Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии
Направленность (профиль) образовательной программы Безопасность информационных систем

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«____»_____2019 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка веб-приложения «Система тестирования» на основе фреймворка Laravel

Исполнитель
студент группы 555-об _____ А.Ю. Манвелян
(подпись, дата)

Руководитель
доцент, канд. техн. наук _____ Т.А. Галаган
(подпись, дата)

Консультант
по безопасности и экологичности
доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль
инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Благовещенск 2019

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВПО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«____» _____

З А Д А Н И Е

К выпускной квалификационной работе студента Манвеляна Артема Юрьевича

1. Тема выпускной квалификационной работы работы: Разработка веб-приложения «Система тестирования» на основе фреймворка Laravel.

(утверждена приказом от 15.04.2019 №847-уч)

2. Срок сдачи студентом законченной работы: _____

3. Исходные данные к выпускной квалификационной работе: отчет о прохождении преддипломной практики, нормативная документация, специальная литература.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): обоснование необходимости разработки и определение требований, проектирование программного продукта, оценка надежности и качества функционирования объекта проектирования, руководство пользователя, описание способов защиты информации для программы, обоснование безопасности и экологичности продукта.

6. Консультанты по выпускной квалификационной работе:
по безопасности и экологичности – Булгаков А.Б., доцент, кандидат технических наук.

7. Дата выдачи задания: _____

Руководитель выпускной квалификационной работы: Галаган Т.А., доцент, кандидат технических наук.

Задание принял к исполнению: _____

РЕФЕРАТ

Бакалаврская работа содержит 94 с., 31 рисунок, 14 таблиц, 3 приложения, 20 источников.

ВЕБ-ФРЕЙМВОРКИ, ВЕБ-ПРИЛОЖЕНИЕ, СИСТЕМА ТЕСТИРОВАНИЯ, ТЕСТ, LARAVEL, PHP, ОБЪЕКТИВНАЯ ОЦЕНКА ЗНАНИЙ, БАЗА ДАННЫХ

Объектом исследования данной выпускной квалификационной работы является разработка современных веб-приложений с помощью веб-фреймворков.

Целью выпускной квалификационной работы является изучение построения приложений с использованием современных веб-технологий, а именно фреймворков, и разработка системы тестирования построенной на основе фреймворка Laravel.

Выполнение работы выполняется в несколько стадий. Первая стадия – исследование предметной области и сравнение с другими похожими продуктами. Вторая стадия – проектирование приложения, определение целей, функционала и требований. Третья стадия – описание разработки веб-приложения, программная реализация продукта. Четвёртая стадия – раскрытие обеспечения информационной безопасности веб-приложения. Пятая стадия – описание безопасности и экологичности.

В результате было разработано и находится в тестировании веб-приложения «Система тестирования», которое позволяет полностью отказаться от ручного создания и проверки тестов.

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

Федеральный закон от 27.07.2006 N 152-ФЗ (ред. от 31.12.2017) "О персональных данных"

СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Требования к помещениям для работы с ПЭВМ.

СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Требования к ПЭВМ.

ГОСТ 30772-2001. Межгосударственный стандарт. Ресурсосбережение. Обращение с отходами. Термины и определения.

ГОСТ 30772-2001. Межгосударственный стандарт. Ресурсосбережение. Обращение с отходами.

Федеральный закон от 21.12.1994 №69-ФЗ (в ред. От 23.06.2016) «О пожарной безопасности».

Постановление Правительства РФ от 25.04.2012 N 390 (ред. от 07.03.2019) "О противопожарном режиме".

Приказ МЧС РФ от 18.06.2003 N 313 "Об утверждении Правил пожарной безопасности в Российской Федерации (ППБ 01-03).

ВВЕДЕНИЕ

Основные технологии и стратегические усилия, стоящие за разработками веб-приложений не так важны для конечного пользователя, но они являются определяющими с технической точки зрения. Чтобы производить хорошо функционирующие, масштабируемые и устойчивые веб-приложения, разработчики должны знать и уметь применять технологические новинки, последние и самые выдающиеся фреймворки, а также следить за конкурентами на рынке. Веб-разработка должна поддерживаться лучшими и последними фреймворками, чтобы ускорить производственный процесс и успевать к времени сдачи проектов.

Объект работы – разработка современных веб-приложений с помощью фреймворков.

Предмет работы – разработка веб-приложения «Система тестирования» для фреймворка Laravel.

Выпускная квалификационная работа преследует цель – изучение построения приложений с использованием современных веб-технологий, а именно фреймворков, и соответственно разработка бесплатной и открытой системы тестирования построенной на основе развивающегося серверного php фреймворка Laravel, который позволил бы в дальнейшем с лёгкостью расширять и модернизировать проект. В качестве основного инструмента разработка клиентской части выступает Vue.js — JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов.

Основными задачами является:

- реализация модуля панели управления системой с использованием управления доступом на основе ролей и соответственно создание роли администратора, имеющего полный доступ ко всему контенту, роли условного преподавателя и тьютора;
- создания модулей для управления вопросами тестов и записи студентов для прохождения тестирования;

- создание личного кабинета студента с возможностью проходить тесты и смотреть результаты;
- создание и описание базы данных;
- ссоздание REST API для получения данных с СУБД;
- внедрение OAuth2 авторизации;

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРИ РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЙ «СИСТЕМА ТЕСТИРОВАНИЯ»

1.1 Актуальность веб-фреймворков для разработки современных защищенных интернет-приложений

Веб-фреймворки сильно повлияли на мир программирования и стали необходимы для быстрой и качественной разработки. Веб-фреймворк — инструмент, облегчающий процесс написания и запуска веб-приложения [1].

В начале эпохи веб-разработки все приложения писались вручную, и только непосредственно разработчик мог изменить или развернуть их. Веб-фреймворки позволили выбраться избавиться от этого недостатка. В середине 90-ых все сложности связанные с изменением приложения была приведены в порядок с помощью появлению общего подхода к разработке веб-приложений. В это время появились языки для интернет-разработок. Сейчас их разнообразие позволяет выбрать подходящий как для любых типов сайтов. В зависимости от поставленной задачи, вы можете выбрать один фреймворк, покрывающий все нужды, или совместить несколько.

У фреймворков есть две основные функции: работа на серверной стороне (бэкенд) и работа на клиентской стороне (фронтенд). Следует выделить следующие типы веб-фреймворков:

- Бэкенд-фреймворки(серверные), которые в основном отвечают за отдельные, но очень важные части приложения, без которых оно не сможет нормально работать.
- Фронтенд-фреймворки(клиенсткие), которые отвечают за внешний вид приложения.

Правила и архитектура серверных фреймворков не даёт возможности создать веб-приложение с динамическим и богатым интерфейсом. Они ограничены в своей функциональности, могут создавать простые страницы и разные формы, также они могут формировать выходные данные и отвечать за безопасность в случае атак. Их основное предназначение состоит в управлении всем на

закрытой серверной части веб-приложения. Бэкенд-фреймворки отвечают за отдельные, но критически важные части приложения, без которых оно не сможет нормально работать. Вот несколько самых популярных фреймворков и языки, с которыми они работают:

- Django – Python;
- Laravel – PHP;
- Express.js – JavaScript;
- Ruby on Rails – Ruby.

В отличие от серверных, клиентские фреймворки никак не связаны с логикой приложения. Этот тип фреймворков работает на стороне пользователя в браузере. С их помощью можно улучшить и внедрить новые пользовательские интерфейсы. Фронтенд-фреймворки позволяют создавать разные анимации и так называемые “Single-page application” . Все клиентские фреймворки отличаются по функциональности и использованию. Вот некоторые самые популярные из них:

- AngularJS;
- React.js;
- Vue.js.

Все эти фреймворки используют JavaScript в качестве языка программирования.

Также существуют многофункциональные фреймворки, например, Meteor, который известен как фулл-стек веб-фреймворк. Он удовлетворяет почти все потребности как со стороны клиента, так и со стороны сервера. Обе стороны – серверная и клиентская – работают на одном языке, и для них используется один и тот же код. Так же он имеет особенность – «режим реального времени» – когда вы что-то меняете в одном интерфейсе, изменения происходят и в остальных.

Архитектура почти всех фреймворков основана на разделении на несколько отдельных модулей, что означает, что вы можете расширять функциональность по мере необходимости и использовать изменённую версию вместе с

кодом фреймворка или использовать сторонние приложения(пакеты). Такая гибкость является ещё одним ключевым преимуществом фреймворков и делает их очень весьма актуальными по сравнению затратами по написанию веб-приложений вручную, и конечно все актуальные фрейморков постоянно обновляются, что говорит о их популярности и востребованности, а вместе с их бесплатностью делает изучение и использование хотя бы одного клиентского и серверного фреймворка необходимостью в современном мире веб-разработок. Существует множество open-source сообществ и коммерческих организаций, которые создают расширения для популярных фреймворков.

Большинство фреймворков в полной степени и частично реализуют модель MVC, что, несомненно, очень удобно и позволяет строить хорошо структурированные и расширяемые интернет-приложения.

Фреймворки обладают отличительными особенностями, которые делают их многофункциональными и удобными при использовании.

Рисунок 1.1 – Особенности веб-фрейморков

Система веб-шаблонов представляет собой набор разных методологий и программного обеспечения, реализованных для создания и развёртывания веб-страниц. Для обработки веб-шаблонов используются шаблонизаторы, являющиеся инструментом фреймворка и отвечающие за публикацию веб-страниц.

Технология сопоставление URL призванная упростить индексацию вашего сайта поисковыми движками, создавая привлекательное название для сайта. Также сопоставление URL может облегчить доступ к страницам вашего сайта.

Веб-кэширование просто помогает хранить разные документы и позволяет избежать перегрузки сервера. Кэширование используется в разных системах и при определённых условиях. Кэшированные веб-страницы можно увидеть в результатах поиска многих поисковых систем.

Скаффолдинг-технология, поддерживается некоторыми MVC-фреймворками. Фреймворк может автоматически сгенерировать типичные части приложения или даже всю струкутру проекта. Это позволяет существенно увеличая скорость разработки и стандартизирует кодовую базу.

Все фрейморки имеют достаточное количество документов, библиотек и руководств, призванных помочь изучить их. Так же существует множество сайтов, официальных для каждого фрейморка и общих для всех, которые предоставляют быстрое введение в любой фреймворк и предоставляют форум и сообщество программистов, активно помогающих начинающим.

Например, Tutorialspoint — собрание разных руководств, покрывающих структуру каждого фрейморка и предоставляющих информацию по разным деталям.

Чтобы найти ответы на возникающие вопросы стоит заглянуть на StackOverflow. Этим сайтом пользуются разработчики по всему миру. Здесь они делятся своим опытом и помогают другим решать их проблемы.

Большинство фреймворков в полной степени и частично реализуют модель MVC, что несомненно очень удобно и позволяет строить хорошо структурированные и расширяемые интернет-приложения.

Архитектура почти всех фреймворков основана на разделении на несколько отдельных модулей, что означает, что можно расширять функциональность по мере необходимости и использовать изменённую версию вместе с кодом фреймворка или использовать сторонние приложения(пакеты). Существует множество open-source сообществ и коммерческих организаций, которые создают расширения для популярных фреймворков. Такая гибкость является ещё одним ключевым преимуществом фреймворков и делает их очень весьма актуальными по сравнению затратами по написанию веб-приложений вручную, и конечно все актуальные фрейморков постоянно обновляются, что говорит о их популярности и востребованности, а вместе с их бесплатностью делает изучение и использование хотя бы одного клиентского и серверного фреймворка необходимостью в современном мире веб-разработок.

1.2 Особенности разработки веб-приложений на фрейморке laravel

Разработка интернет-проекта на Laravel, включает в себя следующие этапы:

- определение ключевых целей и задач;
- формирование окружения (архитектура, код, база данных) для конкретного проект. На данном фреймворке работа над архитектурой сайта ведется по принципу «от основного к второстепенному» (иерархическая логика). Заполнение и настройка базы данных, и написание кода осуществляется с использованием системы Eloquent;
- разработка дизайна проекта, и последующая его привязка к функционалу;
- наполнение веб-ресурса качественным и количественным контентом;
- тестирование работоспособности и исправление недоработок;
- размещение на хостинге и запуск проекта;
- сопровождение и техподдержка.

Перечень основных преимуществ

Разработка сайта на фреймворке Laravel не сильно отличается от интернет-ресурсов на фрейморке Symphony, компоненты которого включает Laravel и при этом имеет ряд достоинств:

- широкий и разнообразный функционал;
- возможность создавать масштабные интернет-проекты, независимо от сложности и направленности, в том числе и многоуровневые веб-сайты;
- возможность красиво решать самые нестандартные задачи;
- возможность тестировать, добавлять обновления, вносить изменения в работу сайта на дополнительной версии. Достигается за счет поддержки различных версий интернет-ресурса;
- простая и понятная система пакетов, что позволяет находить почти любые необходимые компоненты для разрабатываемой системы и освобождает от их написания с нуля;
- надежная защита базы данных от SQL, CSRF, XSS;
- все изменения в PHP и направлений развития веб-сайтов обязательно учитываются в обновлениях исходного кода данного фреймворка и во всех последующих версиях;
- возможность масштабирования проекта.

К недостаткам разработки сайта на фреймворке можно отнести следующие факторы:

- создание сайта занимает больше времени;
- рассчитано больше для профессиональных разработчиков.

В итоге разработка сайта на framework Laravel – это свобода творчества и возможность реализовать практически любой интернет-проект, независимо от его направленности и сложности. Объясняется наличием большого выбора дополнений, а также способностью настройки под очень высокие нагрузки.

1.3 Конфиденциальность и защита персональных данных при разработки многопользовательской веб-системы

Персональные данные – любая информация, относящаяся к прямо или косвенно определенному, или определяемому физическому лицу [2]. Большинство веб-приложений тем или иным образом обрабатывает данные пользователей для различных целей и соответственно их конфиденциальность является обязательным пунктом для соблюдения администрацией сайта или иным лицам, получившим доступ к персональным данным. Также обязательно требование не допускать их распространения без согласия субъекта персональных данных или наличия иного законного основания.

Для соблюдения конфиденциальности и соответственно предотвращения кражи и несанкционированного доступа к данным третьими лицами существуют различные способы защитить веб-приложения:

Валидация входящих данных

Во время проектирования приложения нужно стремиться защитить его от сомнительных входящих данных. Правило, которому нужно следовать, звучит примерно так: “Никогда не верьте тому, что ввёл пользователь”. Хотя большинство пользователей не представляет угрозы, но есть определённый ряд лиц, которые могут попытаться взломать приложение и для предотвращения этого рекомендуется валидация данных как со стороны клиента, так и со стороны пользователя. Laravel предлагает несколько разных подходов к проверке входящих данных. По умолчанию базовый класс контроллера использует типаж ValidatesRequests, который предоставляет удобный метод проверки входящего HTTP-запроса с помощью различных мощных правил проверки;

Защита от XSS атак

Межсайтовый скриптинг или XSS-атака — это атака, основанная на внедрении кода на потенциально уязвимых страницах. Опасность в том, что вредоносный код может быть введён через формы, а затем отображён в браузере.

Для защиты ваших приложений, пропускайте входящие данные через функцию, которая удалит все присутствующие теги и исключит внедрение вредоносного кода. Также существует ещё один инструмент от данного вида атак - Политика безопасности содержимого (CSP). CSP - заголовки сервера, определяющие белый список источников, откуда разрешена загрузка данных для разных типов ресурсов. Благодаря политикам CSP, даже при внедрении вредоносного кода в страницу его выполнение становится невозможным;

Защита от CSRF атак

Следующий вид атаки – это CSRF атака или подделка межсайтовых запросов. Атакующий использует различные трюки для получения конфиденциальной информации или совершения сделки без ведома жертвы. В основном, это происходит на плохо защищённых сайтах, логика строится на GET запросах, которые не предоставляют защиту от и не скрывают данные.

В качестве предотвращения подобного рода атак, используются только POST запросы с генерацией уникальных сессионных CSRF-токенов к тем процессам, которые предназначены для изменения информации в базе данных. Laravel так же предоставляет простой способ защиты вашего приложения от подделки межсайтовых запросов;

Предотвращение SQL инъекций

SQL-инъекция представляет собой выполнение произвольного запроса к базе данных приложения с помощью поля формы или параметра URL. В случае использования стандартного языка SQL и функции mysqli возможно вставить вредоносный код. В результате чего будут получены, изменены или удалены данные таблиц. Чтобы предотвратить это, используйте параметризованные запросы, которые поддерживаются большинством языков веб-программирования. Но большинство фреймворков и, в частности, Laravel делает взаимодействие с базами данных чрезвычайно простым в самых разных серверных базах данных, предоставляя простой в освоении построитель запросов и модуль для работы с моделями базы данных.

Кроме удобного синтаксиса данные модели «из коробки» предоставляют защита от различного рода SQL-инъекций, что позволяет больше сосредоточится на создании приложений, а не на обработки каждого запроса;

Защита данных сессии

Сессии – способ сохранения информации о пользователе между отдельными запросами.

При написании системы с нуля вся информация сессий будет записывается в каталог temp. Если работа происходит на виртуальном хостинге, то кто-то помимо администратора может написать скрипт и считать данные сессий. Поэтому не стоит хранения паролей или номеров кредиток в сессиях.

