

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО «МЦОБ. Онлайн-сервисы»

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-116

курса, группы

Коваля Никиты Георгиевича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка _____

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка _____

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2025 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	5
1.1	Системы тестирования и оценки знаний, основные цели и задачи	5
1.2	Классификация систем тестирования знаний	6
1.3	Основные компоненты системы тестирования знаний	7
1.4	История развития систем тестирования знаний	10
2	Техническое задание	13
2.1	Основание для разработки	13
2.2	Цель и назначение разработки	13
2.3	Требования к данным программной системы	13
2.4	Функциональные требования к программной системе	14
2.5	Моделирование вариантов использования	14
2.5.1	Описание вариантов использования	19
2.5.1.1	Вариант использования «Регистрация»	21
2.5.1.2	Вариант использования «Авторизация»	21
2.5.1.3	Вариант использования «Тестирование»	22
2.5.1.4	Вариант использования «Создание нового теста»	22
2.6	Требования пользователя к интерфейсу приложения	23
2.7	Требования к оформлению документации	25
3	Технический проект	26
3.1	Общая характеристика организации решения задачи	26
3.2	Обоснование выбора технологии проектирования	26
3.2.1	Описание используемых технологий и языков программирования	26
3.2.2	Язык программирования Python	26
3.2.3	Библиотека Tkinter	26
3.2.3.1	Достоинства языка Python	27
3.2.3.2	Недостатки языка Python	27
3.3	Диаграмма компонентов и схема обмена данными между файлами	27
3.4	Содержание информационных блоков. Основные сущности	30

3.5 Проектирование пользовательского интерфейса	33
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

JSON — текстовый формат обмена данными, основанный на JavaScript.

Сериализация — это процесс преобразования объекта в поток байтов для сохранения или передачи в память, базу данных или файл.

Десериализация - это процесс преобразования данных, сериализованных в определенном формате, обратно в объекты или структуры данных, которые они представляют.

ИС – информационная система.

ИТ – информационные технологии.

КТС – комплекс технических средств.

ПО – программное обеспечение.

РП – рабочий проект.

ТЗ – техническое задание.

ТП – технический проект.

1 Анализ предметной области

1.1 Системы тестирования и оценки знаний, основные цели и задачи

Системы тестирования знаний представляют собой программные комплексы, предназначенные для автоматизированной оценки уровня знаний, навыков и компетенций обучающихся в различных областях. Они являются важным инструментом в образовательном процессе, позволяющим преподавателям эффективно контролировать усвоение материала, а студентам - оценивать свои знания и выявлять пробелы.

Основные цели и задачи систем тестирования знаний:

- Автоматизация оценки: Снижение трудовых затрат на проверку работ, особенно при большом количестве обучающихся;
- Объективность оценки: Минимизация субъективного фактора при оценке знаний, основанной на четких критериях и алгоритмах;
- Повышение эффективности обучения: Предоставление оперативной обратной связи студентам о результатах тестирования, что позволяет им своевременно корректировать свою учебную деятельность;
- Дифференцированный подход: Возможность создания тестов различной сложности, учитывающих индивидуальные особенности обучающихся;
- Разнообразие форматов: Поддержка различных типов тестовых заданий (множественный выбор, открытые вопросы, соответствие, перестановка и т.д.), что позволяет оценивать знания с разных сторон;
- Контроль за ходом обучения: Мониторинг успеваемости обучающихся, выявление отстающих и оказание им своевременной помощи;
- Управление знаниями: Создание и поддержание базы тестовых заданий, что позволяет повторно использовать их в различных учебных курсах;
- Статистический анализ: Предоставление данных об уровне усвоения материала по различным темам, что позволяет преподавателям корректировать учебные планы и методы обучения;

- Удобство использования: Интуитивно понятный интерфейс для преподавателей и студентов, облегчающий процесс создания, проведения и прохождения тестирования;
- Безопасность: Обеспечение конфиденциальности результатов тестирования и защита от несанкционированного доступа к системе.

1.2 Классификация систем тестирования знаний

В настоящее время определено, что педагогические тесты помогают получить более объективные оценки уровня знаний, умений, навыков, проверить соответствие требований к подготовке выпускников вузов заданным стандартам, выявить пробелы в подготовке персонала.

Тест – это инструмент, состоящий из квалитметрически выверенной системы тестовых заданий, стандартизированной процедуры проведения и заранее спроектированной технологии обработки и анализа результатов, предназначенной для измерения качеств и свойств личности, изменение которых возможно в процессе систематического обучения. Или тест – краткое стандартизированное испытание, допускающее количественную оценку результатов на основе их статистической обработки. Отбор структура заданий теста зависит от того, какие показатели и факторы интересуют исследователя данной группы лиц. Каждое из заданий теста по своей сути представляет для испытуемого вопрос, проблему. Ответ на вопрос – это всегда устранение некоторых сомнений, колебаний, неопределенности в рассматриваемой ситуации с целью получения новых, более точных знаний.

С точки зрения целей применения можно выделить:

- тесты достижений;
- критериально-ориентированные тесты, позволяющие сопоставить уровень индивидуальных учебных достижений с полным объемом знаний, умений и навыков;
- нормативно-ориентированные тесты, сравнивающие испытуемых по уровню их учебных достижений;
- аттестационные тесты, определяющие степень обученности;

– тесты прогнозирования результатов обучения по выбранной технологии обучения.

1.3 Основные компоненты системы тестирования знаний

В системах тестирования знаний можно выделить несколько ключевых компонентов, каждый из которых выполняет определенную функцию и обеспечивает работоспособность системы в целом.

Ключевые компоненты системы тестирования:

1. Модуль управления пользователями. Отвечает за управление всеми пользователями системы, включая преподавателей, студентов, администраторов и других заинтересованных лиц.

Функции:

– Регистрация пользователей: Предоставление возможности создания новых учетных записей с указанием необходимой информации;

– Авторизация пользователей: Проверка логина и пароля при входе в систему, предоставление доступа к функциональности в соответствии с ролью пользователя;

– Просмотр и редактирование профилей пользователей: Предоставление возможности просмотра и изменения личной информации пользователей;

– Блокировка и удаление учетных записей: Возможность блокировки доступа к системе для определенных пользователей или полного удаления их учетных записей;

– Управление ролями и правами доступа: Назначение ролей пользователям (преподаватель, студент, администратор и т.д.) и определение прав доступа к различным компонентам и функциям системы. Например, преподаватель может иметь право создавать и редактировать тесты, а студент - только проходить их.

2. Модуль создания и редактирования тестов. Предназначен для разработки и модификации тестов с использованием различных типов вопросов и настроек.

