сд - соответственно стоимости одного часа процессорного и дисплейнного времени;

tд и tп - соответственно процессорное и дисплейное время, необходимое для решения задачи (час).

При современных технических средствах запросы обрабатываются с огромной скоростью, поэтому затраты на процессорное время можно не учитывать.

При расчете Мп следует учитывать время на подготовку исходных текстов программ, их отладку и решение контрольных примеров.

На создание системы ушло 59 дней.

Сд примем по тарифной ставке 20 руб/час.

Мп= 59\*8\*20=9440 руб.

Накладные расходы составляют 80-120% от заработной платы персонала занятого эксплуатацией программы.

Н = 50000 \* 100% = 50000 руб.

Рассчитаем капитальные затраты на этапе проектирования по формуле 3:

Kк= 0+214760+9440+50000 =274200‬

Расходы по различным видам работающих определяем по формуле:

Z = nizi\*(1+ Ac/100)\*(1+Ап/100),

где:

ni - численность персонала 1-го вида связанная с выполнением paбот;

Aс - процент отчислений на социальное страхование;

Aп - средний процент премий за год.

Посчитаем расходы на содержание персонала, исходя из условия, что оклад сотрудника составляет 30 тыс руб.

Z = 1 \* 30000 \* (1 + 12% / 100) = 33600 руб.

Прочие расходы составляют 1-3% от суммы всех эксплуатационных расходов: Pпр = (274200‬+4720+40000)\*0,03 = 9567 руб.

Расчет экономии за счет увеличения производительности труда пользователя вычисляется по формуле:

Pi = (ΔTj/(Fj-ΔTj)\*100,

где:

Fj – время, которое планировалось пользователем для выполнения работы j-вида до внедрения программы (час.).

Таблица - Работы пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Вид работ | До авто­матиза­ции, мин Fj | Экономия времени, мин. ΔT | Повышение про­изводительности труда Рi (в %) |
| 1 | Составление вариантов тестов | 15 | 14 | 1400 |
| 2 | Проверка тестов | 15 | 14 | 1400 |

Экономия, связанная с повышением производительности труда пользователя Р определим по формуле: ΔP=Zп\*Σi(Pi/100),

где:

Zп – среднегодовая заработная плата пользователя.

P = 33600 \*21,07 = 707952 руб.

Экономия, связанная с экономией материальной базы:

В эксплуатационные расходы до внедрения ПП входят расходы на бумагу и принтер. В среднем лист бумаги стоит один рубль. За год необходимо потратить как минимум по одному листу на человека. Всего преподаватель составляет двадцать пять тестов для одной группы, в среднем у преподавателя десять групп, а проводимых контрольных мероприятий за год – двенадцать. Таким образом, необходимо потратить 25чел.\*10групп\*12к.т. = 3 тыс. листов бумаги, что составляет 3 тыс. руб. в год. Кроме бумаги, необходимо закупать сменные картриджи для принтера. Один картридж в среднем рассчитан на печать 1750 листов бумаги и стоит 900 рублей, из чего следует, что на 3 тыс. листов бумаги израсходуется печатного материала суммой приблизительно в 1542 руб. Затраты после внедрения ПП сводятся к нулю.

В итоге получаем следующую ожидаемую экономическую эффективность:

Э = (707952 + 3000 + 1542) – (274200+9567 )\* 0,15 = 669929‬ руб.

Таким образом, потратив 283767 рублей мы получаем экономию за год в 669929 рублей.

# ЗАКЛЮЧЕНИЕ

В соответствии с поставленной целью была разработана система тестирования «GenTest».

Для достижения поставленной цели были решены следующие задачи:

1. Проведен анализ существующих на данный момент ИС в данной сфере;
2. Проведен анализ деятельности колледжа, в котором проводится тестирование, и построена функциональная модель «AS-IS»;
3. Выполнена проектировка функциональной модели ИС и разработана диаграмма «TO-BE»;
4. Выполнена проектировка информационной модели ИС и разработаны UML-диаграммы;
5. Выполнена проектировка интерфейса;
6. Разработана база данных для ИС;
7. Разработан программный модуль для генерации тестов;
8. Проведено тестирование ПП;
9. Рассчитана экономическая эффективность.

В ходе выполнения данного дипломного проекта были решены все поставленные задачи.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Анализ предметной области [Электронный ресурс] URL: https://studfile.net/preview/1644616/page:3/ (дата обращения 31.05.2023).
2. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: учебник / А.М.Вендров. – М: МЭСИ, 2018. – 352 с.
3. Внуков, А. А. Основы информационной безопасности: защита информации : учебное пособие для среднего профессионального образования / А. А. Внуков. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 161 с. — (Профессиональное образование). — ISBN 978-5-534-13948- 8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495525 (дата обращения: 31.05.2023).
4. Волкова, А.Н. Функциональная модель: научная статья / А. Н. Волкова // Свободная энциклопедия, 2018.
5. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019 — 513 с. — (Профессиональное образование). — ISBN 978-5-534-11625-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/445766
6. ГОСТ 34.201-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
7. Грекул В.И. Проектирование информационных систем. Практикум. М.:Национальный открытый университет «ИНТУИТ», 2018 — 186 с.
8. Диаграмма вариантов использования [Электронный ресурс] URL: https://habr.com/ru/articles/566218/ (дата обращения 31.05.2023).
9. Диаграмма состояний [Электронный ресурс] URL: https://studfile.net/preview/2567489/page:10/ (дата обращения 31.05.2023).
10. Зенков, А. В. Информационная безопасность и защита информации : учебное пособие для вузов / А. В. Зенков. — Москва : Издательство Юрайт, 2022. — 104 с. — (Высшее образование). — ISBN 978- 5-534-14590-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/497002 (дата обращения: 31.05.2023).
11. Казарин, О. В. Надежность и безопасность программного обеспечения : учебное пособие для вузов / О. В. Казарин, И. Б. Шубинский. — Москва : Издательство Юрайт, 2022. — 342 с. — (Высшее образование). — ISBN 978-5-534-05142-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/493262 (дата обращения: 31.05.2023).
12. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2022. — 312 с. — (Высшее образование). — ISBN 978-5-9916-9043-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/491249 (дата обращения: 31.05.2023).
13. Лучшие IDE для разработки на C# [Электронный ресурс] URL: https://gb.ru/posts/c\_sharp\_ides (дата обращения 31.05.2023).
14. Маклаков С.В. «Моделирование бизнес-процессов с BPwin 4.0»: ДИАЛОГ-МИФИ. – 2018 - 224c
15. Нестеров, С. А. Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва : Издательство Юрайт, 2019 — 230 с. — (Профессиональное образование). — ISBN 978-5-534- 11629-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio- online.ru/bcode/445770
16. Обзор тестовых наборов и комплектов тестов [Электронный ресурс] URL: https://www.ibm.com/docs/ru/elm/6.0.2?topic=suites-test-case-test-suite-overview (дата обращения: 31.05.2023).
17. Описание среды разработки Microsoft Visual Studio [Электронный ресурс] URL: https://studbooks.net/2258619/informatika/opisanie\_sredy\_razrabotki\_microsoft\_visual\_studio (дата обращения 31.05.2023).
18. Определение себестоимости создания ПП [Электронный ресурс] URL: https://studentopedia.ru/informatika/opredelenie-sebestoimosti-sozdaniya-pp--opredelenie-optovoj-i-otpusknoj-ceni-pp---mobilnaya-versiya.html (дата обращения 31.05.2023).
19. Отладка [Электронный ресурс] URL: https://blog.skillfactory.ru/glossary/otladka-debugging/ (дата обращения 31.05.2023).
20. РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов.
21. Репин, В. В. Методики анализа бизнес-процессов: учебное пособие / В. В. Репин. – Москва: Манн, Иванов и Фербер, 2013. – 160 с.
22. Романова, Ю. Д. Информационные технологии в менеджменте (управлении) : учебник и практикум для среднего профессионального образования / Ю. Д. Романова [и др.] ; под редакцией Ю. Д. Романовой. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 411 с. — (Профессиональное образование). — ISBN 978-5-534-13801-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495013 (дата обращения: 31.05.2023).
23. Советов, Б. Я. Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019 — 420 с. — (Профессиональное образование). — ISBN 978-5-534-09324-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/438438
24. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2019 — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/445776
25. Тестовый сценарий против тестового сценария [Электронный ресурс] URL: https://habr.com/ru/company/trinion/blog/245615/ (дата обращения: 31.05.2023).
26. Техническая документация Майкрософт [Электронный ресурс] URL: https://docs.microsoft.com/ru-ru/ (дата обращения 31.05.2023).
27. Требования к программному обеспечению [Электронный ресурс] URL: https://gos-it.fandom.com/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\_%D0%BA\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83\_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E (дата обращения: 31.05.2023).
28. Учебное пособие по диаграммам последовательностей: полное руководство с примерами [Электронный ресурс] URL: https://creately.com/blog/ru/%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0/%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5-%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B5-%D0%BF%D0%BE-%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B9/ (дата обращения 31.05.2023).
29. Фундаментальная теория тестирования [Электронный ресурс] URL: https://habr.com/ru/articles/549054/ (дата обращения 31.05.2023).
30. Чистов, Д. В. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общей редакцией Д. В. Чистова. — Москва : Издательство Юрайт, 2022. — 258 с. — (Профессиональное образование). — ISBN 978-5-534-03173-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/491568 (дата обращения: 31.05.2023).
31. Язык программирования C#: краткая история, возможности и перспективы [Электронный ресурс] URL: https://timeweb.com/ru/community/articles/chto-takoe-csharp (дата обращения 31.05.2023).

# ПРИЛОЖЕНИЕ А

**Техническое задание на разработку прикладного приложения**

**ДЕСКТОПНОЕ ПРИЛОЖЕНИЕ «GenTest»**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**На 6 листах**

**Действует с 09.01.2023**

ЗАКАЗЧИК: Колледж профессионального образования при ПГНИУ

2023

**1. Общие сведения**

Наименование системы

Десктопное приложение для обучающихся колледжа ПГНИУ – «GenTest».

Краткое наименование – «GenTest».

Наименование разработчика и заказчика системы

Разработчик системы – студентка Жилина А.А., г. Пермь.

Заказчик системы – КПО ПГНИУ, г. Пермь.

Сроки начала и окончания работ

Начало работ – 9 января 2023 г.

Окончание работ – 31 мая 2023 г.

Сведения об источниках и порядке финансирования работ

Источником финансирования является КПО ПГНИУ.

Порядок финансирования определяется условиями предприятия ПГНИУ.

**1. Порядок оформления и предъявления заказчику результатов работ по созданию системы.**

Прикладное приложение передается Заказчику в виде архива, в котором хранится загрузчик приложения, а так же дамп базы данных.

**2. Назначение и цели создания системы**

2.1 Назначение системы

Программа «GenTest» осуществляет генерацию различных вариантов тестов, а также проводит тестирование студентов.

2.2 Цели создания системы

Внедрение прикладного приложения должно повысить эффективность проведения тестов, уменьшить возможность списывания студентами, повысить продуктивность работы преподавателя, а также дать возможность вышестоящим лицам контроллировать проведение контрольных мероприятий преподавателем.

**3. Характеристика объектов автоматизации**

В качестве объекта автоматизации выступает процесс, связанный с составлением и проверкой многовариантных контрольных работ.

**4. Требования к системе**

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию системы;

Функции:

Авторизация в системе;

Проведение тестирования;

Выведение результатов тестирования;

Возможность просмотра времени проведения тестирования.

Структура:

Прикладное приложение представляет собой программу, в которой студент может проходить тестирование, преподаватель вводит в базу по возможности новые задания, также вышестоящие лица имеют доступ ко времени проведения тестирований преподавателями.

4.1.1.1 Требования к способам и средствам связи для информационного обмена между компонентами системы.

В качестве основного средства связи между базой данных и приложением должна быть использована проводная или беспроводная сеть.

4.1.1.2 Требования к режимам функционирования системы.

Система должна функционировать непрерывно и круглосуточно при соблюдении соответствующих регламентирующих документов.

4.1.1.3. Перспективы развития, модернизации системы.

Приложение должно иметь перспективы развития, то есть создаваться с учетом возможности модернизации путем дополнения и обновления функций приложения и настройкой имеющихся средств.

4.1.2 Требования к численности и квалификации персонала системы и режиму его работы;

Эксплуатационный персонал приложения должен состоять из преподавателей и студентов.

4.1.3 Требования к надежности.

Должны быть предусмотрены программные средства резервного копирования и восстановления ПО Системы. Минимальная частота полного резервного копирования БД – раз в неделю.

4.1.5 Требования к эргономике и технической эстетике.

Расположение, цвет и яркость элементов интерфейса должны соответствовать зрительным и психофизическим возможностям человека. Надписи должны быть хорошо обозримы.

4.1.6 Требования к защите информации от несанкционированного доступа.

Организационные и технические меры защиты информации, реализуемые в приложении, должны обеспечивать:

 авторизацию пользователей;

 информирование и обучение персонала;

 анализ угроз безопасности информации и рисков от их реализации.

4.1.7 Технические требования к системе.

Приложение должно безопасно и правильно функционировать на ОС Windows 7 и выше.

4.2 Требования к функциям (задачам), выполняемым системой

Система должна обеспечивать выполнение следующих функций:

 Авторизация пользователей;

 Информирование пользователей в реальном времени.

**5. Состав и содержание работ по созданию системы**

Состав и содержание работ по созданию системы включают следующие этапы:

Предпроектное обследование, сбор необходимой информации. Результат: определение целей, задач системы, которые в дальнейшем должны быть решены;

Анализ предметной области. Результат: подробный анализ системы и введение организационных требований к решению задач и целей;

Разработка ТЗ. Результат: документация на разрабатываемую систему, в которой указаны сроки реализации, кем будет реализована, для кого, описаны все необходимые организационные требования к разработке;

Разработка модели программы. Результат: описание данных, определение связей между сущностями, построение концептуальной модели БД, построение логической модели БД;

Разработка рабочего проекта, состоящего из:

Установка платформы разработки;

Написания программы;

Отладка программы;

Корректировка программы;

Проведение тестирования и доработка программного обеспечения по замечаниям и предложениям;

Сдача системы в эксплуатацию с выпуском описания алгоритмов и документации.

Сопровождение системы и устранение возможных неисправностей.

**6. Порядок контроля и приемки системы**

6.1 Общие требования к приемке работ по стадиям.

Сдача-приемка осуществляется комиссией, в состав которой входят представители Заказчика и Исполнителя. По результатам приемки подписывается акт приемо-сдаточных работ.

6.2 Статус приемочной комиссии

Приемочная комиссия определяется Заказчиком.

**7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

В ходе выполнения проекта требуется выполнить работы по подготовке к вводу системы в действие. Необходимо выполнить следующие работы:

Определить подразделения и ответственных должностных лиц за внедрение;

Обеспечить обучение персонала при работе с системой;

Обеспечить выполнение требований, предъявляемых к программно-техническим средствам.