Если же всё-таки необходимо хранить подобные данные в сессии, то лучшей мерой будет шифрование. Это до конца не решает проблему, но зашифрованная информация становится нечитабельной. В PHP есть специальный метод session_set_save_handler(), который позволяет хранить данные сессий по-своему.

Laravel поставляется со множеством различных механизмов сессий, доступных через единый выразительный API, что по умолчанию делает работу с таким важным веб-механизмом защищённой;

Обработка ошибок

Во время разработки приложения могут возникать ошибки, которые от конечных пользователей их нужно скрывать, иначе это делает уязвимым для злоумышленников. Таким образом, лучшим решением будет различная конфигурация для конечного сервера и сервера разработки.

При создании нового проекта Laravel предоставляет уже настроенную обработку ошибок и исключений. Класс, где все исключения, вызванные приложением, регистрируются, а затем оказывается обратно в подробном виде, при этом существует 2 режима работы: для разработчиков и пользователей, что даёт возможность на стадии релиза отключить режим отладки и огородить пользователей от показа каких-либо ошибок и соответственно обезопасить сайт от атак;

Шифрование паролей

Хранить пароли стоит в виде хэша, причём лучше использовать алгоритмы одностороннего хеширования, которые реализованы в Laravel с помощью функции bcrypt . При таком способе для авторизации пользователей сравниваются хешированные значения. Если злоумышленник взломает ресурс и получит хешированные пароли, ущерб будет снижен за счёт того, что хэш имеет необратимое действие и получить из него исходные данные практически невозможно. Но хэши на популярные пароли легко перебираются по словарю, поэтому следует также использовать «соль», уникальную для каждого пароля. Тогда взлом большого количества паролей становится ещё медленнее и требует больших вычислительных затрат;

Контроль процесс загрузки файлов

Загрузка пользователем файлов на веб-сайт, даже если это просто смена фотографии профиля, несёт в себе угрозу информационной безопасности. Загруженный файл, который, на первый взгляд, выглядит безобидно, может содержать скрипт и при выполнении на сервере откроет злоумышленнику доступ к сайту.

Чтобы это предотвратить, нужно запретить исполнение загружаемых файлов пользователями.

Также существуют меры защиты сервера от подобного вида опасностей:

- настройка межсетевого экрана, в том числе на блокировку неиспользуемых портов;
- использование защищённых методов (SFTP, SSH и др.) для передачи файлов и управления сервером извне;
- выделение отдельного сервера для баз данных, который не будет напрямую доступен из внешнего мира;
- отграничение физического доступа к серверу;
- правильное распределение прав доступа к файлам;
- разрешения файла определяют КТО и ЧТО может с ним делать.

Правильным выставлением прав доступа будет: Установка владельцу прав доступа на чтение и запись, группе — на чтение, прочим — запрет доступа, однако часто из-за администраторы выставляют права полного доступа на все файлы всем чтобы сократить время настройки. Этот совет помогает сэкономить время, но открывает серьёзную уязвимость, потому что всем появляется право изменить (вставить вредоносный код) или удалить файлы на сервере;

Запрет большого количества функций неавторизованным пользователям

Данное ограничение позволит отсеять всех заинтересованных пользователей в контенте определённого веб-приложения, так как придётся проходить регистрацию и подтверждать e-mail адрес. Laravel предоставляет удобную систему регистрации пользователей, валидации отправленных ими данных, фильтрации маршрутов, что полезно при создании областей сайта только для авторизованных пользователей, и подтверждения e-mail адреса. Также существует возможность авторизоваться с помощью социальных сетей, что тоже подтверждает реальность пользователя.

Помимо механизма простой аутенфикации в Laravel реализована API аутентификация с помощью Laravel Passport, который предоставляет полную реализацию сервера OAuth2 для вашего приложения.

1.4 Обоснование необходимости создание «Системы тестирования» и сравнение с другими схожими веб-приложениями.

При анализе рынка подобных приложений для фреймворка Laravel построенных на версии Laravel 5.5 и выше, так как ранние версии уже актуальны, и имеющие обновления хотя бы раз в несколько месяцев не было выявлено ни одного похожего бесплатного приложения. Но на одной из торговых площадок в данный момент продается система тестирования стоимостью в 15 долларов, что примерно эквивалентно 1000 рублей, которая использует Laravel 5.5 и фронтенд фрэймворк Vue.js. При ознакомлении с данным продуктом оказалось, что он реализует базовые функции концепции подобной системы, при этом используя отзывчивый дизайн, панель администратора и многие возможности Laravel.

При реализации приложения, которое должно получится в итоге выполнения выпускной квалификационной работы, используется последняя версия фреймморка на данный момент Laravel 5.5, фрэймморк Vue.js, а также другие инструменты, которые в итоге реализуют отзывчивый дизайн для разных устройств. Также планируется использование панели администратора и набор куда более широкого спектра возможностей, что вместе с полной бесплатностью делает реализуемую в данной ВКР приложение куда более привлекательным предложением.

Если рассматривать другие предложения на рынке, построенные не на Laravel, но являющимися веб-приложениями, то стоит отменить модульную объектно-ориентированную динамическую обучающую среду Moodle, свободное веб-приложение, предоставляющее возможность создавать курсы/сайты для онлайн -обучения. Moodle не является системой для тестирования и по умолчанию не имеет такого функционала, но может использовать внешние приложения, внедрённые через свой API или же будет необходимо писать модуль вручную с нуля, что достаточно трудозатратно. Существуют и другие системы дистанционного обучения, например, Blackboard Learn и Mirapolis Virtual Room, но они являются платными и также «из коробки» не реализуют подобный функционал в полной мере. Стоит также отметить, что разработка на Laravel представляет собой современное веб-приложение, реализующее последние стандартны, защита данных, политики конфиденциальности, постоянные обновления фреймворка позволяют быть всегда «в ногу со временем», чего не скажешь о других системах, построенных «с нуля», по типу Moodle.

Также был проведён анализ конкретно систем тестирования. Из наиболее популярных программ для тестирования можно отметить:

MyTestXPro

Комплекс программ (программа тестирования учащихся, редактор тестов и журнал результатов) для создания и проведения компьютерного тестирования, сбора и анализа результатов, выставления оценки по указанной в тесте шкале [2].

NetTest

Программный комплекс NetTest предназначен для проведения компьютерного тестирования знаний. Программа состоит из двух программ – серверной и клиентской. Серверная программа позволяет создавать и редактировать тесты и управлять компьютерным тестированием, обработкой и выводов результатов. Клиентская часть запускается с рабочих станций и предназначена для работы учащегося [3].

«АСТ-Тест»

Комплекс программ для создания банков тестовых заданий, организации и проведения тестирования, обработки результатов и формирования выходных документов в удобной форме [4].

«Айрен»

Это бесплатный программный комплекс, позволяющий создавать тесты для проверки знаний и проводить тестирование в локальной сети, через Интернет или на одиночных компьютерах [5].

Проведём сравнение механизмов тестирования в программных продуктах. Разрабатываемая в работе система носит название TaSFoL, что является аббревиатурой от testing system for Laravel.

Таблица 1.1 – сравнение механизмов тестирования в программных комплексах

Критерий	MyTest XPro	NetTest	«АСТ-Тест»	«Айрен»	TaSFoL
Вопросы с выбором одного верного ответа	+	+	+	+	+
Вопросы с выбором нескольких верных ответов	+	+	-	+	+
Вопросы с вводом ответа	+	+	+	+	+
Вопросы на установление соответствия	+	-	+	+	+
Вопросы с аудио/видеосодержанием	+	-	+	-	-

В итоге, можно сделать вывод о том, что аналогичные или схожие приложения, или являются недостаточно функциональными в этой сфере, или же имеют необходимый минимальный функционал для проведения тестов, но дизайн приложений выглядит довольно скупо, практически отсутствует современная графическая оболочка.

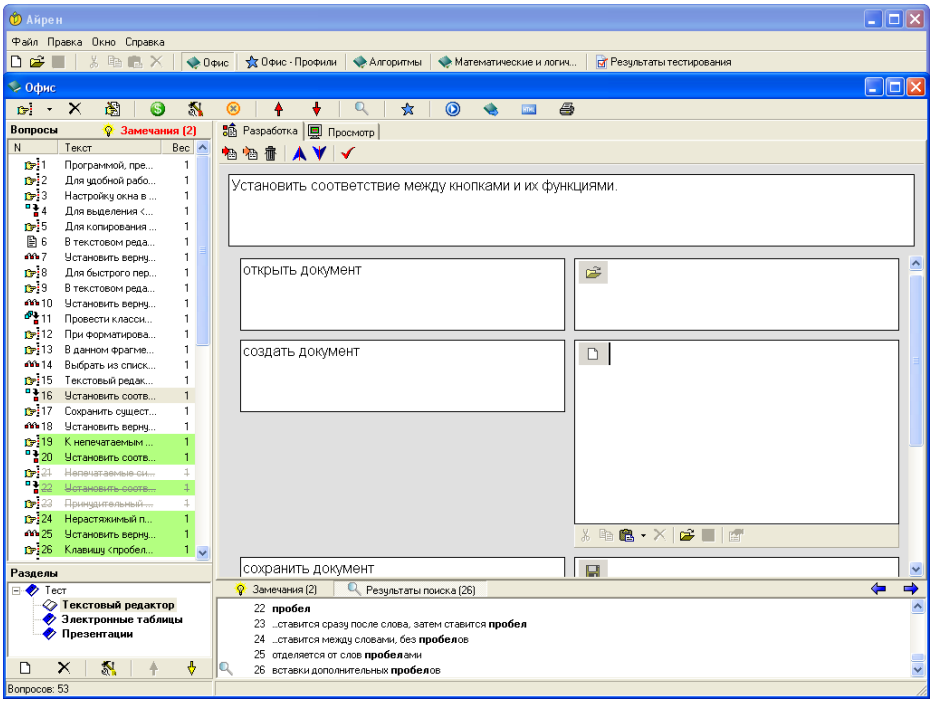

Рисунок 1.2 – Пример интерфейса программы Айрен

Почти все программные продукты имеют недостатки в виде скорости работы или необходимостью платить за них, что делает веб-приложение «система тестирования» построенное на актуальной версии Laravel, которая в дальнейшем будет активно обновляться, довольно конкурентоспособным и необходимым на рынке веб-приложений, сделанных для целей проведения различных тестирований.

2 ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ «СИСТЕМА ТЕСТИРОВАНИЯ»

2.1 Цели, функции системы

Основная цель системы заключается в разработке приложения для объективной оценки знаний с помощью современного фреймворка Laravel. Целью обусловлена тем, что бесплатных и открытых проектов для тестирования различных групп людей под данный фреймворк на данным момент не существует, а необходимость в объективной оценке знаний, будь то сотрудники компании или студенты в университете, существует всегда. Сложные системы управления обучением хоть и предоставляют возможность тестирования, но являются громоздкими и сложными для понимания и не всегда предоставляют интуитивно-понятный интерфейс.

Основной функционал, реализуемый в системе:

- создание тестов с различными типами вопросов;
- разбиение на группы и создание аккаунтов студентов для входа в личный кабинет;
- мгновенное подведение результатов тестирования;
- показ статистики по тестированиям;
- использование надежных методов аутентификации и авторизации пользователей;
- возможность внешней авторизации;
- возможность задания прав для различных групп пользователей на основе ролевой модели разграничения доступа.

Также к системе предъявляются требования, с описанием которых можно подробно ознакомиться в техническом задании к проекту, которое находится в приложении Б.

2.2 Выбор направления проектирования

В качестве модели жизненного цикла была выбрала спиральная стратегия, которая подразумевает разработку в виде последовательности версий, но в

начале проекта определены не все требования. Требования уточняются в результате разработки версий.

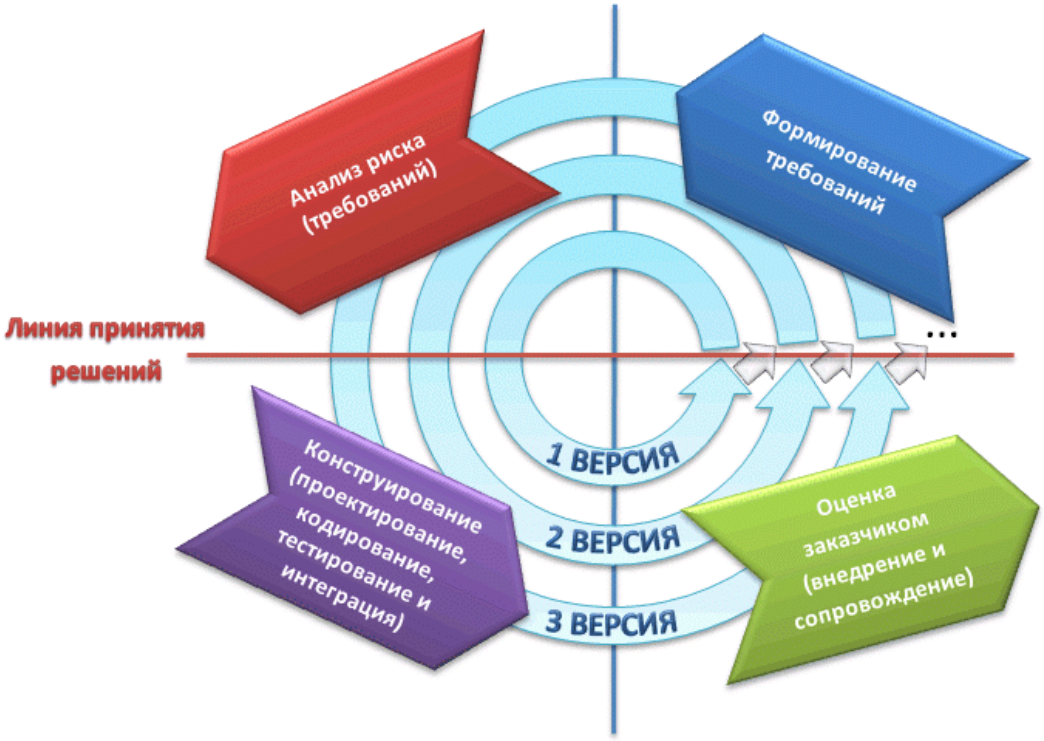

Рисунок 2.1 – Спиральная стратегия

Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Как видно из рисунка 5, развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска. Данная модель идеально подходит для разработки, так как система и, в частности, панель управления не имеют чёткий границ функциональности и расширения, новый функционал может быть придуман и внедрён по ходу разработки, различные версии продукта появляются довольно часто, чтобы можно было оценить уже проделанную работу и уточнить нюансы, внести изменения.

При использовании спиральной модели жизненного цикла наиболее известными методологиями являются методология быстрой разработки приложений (Rapid Application Development, RAD) и экстремальное программирование (eXtreme Programming, XP).

Под RAD-разработкой обычно понимается процесс разработки, содержащий 3 элемента:

- небольшую команду программистов (до 10 человек);
- короткий, но тщательно проработанный производственный график (от 2 до 6 месяцев);
- повторяющийся цикл, при котором разработчики по мере того, как приложение начинает обретать форму, реализуют в продукте требования, полученные через взаимодействие с заказчиком.

Так же методология предполагает использование различных CASE и инструментальных средств, а также шаблонов и библиотек готовых решений как собственной разработки, так и сторонних производителей.

Методология XP. Данный подход ориентирован на разработку информационных систем группами малого и среднего размера в условиях неопределенных или быстро изменяющихся требований.

Отличительными особенностями XP-разработки являются:

- частая смена версий и модификаций (длительность итераций вплоть до часов и минут, а обычно – 2 недели; в RAD – минимум 2 месяца);
- непрерывная связь с заказчиком – в группе все время находится квалифицированный представитель заказчика;
- простое проектирование – при разработке всегда выбирается наиболее простое решение. «Лучше сделать простую вещь сегодня и завтра заплатить еще немного за внесение небольших изменений, чем делать сегодня сложную вещь, которая завтра может не понадобиться» [6];

- простой дизайн – система должна быть спроектирована настолько просто, насколько это возможно на каждый момент времени. Чем интерфейс проще, тем быстрее и качественнее идет освоение системы пользователями;
- коллективное владение кодом – любой, кто видит возможность улучшить какую-то часть кода, может сделать это в любой момент времени. Это подразумевает применение одинаковых стандартов и правил оформления кода с исчерпывающими комментариями, а также и ведение общедоступной истории развития системы;
- программирование в парах – на пару программистов приходится один компьютер. Пока один из них непосредственно программирует, другой обдумывает вопросы реализации требований (функций, БД и т.п.).;
- непрерывное и пересекающееся проектирование, разработка, интеграция и тестирование системы.

Рассмотренные выше методологии направлены на сокращение сроков и расходов по разработке приложений с одновременным повышением качества результатов работы. Главное достоинство этих методологий заключается в наиболее полном и точном удовлетворении требований заказчика за счет обеспечения своевременной обратной связи.

В результате рассмотрения двух методологий, предпочтение было отдано XP методологии, так как именно она предполагает частую смену версий и модификаций, простое проектирование и минималистичный дизайн и также освобождает от строго графика выполнения работ.

2.3 Архитектура веб-приложения «Система тестирования»

Архитектура системы состоит из серверных и клиентских компонентов, соединённых общим интерфейсом API.