Функции:

- Создание новых тестов: Предоставление интерфейса для создания новых тестов с указанием названия, описания, области знаний и других параметров;
- Добавление и редактирование вопросов: Возможность добавления вопросов различных типов (множественный выбор, открытые вопросы, соответствие, перестановка и т.д.) и редактирования их содержимого;
- Настройка параметров вопросов: Указание правильных ответов, баллов за правильные ответы, времени на ответ и других параметров;
- Группировка вопросов по темам и категориям: Организация вопросов по темам или категориям для упрощения поиска и управления;
- Предпросмотр тестов: Возможность предварительного просмотра теста с точки зрения студента для проверки корректности отображения вопросов и ответов;
- Импорт и экспорт тестов: Поддержка импорта тестов из других систем или файлов (например, в формате JSON) и экспорта тестов в различные форматы.

Технические аспекты:

- Использование визуального редактора для создания и редактирования вопросов;
- Поддержка различных типов данных для хранения вопросов и ответов;
- Реализация валидации данных для проверки корректности введенной информации.

3. Модуль проведения тестирования. Обеспечивает возможность студентам проходить тесты и получать результаты.

Функции:

- Доступ к тестам: Предоставление списка доступных тестов для студента;
- Начало и завершение тестирования: Возможность начала и завершения тестирования;

- Отображение вопросов: Отображение вопросов в удобном формате с возможностью выбора ответов или ввода текста;
- Сохранение ответов: Автоматическое сохранение ответов студента в процессе тестирования;
- Предоставление результатов: Отображение результатов тестирования после его завершения, включая количество правильных и неправильных ответов, итоговый балл и оценку;
- Поддержка различных типов тестирования: Возможность настройки типов вопросов: вопрос с самостоятельным вводом ответа; вопрос с единственным выбором ответа, вопрос с множественным выбором ответов.

4. Модуль автоматической проверки. Автоматически проверяет ответы на вопросы, требующие однозначного ответа (например, множественный выбор, соответствие, перестановка), и выставляет предварительные результаты.

Функции:

- Сравнение ответов студента с правильными ответами: Сравнение ответов, выбранных студентом, с эталонными ответами, хранящимися в базе данных;
- Вычисление баллов: Вычисление баллов за правильные ответы на основе настроек, заданных при создании теста;
- Формирование отчета о предварительных результатах: Создание отчета о предварительных результатах тестирования с указанием количества правильных и неправильных ответов, итогового балла и оценки.

Технические аспекты:

- Использование алгоритмов сопоставления строк и логических операций для проверки ответов;
 - Поддержка различных форматов ответов (текст, числа, даты и т.д.);
5. Модуль отчетов и аналитики. Предоставляет инструменты для формирования отчетов о результатах тестирования, анализа статистических данных и визуализации информации.

Функции:

- Генерация отчетов: Формирование отчётов о результатах тестирования для отдельных студентов, групп студентов или всего курса;
- Анализ статистических данных: Анализ статистических данных, таких как средний балл, процент правильных ответов, распределение оценок, корреляция между различными вопросами и т.д.;
- Визуализация информации: Представление данных в виде графиков, диаграмм и таблиц для облегчения восприятия и анализа.

Технические аспекты:

- Использование алгоритмов сопоставления строк и логических операций для проверки ответов;
- Поддержка различных форматов ответов (текст, числа, даты и т.д.);

1.4 История развития систем тестирования знаний

История развития систем тестирования знаний тесно связана с развитием информационных технологий и педагогической науки. Её можно условно разделить на несколько этапов, каждый из которых характеризуется своими особенностями и достижениями:

1. Докомпьютерная эра (до середины XX века).

Традиционные методы оценки: В этот период оценка знаний осуществлялась преимущественно с помощью письменных и устных экзаменов, контрольных работ и рефератов. Характеризуется развитием тестологии: В начале XX века в психологии и педагогике активно развивалась тестология – наука о разработке и применении стандартизированных тестов. Были разработаны первые тесты интеллекта, тесты достижений и другие виды тестов; Первые попытки автоматизации: Появляются первые механические устройства для проверки тестов с множественным выбором, но они были громоздкими и малоэффективными; Недостатки: Субъективность оценки, трудоемкость проверки, ограниченные возможности анализа результатов.

2. Эра мейнфреймов (1950-е – 1970-е годы).

Появление первых компьютеров: С появлением компьютеров стало возможным автоматизировать некоторые этапы тестирования, такие как об-

работка результатов и формирование отчетов. Первые программы для тестирования: Разрабатываются первые программы для проверки тестов с множественным выбором и формирования простых отчетов. Эти программы работали на мейнфреймах и были доступны только в крупных университетах и образовательных учреждениях. Недостатки: высокая стоимость компьютеров, сложность программирования, ограниченные возможности графического интерфейса.

3. Эра персональных компьютеров (1980-е – 1990-е годы).

Распространение персональных компьютеров: Появление персональных компьютеров сделало системы тестирования знаний более доступными для широкого круга пользователей; Разработка программного обеспечения для тестирования: Разрабатываются специализированные программы для создания, проведения и проверки тестов на персональных компьютерах. Появляются первые графические интерфейсы пользователя, что облегчает работу с системой; Развитие баз данных: Развитие баз данных позволяет хранить большие объемы информации о тестовых заданиях, результатах тестирования и пользователях; Недостатки: Ограниченные возможности сетевого взаимодействия, сложность интеграции с другими системами, отсутствие стандартизированных форматов данных.

4. Эра Интернета (с конца 1990-х годов по настоящее время).

Появление веб-технологий: Развитие веб-технологий позволило создавать системы тестирования знаний, доступные через Интернет, что значительно расширило их возможности и охват аудитории; Разработка систем дистанционного обучения (LMS): системы тестирования знаний стали интегрироваться в системы дистанционного обучения (LMS), предоставляя комплексные решения для организации учебного процесса онлайн; Адаптивное тестирование: Развитие алгоритмов машинного обучения и искусственного интеллекта позволило создавать адаптивные системы тестирования знаний, которые автоматически подстраивают сложность тестовых заданий под уровень знаний обучающегося; Мобильное тестирование: Появление мобильных устройств и развитие мобильных технологий позволило проводить те-

стирование с помощью смартфонов и планшетов; Облачные системы тестирования знаний: Развитие облачных технологий позволило развертывать системы тестирования в облаке, что снизило затраты на инфраструктуру и обеспечило высокую доступность и масштабируемость.

Современные тенденции систем тестирования знаний заключаются в следующем:

- Использование искусственного интеллекта (ИИ) для автоматической генерации вопросов и оценки открытых ответов;
- Развитие игровой составляющей в тестировании для повышения мотивации обучающихся;
- Интеграция систем тестирования с социальными сетями и другими онлайн-сервисами;

2 Техническое задание

2.1 Основание для разработки

Полное наименование системы: «Программно-информационная система для оценки и контроля знаний». Основанием для разработки программы является приказ ректора ЮЗГУ от «17» апреля 2025 г. №1828-с «О направлении (допуске) на практику».

2.2 Цель и назначение разработки

Программно-информационная система предназначена для контроля и оценки знаний обучающихся с целью улучшения процесса обучения.