**8. Требования к документированию**

Требования к документации определяются в соответствии ГОСТ 34.201-89 и РД 50-34.698-90

# ПРИЛОЖЕНИЕ Б

Рисунки меню СА:



Рисунок Б 1 - Редактирование данных преподавателя

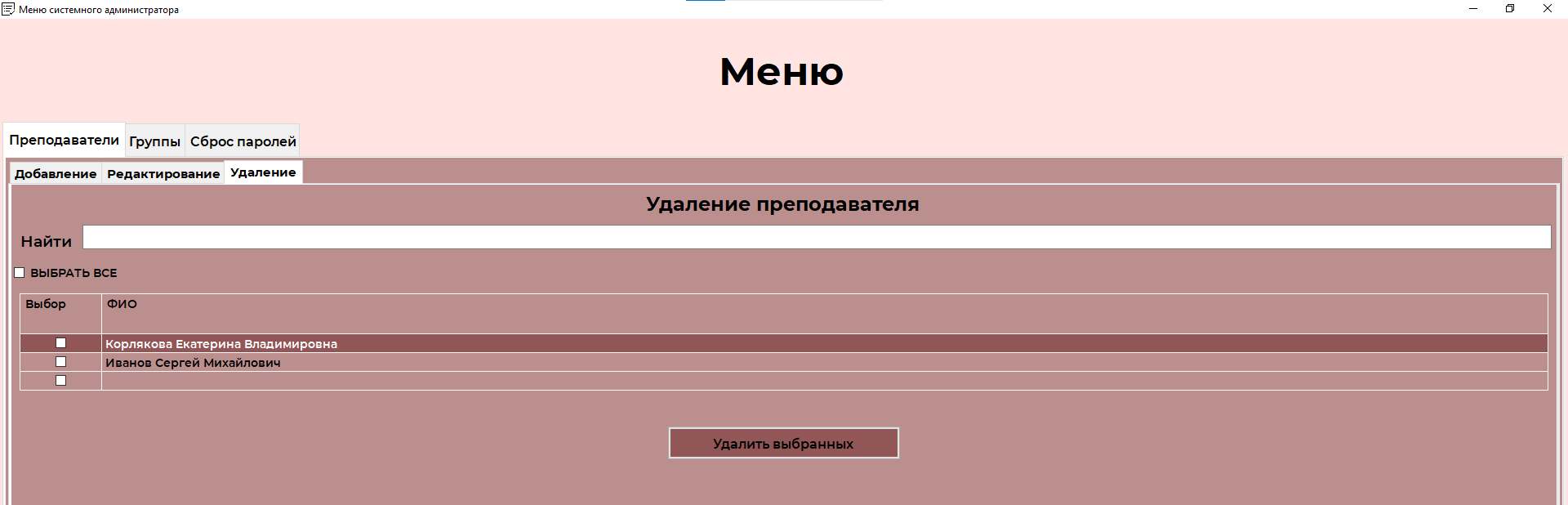


Рисунок Б 2 - Удаление преподавателей



Рисунок Б 3 - Действия с группами

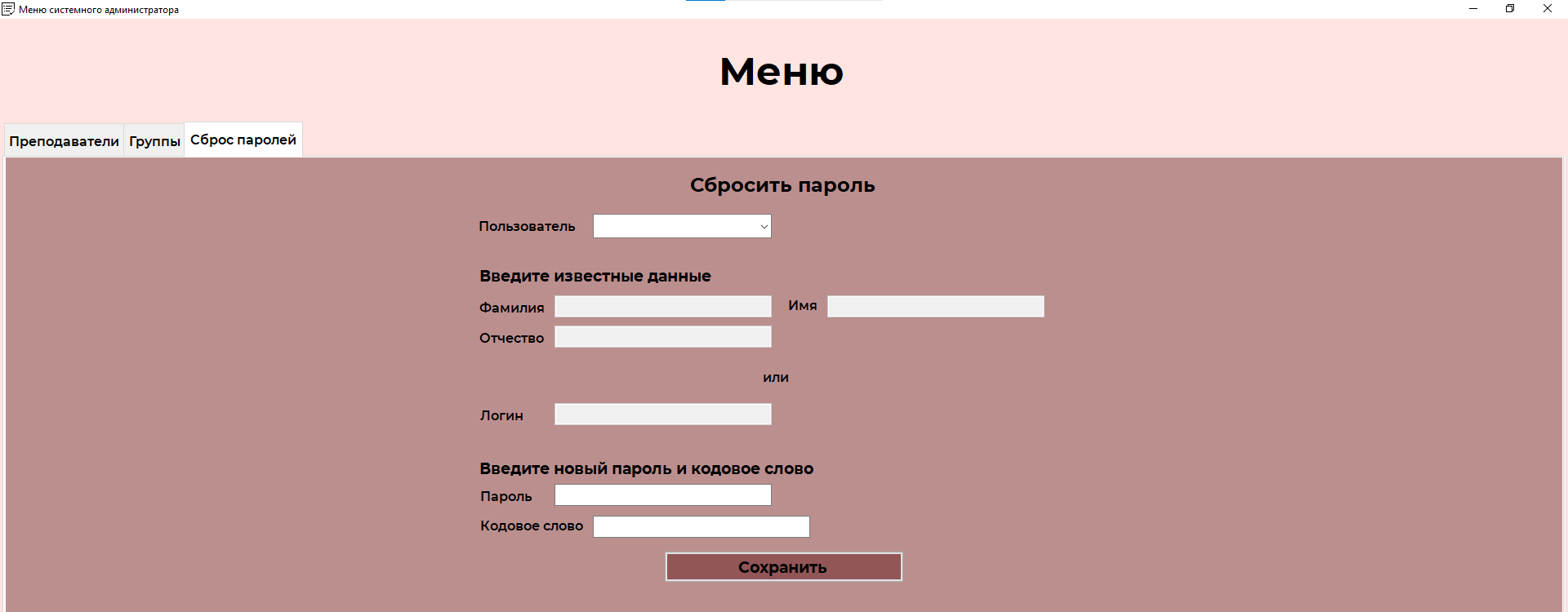


Рисунок Б 4 - Сброс паролей

Рисунки меню преподавателя:

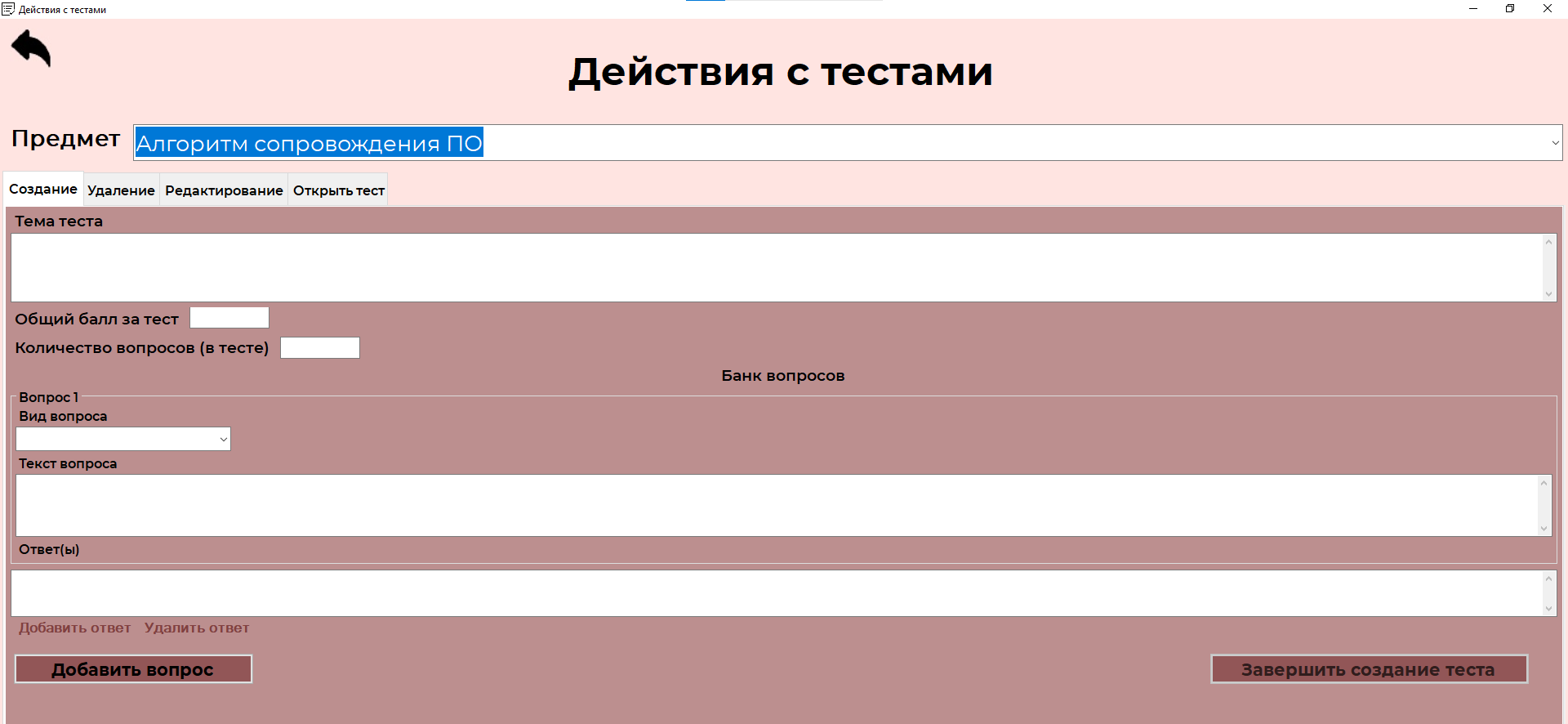


Рисунок Б 5 – Действия с тестами. Создание

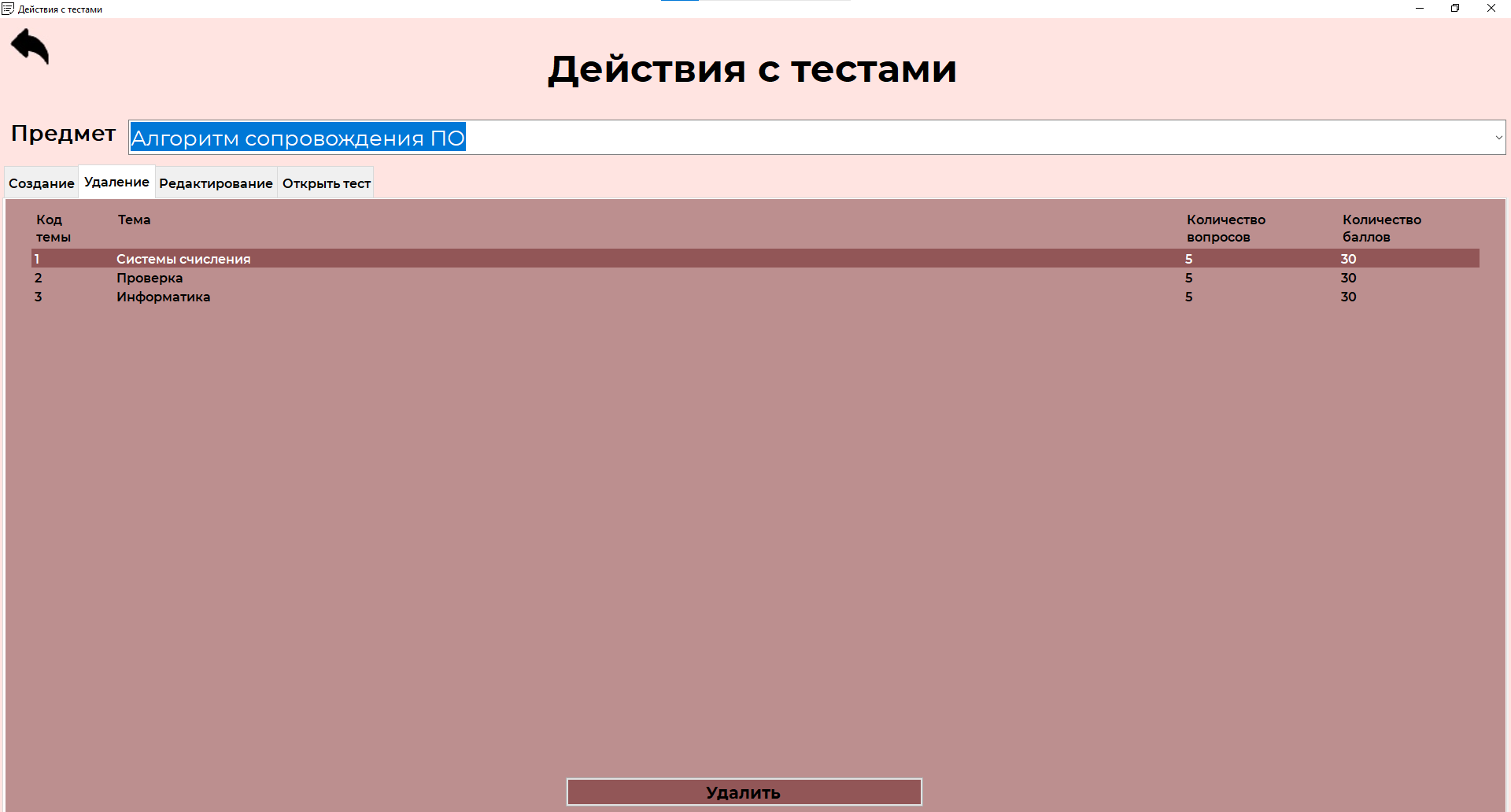


Рисунок Б 6 - Действия с тестами. Удаление

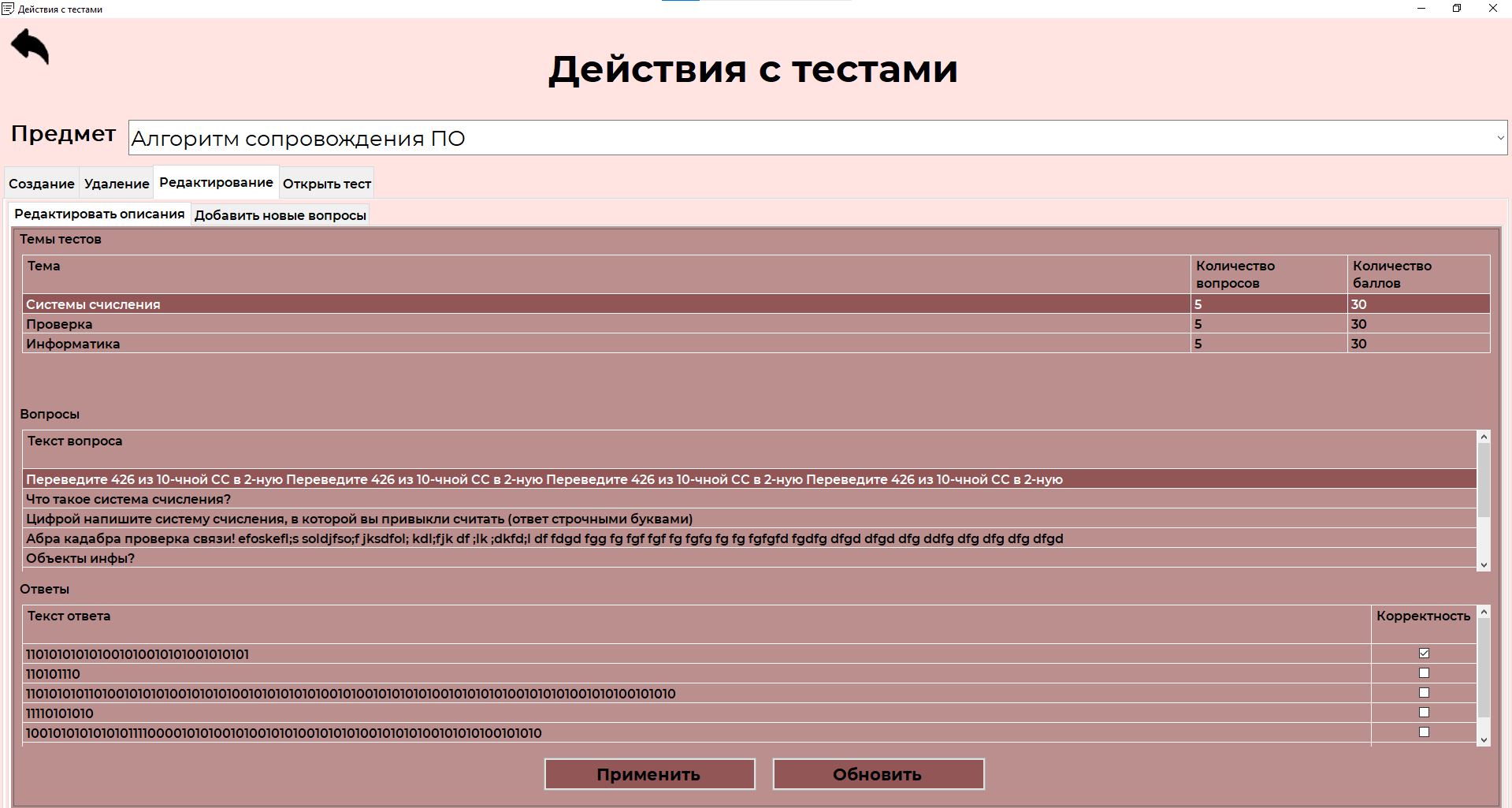


Рисунок Б 7 - Действия с тестами. Редактирование

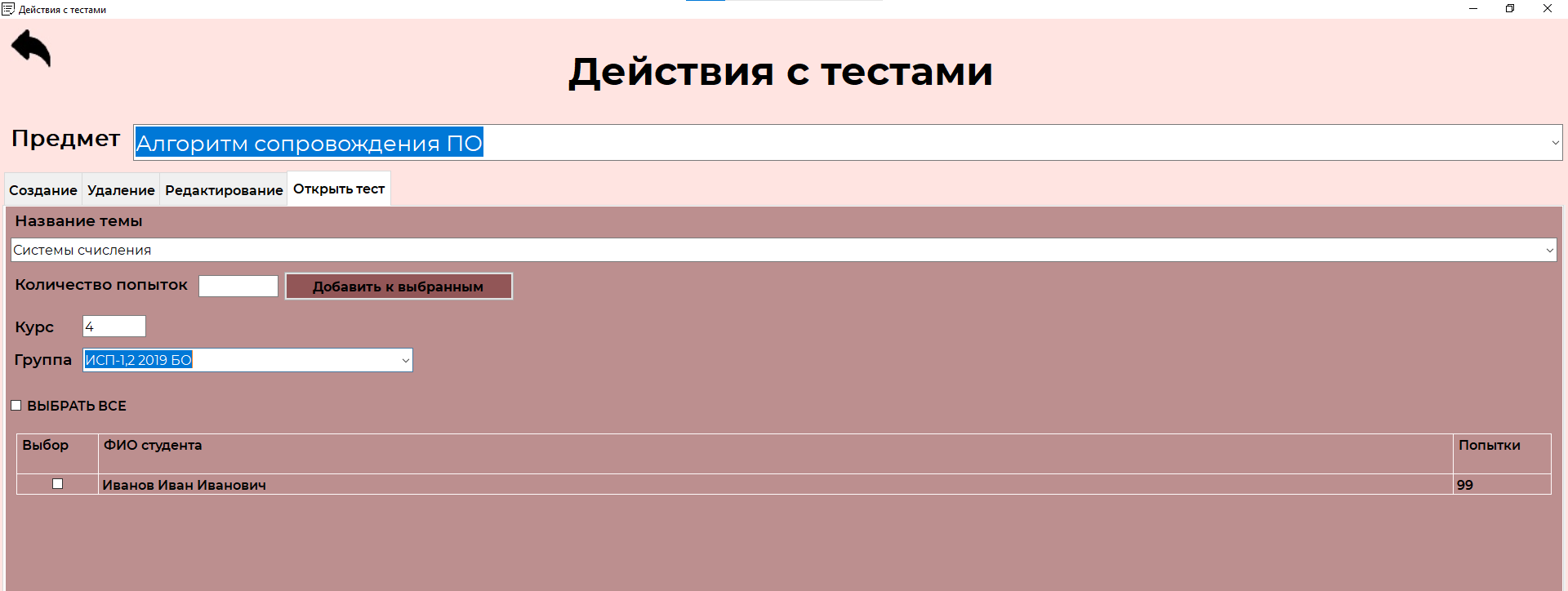


Рисунок Б 8 - Действия с тестами. Открыть тест

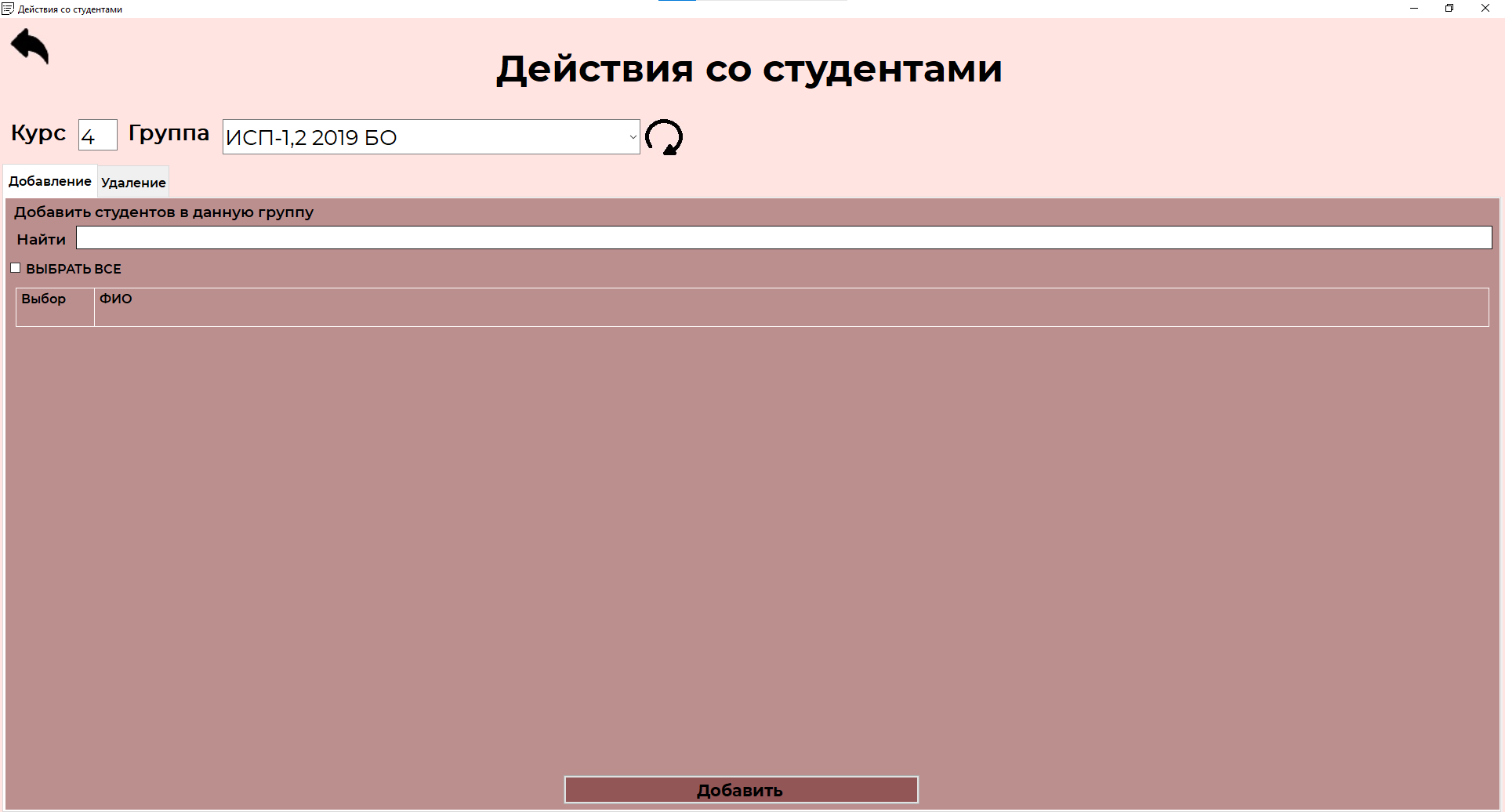


Рисунок Б 9 - Действия со студентами. Добавление

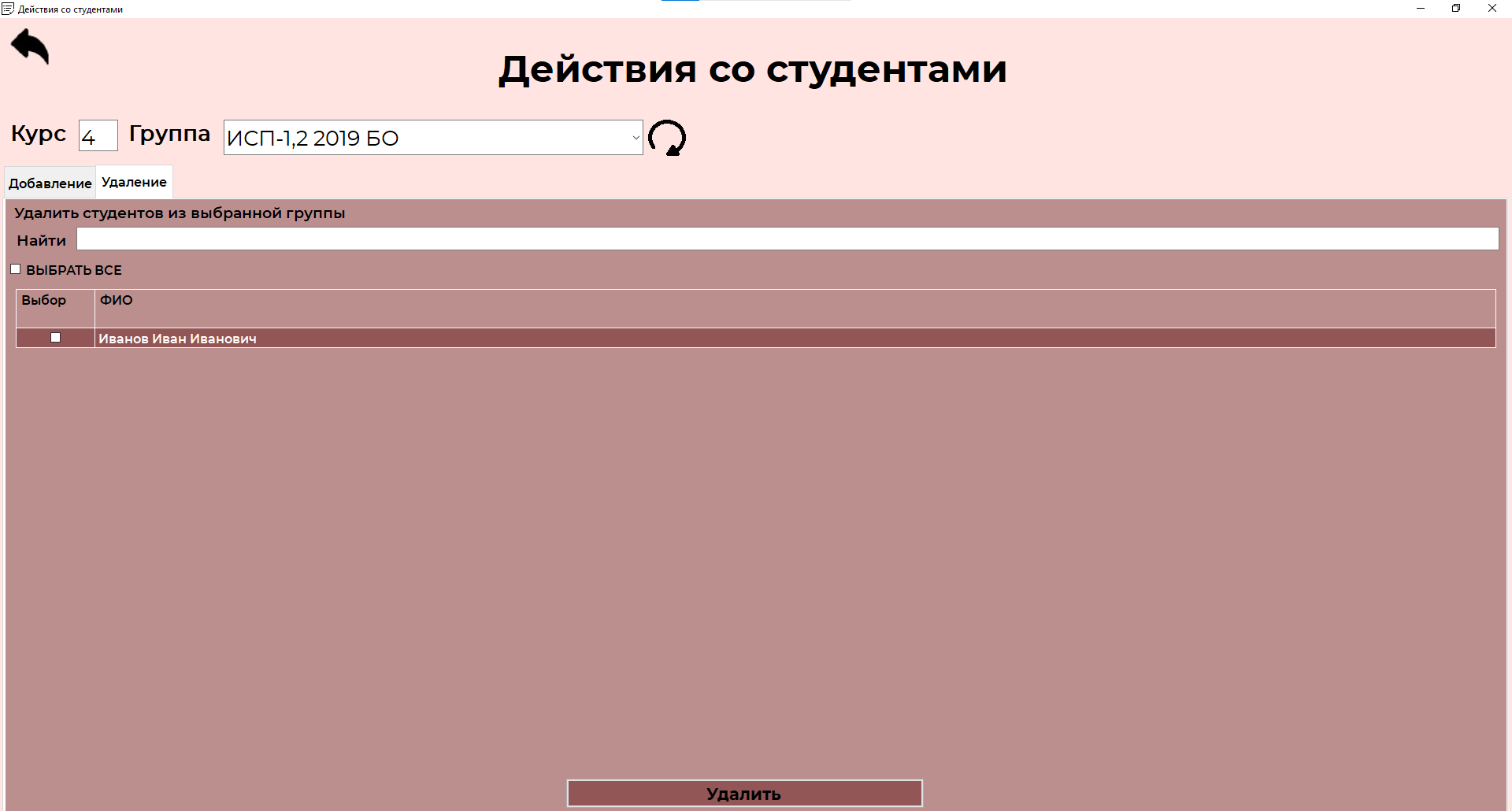


Рисунок Б 10 - Действия со студентами. Удаление

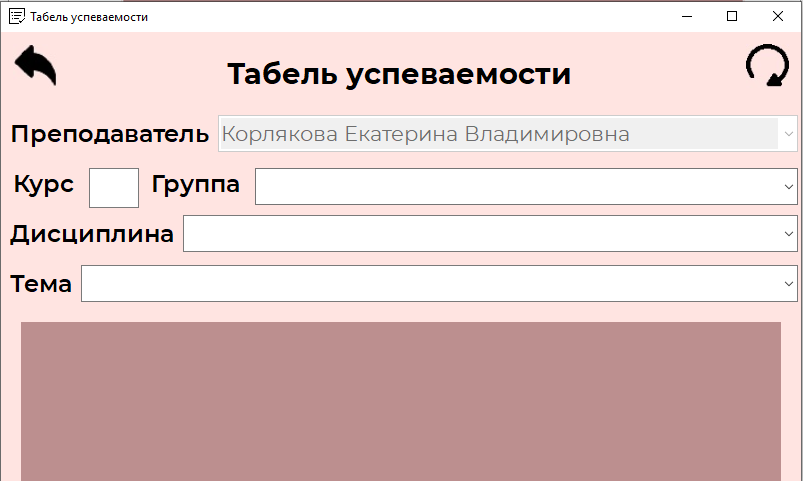


Рисунок Б 11 - Табель успеваемости

Рисунки меню студента:

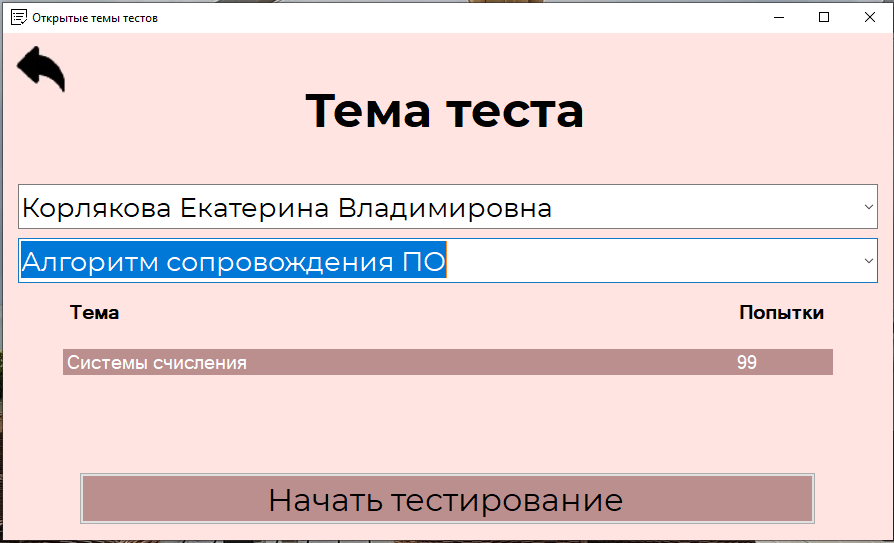


Рисунок Б 12 – Форма тестирование

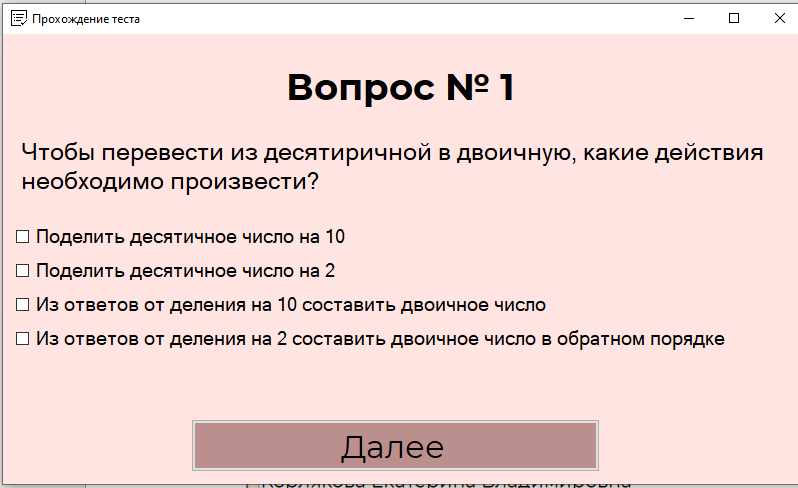


Рисунок Б 13 - Форма прохождения теста

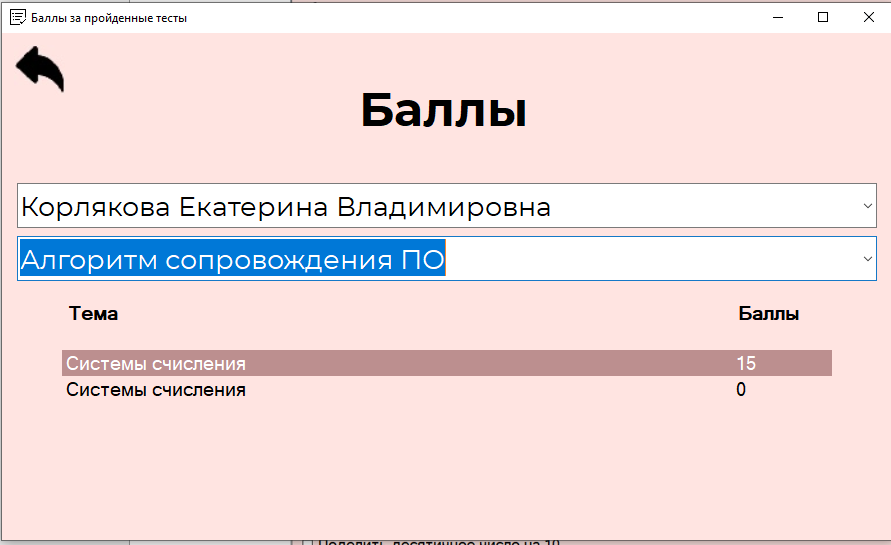


Рисунок Б 14 - Баллы за пройденные тесты

# ПРИЛОЖЕНИЕ В

Таблица В 1 - Тест-кейс работы авторизации

|  |  |
| --- | --- |
| **Номер** | 1 |
| **Заголовок** | Тест-кейс работы авторизации |
| **Предусловие** | Запущено приложение ГенТест |
| **Шаг** | **Ожидаемый результат** |
| Ввод пароля | Символы помечаются звездочками |
| Нажатие на рисунок глазика рядом с вводом пароля | С символов пароля снялась шифровка звездочками |
| Нажатие на кнопку «Регистрация (только для студентов)» | Открытие формы «Регистрация» |
| Нажатие на кнопку «Забыли пароль» | Открытие формы «Сброс пароля» |
| Ввод неверного логина и/или пароля и попытка войти в систему | Запрет н вход и появление окошка с предупреждением о неверно введенных данных |
| Ввод верного логина и/или пароля и попытка войти в систему | Открытие соответствующей формы пользователя (студента, преподавателя, СА или администрации) |
| Вход в систему с выключенным XAMPP | Ошибка с предупреждением о выключенном XAMPP |
| Успешный вход | Логин и пароль стираются, но форма остается работать |

Таблица В 2 - Тест-кейс формы меню СА

|  |  |
| --- | --- |
| **Номер** | 2 |
| **Заголовок** | Тест-кейс работы меню СА |
| **Предусловие** | Открыта форма меню СА |
| **Шаг** | **Ожидаемый результат** |
| Нажатие поочередно на вкладки «Преподаватели», «Группы», «Сброс паролей» | Открытие соответствующих вкладок |

Продолжение таблицы В2

|  |  |
| --- | --- |
| Нажатие во вкладке «Преподаватели» вкладок «Добавление», «Редактирование», «Удаление» | Открытие соответствующих вкладок |
| Нажатие на «+» рядом с введением дисциплины | Появление аналогичного поля для ввода дисциплины |
| На вкладке «Преподаватели»-«Добавление» ввод ФИО, логина, пароля, кодового слова, дисциплины (одного или нескольких полей для ввода) и нажатие кнопки «Добавить» | Появление окна с сообщением о успешно введенном преподаватели и очистка всех полей для ввода, а также, если были добавленны дисциплины больше, чем одна – удаление всех дисциплин до одной |
| Ввод текста в любое поле и нажатие на кнопку в форме обновления, расположенную в правом верхнем углу | Очистка всех полей для ввода |
| Попытка регистрации преподавателя без введения всех необходимых данных (ФИО, логин, пароль, кодовое слово, дисциплины) | Появление окна с предупреждением о не всех введенных данных |
| На вкладке «Преподаватели»-«Редактирование» открыть выборку ФИО | Выведение списка всех преподавателей |
| Выбор одного из существующих в БД преподавателей (то есть выбор из списка) | Заполнение полей ФИО в соответствии с выбранным преподавателем, выведение ведущих им дисциплин |
| Ввод в соседнее поле рядом с фамилией, именем и отчеством нового значения этих данных и нажатие на кнопку «Редактирвание» рядом с полем, где меняем | Меняются соответствующие поля напротив, а также поле с полным ФИО |
| Выбор преподавателя, запись новых дисциплин и нажатик на кнопку «сохранить» | Сохранение в БД новых дисциплин, которые ведет данный преподаватель |
| Выбор преподавателя, выбор дисциплин, нажатие на кнопку «удалить» | Удаление из БД дисциплин, которые ведет данный преподаватель |

Продолжение таблицы В2

|  |  |
| --- | --- |
| Ввод данных преподавателя в поле для ввода рядом с текстом «Найти» | Поиск и выведение в таблицу найденных в БД преподавателей с похожими введенными данными |
| Выбор из таблицы преподавателей с помощью постановки галочки рядом с ФИО и нажатие на кнопку «удалить» | Удаление из БД выбранных преподавателей |
| На вкладке «Группы» после заголовка «Добавить группу» ввод курса и названия группы, нажатие на кнопку «добавить» | Если курс – число, то появление окна о успешном добавлении группы, иначе – об ошибке. |
| На вкладке «Группы» после заголовка «Редактировать группу» введение в поле для ввода номера курса | Появление выборки названий групп в соответствии с введенным курсом |
| Введение в соответствующие поля нового курса и нового названия группы и нажатие на кнопку «редактировать» | Изменение в БД соответствующих данных |
| На вкладке «Группы» после заголовка «Удалить группу» введение в поле для ввода номера курса | Появление выборки названий групп в соответствии с введенным курсом |
| Введение названия группы из выборки и нажатие на кнопку «удалить» | Удаление из БД данной группы |
| На вкладке «Сброс паролей» введение пользователя из выпадающего списка | Активация полей ФИО и г.р. |
| Заполнение всех полей соответствующими данными и нажатие на кнопку «Сохранить» | Сброс пароля и сохранение новых данных пользователя |
| Заполнение НЕ всех полей соответствующими данными и нажатие на кнопку «Сохранить» | Выведение сообщения об ошибке |

Таблица В 3 - Тест-кейс формы меню преподавателя

|  |  |
| --- | --- |
| **Номер** | 3 |

Продолжение таблицы В3

|  |  |
| --- | --- |
| **Заголовок** | Тест-кейс работы меню преподавателя |
| **Предусловие** | Открыта форма меню преподавателя |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Тесты» | Открытие формы «Действия с тестами» |
| Нажатие на кнопку «Студенты» | Открытие формы «Действия со студентами» |
| Нажатие на кнопку «Табель успеваемости» | Открытие формы «Табель успеваемости» |

Таблица В 4 - Тест-кейс формы "Действия с тестами"

|  |  |
| --- | --- |
| **Номер** | 4 |
| **Заголовок** | Тест-кейс работы формы «Действия с тестами» |
| **Предусловие** | Открыта форма меню «Действия с тестами» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню преподавателя |
| Просмотр выпадающего списка напротив надписи «предмет» | Выведение списка с предметами, которые преподает данный преподаватель |
| На вкладке «создание» введение темы теста, общего балла за тест, количества вопросов в тесте, введение вида, текста первого вопроса, ответов на него (для каждого из ответов: введение текста ответа в поле над кнопкой «добавить ответ» и нажатие на нее) и отметка верного(ых) вопроса(ов). Затем нажать на кнопку «Добавить вопрос» | Если тест не был создан, то выведется сообщение с вопросом подтверждения. После подтверждения, если все необходимые данные для создания вопроса есть, то обновятся все поля для создания второго вопроса и так после каждого. Если нет, то выведутся сообщения с соответствующими ошибками. Поля для ввода темы теста, общего балла, количества вопросов в тесте становятся неактивными |

Продолжение таблицы В4

|  |  |
| --- | --- |
| Ввести в поле под ответами ответ, который необходимо удалить и нажать на кнопку «Удалить ответ» | Удаление необходимого ответа |
| Создание минимального количества вопросов | Кнопка «завершить создание теста» становится активной |
| Нажатие на кнопки «Завершить создание теста» | Все поля для ввода становятся пустыми. Неактивные поля становятся активными и готовыми для создания нового теста |
| Выбрать из списка предмет, на вкладке «Удаление» выбрать нужные темы и нажать на кнопку «Удалить» | Удаление выбранных тем из БД |
| На вкладке «Редактирование» - «Редактировать описания» есть возможность выбрать тему тестов, изменить ее название, в соответствии с ней название вопросов, а в соответствии с каждым вопросом – названий ответов. После изменения тем/вопросов/ответов нажать на кнопку «Применить» | Внесение изменений в названии тем/вопросов/ответов в БД |
| Нажатие на кнопку «Обновить» | Сброс выбранных тем |
| На вкладке «Редактирование» - «Добавить новые вопросы» при выбранном предмете открыть выборку тем тестов и выбрать нужный из них. Далее аналогично с созданием теста при введении вида вопроса, текста вопроса и ответов нажатие на кнопку «Добавить вопрос» | Добавление вопроса к нужной теме |
| На вкладке «Открыть тест» выбрать курс | Появление в соответствии с курсом выборки групп |
| Выборать группу | Появление в таблице списка студентов данной группы |
| Нажать на «ВЫБРАТЬ ВСЕ» | Выбор всех студентов из группы |

Продолжение таблицы В4

|  |  |
| --- | --- |
| При выбранном предмете из предложенных названий тем выбрать нужную. Напротив количества попыток вписать количество, которое необходимо добавить, выбрать нужных студентов и нажать на кнопку «Добавить к выбранным» | Добавление количества попыток прохождения данного теста данным студентам |

Таблица В 5 - Тест-кейс формы «Действия со студентами»

|  |  |
| --- | --- |
| **Номер** | 5 |
| **Заголовок** | Тест-кейс работы формы «Действия со студентами» |
| **Предусловие** | Открыта форма «Действия со студентами» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню преподавателя |
| Ввод курса | Появление в соответствии с курсом выборки групп |
| Ввод группы из списка | Появление в таблице списка студентов данной группы |
| На вкладке «Добавление» выбрать необходимых студентов и нажать на кнопку «Добавить» | Присвоение выбранной группы выбранным студентам |
| На вкладке «Удаление» выбрать необходимых студентов и нажать на кнопку «Удалить» | Удаление выбранных студентов из выбранной группы |
| При необходимости (на обеих вкладках) ввести в строку «Найти» ключевых слов для поиска нужных студентов | Выборка из БД нужных студентов в соответствии с введенными данными |
| Нажатие на кнопку «Обновление» | Обновление таблиц со студентами |

Таблица В 6 - Тест-кейс формы "Табель успеваемости" из меню преподавателя

Продолжение таблицы В6

|  |  |
| --- | --- |
| **Номер** | 6 |
| **Заголовок** | Тест-кейс работы формы «Табель успеваемости» |
| **Предусловие** | Открыта форма «Табель успеваемости» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню преподавателя |
| Ввод курса | Появление в соответствии с курсом выборки групп |
| Ввод дисциплины из списка | Появление в соответствии с дисциплиной выборки тем |
| Ввод группы и темы из списка | Выведение списка успеваемости студентов из данной группы по данной теме |
| Нажатие на кнопку «Обновление» | Очистка всех полей |

Таблица В 7 - Тест-кейс формы меню студента

|  |  |
| --- | --- |
| **Номер** | 7 |
| **Заголовок** | Тест-кейс работы формы меню студента |
| **Предусловие** | Открыта форма меню студента |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Тестирование» | Открытие формы «Тестирование» |
| Нажатие на кнопку «Просмотр баллов» | Открытие формы «Просмотр баллов» |

Таблица В 8 - Тест-кейс формы "Открытые темы тестов "

|  |  |
| --- | --- |
| **Номер** | 8 |
| **Заголовок** | Тест-кейс работы формы «Открытые темы тестов» |
| **Предусловие** | Открыта форма «Открытые темы тестов» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню студента |
| Из списка преподавателей выбрать преподавателя | Появление списка предметов, которые ведет данный преподаватель |

Продолжение таблицы В8

|  |  |
| --- | --- |
| Выбор предмета из списка | Появление таблицы тем и оставшееся количество попыток их прохождения |
| Выбор темы из списка и нажатие на кнопку «Начать тестирование» | Переход на форму «Тестирование» |