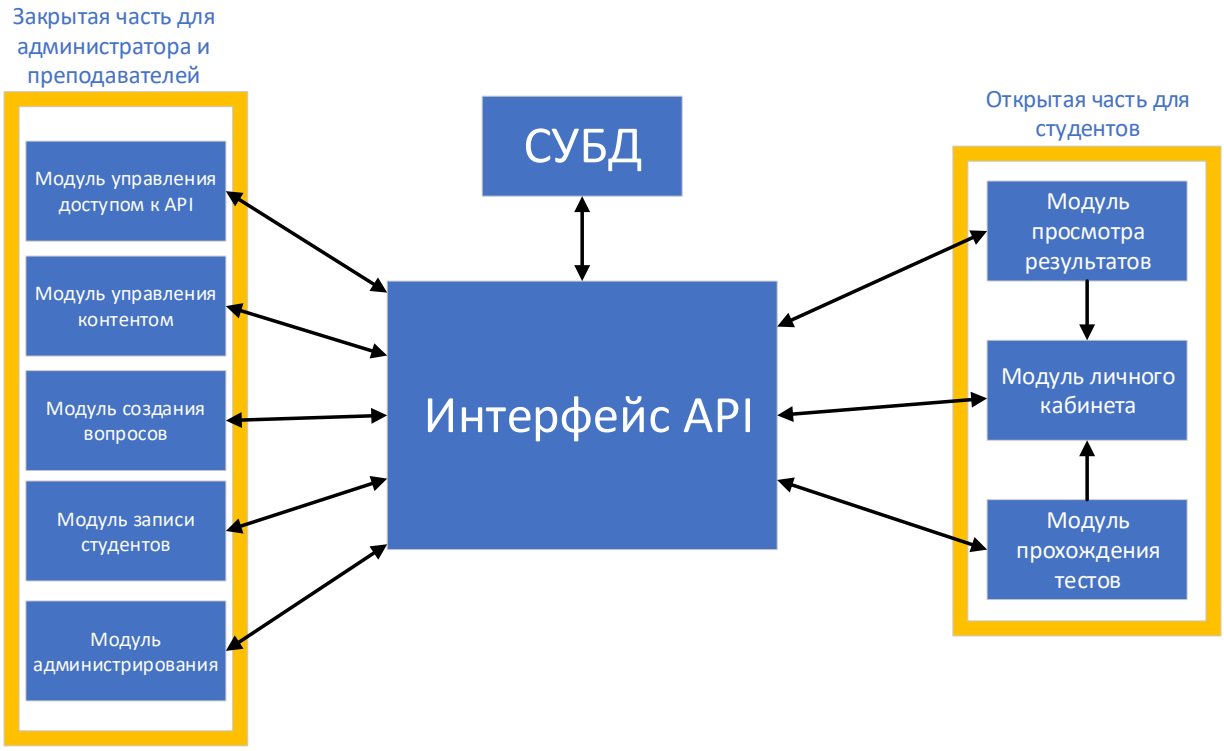

Рисунок 2.2 – Взаимодействие модулей в веб-приложении «Система тестирования».

Как видно из рисунка система разделяется на панель управления контентом, являющаяся закрытой частью для администратора и преподавателей и открытой частью, в которой проходят тесты обучающиеся или сотрудники организации.

2.3.1 Функциональные подсистемы

2.3.1.1 Интерфейс API.

API представляет собой промежуточное звено между СУБД и клиентской частью приложения, которое обеспечивает взаимодействие модулей с базой данных и объединяет их в единую систему.

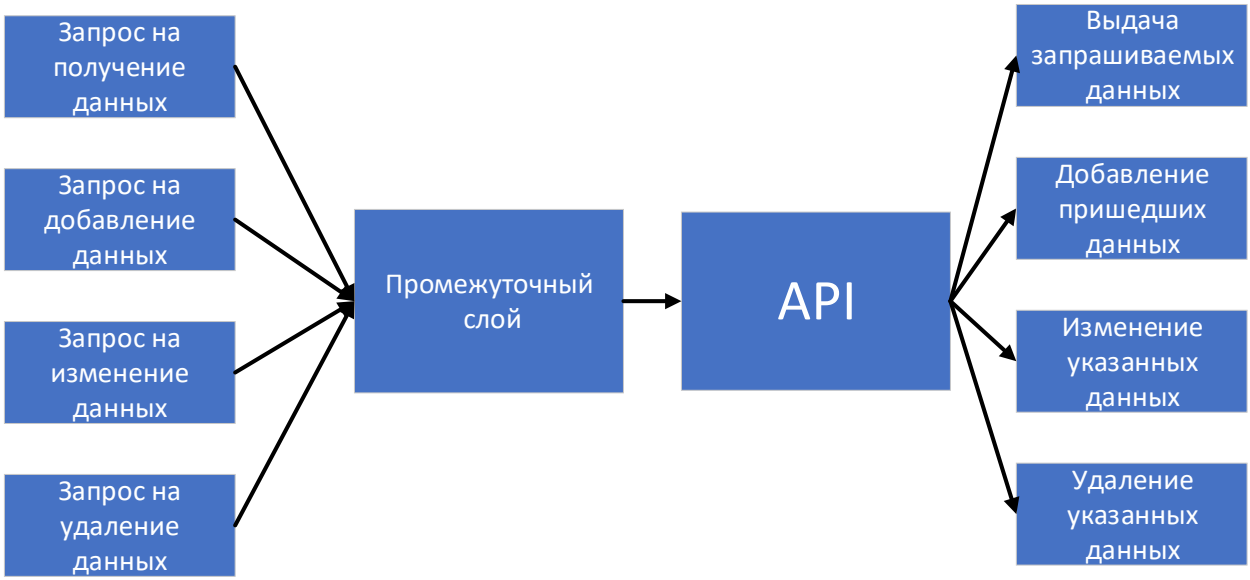

Рисунок 2.3 – Схема работы API системы

Как видно из рисунка 2.3 API получает внешний запрос на одно из действий, таких как получение, добавление, обновление и удаление. Далее запрос проходит через промежуточный слой, который является авторизацией для проверки подлинности пользователя, так, например это может быть токен авторизации. После проверки пользователя API выполняет соответствующий запрос и возвращает ответ с кодом успешного выполнения или ошибки.

2.3.1.2 Модуль панели управления

Конструктор вопросов является отдельным модулем панели управления, который позволяет быстро и удобно создавать вопросы различных типов.

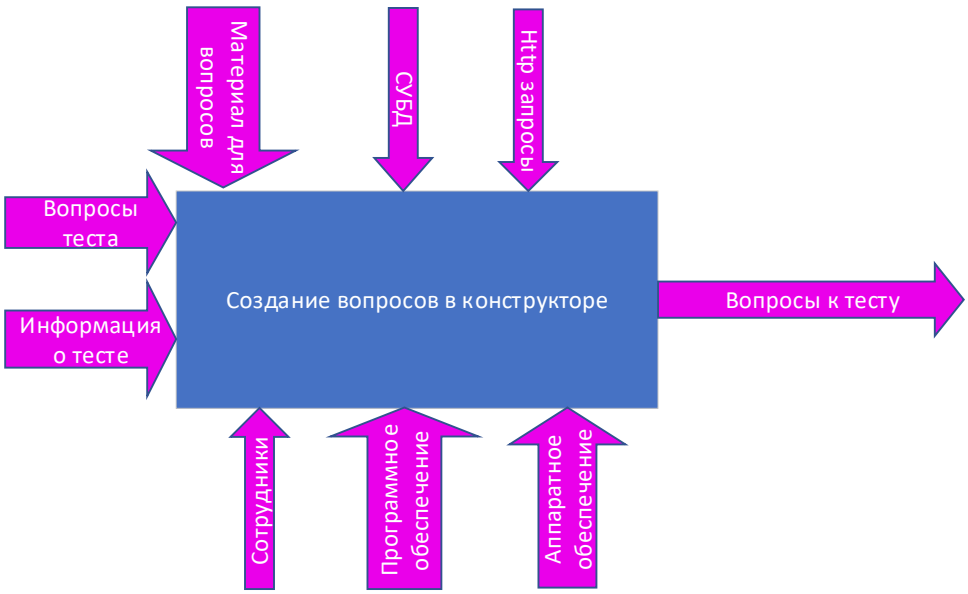

Рисунок 2.4 – Функциональная схема IDEF0 конструктора вопросов

В приложении Б.1 можно подробнее рассмотреть декомпозицию функциональной схемы IDEF0 конструктора вопросов

2.3.1.3 Модуль записи студентов для прохождения тестов

Конструктор вопросов является отдельным модулем панели управления, который позволяет быстро и удобно создавать вопросы различных типов.

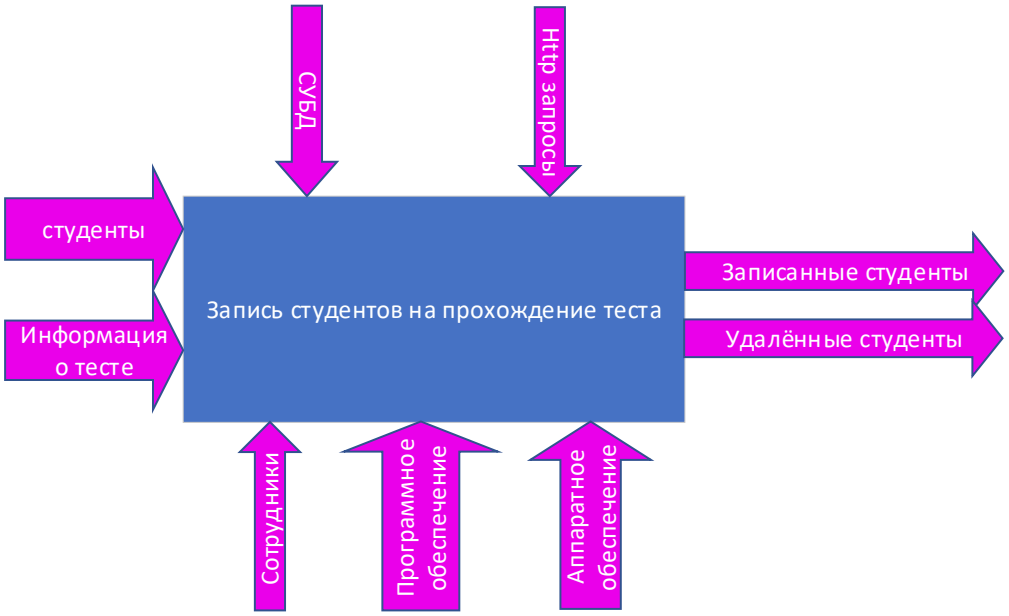

Рисунок 2.5 – Функциональная схема IDEF0 модуля записи студентов

В приложении Б.2 можно подробнее рассмотреть декомпозицию функциональной схемы IDEF0 модуля записи студентов на прохождение тестов.

2.3.1.4 Модуль OAuth2 авторизации

API работает с протоколом OAuth2, который предоставляет возможность внешней доверенной авторизации различных устройств.

Рисунок 2.6 – Абстрактное описание протокола AOuth2

2.3.1.5 Модуль личного кабинета пользователя

Модуль личного кабинета является отдельным модулем открытой части системы, который взаимодействует с модулями прохождения тестов и просмотра результатов и является связующим звеном между ними. Он позволяет авторизоваться и удобно проходить тестирование и просматривать результаты.

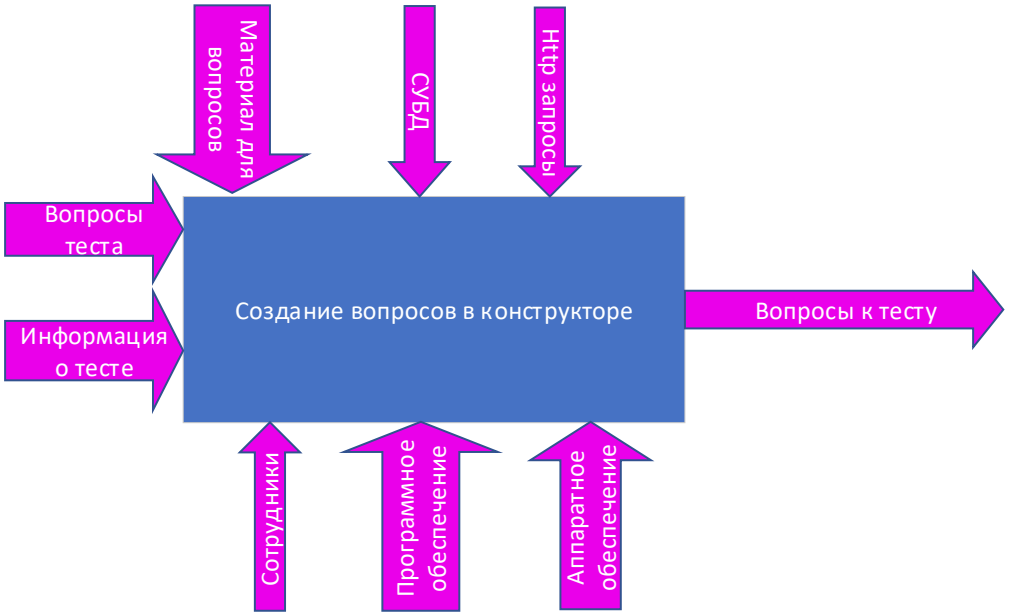

Рисунок 2.7 – Функциональная схема IDEF0 модуля личного кабинета пользователя

В приложении Б.3 можно подробнее рассмотреть декомпозицию функциональной схемы IDEF0 личного кабинета пользователя

2.3.2 Обеспечивающие подсистемы

В состав обеспечивающих подсистем для веб-приложения система тестирования входят

Техническое обеспечение

В качестве технического обеспечения для работы системы может выступать любой компьютер или сервер с минимальным набором аппаратных требований, твердотельные накопители или обычные жесткие диски, необходимость в постоянном кабельном или беспроводном быстром интернете(роутер) и при необходимости принтер для печати необходимой информации.

Программное обеспечение

Для разработки веб-приложения использовались общесистемные и специальные программные средства. К общесистемным относятся: операционная система Windows 10, Движок PHP ZEND который транслирует код, проверяя синтаксис в специальный байт-код и исполняет, в качестве среды разработки используется JetBrans PhpStorm, который обеспечивает необходимые функции для разработки веб-приложений, СУБД является MySQL и управляется с помощью веб-интерфейса для администрирования phpMyAdmin.

В качестве специальных программных средств для создания системы использовался веб-фреймворк Laravel для серверной части приложения, клиентская часть использует JavaScript, библиотеку jQuery и Vue.js – фреймворк для создания пользовательских интерфейсов. Для создания и конфигурирования виртуальной локальной среды разработки используется Vagrant вместе с виртуальной машиной Homestead, предоставляющей всё необходимое для разработки. Помимо всего этого также используется система контроля версий Git и программа для быстрого редактирования host-файла Windows.

Информационное обеспечение

Для информационного обеспечения использовались документации к Laravel, Vue.js, jQuery и другим программным пакетам, также официальная документация к языку PHP, JavaScript и СУБД MySQL. Для решения проблем во время разработки использовались форумы GitHub и Stack Overflow. Приложение должно содержать данные студентов, преподавателей, материалы по тестированию.

Лингвистическое обеспечение

В качестве основного языка системы выступает русский язык, но при необходимости система может быть расширена и другими. Для работы с ЭВМ используются командная строка Bash, язык SQL вместе с конструктором запросов Laravel, языки PHP и JavaScript.

2.4 Проектирование базы данных

Для работы всех функциональных модулей системы необходима база данных со следующими сущностями:

Таблица 2.1 – Атрибуты таблицы disciplines

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер дисциплины	int(10)
name	Название дисциплины	varchar(255)
description	Описание дисциплины	longtext

Таблица 2.2 – Атрибуты таблицы group_students

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер теста	int(10)
name	Название группы	varchar(255)
headman	Староста группы	text

Таблица 2.3 – Атрибуты таблицы students

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер теста	int(10)
group_student_id	Уникальный идентификационный номер группы студента(внешний ключ)	int(10)
name	ФИО студента	varchar(255)
email	email студента	varchar(255)
password	Пароль студента	varchar(255)
active	Указатель на активность аккаунта	tinyint(1)
tel	Телефон студента	varchar(255)
city	Город студента	varchar(255)

Таблица 2.4 – Атрибуты таблицы tests

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер теста	int(10)
discipline_id	Уникальный идентификационный номер дисциплины	int(10)
student_id	Уникальный идентификационный номер студента	int(10)
name	Название теста	varchar(255)
description	Описание теста	text
total_time	Время для прохождения теста	time
ids_students	Привязанные к тесту студенты	json

Таблица 2.5 – Атрибуты таблицы questions

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер вопроса	int(10)
test_id	Уникальный идентификационный номер теста (внешний ключ)	int(10)
content	Содержание вопроса	text
type	Тип вопроса	varchar(255)
answer	Ответы	varchar(255)
correct_answer	Правильные ответы	json
answer_explanation	Пояснение к ответам	text
score	Количество баллов	double

Таблица 2.6 – Атрибуты таблицы anon_reviews

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер отзыва	int(10)
test_id	Уникальный идентификационный номер теста (внешний ключ)	int(10)
content	Содержание отзыва	text

Таблица 2.7 – Атрибуты таблицы results

Название атрибута	Описание атрибута	Тип данных
id	Уникальный идентификационный номер вопроса	int(10)
test_id	Уникальный идентификационный номер теста (внешний ключ)	int(10)
student_id	Уникальный идентификационный номер студента (внешний ключ)	int(10)
lead_time	Время выполнения теста	text
data_test	Содержание теста	json
correct_answers	Количество правильных ответов	int(11)
final_score	Итоговый балл	double

Как видно, база данных реляционной и реализует 7 основных сущностей, которые используются для основных действий для работы с API: просмотр, добавление, редактирование и удаление данных. Каждая сущность имеет уникальный идентификационный номер, являющийся первичным ключом, используя который можно найти какую-либо запись. Также в базе данных поддерживается целостность данных, с помощью использования ряда инструментов. В их число, помимо первичных ключей, входят внешние ключи, ограничения «Not NULL», «Unique», «Default». Ниже представлена логическая схема базы данных, необходимая для понимания работы информационных потоков в системе.