Задачами данной разработки являются:

1. создание модуля для создания новых тестов;
2. создание модуля для создания вопросов;
3. создание модуля для редактирования вопросов;
4. создание модуля для авторизации;
5. создание модуля для регистрации;
6. создание модуля для просмотра результатов пройденных тестов;
7. создание модуля для управления пользователями;
8. создание модуля для пользователя;
9. создание модуля для администратора;
10. реализация системы оценивания по результатам теста;
11. реализация системы хранения тестов и результатов тестов;

2.3 Требования к данным программной системы

Входными данными для системы являются:

- данные имеющихся тестов;
- данные о зарегистрированных пользователях;
- данные администратора;

Выходными данными для системы являются:

- графическое отображение вопросов тестов;

- данные создания/изменении тестов;
- данные о новых зарегистрированных/добавленных администратором пользователях;
- данные о изменении паролей пользователей;
- данные результатов пройденных тестов.

2.4 Функциональные требования к программной системе

В разрабатываемой программно-информационной системе должно быть предусмотрено наличие два класса пользователей: пользователь и администратор.

Пользователю должны быть доступны следующие функции программы:

1. Авторизация.
2. Регистрация.
3. Прохождение выбранного теста.
4. Просмотр собственных результатов пройденных тестов.

Администратору должны быть доступны следующие функции программы:

1. Авторизация.
2. Изменение пароля администратора.
3. Прохождение выбранного теста.
4. Просмотр результатов всех пользователей.
5. Управление пользователями.
6. Создание новых тестов.
7. Редактирование имеющихся тестов.

2.5 Моделирование вариантов использования

Для разрабатываемой системы тестирования знаний была реализована модель, которая обеспечивает наглядное представление вариантов использования системы.

Она помогает в физической разработке и детальном анализе взаимосвязей объектов.

Диаграмма вариантов описывает функциональное назначение разрабатываемой системы. То есть это то, что система будет непосредственно делать в процессе своего функционирования. Она является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Проектируемая система представляется в виде ряда прецедентов, предоставляемых системой актерам или сущностям, которые взаимодействуют с системой. Актером или действующим лицом является сущность, взаимодействующая с системой извне (например, человек, техническое устройство). Прецедент служит для описания набора действий, которые система предоставляет актеру.

На рисунке 2.1 в виде диаграммы прецедентов представлены функциональные требования к системе, доступные для пользователя.

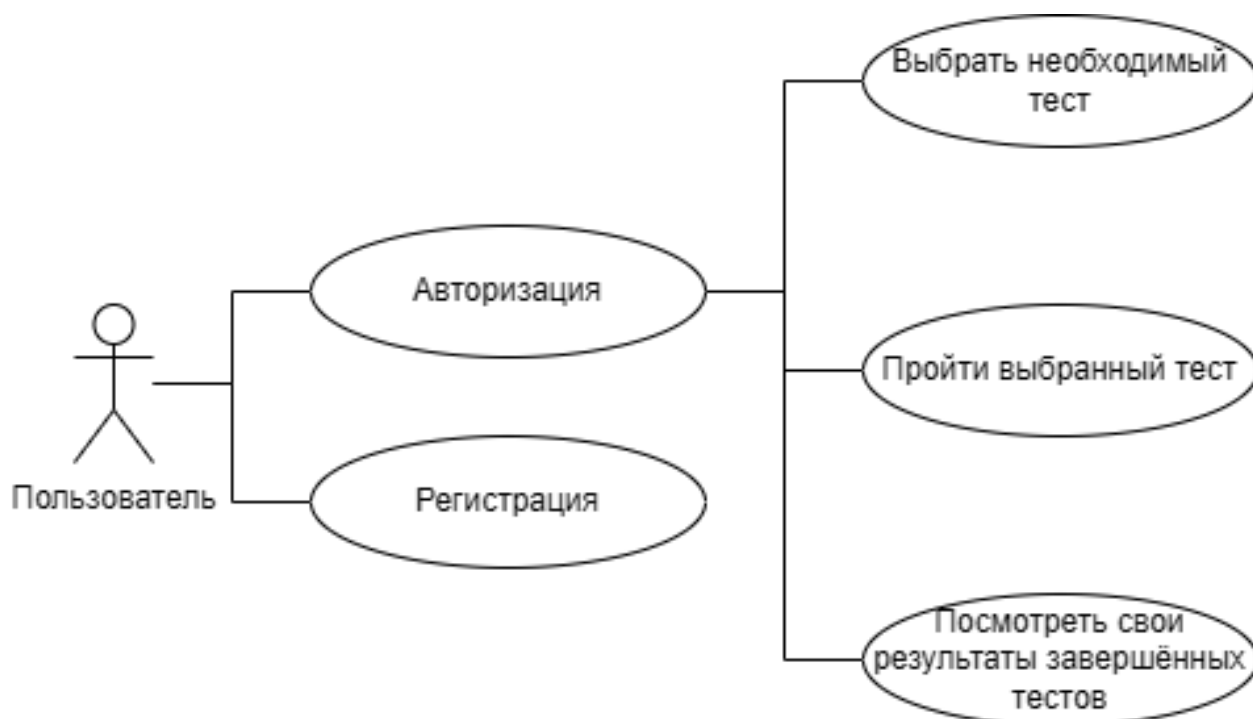


Рисунок 2.1 – Диаграмма прецедентов пользователя

На рисунке 2.2 представлены дополнительные функциональные требования к системе для администратора.