Таблица В 9 - Тест-кейс формы " Тестирование"

|  |  |
| --- | --- |
| **Номер** | 9 |
| **Заголовок** | Тест-кейс работы формы «Тестирование» |
| **Предусловие** | Открыта форма «Тестирование» |
| **Шаг** | **Ожидаемый результат** |
| Выбор верного(ых) ответа(ов) и нажатие на кнопку «Далее» | Смена номера вопроса n на n+1, появление нового вопроса |
| Выбор верного(ых) ответа(ов) и нажатие на кнопку «Далее» или (после введения ответов на последний вопрос) «Завершить тестирование» | Появление окна сообщения о прохождении теста и открытие меню студента |

Таблица В 10 - Тест-кейс формы "Просмотр баллов"

|  |  |
| --- | --- |
| **Номер** | 10 |
| **Заголовок** | Тест-кейс работы формы «Просмотр баллов» |
| **Предусловие** | Открыта форма «Просмотр баллов» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню студента |
| Из списка преподавателей выбрать преподавателя | Появление списка предметов, которые ведет данный преподаватель |
| Выбор предмета из списка | Появление таблицы тем и баллов за пройденный тест |

Таблица В 11 - Тест-кейс формы "Табель успеваемости" из меню администрации

Продолжение таблицы В11

|  |  |
| --- | --- |
| **Номер** | 11 |
| **Заголовок** | Тест-кейс работы формы «Табель успеваемости» |
| **Предусловие** | Открыта форма «Табель успеваемости» |
| **Шаг** | **Ожидаемый результат** |
| Нажатие на кнопку «Назад» | Открытие формы меню преподавателя |
| Ввод курса | Появление в соответствии с курсом выборки групп |
| Ввод преподавателя из списка | Появление выборки дисциплин |
| Ввод дисциплины из списка | Появление в соответствии с дисциплиной выборки тем |
| Ввод группы и темы из списка | Выведение списка успеваемости студентов из данной группы по данной теме |
| Нажатие на кнопку «Обновление» | Очистка всех полей |

# ПРИЛОЖЕНИЕ Г

**1. Руководство системного программиста**

1. **Общие сведения о программном продукте**

Программа «GenTest» будет использована в колледже профессионального образования при ПГНИУ и направлена на осуществление генерации различных вариантов тестов и прохождение тестирований студентами.

Программа требует объема оперативной памяти не менее 4Гб, 1Гб свободного места на диске, 32 или 64-разрядные системы: процессор с тактовой частотой 1,4 ГГц или выше. Программа ставится только на операционную систему Windows не менее 7 версии. Обязательно наличие СУБД MySQL. Выполнить вход в программу могут только зарегестрированные пользователи. Регистрацию преподавателей осуществляет системный администратор. Студенты регистрируются самостоятельно.

1. **Структура программы**

Программа на C# состоит из одного или нескольких файлов с исходным текстом. Каждый файл имеет расширение .CS и представляет собой класс.

Файлы-ресурсы для классов имеют расширения . resx.

Файлы, имеющие расширения .Designer.cs содержат информацию об элементах управления, размещенных на форме.

Файл с расширением .csproj хранит настройки проекта и ссылки на все ресурсы, используемые в проекте.

Файл .sln хранит всю информацию о решении (проекты, ссылки..).

Файлы с расширением .jpg и .ico – ресурсы приложения. Это картинки на формах и иконка самого приложения.

1. **Настройка программы**

Для начала работы программы необходимо установить на сервер БД с названием базы gentest.

1. **Проверка на тестовом примере**

Для каждого отдельного пункта и подпункта меню было выполнено несколько раз испытание на правильное функционирование. В ходе испытаний при корректном введении данных ошибок или сбоев найдено не было. При некорректном введении данных программа выдает сообщения об ошибках.

1. **Дополнительные возможности программы**

Дополнительных возможностей программы нет.

1. **Сообщения системному программисту**

Для работы приложения необходимо установить XAMPP на сервере через веб-интерфес PhpMyAdmin  создать БД под названием «gentest» c кодировкой utf8\_general\_ci, как представлено на рисунке Г1.



Рисунок Г - Создание БД

Затем необходимо импортировать в созданную БД файл с названием «gentest.sql», как представлено на рисунке Г2.

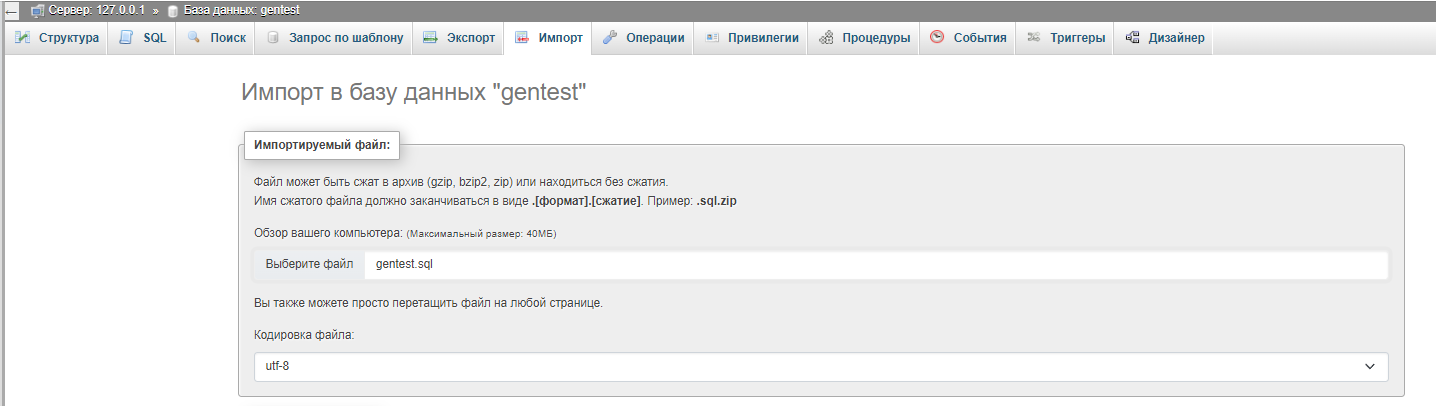


Рисунок Г - Импорт БД

После импорта БД необходимо изменить логин и пароль системного администратора и администрации в таблице «users», как показано на рисунке Г3. Пароль должен быть зашифрован с помощью алгоритма хэширования MD5.

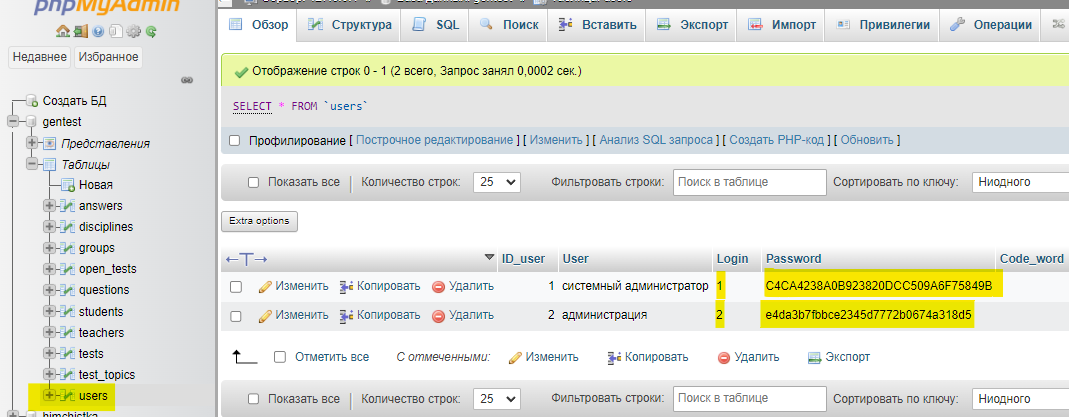


Рисунок Г - Редактирование данных для авторизации

Для корректного обращения к хосту, необходимо загрузить Visual Studio, открыть файл «Peremennye.cs» и изменить наименование хоста, как показано на рисунке Г4.

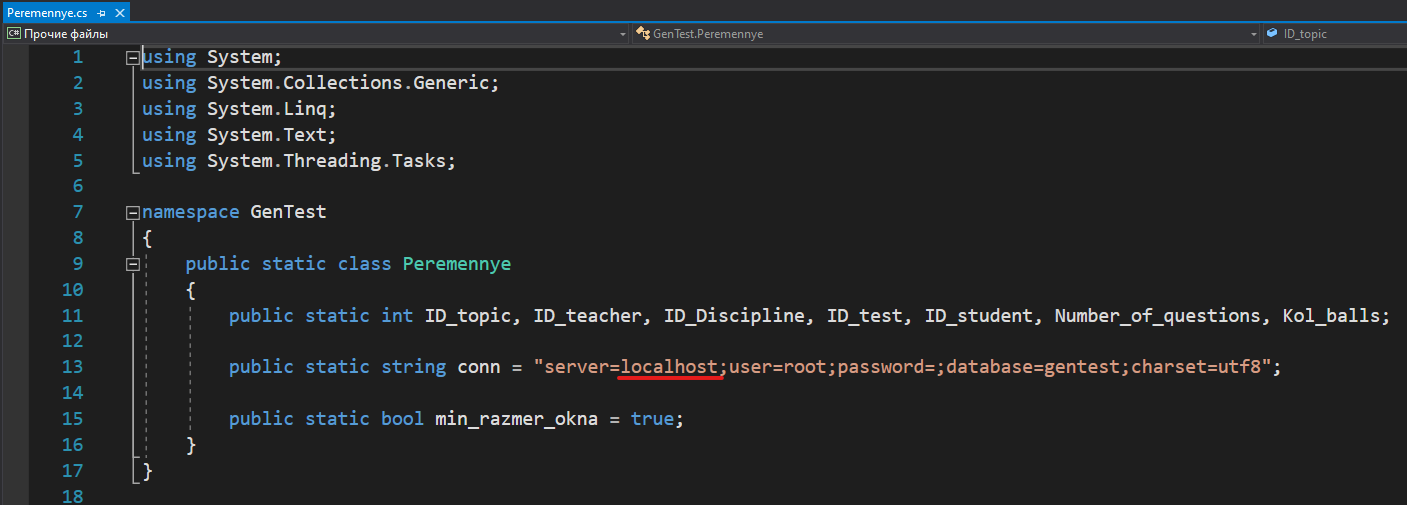


Рисунок Г - Редактирование хоста

Далее открыть файл «GenTest.sln» пересобрать решение, как показано на рисунке Г5.

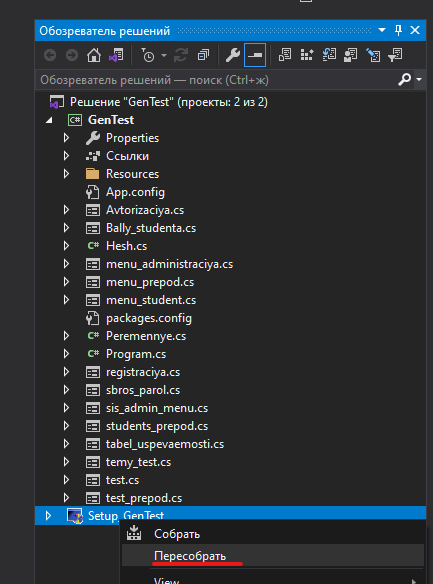


Рисунок Г - Создание установщика

Теперь для дальнейшей установки программы у нас есть установщик, который распологается по пути: …GenTest\Setup\_GenTest.

После создания БД и изменения приложение настраивать не нужно.

Программа устанавливается с помощью установщика.

Визуально работоспособность программы программист может проверить следующим способом:

* при запуске программы на экране монитора появляется главное окно программы;
* при щелчке по пунктам меню происходят соответствующие действия;
* выдаются все предупреждения и сообщения об ошибках.

**2. Руководство оператора**

* 1. **Общие сведения о программном продукте**

Программа «GenTest» будет использована в колледже профессионального образования при ПГНИУ и направлена на осуществление генерации различных вариантов тестов и прохождение тестирований студентами.

К основным функциональным требованиям системы можно отнести:

* Авторизация в системе;
* Создание тестовых заданий;
* Проведение тестирования;
* Выведение результатов тестирования;
* Возможность просмотра результатов и времени проведения тестирования.
  1. **Описание установки приложения**

Приложение устанавливается с помощью мастера установки. Заходим по пути …GenTest\Setup\_GenTest\Release, выбираем .exe.

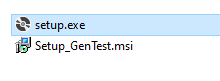


Рисунок Г 6 - Установщик ПО

Соглашаемся со всем, что предлагает установщик ПО. Таким образом, программа установится автоматически.

* 1. **Описание запуска приложения**

Приложение запускается двойным нажатием на его иконку. Иконка изображена на рисунке Г7.



Рисунок Г - Иконка приложения

* 1. **Описание пользовательского интерфейса**

**2.4.1 Авторизация**

При запуске приложения откроется окно авторизации, которое изображено на рисунке Г8.

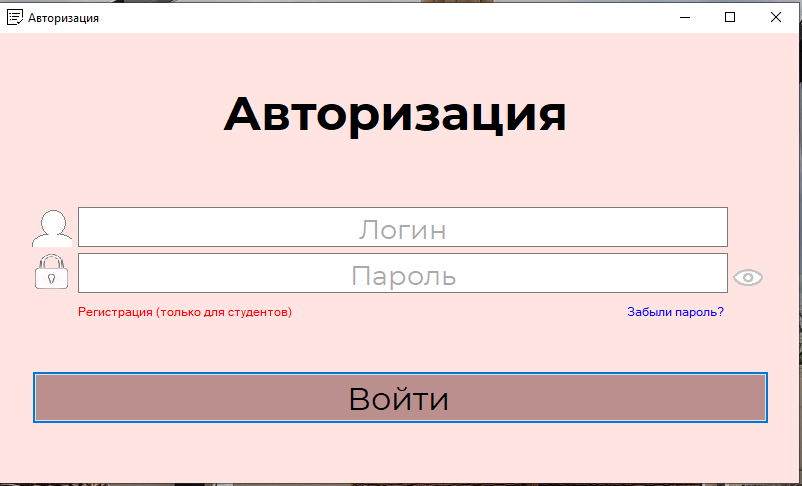


Рисунок Г - Авторизация

**2.4.2 Регистрация студентов**

Кликнув на кнопку «Регистрация» откроется окно, в котором студент может зарегистрироваться. Форма изображена на рисунке Г9. Здесь студент вводит все данные. Есть возможность просмотра введенного пароля. При наведении на кодовое слово выплывает подсказка о данных, которые желательно ввести.