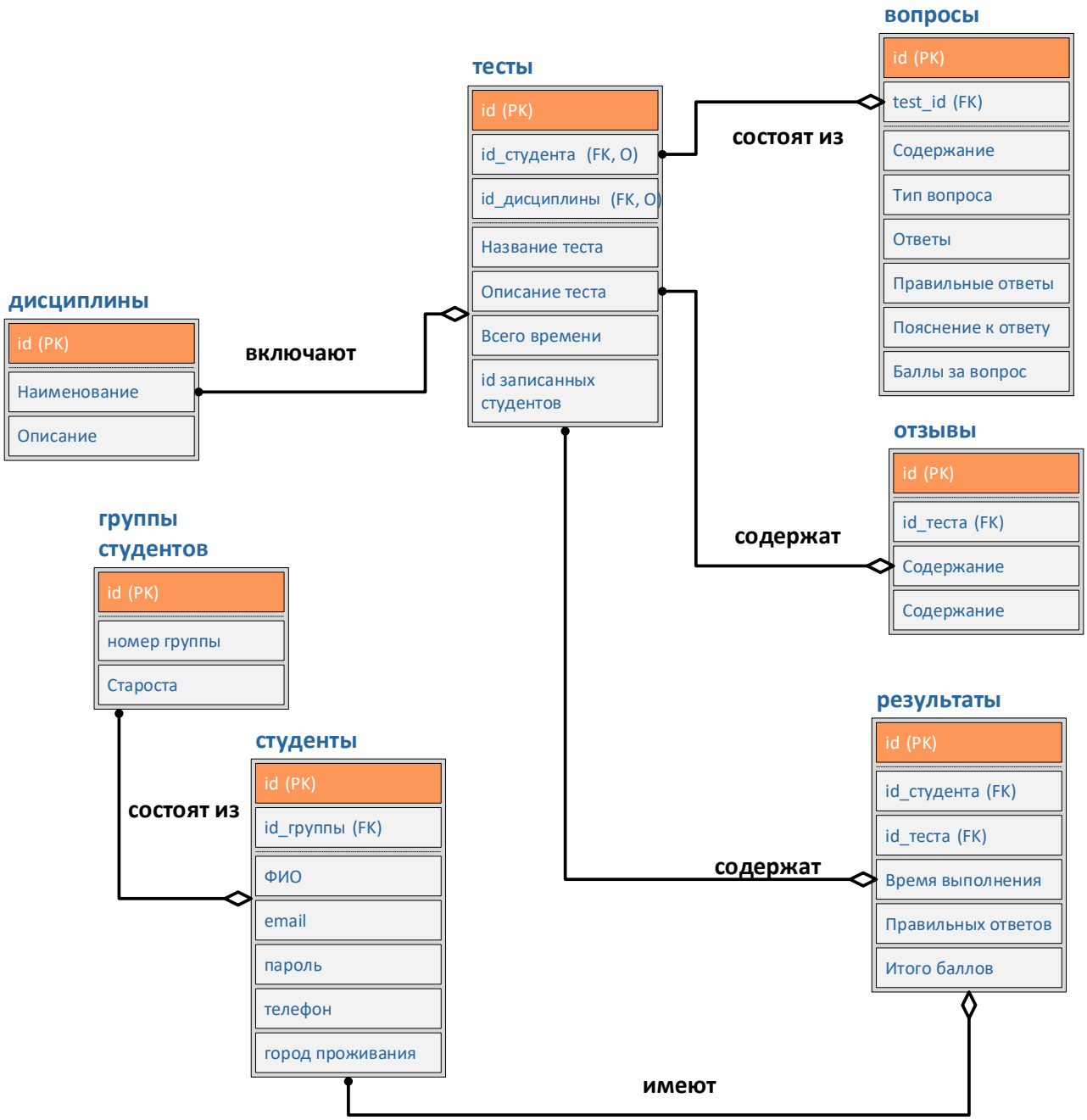

Рисунок 2.8 — логическая структура базы данных модуля в нотации IDEF1X

Физическая модель базы данных модуля представлена в приложении А.

3 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «СИСТЕМА ТЕСТИРОВАНИЯ»

3.1 Аппаратные требования и обоснование выбора средств разработки

Система ориентирована на любое клиентское аппаратное обеспечение способное приемлемо работать с современными браузерами IE10 и выше. Для серверного аппаратного обеспечения рекомендуется использовать:

- двухъядерный процессор и выше;
- от 1 гигабайта оперативной памяти;
- предпочтительно твердотельные накопители и HDD для резервных копий блок питания от 500 Вт;
- просторный проветриваемый корпус и кулер охлаждение для процессора с рассеиваемой мощностью не ниже, чем заявленное TDP процессора.

В качестве основного средства разработки для всего проекта веб-приложения «Система тестирования» был выбран полностью бесплатный серверный PHP-фреймворк Laravel. Фреймворк имеет открытый код, работу с архитектурной моделью MVC, удобный и понятный интерфейс, расширенную функциональность и подробную документацию с отзывчивым сообществом.

Laravel полностью реализует архитектурную модель MVC, что несомненно является его большим преимуществом. MVC – это конструкционный шаблон, который описывает способ построения структуры приложения.

Основные преимущества Laravel:

- Laravel построен на базе известных и надёжных компонентов Symfony;
- система объектно-реляционного отображения Eloquent ORM позволяет полностью обезопасить себя от атак типа SQL Injection, а также загружать данные из нескольких таблиц (решая проблему N+1) или же обрабатывать данные из БД частями;
- фреймворк имеет защиту от подделки запросов (CSRF);

- необходимые модули для фреймворка подключаются в виде отдельных пакетов-провайдеров, которые достаточно лишь установить, и они сразу будут доступны, без необходимости что-либо писать в коде;
- код фреймворка отделён от кода разработчика, каждый компонент легко расширяется;
- код веб-проекта, CSS, JS, HTML-код страниц разделены в отдельные директории. Фреймворк использует удобный шаблонизатор Blade, который позволяет отделить вёрстку от PHP-кода;
- удобная маршрутизация, валидация входящих параметров;
- кеширование, работа с хранилищами файлов, работа с различными БД;
- миграции для базы данных. Вы можете изменять структуру БД и откатывать изменения из кода;
- кэширование файлов маршрутизации, файлов конфигурации, шаблонов. Это ускоряет работу фреймворка;
- отправка уведомлений различными способами: почта, Slack и т.д;
- поддержка мультиязычности: легко добавляйте любые языки, а языковой пакет по умолчанию уже содержит множество переводов;
- интерфейс командной строки artisan, который позволяет генерировать модели БД, контроллеры, уведомления, запускать задания из очереди заданий и многое другое;
- Laravel Tinker – дополнительный пакет, который позволяет работать с кодом проекта из командной строки;
- огромные возможности для тестирования веб-проекта, включая заполнение базы данных тестовыми данными;
- модуль Auth – позволяет достаточно просто создавать авторизацию на сайте. Так же имеется возможность внедрения OAuth2 авторизация с помощью пакета Passport;
- использование в полной мере одностороннего шифрования и двустороннего шифрования.

Данный фреймворк был выбран как основа для создания веб-приложения, так как является полностью бесплатным и одним из самых быстрых по скорости работы среди серверных фреймворках, стремительно развивается и имеет огромное множество встроенных и пользовательских пакетов, которые максимально облегчают разработку, а также имеет сильную систему безопасности и защиты пользовательских данных, что является одним из главных аспектов при разработки многопользовательской системы.

Также вместе с Laravel используются различные пакеты, которые предоставляют множество возможностей для ускорения разработки. Так, например, в серверной части панели управления используется платформа, реализующая метод scaffolding, позволяющий указывать спецификации и генерировать определённые участки кода для MVC платформ, что позволяет полностью сосредоточится на разработке приложения.

Клиентская часть приложения реализуется с помощью JavaScript-фреймворка - Vue.js, который по умолчанию встроен в ядро Laravel. Инструмент является довольно молодым и включает в себя сильные стороны самых популярных долгоиграющих клиентских фреймворков React и Angular.

Vue Обладает следующими основными преимуществами:

- может работать как самостоятельный фреймворк, так и как библиотека для сторонних проектов;
- обеспечивает хорошее быстродействие и малый объём файлов;
- обладает двусторонней реактивностью, что позволяет удобно манипулировать данными;
- обеспечивает работу с помощью однофайловых компонентов, что достаточно удобно для разветвлённой архитектуры;
- является хорошо расширяемым с помощью отдельных плагинов;
- обладает широкой документацией на разных языках, в том числе и русском;
- может быть без проблем внедрён в разрабатываемый проект;
- достаточно низкий порог вхождения.

На рисунке 3.1 можно увидеть схему средств разработки модулей веб-приложения «Система тестирования». Исходя из рисунка видно, что весь интерфейс API, через который работают все модули, как клиентские, так и серверные, сделан с использованием Laravel, что делает его главным средством разработки программного продукта.

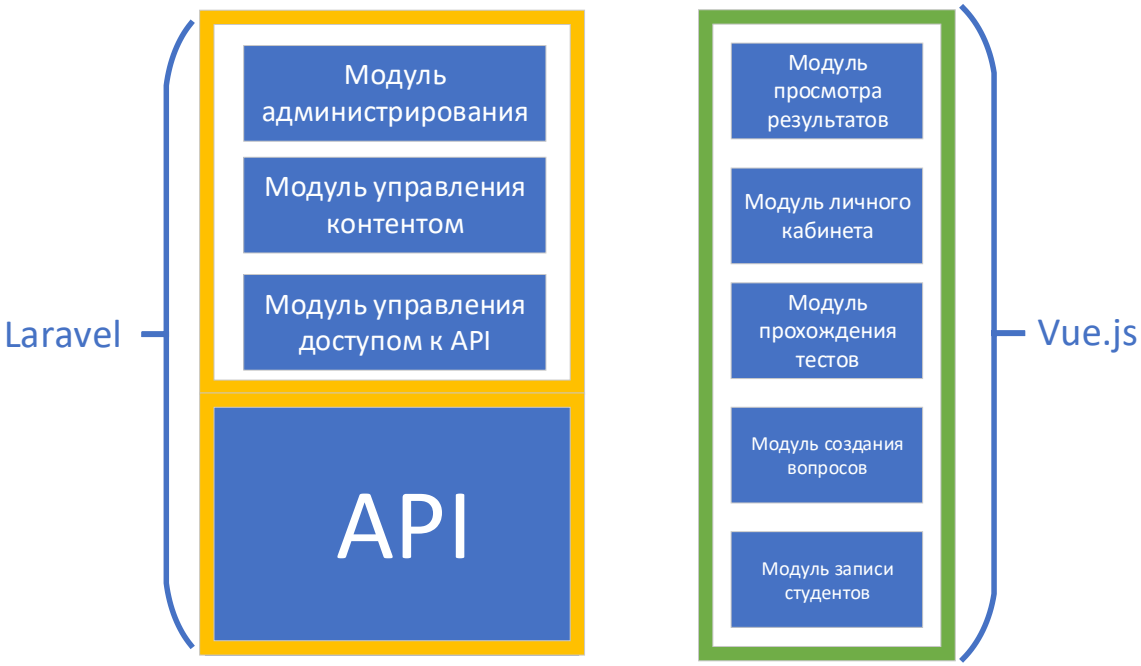

Рисунок 3.1 – Средства разработки модулей приложения

Основываясь на всем вышесказанном, можно сделать вывод, что все выбранные средства разработки являются современными, поддерживаемыми разработчиками и сообществом, а также активно развивающимися. Фреймворки содержат все необходимые средства для разработки системы тестирования и модулей к ней.

3.2 Реализация базы данных

База данных модуля основывается на СУБД MySQL, который которая является свободно-распространяемой и довольно гибкой системой управления базами данных. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц. База данных с использованием мощного компонента фреймворка Laravel – миграций. Миграции являются системой, на систему контроля версий для вашей базы данных.

Они позволяют команде программистов изменять структуру БД, в то же время зная об изменениях других участников.

Рисунок 3.2 – Процесс создания миграций к базе данных

С помощью миграций была создана база данных, которую можно гибко изменить, переписав код и выполнив команду обновления базы данных.

База данных модуля состоит из 7 таблиц, связанных между собой связями один-ко-многим. Такие таблицы как «дисциплины» и «тесты» имеют необязательную связь, так как, например, тест может быть не привязан к конкретной дисциплине.

3.3 Реализация модулей

3.3.1 REST API

Для создания API использовались RESTful API контроллеры основанные на ORM Eloquent – красивой и простой реализации паттерна ActiveRecord для работы с базой данных. Каждая таблица имеет соответствующий класс-модель, который используется для работы с этой таблицей. Модели позволяют читать данные из таблиц и записывать данные в таблицу. Паттерн также включает в себя технологию Resource API, которая позволяет легко модифицировать возвращаемую JSON коллекцию и также используется в модуле.

Контроллер каждой модели реализует 5 основных функций:

Таблица 3.1 – Основные функции API контроллера модели тестов

Запрос	URL	Действие
GET	api/tests	Получить всё
POST	api /tests	Добавить запись
GET	api /tests/{test}	Получить конкретный элемент
PUT/PATCH	api /tests/{test}	Обновить конкретный элемент
DELETE	api /tests/{test}	Удалить конкретный элемент

Используя данные URL можно получать данные из базы данных. Вместо фигурных скобок принимается объект модели. Так, например GET запрос возвращает JSON коллекцию, составленную с помощью технологии API Resource и возвращающую все элементы модели Test.

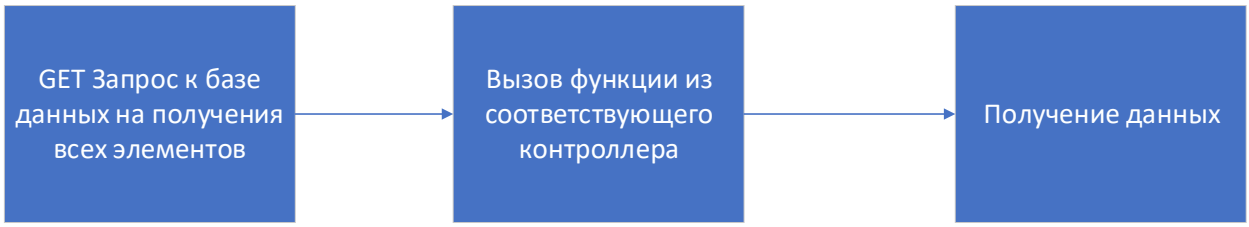

Рисунок 3.3 – Схема GET запроса получения данных с API

3.3.2 Модули панель администрирования и управления контентом

Данные модули реализуются с применением Scaffolding пакета Laravel, который предоставляет функционал администрирования и позволяет генерировать некоторые участки кода, тем самым освобождая от рутинной работы. Также пакет обладает своей системой контроллеров и жизненного цикла, что позволяет очень удобно модернизировать его собственным функционалом.

Основной функционал модуля:

Администратор

1) возможность создавать роли и пользователей системы;

2) управление статистикой;

3) просмотр логов;

4) модификация представлений по управлению таблицами данных;

5) управление настройками системы;

6) возможность просматривать всё, что включают в себя другие нижестоящие по привилегиям роли.

Преподаватель

1) возможность управлять создание групп и студентов;

2) полный доступ ко всем возможность по созданию тестов и вопросов.

Тьютор

1) возможность управлять создание групп и студентов;

2) возможна частичное управление созданием тестов (по требованию преподавателя).

3.3.3 Модули, построенные с использованием фреймворка Vue.js

Такие модули как: конструктор запросов, запись студентов на тест, личный кабинет пользователя, включающий модули прохождение теста и просмотра вопросов построены как однофайловые компоненты клиентского фреймворка Vue.js с использованием Vuex – паттерна управления состоянием для приложений на Vue.js. Он служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только предсказуемым образом.

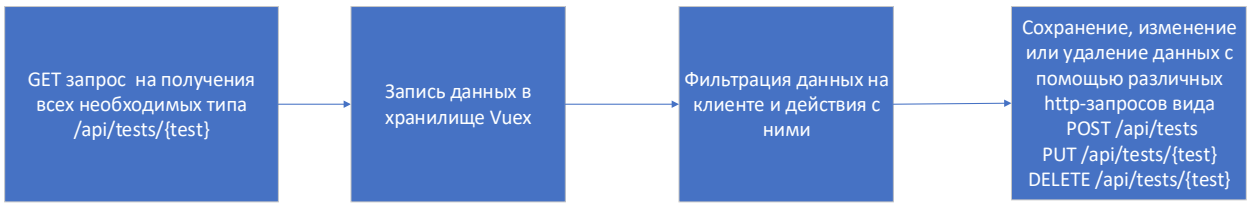

Рисунок 3.4 – Схема работы модулей построенных на Vue.js

3.3.4 Модуль Oauth2 авторизации

Так как приложение имеет свой API, то возникает вопрос его защиты от кражи конфиденциальной информации. В системе внедрён модуль OAuth2, который обеспечивает внешнюю авторизацию и соответственно защищает API так, чтобы только авторизованные модулем пользователи имели доступ к нему. За реализацию подмодуля отвечает пакет Laravel Passport, созданный на основе сервера League OAuth2.