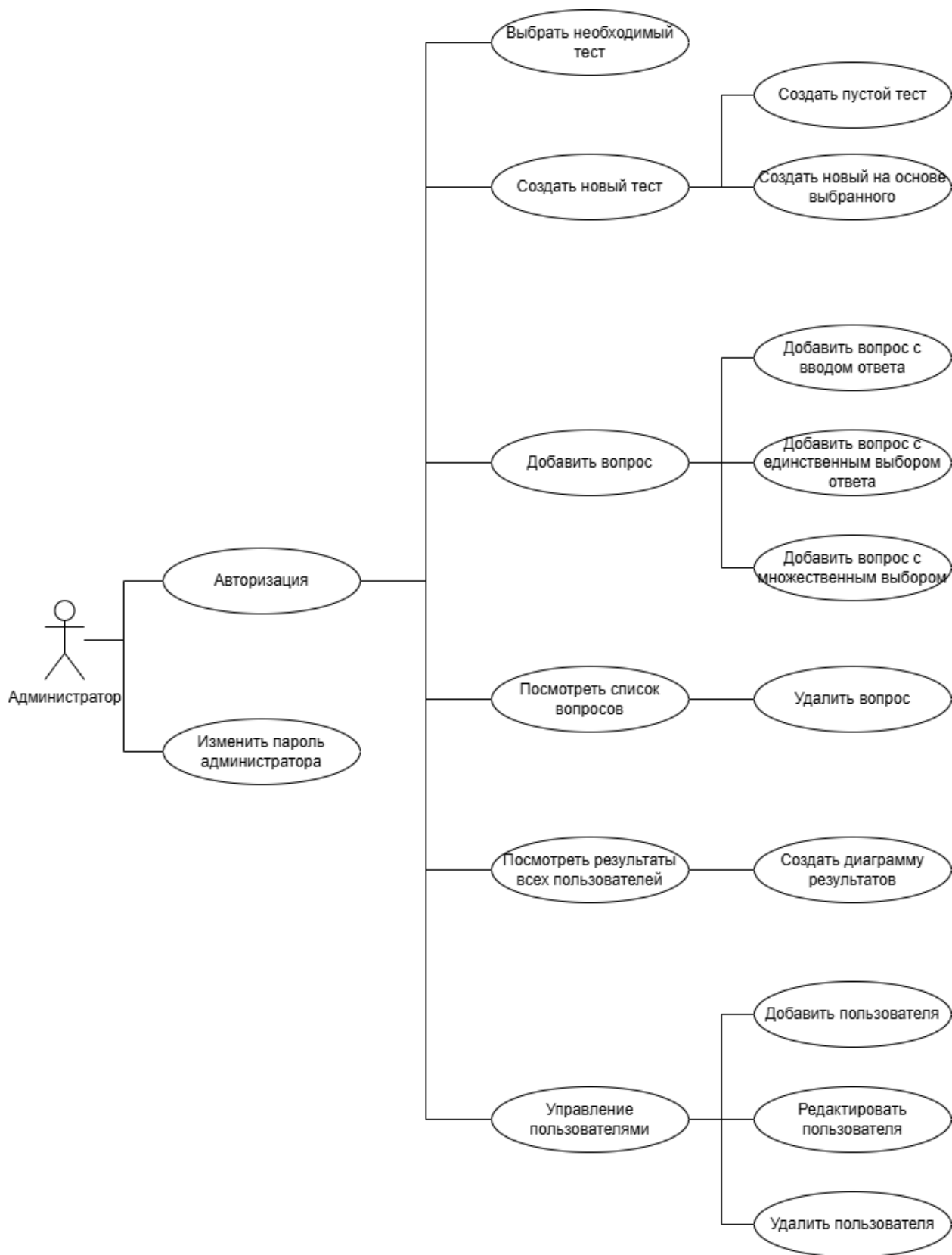


Рисунок 2.2 – Диаграмма прецедентов администратора

На основании анализа предметной области в программе должны быть реализованы следующие варианты использования: Для пользователя:

1. ВИ «Авторизация». Данный прецедент позволяет пользователю авторизоваться.
2. ВИ «Регистрация». Данный прецедент позволяет зарегистрироваться.
3. ВИ «Выбрать желаемый тест». Данный прецедент позволяет пользователю выбрать желаемый тест.
4. ВИ «Пройти выбранный тест». Данный прецедент позволяет пользователю приступить к прохождению выбранного теста.
5. ВИ «Добавить вопрос». Данный прецедент позволяет пользователю добавить вопрос в выбранном тесте.
6. ВИ «Удалить вопрос». Данный прецедент позволяет пользователю удалить вопрос в выбранном тесте.
7. ВИ «Просмотреть результаты». Данный прецедент позволяет пользователю посмотреть собственные результаты пройденных тестов.
8. ВИ «Создать новый тест». Данный прецедент позволяет пользователю создать новый тест.

Для администратора:

1. ВИ «Авторизоваться как администратор». Данный прецедент позволяет авторизоваться как администратор и получить соответствующие права.
2. ВИ «Изменить пароль администратор». Данный прецедент позволяет изменить пароль администратор. Требуется ввести старый пароль и два раза повторить новый.
3. ВИ «Выбрать желаемый тест». Данный прецедент позволяет администратору выбрать желаемый тест.
4. ВИ «Пройти выбранный тест». Данный прецедент позволяет администратору приступить к прохождению выбранного теста.
5. ВИ «Добавить вопрос». Данный прецедент позволяет администратору добавить вопрос в выбранном тесте. Возможно добавить три варианта

вопроса: с вводом ответа, с единственным выбором из имеющихся ответов, с множественным выбором из имеющихся ответов.

6. ВИ «Просмотреть набор выбранного теста». Данный прецедент позволяет администратору посмотреть состав теста.

7. ВИ «Удалить вопрос». Данный прецедент позволяет администратору удалить выбранный вопрос.

8. ВИ «Просмотреть результаты». Данный прецедент позволяет администратору посмотреть результаты пройденных тестов всех пользователей.

9. ВИ «Создать новый тест». Данный прецедент позволяет администратору создать новый тест. Можно создать пустой тест или копировать текущий набор выбранного теста.

10. ВИ «Управление пользователями». Данный прецедент позволяет администратору добавить нового пользователя, удалить или редактировать уже существующего.

2.5.1 Описание вариантов использования

Данные варианта использования «Пройти выбранный тест».

Входными данными является выбранный тест, имя пользователя.

Выходными данными прецедента «Пройти выбранный тест» являются вопросы с ответами и результат прохождения теста.

Основной исполнитель: Пользователь.

Заинтересованные лица и их требования: Пользователь хочет пройти тест.

Предусловие: поля для ввода имени, ответа на вопрос должны быть заполнены.

Постусловие: приложение проверит введены ли данные, если нет, сообщит об этом пользователю.

Основной успешный сценарий:

1. Пользователь запускает программу.
2. Пользователь проходит авторизацию, введя своё имя.
3. Пользователь попадает в меню.

4. Пользователь выбирает желаемый тест.
5. Пользователь нажимает кнопку «Пройти тест».
6. Пользователь попадает в окно теста.
7. Пользователь выбирает или вводит ответы.
8. Пользователь по окончании теста видит его результат.
9. Пользователь закрывает приложение.

Данные варианта использования «Создать новый тест».

Входными данными является пароль администратора, данные для нового теста.

Выходными данными прецедента «Создать новый тест» являются новый тест с вопросами и ответами.

Основной исполнитель: Администратор.

Заинтересованные лица и их требования: Администратор хочет создать тест и добавить в него вопрос

Предусловие: поля для ввода пароля администратора, текста вопроса и его ответа должны быть заполнены.

Постусловие: приложение проверит введены ли данные, если нет, сообщит об этом администратору.

Основной успешный сценарий:

1. Администратор запускает программу.
2. Администратор проходит авторизацию, введя пароль.
3. Администратор попадает в меню.
4. Администратор нажимает кнопку «Создать тест».
5. Администратор попадает в окно создания теста.
6. Администратор нажимает кнопку «Создать новый тест».
7. Администратор нажимает кнопку «Вернуться в меню».
8. Администратор попадает в меню.
9. Администратор нажимает кнопку «Добавить вопрос».
10. Администратор попадает в меню добавления вопроса.
11. Администратор нажимает кнопку добавить базовый вопрос.

12. Администратор в окне создания базового вопроса вводит текст и ответ вопроса.

13. Администратор нажимает кнопку добавить вопрос.

14. Администратор закрывает приложение.