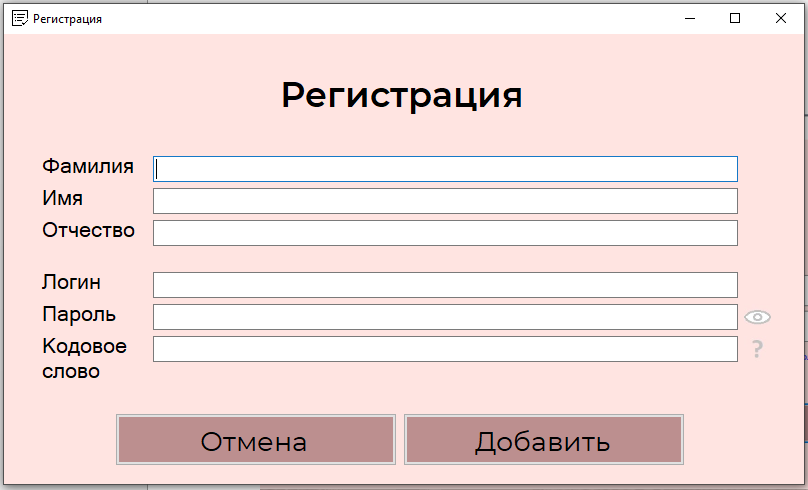


Рисунок Г - Регистрация студента

**2.4.3 Сброс паролей**

Кликнув на кнопку «Забыли пароль?» откроется окно для сброса пароля. Форма изображена на рисунке Г10. Пользователь вводит известные данные. Если не удается вспомнить кодовое слово, пользователю нужно обратиться к системному администратору. После обращения нужно обязательно вновь поменять пароль, чтобы ограничить доступ системного администратора к аккаунту пользователя.

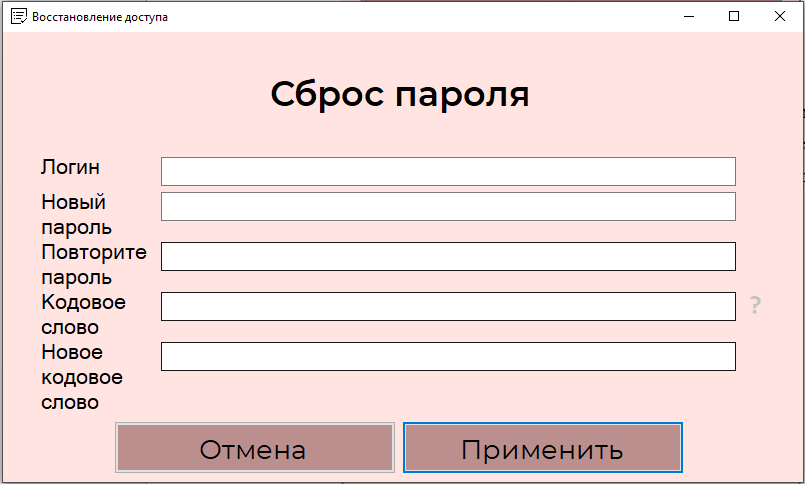


Рисунок Г - Сброс пароля

**2.4.4 Меню системного администратора**

Меню системного администратора содержит три вкладки: «Преподаватели», «Группы», «Сброс паролей». Первая вкладка в свою очередь делится еще на три вкладки: «Добавление», «Редактирование» и «Удаление» преподавателей. Меню изображено на рисунке Г11.

**2.4.4.1 Вкладка «Добавление» меню системного администратора**

Вкладка «Добавление» вкладки «Преподаватели» изображена на рисунке Г11. Во вкладке «Добавление» вкладки «Преподаватели» нужно заполнить все пустые поля. Если создаваемый преподаватель ведет несколько дисциплин, необходимо нажать на «+» рядом с добавлением дисциплин и в появившуюся строку ввести название дисциплины. При завершении ввода данных о преподавателе нажать на кнопку «Добавить». При необходимости очистки всех полей, можно обновить вкладку «Добавление», нажав на кнопку стрелочки в правом верхнем углу.

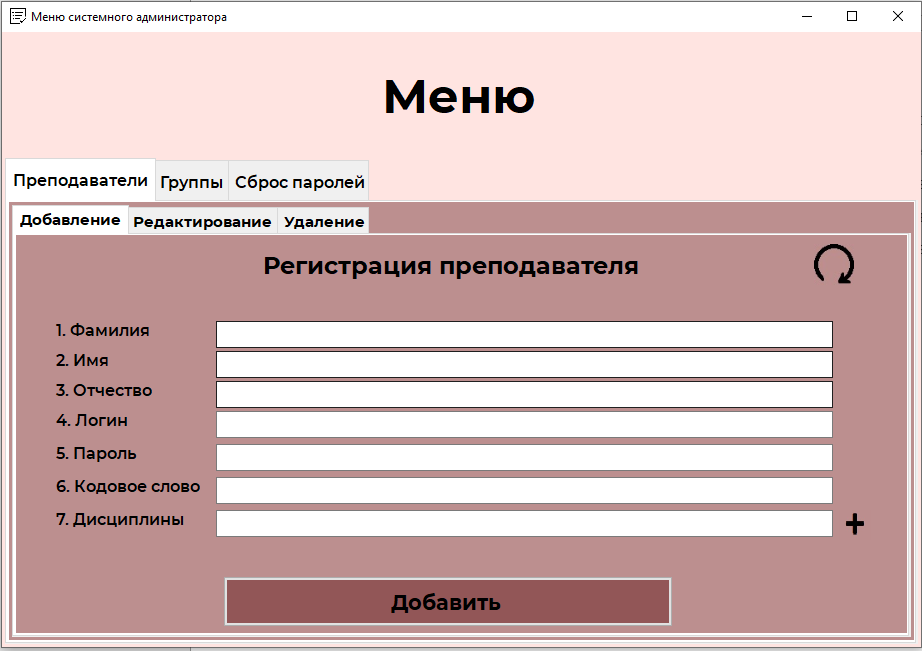


Рисунок Г - Меню системного администратора. Вкладка «Добавление» преподавателя

**2.4.4.2 Вкладка «Редактирование» меню системного администратора**

Вкладка «Редактирование» вкладки «Преподаватели» изображена на рисунке Г12. Во вкладке «Редактирование» из списка преподавателей в строке ФИО для редактирования его данных необходимо выбрать нужного. При редактировании ФИО напротив каждого из данных есть поле для ввода нового значение. При сохранении нажать «Редактировать», также расположенного рядом. Есть возможность редактировать имеющиеся, добавить или удалить дисциплины. Для сохранения изменений дисциплин (при добавлении или редактировании) внизу формы нажать «Сохранить». Для удаления дисциплин поставить галочки напротив нужных и нажать внизу формы кнопку «Удалить».

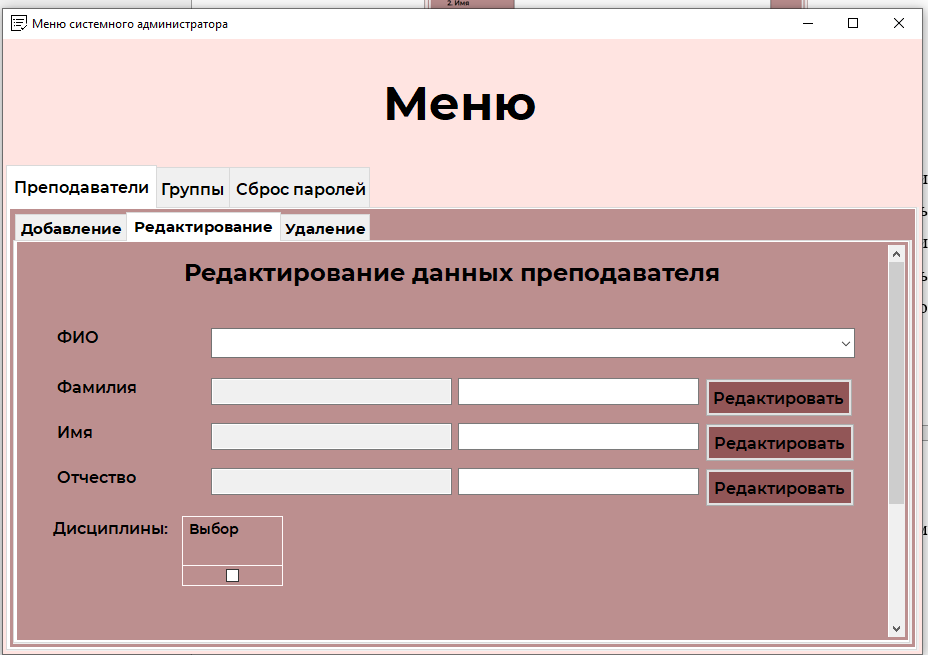


Рисунок Г - Меню системного администратора. Вкладка "Редактирование" преподавателя

**2.4.4.3 Вкладка «Удаление» меню системного администратора**

Вкладка «Удаление» вкладки «Преподаватели» изображена на рисунке Г13. Для быстрого поиска преподавателя, можно ввести его ФИО в поле для поиска, рядом с надписью «Найти». Для удаления преподавателей необходимо поставить галочку с нужным(и) и нажать кнопку «Удалить выбранных».

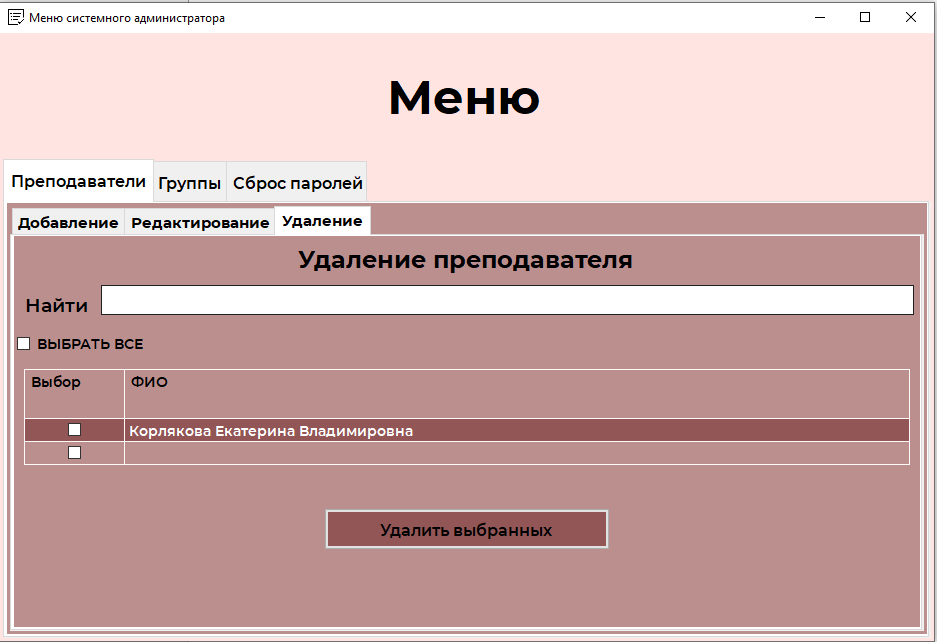


Рисунок Г - Меню системного администратора. Вкладка "Удаление" преподавателя

**2.4.4.4 Вкладка «Группы» меню системного администратора**

Вкладка «Группы» содержит сразу все функции по действиям с группами: добавить, удалить, редактировать группу. Вкладка изображена на рисунке Г14. Для выполнения всех действий необходимо ввести курс и название группы, а затем нажать на соответствующую кнопку.

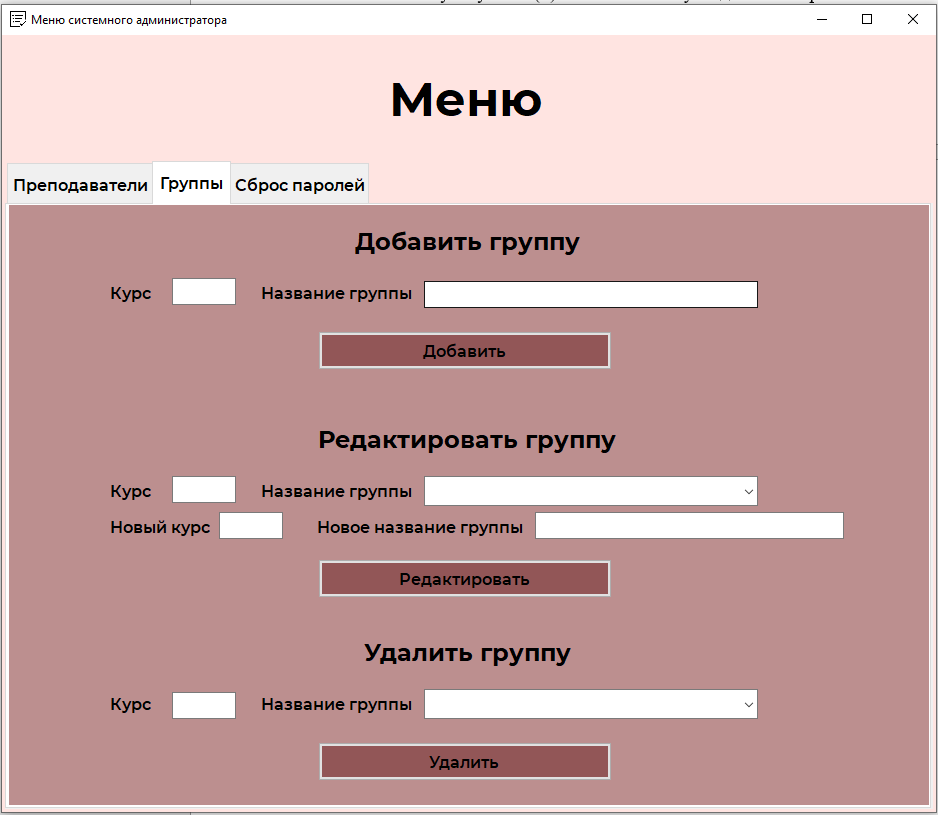


Рисунок Г - Меню системного администратора. Вкладка "Группы"

**2.4.4.5 Вкладка «Сброс паролей» меню системного администратора**

Вкладка «Сброс паролей» изображена на рисунке Г15. Для сброса пароля необходимо ввести пользователя (выбираем из списка – студент или преподаватель), ФИО, год рождения, а также новые данные: пароль и кодовое слово. После ввода данных, нажать на кнопку «Сохранить».