Проанализировав документацию к данному пакету было выявлено, что пакет в соответствии с структурой сервера League OAuth2 умеет создавать и хранить клиентов, коды авторизации, выдающиеся приложениям, если конечный пользователь разрешим им доступ. На основании этих кодов создаются токены доступа и обновления, которые могут как общими, так и персональными.

Интерфейс позволяет создавать новых клиентов, подтверждать их токены, обновлять, создавать персональные токены.

В итоге, связка API и пакет Laravel Passport даёт хорошо спроектированный и надёжно защищенный от внешних атак интерфейс взаимодействия компонентов.

3.4 Интерфейс приложения

Интерфейс строится с использованием Blade — мощного шаблонизатор, поставляемый с Laravel. В отличие от других популярных шаблонизаторов для PHP Blade не ограничивает в использовании чистого PHP-кода в представлениях. Все представления Blade скомпилированы в чистый PHP-код и кешированы пока в них нет изменений, что означает, что Blade практически не нагружает систему.

В системе также используется Scaffolding платформа, составной частью которой являются Blade шаблоны. Она позволяет сгенерировать некоторые части кода, описывав в коде как они должны выглядеть. Так, например, можно сгенерировать кнопку с некоторым действием, например, динамическое создание Blade шаблона с получением каких-либо данных. Далее с помощью HTML указать необходимые поля для формы и создать функцию обработки входных

данных. Данная платформа позволяет избавиться от некоторых рутинных вещей и сосредоточиться на логике работы модуля.

3.5 Документация по сопровождению системы

3.5.1 Руководство для администратора и преподавателя

Рассмотрим руководство пользователя/администратора модуля панели управления и администрирования.

Экран входа

Экран входа в систему находится по адресу http(s)://xxxxxx.xxx/admin/login

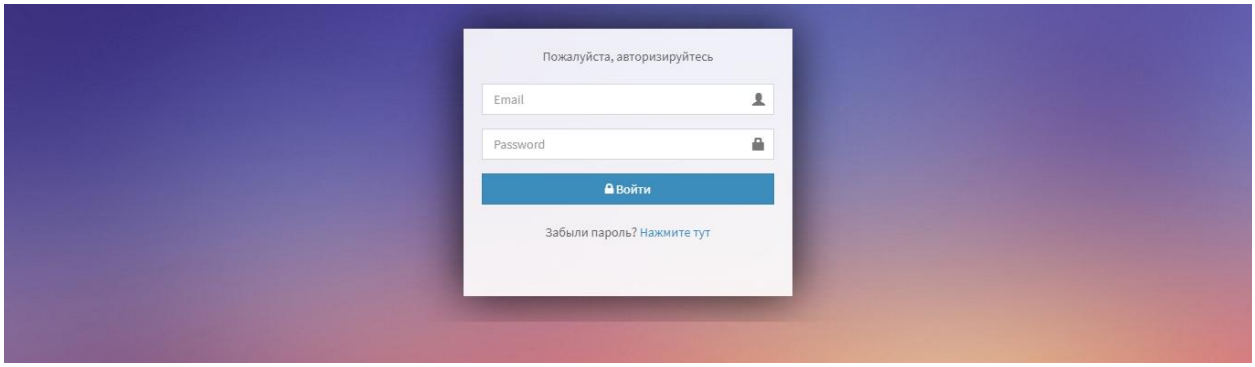

Рисунок 3.5 – Экран входа в панель управления

В форме необходимо ввести свой e-mail и пароль, для прохождения аутентификации и авторизации в системе и нажать на кнопку войти. Если же пользователь введёт неправильный пароль, то получит ошибку.

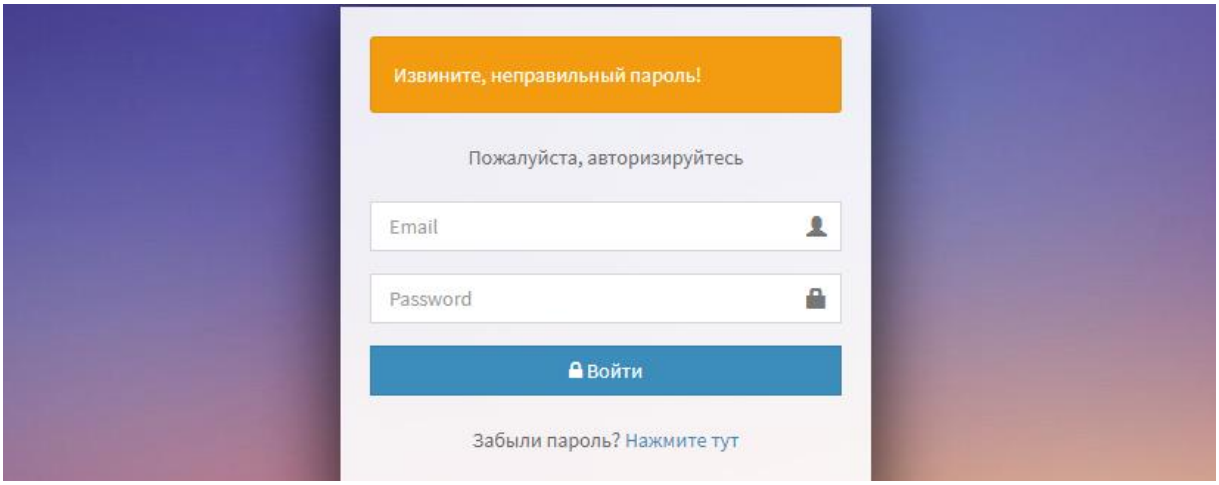

Рисунок 3.6 – Ошибка при вводе неправильного пароля

Работа в панели управления

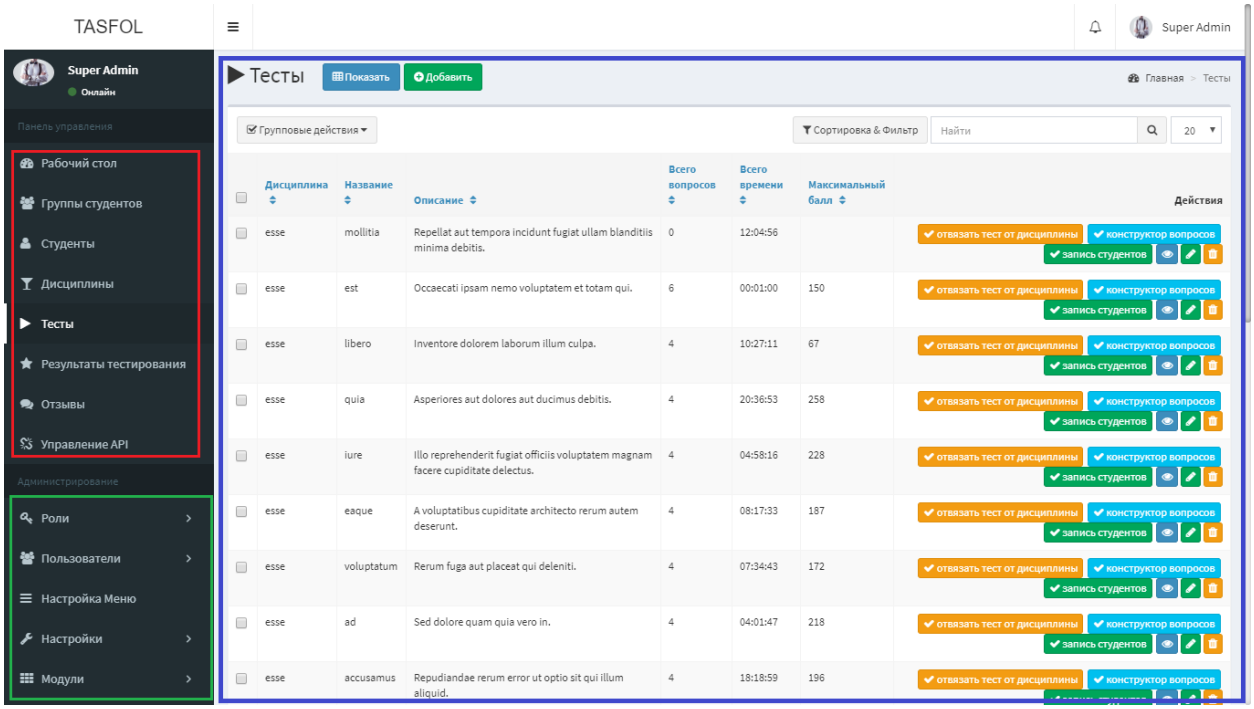

Рисунок 3.7 – Экран панели управления

На экране панели управление представлено боковое меню. Часть, отмеченную красным, видят преподаватели и тьюторы в зависимости от своих ролей и прав. В этой части располагаются все таблицы (так синим отмечен вид модели таблицы «Тесты») и соответствующий функционал для записи студентов, создания тестов и вопросов. Зеленым отмечена часть, которую видит только администратор, тут располагаются основные функции администрирования системы, такие как, например: управление ролями, меню, просмотр логов пользователей. Администратор видит всё, что видят преподаватели и другие роли, может менять права доступа к роли, удалять пользователей или добавлять. Панель также имеет возможность просмотреть свой профиль и изменить его. Реализована функция неполного выхода из система, которая может быть полезна в случае, когда пользователю панели управления нужно отойти от своего рабочего места и при этом оставить систему в безопасном состоянии, так как для входа потребуется повторно ввести пароль.

Пользователей панели может создавать вопросы к определённому тесту используя специальный конструктор вопросов.

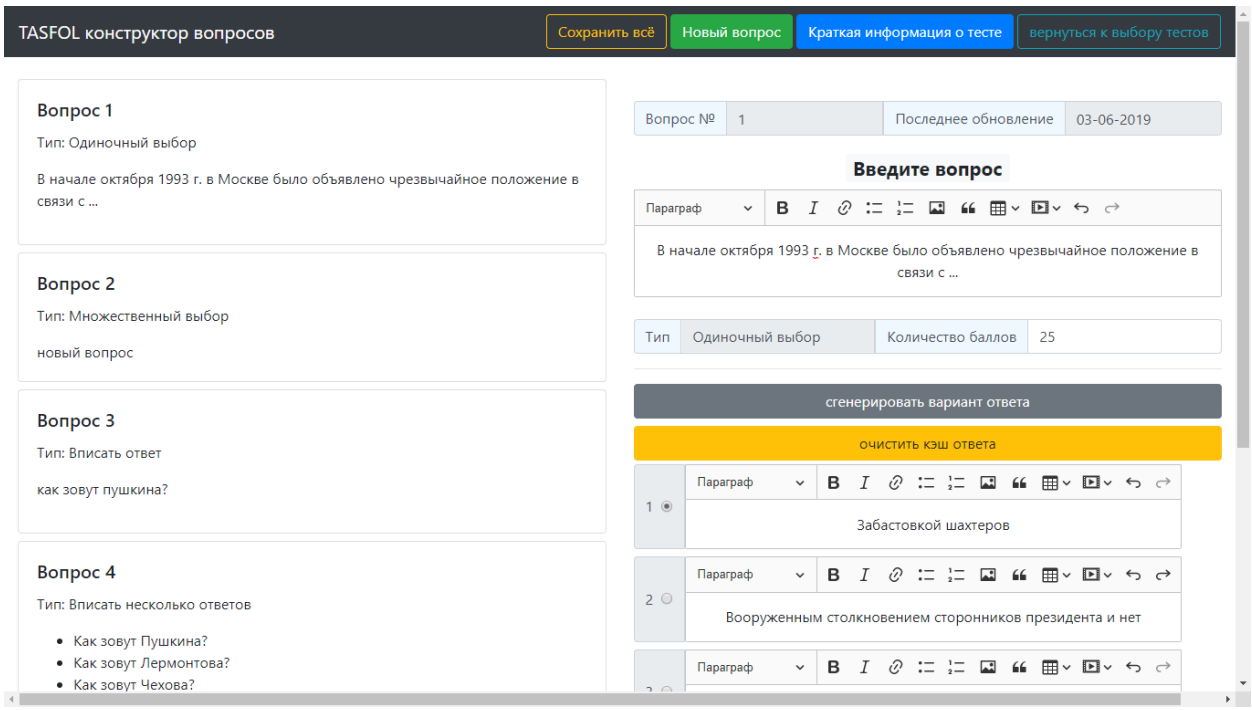

Рисунок 3.8 – Экран конструктора запросов

Как видно из экрана конструктора запросов в столбце слева располагаются сами вопросы, а в столбце слева элемент для их редактирования. Конструктор реализует 8 различных типов вопросов и имеет функции просмотра, создания, редактирования и удаления вопросов.

Администратор или преподаватель имеют право создавать группы и добавлять в них студентов.

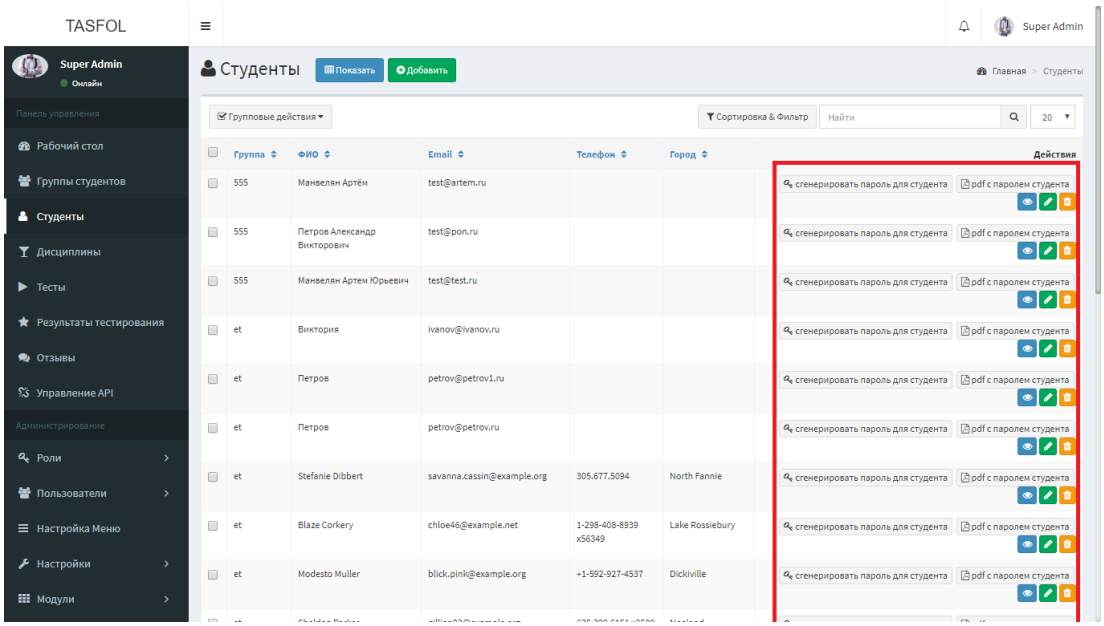

Рисунок 3.9 – Экран просмотра студентов

Красным цветом на экране просмотра студентов отмечены кнопки с функциями создания пароля для студента и получения данных для входа в виде pdf файла. Такие функции реализованы и для групп студентов.

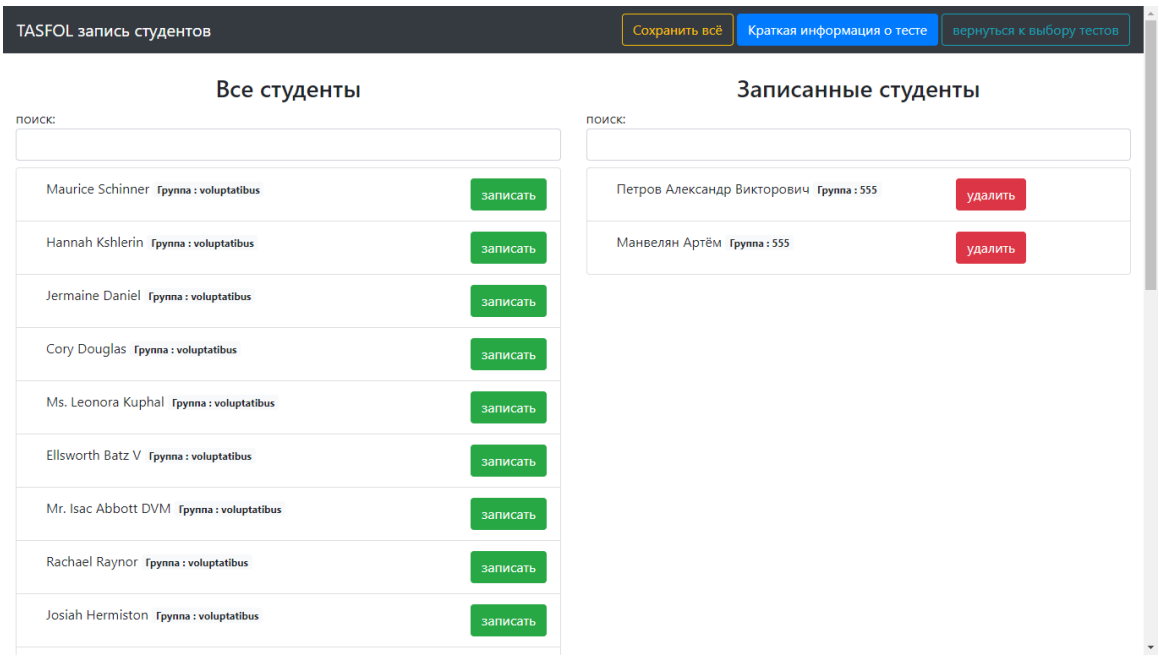

Рисунок 3.10 – Экран модуля записи студентов

Крупным является модуль записи студентов на прохождения тестов. Имеется возможность выбрать из всех студентов и записать конкретных для прохождения конкретного теста.