2.5.1.1 Вариант использования «Регистрация»

Заинтересованные лица и их требования: пользователь желает зарегистрироваться в приложении. Предусловие: открыто окно «Авторизация пользователя». Постусловие: создаётся аккаунт нового пользователя на основании введённых данных для регистрации, имени и пароля. Основной успешный сценарий:

1. Пользователь переходит в окно «Регистрация».
2. Приложение отображает соответствующее окно регистрации с полями для ввода имени и пароля.
3. Пользователь заполняет текстовые поля «Имя» и «Пароль». Данные поля обязательны для заполнения. Имя должно состоять только из символов без цифр, минимум два символа. Имя должно быть уникальным.
4. Пользователь нажимает кнопку «Зарегистрироваться».
5. Приложение проверяет введённые пользователем имя и пароль. Если данные некорректны, сообщает об этом пользователю. При корректных данных приложение создаёт новый аккаунт и сообщает пользователю об успешной регистрации.

2.5.1.2 Вариант использования «Авторизация»

Заинтересованные лица и их требования: пользователь желает авторизоваться в приложении. Предусловие: открыто окно «Авторизация пользователя». Постусловие: авторизация произведена, отображается окно «Меню пользователя». Основной успешный сценарий:

1. Пользователь переходит в окно «Авторизация пользователя».
2. Приложение отображает соответствующее окно авторизации с полями для ввода имени и пароля.

3. Пользователь заполняет текстовые поля «Имя» и «Пароль». Данные поля обязательны для заполнения.

4. Пользователь нажимает кнопку «Перейти в меню».

5. Приложение проверяет введенные пользователем имя и пароль. Если данные некорректны, сообщает об этом пользователю. При корректных данных приложение закрывает текущее окно авторизации и открывает окно «Меню пользователя».

2.5.1.3 Вариант использования «Тестирование»

Заинтересованные лица и их требования: пользователь желает пройти тестирование. Предусловие: открыто окно «Меню пользователя». Постусловие: Даны ответы на все вопросы, приложение отображает результаты тестирования пользователю и сохраняет отчет результатов. Основной успешный сценарий:

1. Пользователь выбирает желаемый тест в выпадающем списке и переходит в окно «Тест».

2. Приложение отображает окно теста и выводит текст первого вопроса. Отображение вопроса зависит от его типа: для вопроса с вводом ответа будет создано поле для ввода; для вопроса с единственным выбором ответа будут созданы кнопки с ответами, только одну можно выбрать для ответа; для вопроса с множественным выбором будут созданы кнопки с ответами, можно выбрать сразу несколько ответов.

3. Пользователь отвечает на вопрос и нажимает кнопку «Ответить».

4. Приложение проверяет ответ пользователя и выводит следующий вопрос.

5. Когда пользователь отвечает на последний вопрос, приложение выводит результат прохождения теста и сохраняет его.

2.5.1.4 Вариант использования «Создание нового теста»

Заинтересованные лица и их требования: администратор желает создать новый тест и добавить в него вопрос. Предусловие: открыто окно «Ме-

ню администратора». Постусловие: Создан и сохранён новый тест с добавленным вопросом. Основной успешный сценарий:

1. Администратор переходит в окно «Создать тест».
2. Приложение отображает окно с двумя кнопками: «Создать новый тест» и «Копировать текущий».
3. Администратор нажимает кнопку «Создать новый тест» и вводит имя нового теста всплывающем окне.
4. Приложение оповещает администратора об успешном создании теста.
5. Администратор возвращается в меню, где выбирает созданный тест в выпадающем списке и нажимает кнопку «Добавить вопрос».
6. Приложение отображает окно с тремя кнопками: «Вопрос с вводом», «Вопрос с единственным выбором» и «Вопрос с множественным выбором».
7. Администратор нажимает кнопку добавить «Вопрос с единственным выбором».
8. Приложение отображает окно для создания соответствующего вопроса.
9. Администратор заполняет поле ввода для текста вопроса и переходит к созданию списка ответов: вводит ответ на вопрос и при помощи галочки в элементе интерфейса CheckBox указывает правильность ответа. В данном типе вопроса правильный ответ может быть только один. После чего нажимает кнопку «Добавить ответ».
10. Приложение отображает добавленный вопрос и отмечает его зелёным цветом, если при создании он был выбран правильным.
11. После добавления необходимого количества вопросов администратор нажимает кнопку «Добавить вопрос».
12. Приложение добавляет вопрос с введённым текстом и созданным списком ответов в выбранный тест.

2.6 Требования пользователя к интерфейсу приложения

Приложение должно иметь следующие основные экраны:

1. Окно «Авторизация» - экран при открытии приложения, где пользователь может ввести имя и пароль для входа в свой аккаунт. Имеет переход в окна: «Меню пользователя», «Регистрация», и «Авторизоваться как администратор».

2. Окно «Регистрация» - экран приложения, на котором пользователь может зарегистрироваться. Имеет переход в окно «Авторизация».

3. Окно «Авторизоваться как администратор» - экран приложения, на котором требуется ввести пароль администратора. После успешного ввода пароля позволяет перейти в специальное меню администратора с расширенными возможностями. Имеет переход в окна: «Меню администратора», «Изменить пароль администратора» и «Вернуться к авторизации пользователя».

4. Окно «Изменить пароль администратора» - экран приложения, на котором пользователь может изменить пароль администратора, требуется ввести старый пароль и два раза новый. Имеет переход к окну «Авторизоваться как администратор».

5. Окно «Меню пользователя» - экран приложения, на котором пользователь может выбрать тест и перейти к его прохождению, просмотреть результаты своих завершённых тестов. Имеет переходы в окна: «Тест», «Результаты».

6. Окно «Тест» - экран приложения, на котором динамически отображаются вопросы теста и требуется ввести или выбрать соответствующий ответ.

7. Окно «Результаты» - экран приложения, на котором в зависимости от прав доступа отображаются результаты пройденных тестов. Для пользователя отображаются только собственные результаты, для администратора - результаты всех пользователей с возможностью создания и выгрузки диаграммы, отображающей лучшие попытки прохождения тестов.

8. Окно «Меню администратора» - экран приложения, на котором доступны все возможности системы тестирования знаний: создание и редактирование тестов, управление пользователями, просмотр и прохождение тестов, результаты тестов всех пользователей. Имеет переходы в окна: «Создать

тест», «Добавить вопрос», «Список вопросов», «Тест», «Управление пользователями», «Результаты».

9. Окно «Создать тест» - экран приложения, на котором можно создать новый тест с пустым списком вопросов или копировать список вопросов из выбранного теста.

10. Окно «Добавить вопрос» - экран приложения, на котором можно добавить три типа вопроса: с вводом ответа, с единственным выбором ответа, с множественным выбором.

11. Окно «Список вопросов» - экран приложения, на котором отображен список всех вопросов теста с возможностью удаления любого вопроса из списка.

12. Окно «Управление пользователями» - экран приложения, на котором отображены все зарегистрированные пользователи. Позволяет администратору добавлять, удалять и редактировать пользователей.