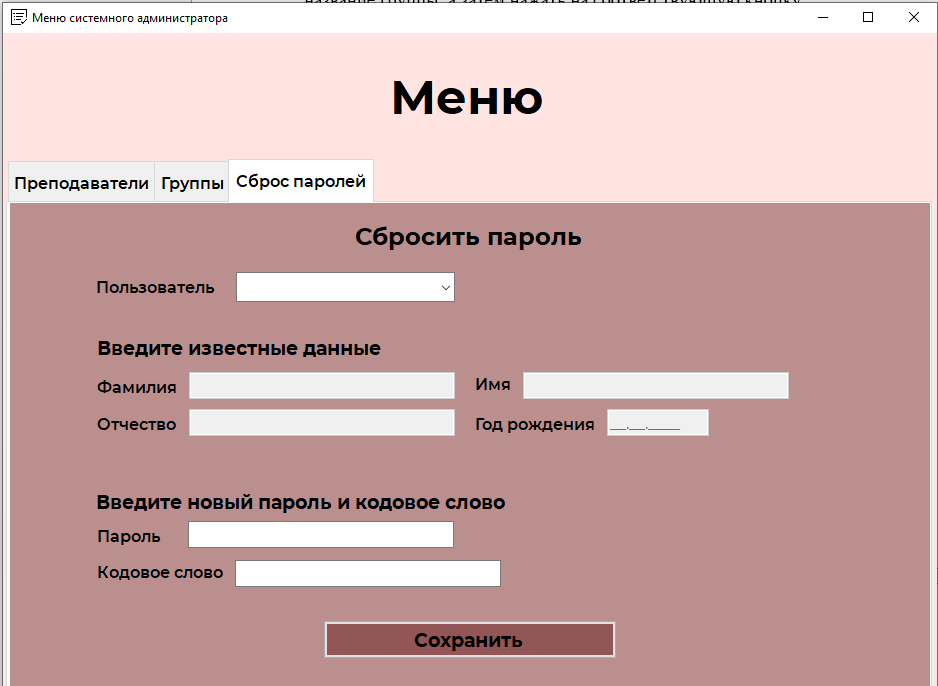


Рисунок Г - Меню системного администратора. Сброс паролей

**2.4.5 Меню администратора**

Меню администратора изображено на рисунке Г16. Оно содержит только форму «Табель успеваемости». Для просмотра результатов тестирований по определенной теме какой-либо дисциплины в каждой из групп студентов, необходимо ввести эти данные из предложенных списков в поля для ввода. При желании очистить поля нажать на кнопку в виде стрелочки в правом верхнем углу.

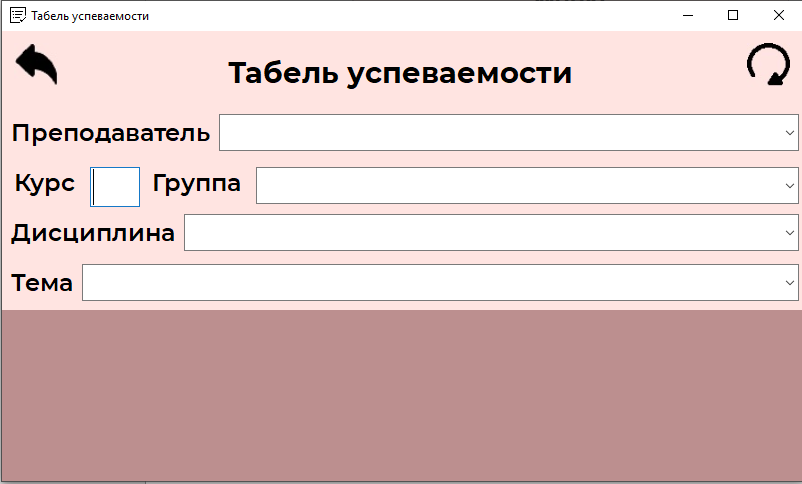


Рисунок Г - Меню администрации

**2.4.5 Меню преподавателя**

**2.4.5.1 Главное меню преподавателя**

Главное меню преподавателя изображено на рисунке Г17. Оно содержит три кнопки: «Тесты», «Студенты» и «Табель успеваемости». Под каждой кнопкой прописаны возможности каждой формы, открывающейся нажатием на них.



Рисунок Г - Меню преподавателя

При нажатии на кнопку «Тесты» откроется окно «Действия с тестами». На форме есть четыре вкладки: «Создание», «Удаление», «Редактирование», «Открыть тест».

**2.4.5.2 Вкладка «Создание» из меню «Действия с тестами» преподавателя**

Вкладка «Создание» представлена на рисунке Г18. Во вкладке «Создание» происходит создание теста. Вводится тема, общий балл и количество вопросов в соответствующие поля для ввода. Далее вводятся вопросы, пока не достигнут пика – введенного количества вопросов. Вопросы вводятся по следующему алгоритму: вводится вид вопроса из предложенных вариантов, текст вопроса и поочередно ответы с отметкой верного(ых). Ответы вводятся в поле, расположенной под текстом «Ответ(ы)», а затем нажимается кнопка «Добавить ответ». При желании удалить – текст ответа вводится в то же поле и нажимается кнопка «Удалить ответ». После заполнения всех полей нажимается кнопка «Добавить вопрос». После добавления всех вопросов нажимается кнопка «Завершить создание теста».

Чтобы вернуться к главному меню, достаточно нажать на стрелочку в верхнем левом углу.

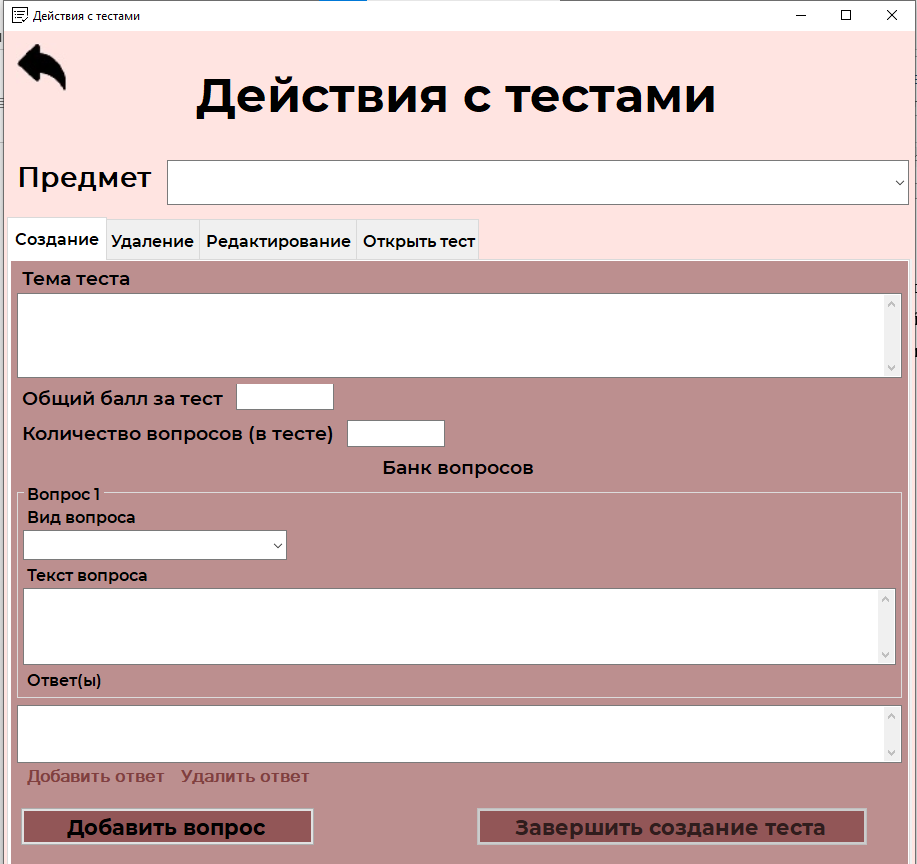


Рисунок Г - Меню преподавателя. Действия с тестами. Создание

**2.4.5.3 Вкладка «Удаление» из меню «Действия с тестами» преподавателя**

Вкладка «Удаление» представлена на рисунке Г19. Во вкладке «Удаление» происходит удаление теста. Для этого необходимо выбрать предмет из списка имеющихся, выбрать тест, а затем нажать на кнопку «Удалить».

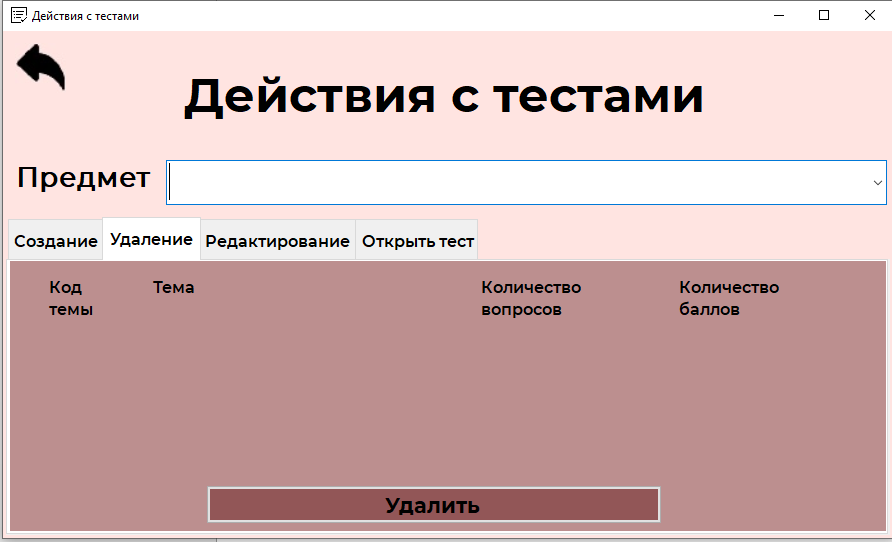


Рисунок Г - Меню преподавателя. Действия с тестами. Удаление

**2.4.5.4 Вкладка «Редактирование» и подвкладка «Редактирование описания» из меню «Действия с тестами» преподавателя**

Вкладка «Редактирование» представлена на рисунке Г20. Вкладка «Редактирование» делится так же на две вкладки: «Редактирование описания» и «Добавить новые вопросы». Во вкладке «Редактирование»-«Редактирование описания» происходит редактирование текста тем, вопросов и ответов теста. Для этого необходимо выбрать предмет из списка имеющихся, щелкнуть на тему, за ней появляются все имеющиеся у данной темы вопросы и соответственно этому вопросу ответ, изменить нужные данные, а затем нажать на кнопку «Применить». Чтобы сбросить выбранные вопросы и ответы, можно нажать на кнопку «Обновить».

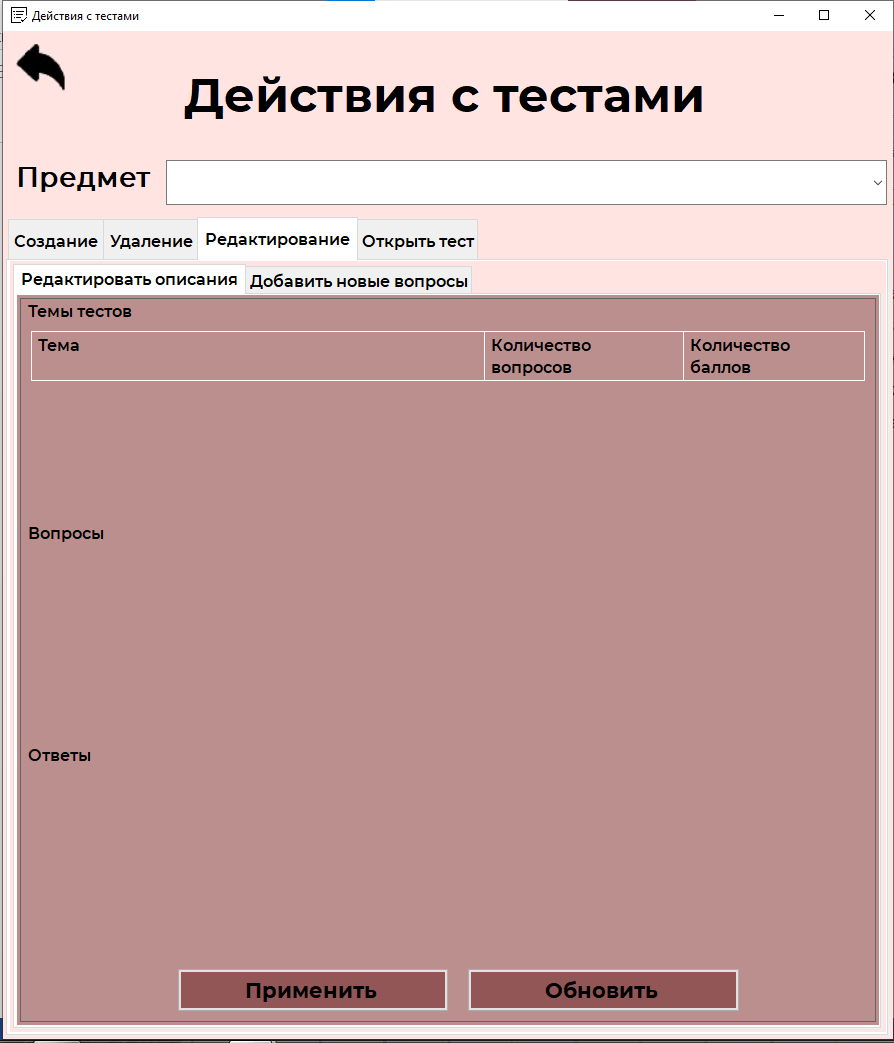


Рисунок Г - Меню преподавателя. Действия с тестами. Редактирование - Редактировать описания

**2.4.5.5 Вкладка «Редактирование» и подвкладка «Добавить новые вопросы» из меню «Действия с тестами» преподавателя**

Владка «Редактирование» - «Добавить новые вопросы» изображена на рисунке Г21. Аналогично созданию теста, при выборе предмета и названия темы из списка имеющихся, добавляем новые вопросы.

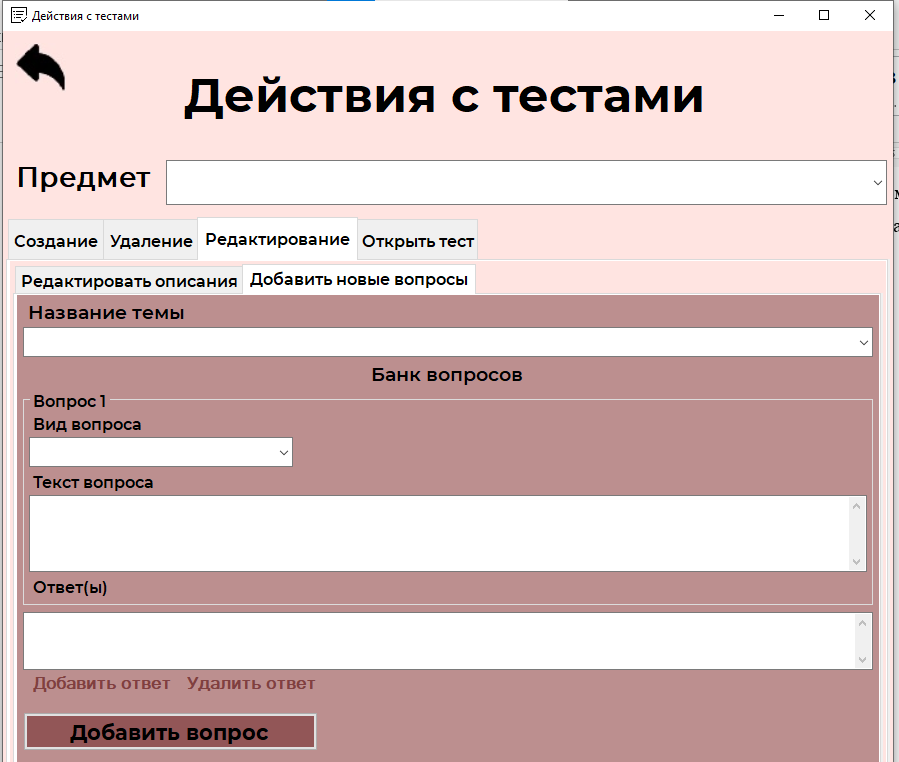


Рисунок Г - Меню преподавателя. Действия с тестами. Редактирование - Добавить новые вопросы

**2.4.5.6 Вкладка «Открыть тест» из меню «Действия с тестами» преподавателя**

Вкладка «Открыть тест» представлена на рисунке Г22. Во вкладке «Открыть тест» происходит предоставление доступа студенту(ам) к прохождению определенного теста. Для этого вводится предмет и название темы из списка имеющихся, вводится курс, группа из списка имеющихся, ставятся галочки напротив нужных студентов, если нужно поставить галочки всей группе – нужно нажать на «Выбрать все». Затем ввести количество попыток и нажать на кнопку «Добавить к выбранным».