Отдельным является модуль, который предоставляет API для внешних устройств.

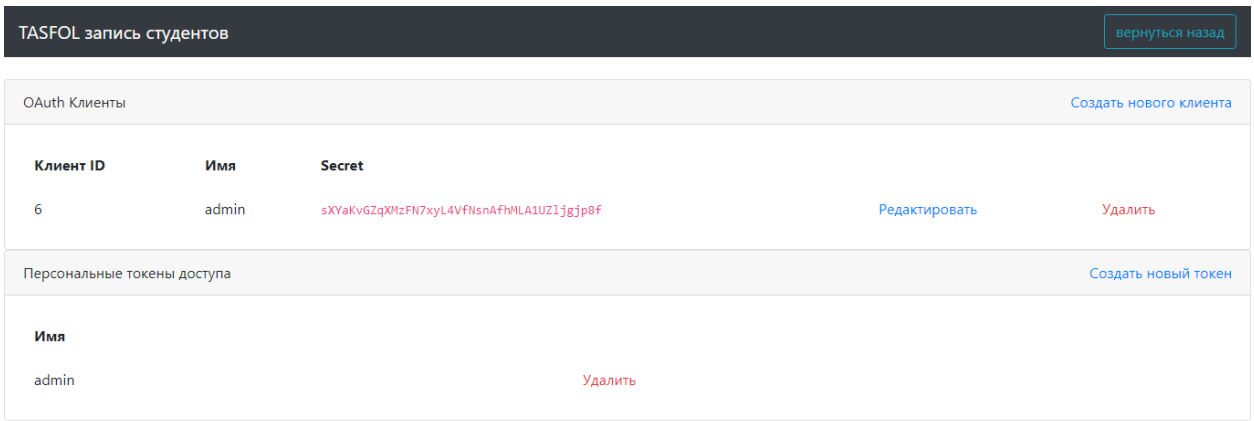

Рисунок 3.11 – Экран модуля записи студентов

В нём реализуется функционал пакета Laravel Passport, который реализует OAuth2 авторизацию для внешних устройств выдавая общие или персональные токены доступа.

Каждый пользователей панели имеет возможность редактировать свой профиль (кнопка отмечена красным).

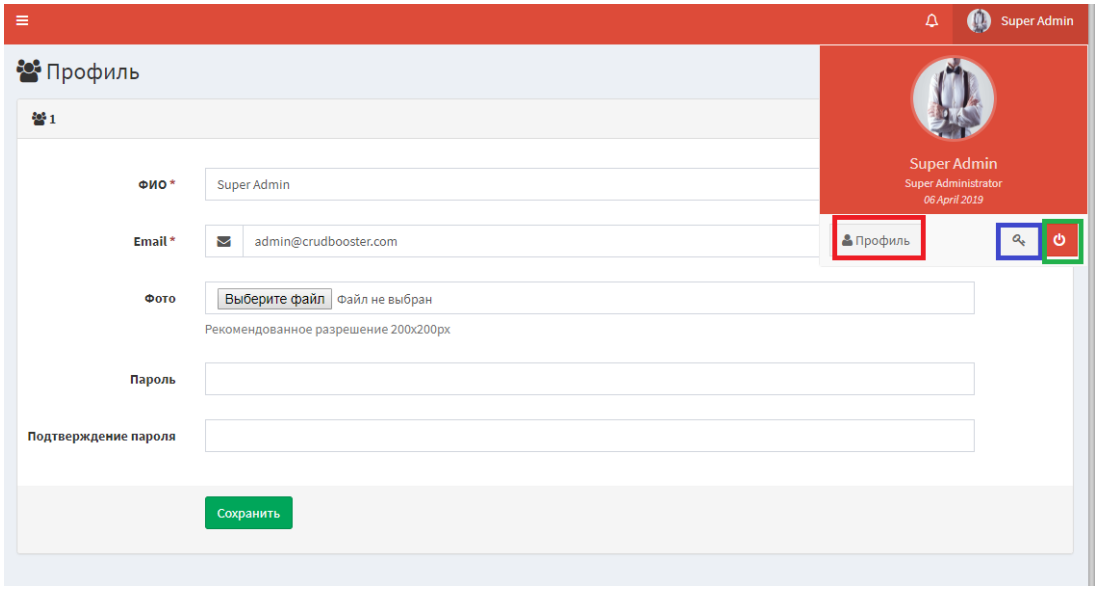

Рисунок 3.12 – Экран редактирования профиля

Помимо редактирования в всплывающем окне пользователя имеется кнопка выйти из системы (отмечена зелёным) и кнопка LockScreen (отмечена синим), которая позволит ввести систему в спящий режим.

3.5.2 Руководство для студента

Экран входа в личный кабинет студента находится по адресу http(s)://xxxxxx.xxx/login.

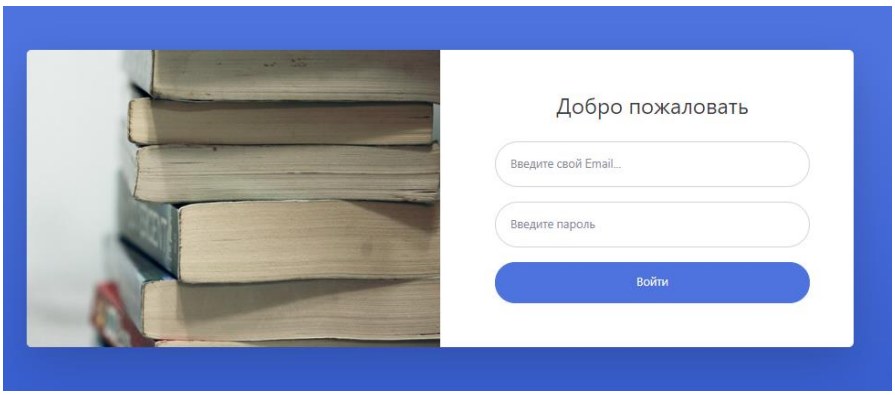

Рисунок 3.13 – Экран входа в личный кабинет пользователя

В форме необходимо ввести данные, которые были выданы для входа в личный кабинет, для прохождения аутентификации и авторизации.

После пользователь попадает на страницу доступных ему тестов, где также присутствует краткая статистика о прохождении тестов.

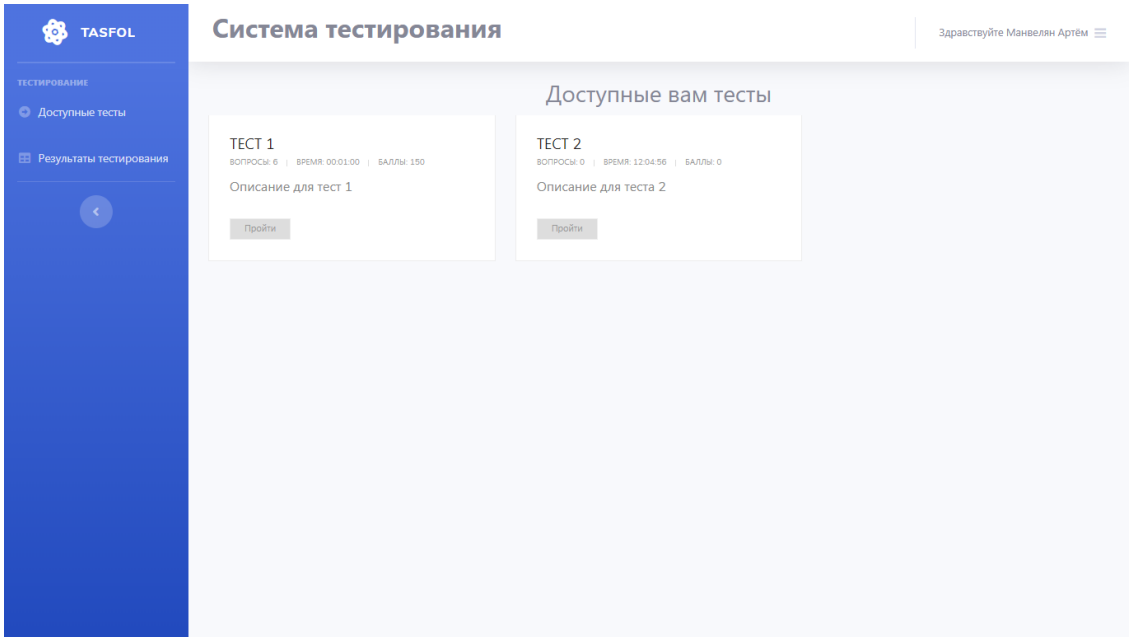

Рисунок 3.14 – Экран списка тестов

На данной странице пользователя выбирает тест, которых хочет пройти и проходит его, отвечая на вопросы.

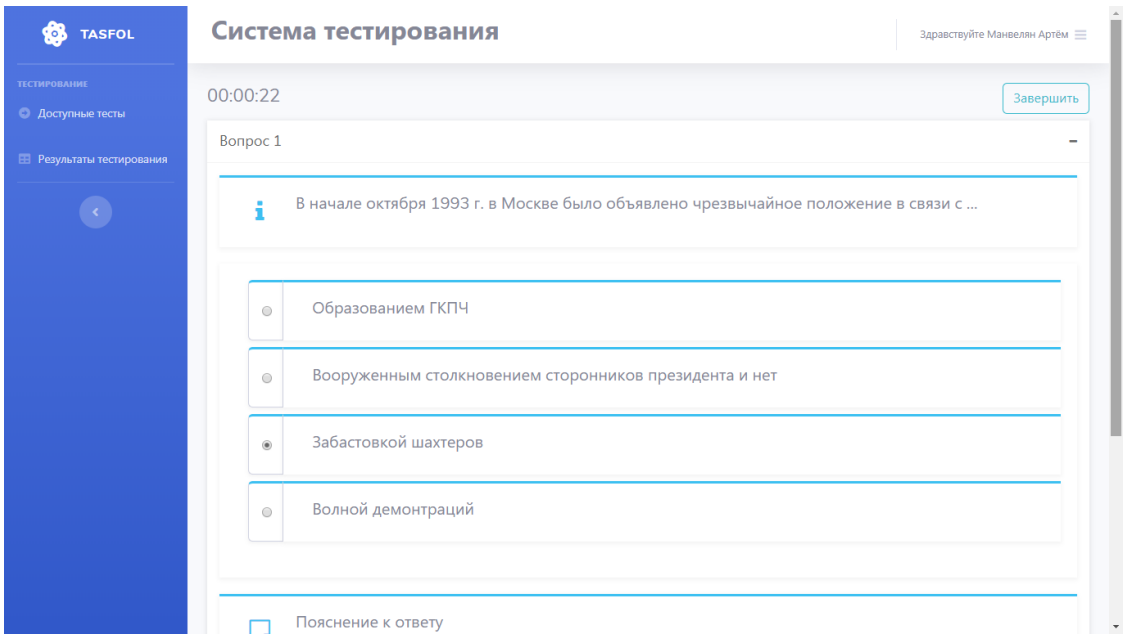

Рисунок 3.15 – Экран прохождения теста

После прохождение необходимых тестов пользователь может увидеть свои результаты в разделе «Результаты тестирований».

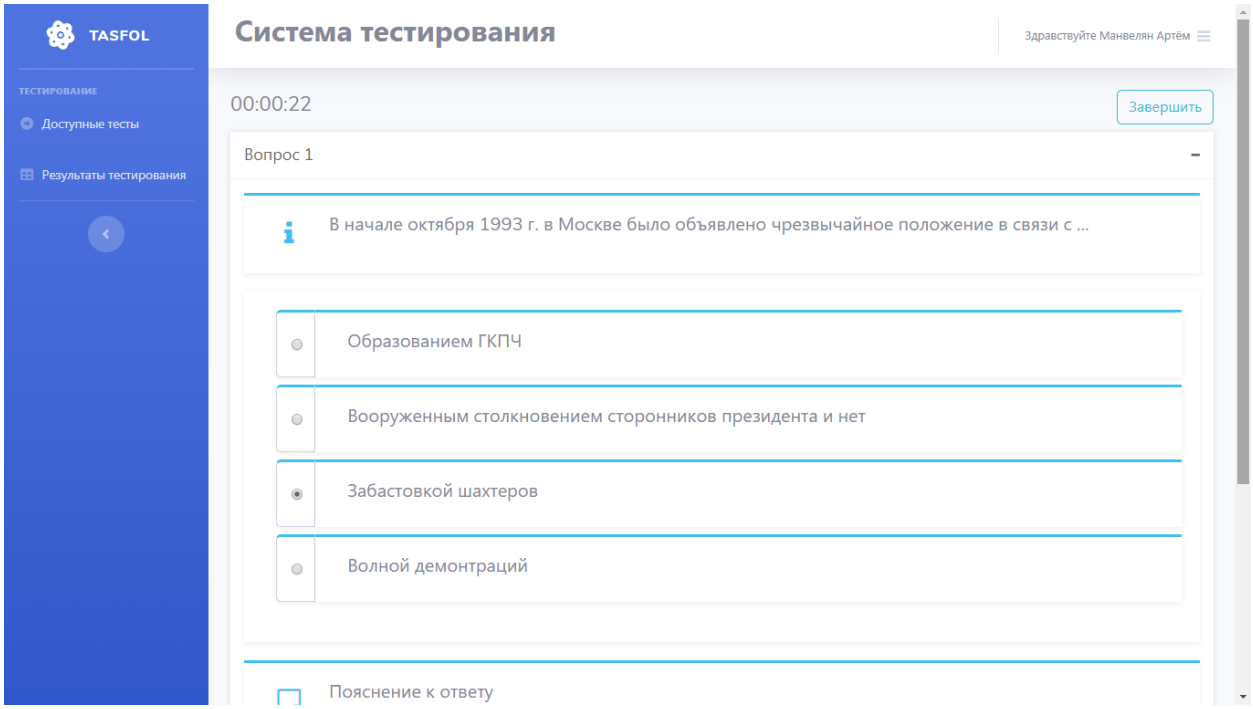

Рисунок 3.16 – Экран результатов тестирований

4 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ВЕБ-ПРИЛОЖЕНИЯ «СИСТЕМА ТЕСТИРОВАНИЯ

4.1 Модель угроз информационной безопасности

Модель угроз информационной безопасности – это описание существующих угроз ИБ, их актуальности, возможности реализации и последствий.

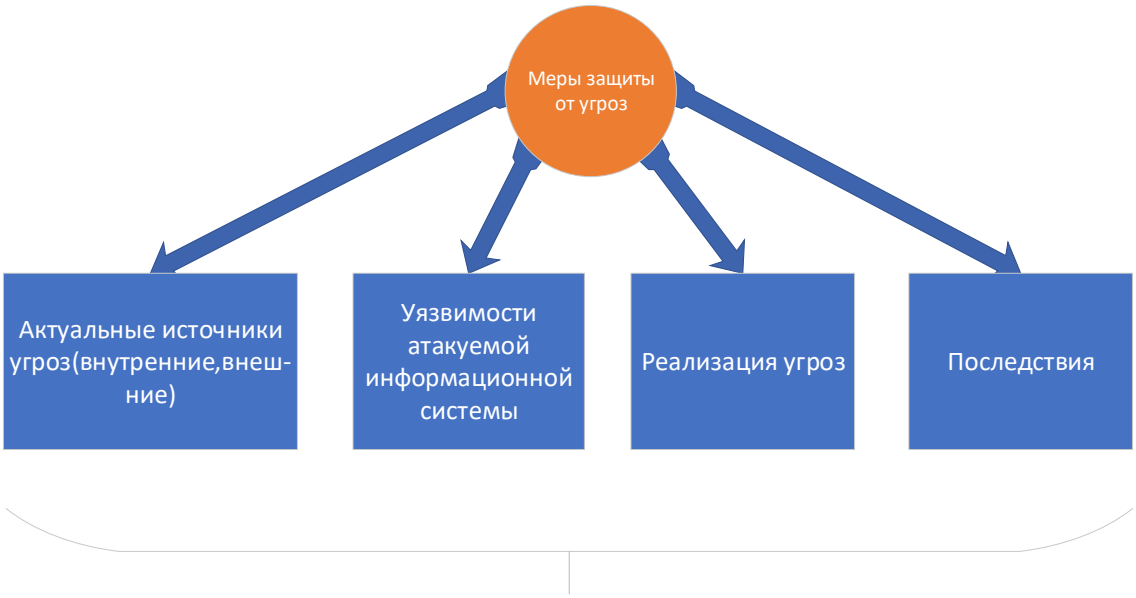

Рисунок 4.1 – жизненный цикл угроз информационной безопасности

В модели должны учитываться все актуальные угрозы на всех стадиях их жизненного цикла

У различных информационных систем, а также объектов одной информационной системы может быть разный спектр угроз, определяемый особенностями конкретной информационной системы и её объектов и характером возможных действий источника угрозы.