2.7 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Необходимо спроектировать и разработать систему тестирования для контроля и оценки знаний.

Система тестирования представляет собой набор взаимосвязанных окон, содержащих текстовую и графическую информацию, которые сгруппированы по разделам.

3.2 Обоснование выбора технологии проектирования

На сегодняшний день информационный рынок, поставляющий программные решения в выбранной сфере, предлагает множество продуктов, позволяющих достигнуть поставленной цели – разработки системы тестирования.

3.2.1 Описание используемых технологий и языков программирования

В процессе разработки системы тестирования используются программные средства и язык программирования. Каждое программное средство и язык программирования применяется для круга задач, при решении которых они необходимы.

3.2.2 Язык программирования Python

Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML). Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах.

3.2.3 Библиотека Tkinter

Tkinter — это кроссплатформенный графический интерфейс Python, позволяющий работать с библиотекой Tk. Он содержит элементы графиче-

ского интерфейса пользователя (GUI — Graphical User Interface), с помощью которых можно создавать различные приложения.

3.2.3.1 Достоинства языка Python

К плюсам Python относятся: Простота. Его часто советуют в качестве первого “базового” языка, так как он очень прост в изучении и исполнении. В процессе написания программы не требуется использование фигурных скобок, как в других языках, что позволяет не отвлекаться на переключение между клавишами уделять больше внимания разработке программы. Обширность применения.

3.2.3.2 Недостатки языка Python

Низкая скорость выполнения (причины: интерпретация, динамическая типизация) по сравнению с Delphi, C/C++, C#, Java. Отсутствие библиотек для создания ”родных”интерфейсов для Windows.

3.3 Диаграмма компонентов и схема обмена данными между файлами

Диаграмма компонентов описывает особенности физического представления разрабатываемой системы. Она позволяет определить архитектуру системы, установив зависимости между программными компонентами, в роли которых может выступать как исходный, так и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы, а также зависимости между ними. На рисунке 3.1 изображена диаграмма компонентов для проектируемой системы. Она включает в себя сервер с операционной системой, на которой установлена система управления содержимым, включающая в себя интерфейс. Помимо этого на диаграмме изображен клиентский компьютер с операционной системой, на которой установлено приложение.

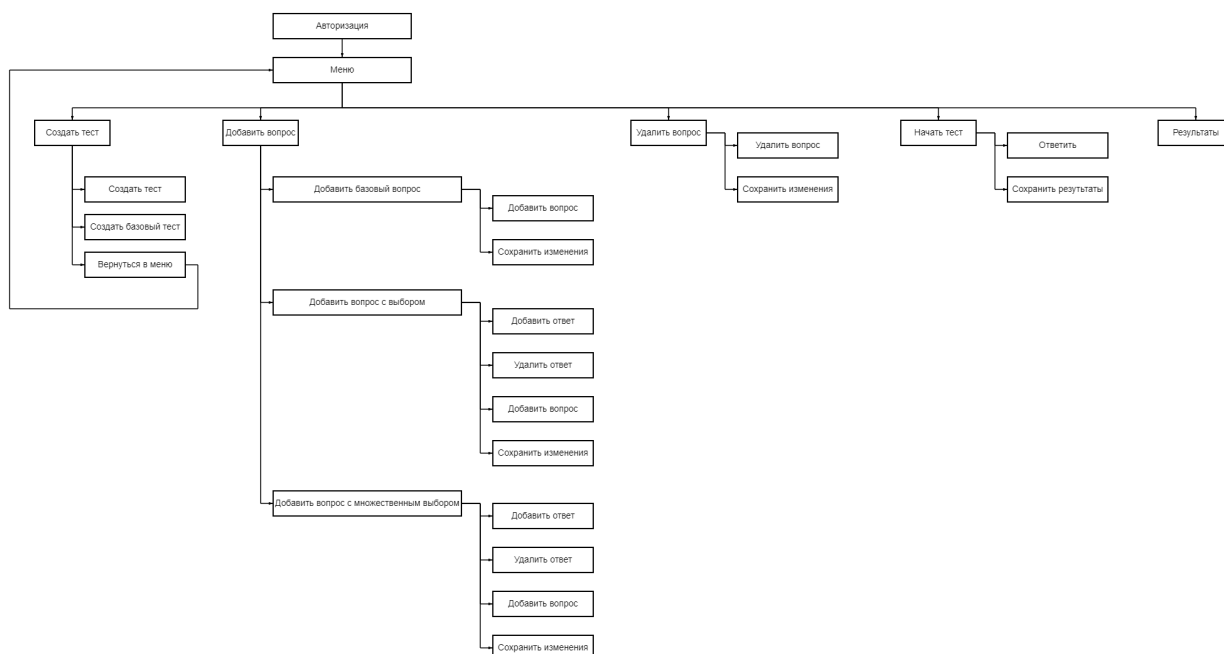


Рисунок 3.1 – Диаграмма компонентов

Любой компонент должен быть вызван в сценарии окна системы тестирования. Окно передает данные компоненту в момент вызова последнего.

На рисунке 3.2 представлена схема обмена данными между сценариями компонента при вызове компонента на текущем окне интерфейса.