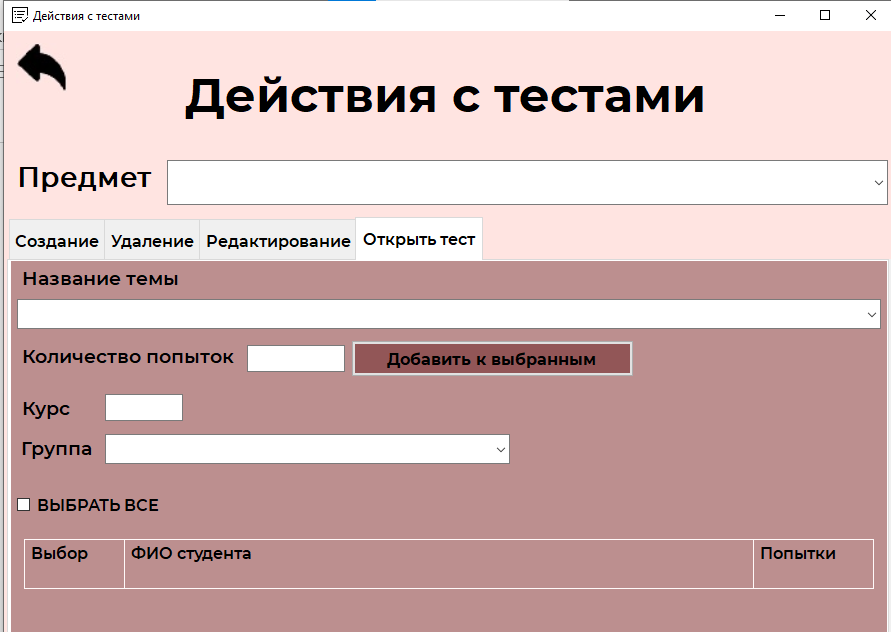


Рисунок Г - Меню преподавателя. Действия с тестами. Открыть тест

**2.4.5.7 Вкладка «Добавление» из меню «Действия со студентами» преподавателя**

Вкладка «Добавление» представлена на рисунке Г23. Во вкладке «Добавление» происходит добавление студента в определенную группу. Для этого необходимо ввести номер курса и из предложенного списка в соответствии с курсом ввести группу. Для поиска студентов есть возможности найти их по ФИО. Далее необходимо выбрать нужных студентов из списка и нажать на кнопку «Добавить». Чтобы вернуться к главному меню, достаточно нажать на стрелочку в верхнем левом углу. При необходимости очистки всех полей нажать на стрелочку в правом верхнем углу.

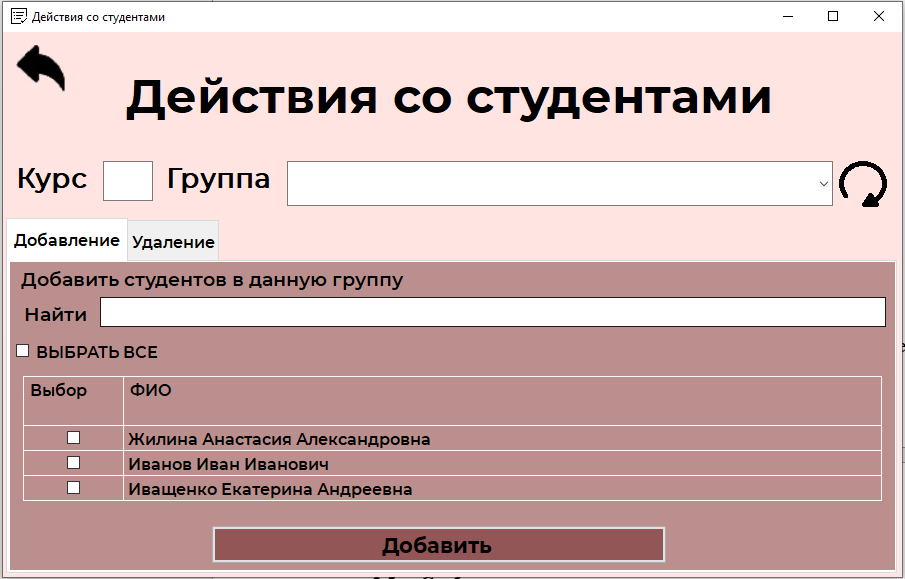


Рисунок Г - Меню преподавателя. Действия со студентами. Добавление

**2.4.5.7 Вкладка «Удаление» из меню «Действия со студентами» преподавателя**

Вкладка «Удаление» представлена на рисунке Г24. Во вкладке «Удаление» происходит удаление студента из определенной группы. Для этого необходимо ввести номер курса и из предложенного списка в соответствии с курсом ввести группу. Для поиска студентов есть возможности найти их по ФИО. Далее необходимо выбрать нужных студентов из списка и нажать на кнопку «Удалить».

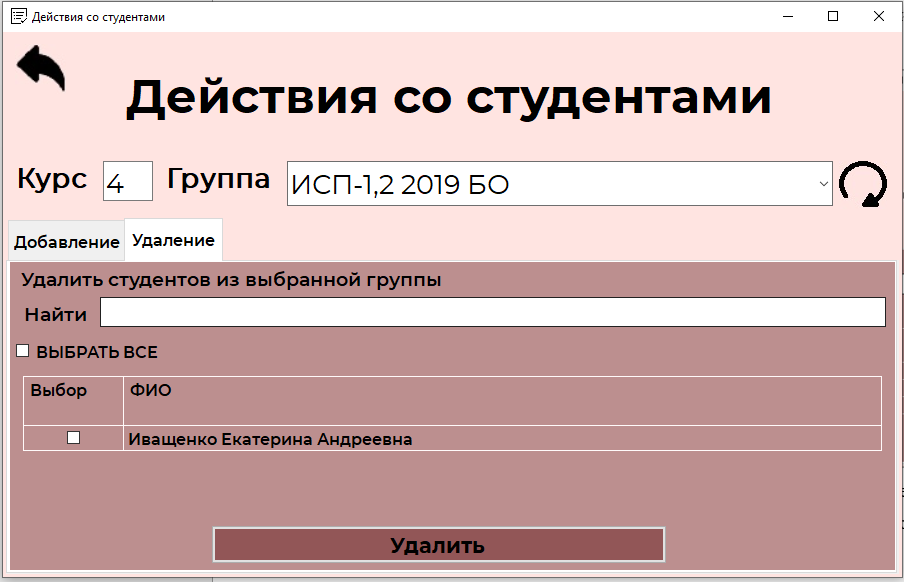


Рисунок Г - Меню преподавателя. Действия со студентами. Удаление

**2.4.5.7 Форма «Табель успеваемости» преподавателя**

Форма «Табель успеваемости» представлена на рисунке Г25. На форме «Табель успеваемости» открывается возможность просмотра результатов тестирований в определенной группе. Для этого необходимо ввести номер курса и из предложенного списка в соответствии с курсом ввести группу. Из списка ввести дисциплину и в соответствии с ним тему. Чтобы вернуться к главному меню, достаточно нажать на стрелочку в верхнем левом углу. При необходимости очистки всех полей нажать на стрелочку в правом верхнем углу.

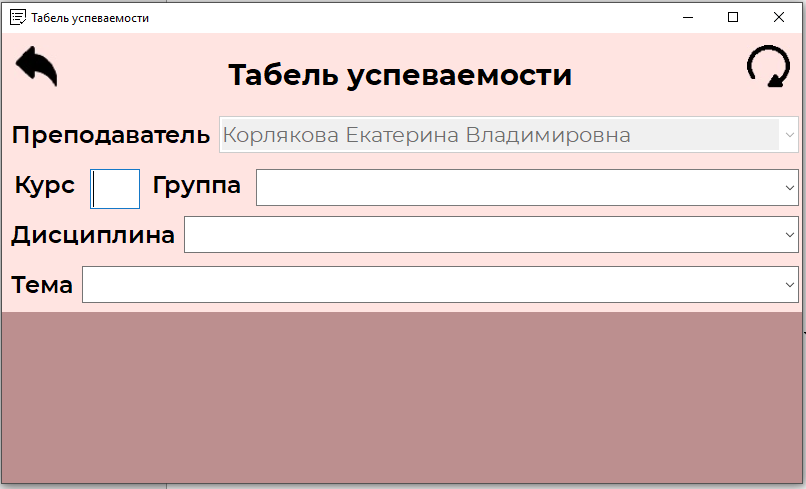


Рисунок Г - Меню преподавателя. Табель успеваемости

**2.4.6 Меню студента**

Главное меню студента изображено на рисунке Г26. Оно содержит две кнопки: «Тестирование» и «Просмотр баллов».

При нажатии на кнопку «Тестирование» откроется окно «Тестирование». На форме есть четыре вкладки: «Создание», «Удаление», «Редактирование», «Открыть тест».

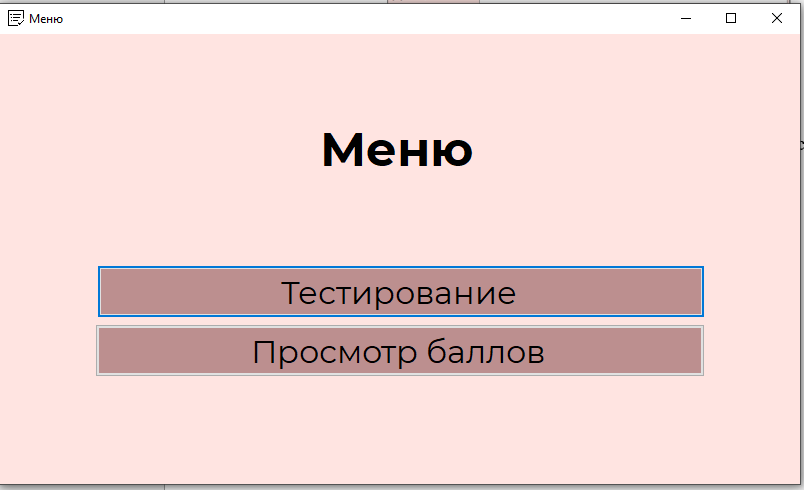


Рисунок Г - Главное меню студента

**2.4.6.1 Форма «Открытые темы тестов»**

Форма «Открытые темы тестов» изображена на рисунке Г27. Для начала теста необходимо ввести преподавателя и предмет из предложенных вариантов, затем выбрать тему теста и нажать «Начать тестирование». Чтобы вернуться к главному меню, достаточно нажать на стрелочку в верхнем левом углу.

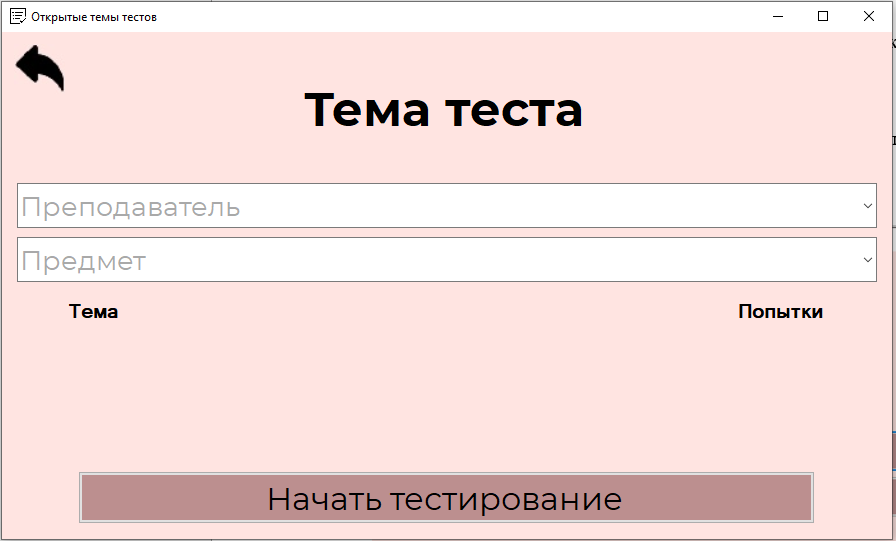


Рисунок Г - Меню студента. Открытые темы тестов

**2.4.6.2 Форма «Баллы за пройденные тесты»**

Форма «Баллы за пройденные тесты» изображена на рисунке Г28. Для просмотра баллов необходимо ввести преподавателя и предмет из предложенных вариантов. Чтобы вернуться к главному меню, достаточно нажать на стрелочку в верхнем левом углу.

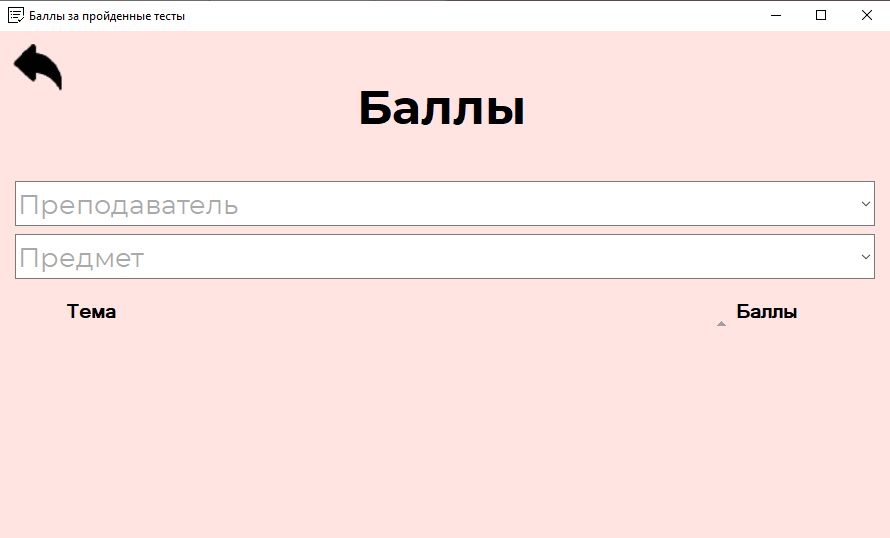


Рисунок Г 28 - Меню студента. Баллы за пройденные тесты

* 1. **Сообщения пользователю**

При неверном введении данных появляются окна сообщений, указывающие на ошибку. Пример такого окна представлен на рисунке Г29.

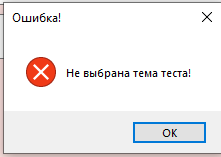


Рисунок Г 29 - Сообщение пользователю