Процедура построения модели угроз информационной безопасности состоит из нескольких последовательных шагов:

- определение источников угроз;
- выявление критических объектов информационной системы;
- определение перечня угроз для каждого критического объекта;
- выявление способов реализации угроз;

- оценка материального ущерба и других последствий возможной реализации угроз.

Модели угроз составляются на основе постоянно меняющихся данных и поэтому должны регулярно пересматриваться и обновляться.

При построении данной модели использовались каталоги и перечни угроз, содержащиеся в официальных стандартах информационной безопасности и методических документах ФСТЭК и ФСБ России как, например, Банк угроз безопасности информации ФСТЭК России и документ – «Методика определения угроз безопасности информации в информационных системах».

Таблица 4.1 – модель угроз информационной безопасности веб-приложения

Название угрозы	**Реализация угрозы**	**Источники угрозы**	**Объект воздействия**	**Нарушение конфиденциальности (ущерб)**	**Нарушение целостности (ущерб)**	**Нарушение доступности (ущерб)**
Угроза внедрения кода или данных	внедрение нарушителем в дискредитируемую информационную систему вредоносного кода	Внешний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение	средний	средний	средний
Угроза восстановления аутентификационной информации	подбор (например, путём полного перебора или перебора по словарю) аутентификационной информации дискредитируемой учётной записи пользователя в системе.	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, микропрограммное обеспечение, учётные данные пользователя	средний	—	—

Продолжение таблицы 4.1

Название угрозы	**Реализация угрозы**	**Источники угрозы**	**Объект воздействия**	**Нарушение конфиденциальности (ущерб)**	**Нарушение целостности (ущерб)**	**Нарушение доступности (ущерб)**
Угроза доступа к локальным файлам сервера при помощи URL	передачи нарушителем дискредитируемому браузеру запроса на доступ к файловой системе пользователя вместо URL—запроса.	Внешний нарушитель со средним потенциалом	Сетевое программное обеспечение	средний	—	—
Угроза доступа/перехвата/изменения HTTP cookies	осуществление нарушителем несанкционированного доступа к защищаемой информации (учётным записям пользователей, сертификатам и т.п.), содержащейся в cookies—файлах	Внешний нарушитель с низким потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение	средний	—	средний
Угроза межсайтового скриптинга	внедрение нарушителем участков вредоносного кода на сайт дискредитируемой системы таким образом, что он будет выполнен на рабочей станции просматривающего этот	Внешний нарушитель с низким потенциалом	Сетевой узел, сетевое программное обеспечение	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	**Реализация угрозы**	**Источники угрозы**	**Объект воздействия**	**Нарушение конфиденциальности (ущерб)**	**Нарушение целостности (ущерб)**	**Нарушение доступности (ущерб)**
	сайт пользователя.					
Угроза межсайтовой подделки запроса	отправки нарушителем дискредитируемому пользователю ссылки на содержащий вредоносный код веб—ресурс, при переходе на который автоматически будут выполнены неправомерные вредоносные действия от имени дискредитированного пользователя.	Внешний нарушитель со средним потенциалом	Сетевой узел, сетевое программное обеспечение	средний	средний	средний
Угроза неправомерного/некорректного использования интерфейса взаимодействия с приложением	осуществление нарушителем деструктивного программного воздействия на API в целях реализации функций, изначально не предусмотренных дискредитируемым приложением.	Внешний нарушитель со средним потенциалом, Внутренний нарушитель со средним потенциалом	Системное программное обеспечение, прикладное программное обеспечение, сетевое программное обеспечение, микропрограммное обеспечение, реестр	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	**Реализация угрозы**	**Источники угрозы**	**Объект воздействия**	**Нарушение конфиденциальности (ущерб)**	**Нарушение целостности (ущерб)**	**Нарушение доступности (ущерб)**
Угроза несанкционированного доступа к защищаемым виртуальным машинам	осуществление деструктивного программного воздействия на защищаемые виртуальные машины	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Виртуальная машина	средний	средний	средний
Угроза несанкционированного копирования защищаемой информации	неправомерное получения нарушителем копии защищаемой информации путём проведения последовательности неправомерных действий, включающих: несанкционированный доступ к защищаемой информации, копирование найденной информации на съёмный носитель.	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы, машинный носитель информации	средний	—	—
Угроза несанкционированного создания учётной записи пользователя	создание нарушителем в системе дополнительной учётной записи пользователя и её дальнейшего использования в собственных	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
Угроза обхода некорректно настроенных механизмов аутентификации	неправомерных целях получение нарушителем привилегий в системе без прохождения процедуры аутентификации за счёт выполнения действий, нарушающих условия корректной работы	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, сетевое программное обеспечение	средний	средний	средний
Угроза перехвата вводимой и выводимой на периферийные устройства информации	осуществления нарушителем несанкционированного доступа к информации, вводимой и выводимой на периферийные устройства, путём перехвата данных, обрабатываемых контроллерами периферийных устройств.	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Системное программное обеспечение, прикладное программное обеспечение, аппаратное обеспечение	средний	—	—
Угроза подмены действия пользователя путём обмана	выполнение неправомерных действий в системе от имени другого пользователя с	Внешний нарушитель со средним потенциалом	Прикладное программное обеспечение, сетевое программное обеспечение	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
	помощью методов социальной инженерии					
Угроза «кражи» учётной записи доступа к сетевым сервисам	неправомерное ознакомления нарушителем с защищаемой информацией пользователя путём получения информации идентификации/аутентификации, соответствующей учётной записи доступа пользователя к сетевым сервисам (социальной сети, облачным сервисам и др.), с которой связан неактивный/несуществующий адрес электронной почты.	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	средний	—	средний
Угроза распространения «почтовых червей»	нарушение безопасности защищаемой информации пользователя вредоносными программами, скрытно	Внешний нарушитель с низким потенциалом	Сетевое программное обеспечение	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
	устанавливаемыми при получении пользователями системы электронных писем, содержащих вредоносную программу типа «почтовый червь», а также невольного участия в дальнейшем противоправном распространении вредоносного кода.					
Угроза «фарминга»	неправомерное ознакомления нарушителем с защищаемой информацией (в т.ч. идентификации/аутентификации) пользователя путём скрытного перенаправления пользователя на поддельный сайт (выглядящий одинаково с оригинальным), на котором от	Внешний нарушитель с низким потенциалом	Рабочая станция, сетевое программное обеспечение, сетевой трафик	средний	—	—

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
	дискредитируемого пользователя требуется ввести защищаемую информацию.					
Угроза «фишинга»	неправомерное ознакомления нарушителем с защищаемой информацией (в т.ч. идентификации/аутентификации) пользователя путём убеждения его с помощью методов социальной инженерии (в т.ч. посылкой целевых писем (т.н. spear—phishing attack), с помощью звонков с вопросом об открытии вложения письма, имитацией рекламных предложений (fake offers) или различных приложений (fake apps))	Внешний нарушитель с низким потенциалом	Рабочая станция, сетевое программное обеспечение, сетевой трафик	средний	—	—

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
	зайти на поддельный сайт (выглядящий одинаково с оригинальным), на котором от дискредитируемого пользователя требуется ввести защищаемую информацию или открыть заражённое вложение в письме.					
Угроза несанкционированной модификации защищаемой информации	нарушение целостности защищаемой информации путём осуществления нарушителем деструктивного физического воздействия на машинный носитель информации или деструктивного программного воздействия	Внешний нарушитель с низким потенциалом, Внутренний нарушитель с низким потенциалом	Объекты файловой системы	—	средний	—
Угроза внедрения вредоносного кода за счет посещения	осуществление нарушителем внедрения вредоносного кода в компьютер	Внешний нарушитель со средним потенциалом	Сетевое программное обеспечение	средний	средний	средний

Продолжение таблицы 4.1

Название угрозы	Реализация угрозы	Источники угрозы	Объект воздействия	Нарушение конфиденциальности (ущерб)	Нарушение целостности (ущерб)	Нарушение доступности (ущерб)
зараженных сайтов в сети Интернет	пользователя при посещении зараженных сайтов.					

В таблице представлена модель угроз, описывающая угрозы, их источники, объект воздействия, реализацию и примерную оценку ущерба для триады ИБ: конфиденциальности, целостности, доступности. В основном ущерб оценивается как средний от всех угроз ИБ.

Кроме приведённых угроз, существуют и другие, менее значимые, как, например, некоторые из них:

- угроза воздействия на программы с высокими привилегиями;
- угроза доступа к защищаемым файлам с использованием обходного пути;
- угроза использования механизмов авторизации для повышения привилегий;
- угроза некорректного использования прозрачного прокси-сервера за счёт плагинов браузера;
- угроза несанкционированного доступа к виртуальным каналам передачи;
- угроза несанкционированного изменения аутентификационной информации.

4.2 Модель нарушителя информационной безопасности

Модель нарушителя информационной безопасности – это набор предположений об одном или нескольких возможных нарушителях информационной безопасности, их квалификации, их технических и материальных средствах и т. д. Рассмотрим таблицу возможных нарушителей информационной безопасности.

Таблица 4.2 – модель нарушителя ИБ.

Нарушители	**Категория нарушителя**	**Потенциал нарушителя**	**Возможная мотивация**	**Предполагаемые возможности**
Бывшие пользователи,администраторы или преподаватели	Внешний нарушитель	с низким потенциалом	Причинение имущественного ущерба путем мошенничества или иным преступным путем. Месть за ранее совершенные действия.	Возможность самостоятельно осуществлять создание способов атак, подготовку и проведение атак только за пределами контролируемой зоны Возможность получить информацию об уязвимостях отдельных компонент информационной системы, опубликованную в общедоступных источниках Возможность получить информацию о методах и средствах реализации угроз безопасности информации (компьютерных атак), опубликованных в общедоступных источниках, и (или) самостоятельно осуществлять создание методов и средств реализации атак и реализацию атак на информационную систему
Лица, обеспечивающие функционирование информационных систем или обслуживающих инфраструктуру оператора (администрация, охрана, уборщики,пользователи и т.д.)	Внутренний нарушитель	с низким потенциалом	Причинение имущественного ущерба путем обмана или злоупотребления доверием. Непреднамеренные, неосторожные или неквалифицированные действия.	Возможность получить информацию об уязвимостях отдельных компонент информационной системы, опубликованную в общедоступных источниках Возможность получить информацию о методах и средствах реализации угроз безопасности информации (компьютерных атак), опубликованных в общедоступных источниках, и (или) самостоятельно осуществлять создание методов и средств реализации атак и реализацию атак на информационную систему.

Продолжение таблицы 4.2

Нарушители	**Категория нарушителя**	**Потенциал нарушителя**	**Возможная мотивация**	**Предполагаемые возможности**
Администраторы информационной системы и администраторы безопасности	Внутренний нарушитель	со средним потенциалом	Причинение имущественного ущерба путем мошенничества или иным преступным путем. Любопытство или желание самореализации (подтверждение статуса). Месть за ранее совершенные действия. Выявление уязвимостей с целью их дальнейшей продажи и получения финансовой выгоды. Непреднамеренные, неосторожные или неквалифицированные действия.	Возможность самостоятельно осуществлять создание способов атак, подготовку и проведение атак за пределами контролируемой зоны Возможность получить информацию об уязвимостях отдельных компонент информационной системы, опубликованную в общедоступных источниках Возможность получить информацию о методах и средствах реализации угроз безопасности информации (компьютерных атак), опубликованных в общедоступных источниках, и (или) самостоятельно осуществлять создание методов и средств реализации атак и реализацию атак на информационную систему Имеют осведомленность о мерах защиты информации, применяемых в информационной системе данного типа. Имеют возможность получить информацию об уязвимостях отдельных компонент информационной системы путем проведения, с использованием имеющихся в свободном доступе программных средств, анализа кода прикладного программного обеспечения и отдельных программных компонент общесистемного программного обеспечения. Имеют доступ к сведениям о структурно-функциональных характеристиках и особенностях функционирования информационной системы.

Для разрабатываемого веб-приложения преобладают внутренние нарушители, и в большинстве случаев это будут недовольные пользователи системы, которых не устроит результат тестирования или сами задания.

В некоторых случаях это могут быть бывшие пользователи системы, занимающие разные роли в ней и которых что-то не устроило.

4.3 Средства обеспечения ИБ

Для обеспечения информационной безопасности в системе используются различные средства и методы.

Фреймворк Laravel построен с использованием современных стандартов защиты. Он защищён от всех распространённых веб-атак и предоставляет злоумышленникам минимальные шансы на успех. В данном случае большое значение имеют пользователи сети и их подготовленность к методам социальной инженерии.

Панель управления приложением использует парольную защиту, в которой нельзя использовать слабые пароли, данное явление тщательно проверяется сервером в процессе валидации данных. Так же имеется возможность подключения двухэтапной аутентификации от Google, которая обеспечит большую защищённость с помощью одноразовых кодов входа. Коллаборация данных средств достаточно для защиты всех данных веб-приложения, а также соответственно данных пользователей.

Веб-приложение имеет возможность авторизации извне. Авторизация основана на OAuth2 — протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе, например получение данных мобильным приложением, для построения статистики прохождения тестов. Протокол достаточно давно разработан и позволяет достаточно защищённо предоставлять доступ сторонним сервисам. Данным протокол также защищает API системы от удалённого доступа неавторизованного пользователя.

Часть системы, предназначенная для студентов, имеет парольную защиту со заранее созданными сложными паролями, что обеспечивает достаточную защиту для аккаунтов потенциальных студентов.

4.4 Политика безопасности

Проанализировав модели угроз и нарушителей, была составлена политика безопасности приложения, кратко поясняющая основные аспекты ИБ в системе.

Общие положения

Система является открытой и распространяется с помощью лицензии MIT. Данное программное обеспечение предоставляется «как есть», без каких-либо гарантий, явно выраженных или подразумеваемых, включая гарантии товарной пригодности, соответствия по его конкретному назначению и отсутствия нарушений, но не ограничиваясь ими. Ни в каком случае авторы или правообладатели не несут ответственности по каким-либо искам, за ущерб или по иным требованиям, в том числе, при действии контракта, деликте или иной ситуации, возникшим из-за использования программного обеспечения или иных действий с программным обеспечением

Система построена на современных инструментах веб-разработки, с использованием различных технологий и валидаций данных, что обеспечивает достаточную защищенность от большинства внутренних и внешних угроз, как например фишинг, межсайтовый скриптинг, межсайтовая подделка запроса и др.

В случае несогласия с условиями Политики конфиденциальности Пользователь должен прекратить использование сайта Интернет-магазина.

Предмет политики конфиденциальности

Администраторы веб-приложения «Система тестирования» гарантируют обеспечение полной конфиденциальности информации, полученной от зарегистрированных пользователей. Вся информация, которую Вы укажите при регистрации в системе, будет храниться в защищенной базе данных. Все пароли используют одностороннее шифрование;

Персональные данные, разрешённые к обработке в рамках настоящей Политики конфиденциальности, предоставляются Пользователем путём предоставления данных администратору или преподавателю и включают в себя следующую информацию:

1) ФИО пользователя;

2) контактный телефон Пользователя;

3) адрес электронной почты (e-mail);

4) город проживания;

5) место жительство Пользователя.

Ваши регистрационные данные нам нужны исключительно для того, чтобы администраторы системы зарегистрировали вас как условного студента, и преподаватели могли с вами контактировать.

Представители «системы тестирования» никогда не попросят у Вас данные пластиковой карты или другие конфиденциальные реквизиты.

В системе могут быть размещены ссылки на другие ресурсы. Администрация не несем ответственности за сведения, публикуемые на этих сайтах, и угрозы безопасности, которые могут на них возникнуть.

Личные сведения и безопасность

Администрация, использующая систему, гарантирует, что никакая полученная от Вас информация никогда и ни при каких условиях не будет предоставлена третьим лицам, за исключением случаев, предусмотренных действующим законодательством Российской Федерации.

Личные сведения можно изменить, обновить в любое время в разделе "Личный кабинет".

Ответственность сторон

Администрация сайта, не исполнившая свои обязательства, несёт ответственность за убытки, понесённые Пользователем в связи с неправомерным использованием персональных данных, в соответствии с законодательством Российской Федерации.

В случае утраты или разглашения Конфиденциальной информации Администрация сайта не несёт ответственность, если данная конфиденциальная информация стала публичным достоянием до её утраты или разглашения.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1. Безопасность

Для нормальной работы веб-приложения «Система тестирования» необходимо соблюдение ряда правил по безопасности. Проведём анализ типичного рабочего места работы за ПЭВМ в помещении. Для конкретного рассмотрения возьмём рабочее место офисного бизнес-центра, так как зачастую, из-за перегруженности графиков, люди забывают о каких-либо санитарных нормах при работе за компьютером, тем самым подвергая своё здоровье риску. Анализ проводится на основании документа СанПиН 2.2.2/2.4.1340-03.

5.1.1 Требования к ПЭВМ

Для нормальной работы веб-приложения «Система тестирования» необходимо рабочее место пользователя, которое может располагаться в любом месте: в офисе компании, в кабинете школы или дома у пользователя. Проведём анализ типичного рабочего места работы за ПЭВМ. Для конкретного рассмотрения возьмём рабочее место офисного бизнес-центра, так как зачастую, из-за перегруженности графиков, люди забывают о каких-либо санитарных нормах при работе за компьютером, тем самым подвергая своё здоровье риску. Анализ проводится на основании документа СанПиН 2.2.2/2.4.1340-03.

5.1.2 Требования к помещению для работы с ПЭВМ

Рассмотрим типичное рабочее место в офисе.