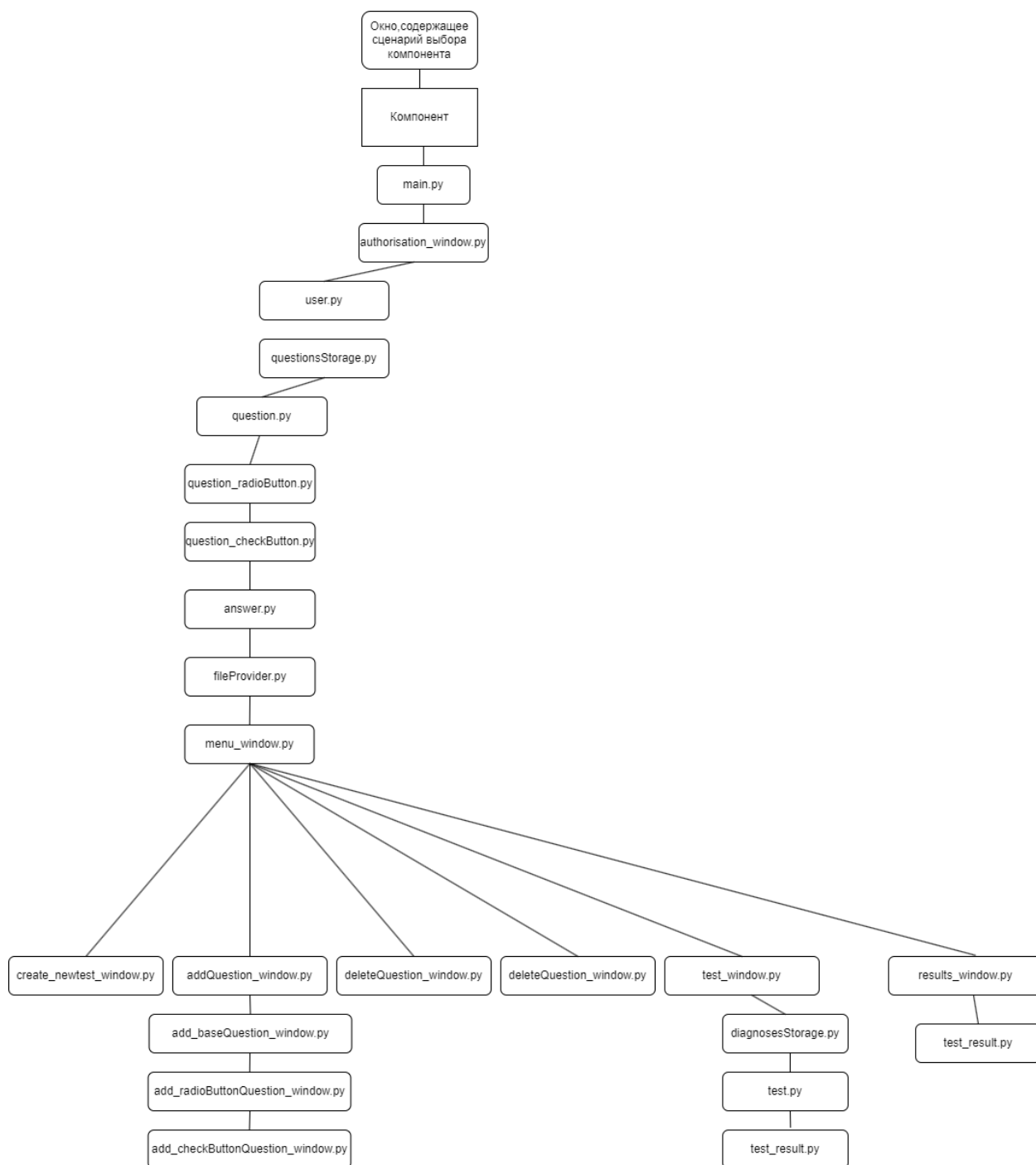


Рисунок 3.2 – Схема обмена

При вызове компонента в сценарии системы тестирования указывается значения имя пользователя и набор вопросов соответствующий выбранному тесту, которые далее посредством ссылок на объекты классов, описывающих данную информацию передаются в следующее окно.

В зависимости от типа вопроса инициализируется соответствующее отображение с вариантами ответа.

Работа компонента заканчивается в момент закрытия главного окна.

Таблица 3.1 – Сравнение языков разметки JSON и XML

	JSON	XML
Удобочитаемость кода	+	-
Простота создания	+	+
Простота использования	+	-
Расширяемость	+	-
Отладка и исправление ошибок	-	+
Безопасность	-	+
Описание	JSON (англ. JavaScript Object Notation) — формат обмена данными, легко читаем людьми, легко обрабатывается и генерируется программами.	XML - это язык разметки подобный HTML. Расшифровывается как (англ. Extensible Markup Language - Расширяемый Язык Разметки) и является рекомендацией сообщества W3C в качестве языка разметки общего назначения (W3C recommended).

3.4 Содержание информационных блоков. Основные сущности

Проанализировав требования, можно выделить шесть основных сущностей:

- «Пользователь»;
- «Вопрос»;
- «Ответ»;

- «Тест»;
- «Результат»;
- «Диагноз»;

В состав сущности «Пользователь» можно включить атрибуты, представленные в таблице 3.2.

Таблица 3.2 – Атрибуты сущности «Пользователь»

Поле	Тип	Обязательное	Описание
1	2	3	4
name	String	true	Имя пользователя

В состав сущности «Вопрос» можно включить атрибуты, представленные в таблице 3.3.

Таблица 3.3 – Атрибуты сущности «Вопрос»

Поле	Тип	Обязательное	Описание
1	2	3	4
text	String	true	Текст вопроса
answers	List[Answer]	true	Список вопросов

В состав сущности «Ответ» можно включить атрибуты, представленные в таблице 3.4.

Таблица 3.4 – Атрибуты сущности «Ответ»

Поле	Тип	Обязательное	Описание
1	2	3	4
text	String	true	Текст ответа

1	2	3	4
is correct	Bool	false	Корректность вопроса

В состав сущности «Тест» можно включить атрибуты, представленные в таблице 3.5.

Таблица 3.5 – Атрибуты сущности «Тест»

Поле	Тип	Обязательное	Описание
1	2	3	4
name	String	true	имя теста
questions	List[Question]	true	Список вопросов
diagnoses storage	DiagnosesStorage	false	Хранилище диагнозов
score	Integer	false	Счёт
is started	Bool	false	Состояние теста
current question id	Integer	false	ID текущего вопроса
current answers	List[Answer]	false	Список ответов текущего вопроса

В состав сущности «Результат» можно включить атрибуты, представленные в таблице 3.6.

Таблица 3.6 – Атрибуты сущности «Результат»

Поле	Тип	Обязательное	Описание
1	2	3	4
user	User	true	Пользователь
right answers count	Integer	true	Количество правильных ответов
completion time	datetime	false	Время завершения теста

В системе предусмотрен внутренний механизм связи между разделами и элементами информационных блоков, поэтому введения дополнительных идентификаторов при реализации связей между сущностями не предполагается.

Экземпляры сущностей реализуются в информационных блоках посредством элементов, атрибуты сущности – посредством полей и свойств элемента.

3.5 Проектирование пользовательского интерфейса

На основании требований к пользовательскому интерфейсу, представленных в пункте 2.7 технического задания, был разработан графический интерфейс десктопного приложения. Для создания пользовательского интерфейса используется библиотека Custom Tkinter. На рисунке 3.3 представлен макет интерфейса вкладки «Авторизация пользователя». Макет содержит следующие элементы:

1. Текстовое поле для ввода имени пользователя.
2. Текстовое поле для ввода пароля пользователя.
3. Кнопка авторизации, проверяет введённые данные пользователя, если они верны открывает меню пользователя.
4. Кнопка регистрации, открывает окно регистрации пользователя.
5. Кнопка для авторизации администратора.

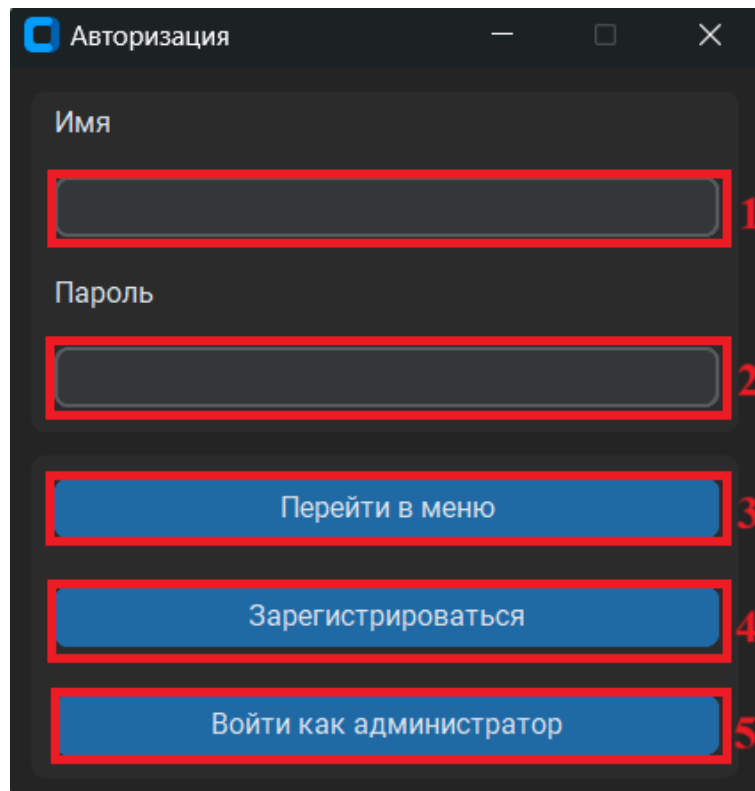


Рисунок 3.3 – Макет авторизации пользователя

На рисунке 3.4 представлен макет интерфейса вкладки «Авторизация администратора». Макет содержит следующие элементы:

1. Текстовое неизменяемое поле с надписью "Администратор".
2. Текстовое поле для ввода пароля администратора.
3. Кнопка авторизации, проверяет введенные данные пользователя, если они верны открывает меню администратора.
4. Кнопка смены пароля администратора.
5. Кнопка для авторизации пользователя.

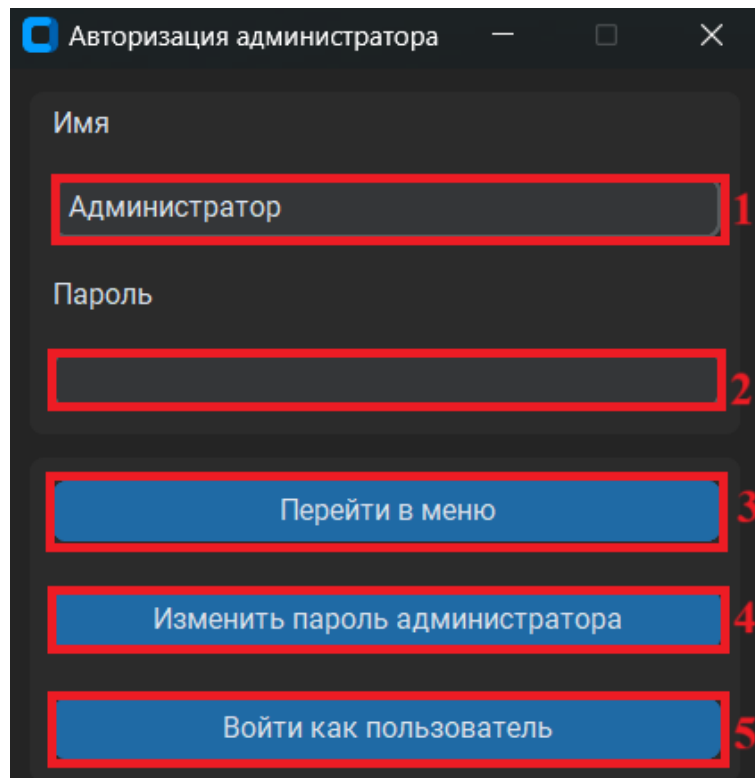


Рисунок 3.4 – Макет авторизации администратора

На рисунке 3.5 представлен макет интерфейса вкладки «Меню пользователя». Макет содержит следующие элементы:

1. Выпадающий список тестов для выбора.
2. Кнопка запуска теста.
3. Кнопка перехода к окну результатов пройденных тестов.
4. Кнопка возврата к авторизации.

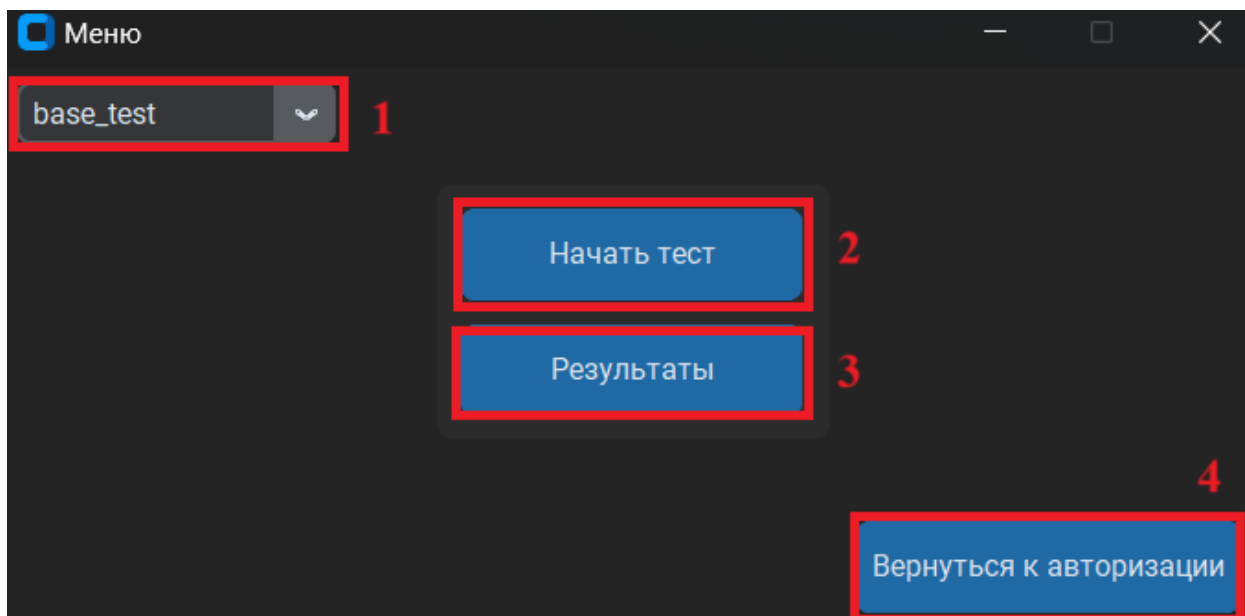


Рисунок 3.5 – Макет меню пользователя

На рисунке 3.6 представлен макет интерфейса вкладки «Меню администратора». Макет содержит следующие элементы:

1. Выпадающий список тестов для выбора.
2. Кнопка создания нового теста.
3. Кнопка добавления нового вопроса в выбранный тест.
4. Кнопка перехода к окну со списком вопросов теста.
5. Кнопка запуска теста.
6. Кнопка просмотра результатов всех пользователей.
7. Кнопка перехода в окно управления пользователями.
8. Кнопка возврата к окну авторизации.



Рисунок 3.6 – Макет меню администратора

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
3. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
4. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
7. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).
9. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестселлеры O'Reilly»).