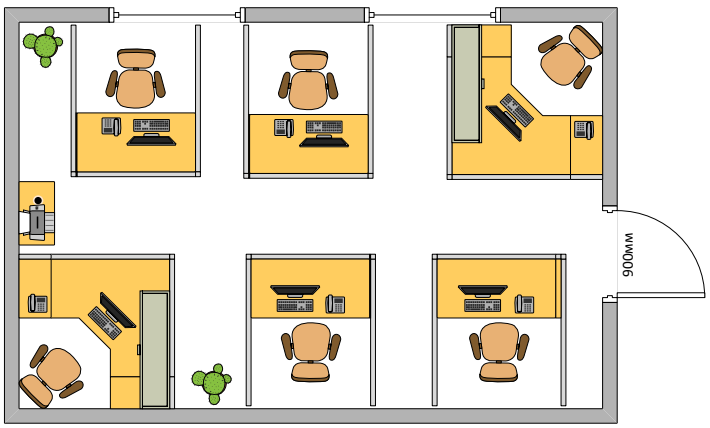

Рисунок 5.1 — рабочее место в офисе (как есть)

Помещение обладает необходимым искусственным освещением, но содержит достаточно маленькие окна, которые также перекрываются рабочими местами и не предоставляют достаточного проветривания.

Так как все современные мониторы жидкокристаллические или светодиодные и являются практически полностью безопасными, то площадь на одно рабочее место пользователя ПЭВМ в помещениях учреждений с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м^2, что как видно из схемы не соблюдается, так как при площади помещения 24 м^2 максимальное количество рабочих мест должно быть четыре, чтобы оставить остальное место под коридор и выходы из-за мест.

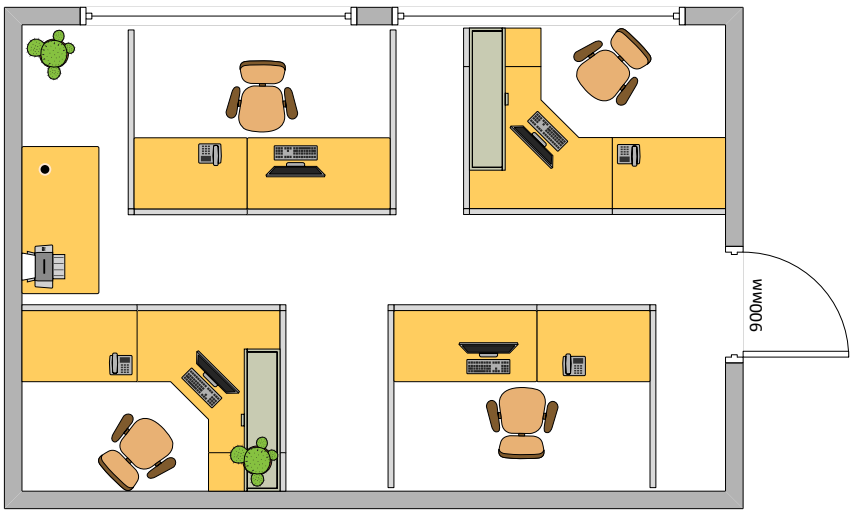

Рисунок 5.2 — рабочее место в офисе (как должно быть)

Представленная схема (как должно быть) решает описанные выше недочёты и позволяет комфортно работать в системе находясь в офисном помещении. Теперь окна занимают большую часть одной из стен, что позволяет получать достаточное освещение и хорошую вентиляцию, а на каждое место выделено примерно по 4,5 м^2

5.1.3 Требования к организации рабочего места возле ЭВМ

СанПиН 2.2.2/2.4.1340-03 также описывает допустимые уровни звукового давления и уровней звука, создаваемого ПЭВМ, не должны превышать значений, указанных в таблице 5.1

Таблица 5.1 – допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
86 дБ	71 дБ	61 ДБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

Также должны соблюдаться временные допустимые уровни электромагнитных полей (ЭМП), создаваемых ПЭВМ, которые не должны превышать значений, представленных в таблице 5.2.

Таблица 5.2 – временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25нТл
Напряженность электростатического поля		15 кВ/м

Стоит отметить, что современная техника достаточно тихая и излучаем минимальные уровни ЭМП, что сводит к минимуму нарушения в этих параметрах.

Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана монитора. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие устройства ПЭВМ должны иметь матовую поверхность. Данные требования стали неким

стандартом, для производителей офисной техники и соблюдаются почти повсеместно, как в офисах, так и в образовательных учреждениях.

Немало важным является высота рабочего места, а именно стола, которая должна быть подходящей для комфортной работы. В таблице 5.3 приведены примерные высоты стола для работы за вычислительной техникой.

Таблица 5.3 – временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах

Рост с	Высота над полом, мм	пространство для ног, не менее
146-160	640	580
161-175	700	640
выше 175	760	700

Исходя из этих данных можно подобрать рабочее место для каждого работника, чтобы обеспечить его максимальным удобством и соответственно получить максимальную отдачу.

5.1.4 Комплексы упражнений для работающих на компьютере

- исходное положение – сидя на стуле, руки на пояс. 1 – повернуть голову направо. 2 – исходное положение – то же налево. Повторить 6-8 раз. Темп медленный;

- исходное положение – руки в стороны. 1 – 4-восьмеркообразные движения руками. 5-8 – то же, но в другую сторону. Руки не напрягать. Повторить 4-6 раз. Темп медленный. Дыхание произвольное;

- исходное положение – стойка ноги врозь, руки на поясе. 1 – 3-три пружинящих движения тазом вправо, сохраняя исходное положение. плечевого пояса. 4 – исходное положение. Повторить 4-6 раз в каждую сторону. Темп средний. Дыхание не задерживать;

- исходное положение - основная стойка. 1 - руки в стороны, туловище и голову повернуть налево. 2 - руки вверх. 3 - руки за голову. 4 - и.п. Повторить 4-6 раз в каждую сторону. Темп медленный;

- исходное положение – основная стойка. 1 – шаг влево, руки к плечам, прогнуться. 2 – исходное положение. 3-4 – то же в другую сторону. Повторить 6-8 раз. Темп медленный.

5.2 Экологичность

Утилизация устаревшей или сломанной компьютерной техники является необходимой процедурой во множестве стран.

Отдельного положения, регулирующего утилизацию именно техники и компьютеров, в законодательстве нет. Утилизация и переработка осуществляются согласно общим положениям. Компьютер и комплектующие содержат не только ценные цветные металлы, но и целый набор опасных для окружающей среды веществ. Это производные газов, тяжелые металлы, среди которых кадмий, ртуть и свинец. Все эти вещества проникая в почву, отравляют воздух и воду. Также в процессе утилизации из техники извлекаются и материалы, которые могут быть использованы как вторичные ресурсы: материальные накопления сырья, веществ, материалов и продукции, образованные во всех видах производства и потребления, которые не могут быть использованы по прямому назначению, но потенциально пригодные для повторного использования в народном хозяйстве для получения сырья, изделий и/или энергии.

Отходы, содержащие в себе части цветных металлов, относятся к категории металлолома и подпадают под соответствующий регламент.

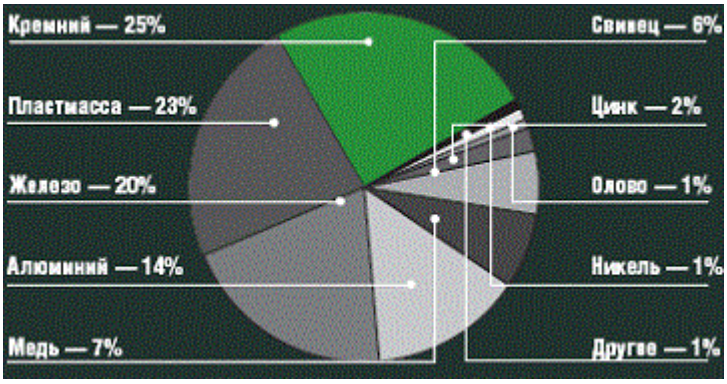

Рисунок 5.3 — основные виды металлов, использующиеся в технике и компьютерах

В случае, если техника содержит в себе опасные для окружающей среды вещества – тяжелые металлы, соли, – она подпадает под категорию опасных отходов. Самостоятельный вывоз техники в места скопления отходов без дальнейшей утилизации является нарушением законов, предусматривающие административную ответственность и штрафы.

Законодательными нормами не предполагается экспертиза на предмет выявления в технике веществ, подпадающих под особые положения, поэтому при утилизации следует ориентироваться на рекомендации от производителя, а также на нормы эксплуатации, предполагающие списание техники после определённого срока эксплуатации.

Так как компьютеры не подпадают под общее положение, они утилизируются в определенном порядке и только на предприятиях, имеющих официальные разрешения на переработку опасных отходов и металлолома.

Порядок по утилизации компьютерной техники

а) создание комиссии на предприятии, имеющем технику, которая определит какая именно техника нуждается в утилизации;

б) составление экспертного заключения о том, что техника действительно подлежит списанию;

в) составление акта технической экспертизы, подтверждающего действительность и целесообразность утилизации;

г) составление акта списания компьютерной техники с отображением в бухгалтерском учете предприятия;

д) утилизация техники на специализированном предприятии;

е) получение официального подтверждения в виде документа, сообщающего о том, что техника была утилизирована в соответствующем порядке.

Как видно из перечисленного списка действий, процесс утилизации достаточно трудоёмок, но необходим для соблюдения законодательства и предотвращения загрязнения среды, что является немало важной причиной. При сдаче на переработку техники количество не переработанных опасных отходов снижается.

К тому же пластик, различные черные и драгоценные металлы могут использоваться во вторичном производстве. Из «электронного мусора» можно выделить драгоценные металлы и дать старому «железу» новую жизнь.

5.3 Безопасность при возникновении чрезвычайных ситуаций

Одним из вероятных чрезвычайных ситуаций, возникающих при работе с ЭВМ и не любых других бытовых ситуациях – это пожар. В соответствии со ст. 1 Федерального закона от 21.12.1994 №69-ФЗ, пожарная безопасность – состояние защищенности личности, имущества, общества и государства от пожаров. Соответственно каждая организация, которая использует компьютерную технику, должна выполнять соответствующие требования, быть оснащена и подготовлена к подобным ситуациям.

Согласно Нормам пожарной безопасности (НПБ 105-95), помещения с ЭВМ относятся к категории В (пожароопасные). Основываясь на СНиП 21-01-97, вычислительные центры должны располагаться в зданиях не ниже II степени огнестойкости, залы электронно-вычислительных машин – не ниже первого этажа.

Помещения должны содержаться в чистоте. Все отходы необходимо ежедневно убирать в специально отведённые места. Коридоры, лестничные клетки, двери эвакуационных выходов, подходы к средствам тушения всегда должны быть свободны и ничем не загромождены. Мебель в помещениях не должна препятствовать быстрой эвакуации людей. Расположение электрических сетей должно исключать их повреждение и как следствие поражение работников электрическим током.

Помещения с ПЭВМ должны оснащаться аптечкой первой помощи и огнетушителями, количество и состав которых выбирают согласно Правилам пожарной безопасности ППБ-01-93. Согласно требованиям Правил ППБ-01-93, расстояние от возможного очага возгорания до места размещения огнетушителя не должно превышать 20 м, если ПЭВМ установлены в общественных зданиях и сооружениях; 30 м – для помещений ВЦ.

5.3.1 Первичные средства и установки пожаротушения

Для тушения пожара в начальной стадии развития в здания используются огнетушители в основном двух видов, а именно углекислые, предназначенные для тушения любых предметов, электроустановок под напряжением и ЭВМ, порошковые, который имеет схожее применение, но больше предназначен для тушения пожаров класса А, когда горят твердые сгораемые материалы, каковыми являются различные палки, доски, мебель, ковры, кучи бумажной документации, очаги, по типу лужи бензина и др. На рисунке 5.4 указана полная классификация огнетушителей по классам пожара.

Рисунок 5.4 - классификация огнетушителей по классам пожара

Наиболее же вероятные классы пожаров в помещениях с ПЭВМ - «А» (горение твёрдых веществ) и «Е» (возгорание электроустановок). При защите помещений с ЭВМ следует учитывать специфику взаимодействия огнетушащих веществ с защищаемыми оборудованием, изделиями, материалами и т.п. Данные помещения следует оборудовать хладоновыми и углекислотными или порошковыми огнетушителями с учетом предельно допустимой концентрации огнетушащего вещества.

Помещения с ПЭВМ, как правило, оснащают автоматической системой газового пожаротушения, однако в труднодоступных местах или местах хранения информации также рекомендуется устанавливать огнетушители ОСП.

В замкнутых помещениях объемом до 50 м3 вместо переносных огнетушителей можно использовать подвесные автоматически срабатывающие порошковые огнетушители (ОСП и другие).

5.3.2 Пожарная сигнализация

Для оповещения о пожарах используются ручные и автоматические средства. По способу передачи сигнала пожарная сигнализация может быть электрической и автоматической.

Для повышения безопасности (при отсутствии системы автоматического извещения о пожаре), особенно в небольших помещениях, рекомендуется устанавливать противопожарные дымовые датчики. Они крепятся на стену и имеют малые габариты. При задымлении помещения издают сигнал с уровнем звука 85 дБА.

Для избежания паники и быстрой безопасной эвакуации персонала (при возможном задымлении помещений и коридоров), в соответствии с требованиями НПБ-96, у дверных проемов, выключателей, рубильников, по пути возможной эвакуации для быстрого обнаружения шкафов с первичными средствами пожаротушения и т.п. следует размещать фотолюминесцентные эвакуационные знаки, выполненные по ГОСТ 12.4.026-01[7].

5.3.3 Мероприятия для обеспечения пожарной безопасности

Комплекс помещений вычислительных центров должен иметь не менее двух эвакуационных выходов. Двери, ведущим на лестничные клетки должны быть пожароустойчивыми.

Для звукоизоляции и акустической отделки стен и потолков должны применяться негорючие материалы, такие как пенопласт или стекловата.

Хранилища данных размещают в помещениях, оборудованных стеллажами из негорючих материалов.

Источники электрической энергии располагают в отведённых помещениях. Прокладка кабелей через перекрытия, стены должна осуществляться в стальных трубах с уплотнением из негорючих материалов. Аварийные сети освещения, дистанционного и автоматического пуска противопожарных систем и сигнализации прокладывают отдельно от силовых и других электрических сетей, а при совместной прокладке разделяют перегородками из негорючих материалов.

Система электропитания ЭВМ должна иметь защиту от перегрева, обеспечивающую отключение ее в случае остановки системы охлаждения и кондиционирования.

Воздуховоды выполняют из негорючих материалов. Система вентиляции должна быть оборудована устройством, закрывающим поступление воздуха в помещения с ЭВМ в случае возникновения пожара. Целесообразно установить датчики влажности (опасность коррозии, короткого замыкания и начальный риск затопления).

В помещениях для ЭВМ, размещаемых в одном помещении площадью не более 20 $м^2$, следует предусматривать автоматическую пожарную сигнализацию.

Для автоматического обнаружения пожаров могут быть использованы любые извещатели. Основные требования к ним состоят в том, чтобы они реагировали на изменения среды. Также такие датчики должны быть устойчивы к пожару.

Предусматривается, что в случае отказа основной системы оповещения, сработает дублирующая система. Они должны быть связаны с центральным постом безопасности и контроля.

Необходимы меры, обеспечивающие быстрое освобождение операционных залов от дыма, так как разъедающее влияние дыма может быть очень вредным для технических средств и персонала. Для этого применяются вытяжные системы.

Эффективность эвакуации людей может быть обеспечена достаточным количеством запасных выходов с хорошим освещением и многочисленными сигнальными огнями, и указателями на стенах, и т.д.

5.3.4 Действия при пожаре

При обнаружении пожара следует немедленно сообщить в пожарную службу:

- причину возгорания;
- адрес объекта;
- есть ли опасность для людей;
- назвать свою фамилию;
- немедленно обесточить всю электротехнику в помещении;
- обеспечить эвакуацию людей.

Далее оповестить всех работников организации, взять необходимо важные документы и материальные ценности и покинуть здание через запасные выходы. В случае невозможности покинуть здание, необходимо закрыть все двери, перекрыть вентиляцию, открыть окно и ожидать эвакуации. При ожогах обработать раны антисептиком.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе был поведён анализ предметной области, в результате которого были рассмотрены: актуальность использования фреймворков при разработке веб-приложений, правила конфиденциальности и защиты персональных данных, особенности Laravel и проведено сравнение с другими аналогами. Следующим этапом проведено проектирования веб-приложения «Система тестирования», итогом которого стало выявление целей и функционала системы, выбор направление проектирования и описание архитектуры приложения и базы данных. При описании разработки приложения выявлены минимальные аппаратные требования и обоснован выбор средств разработки, также описан процесс разработки базы данных, интерфейса и модулей приложения, создана документация по сопровождению системы. Далее был проведен анализ информационной безопасности системы, выявивший модель угроз и нарушителя, основные средства обеспечения информационной безопасности, а также была составлена политика безопасности для веб-приложения. Завершающим этапом стало описание главы по безопасности и экологичности, содержащей требования при работе за ЭВМ, действия для сохранения экологии при утилизации компьютерной техники и правила пожарной безопасности.

В результате было разработано и находится в тестировании веб-приложения «Система тестирования», которое позволяет полностью отказаться от ручного создания и проверки тестов, что увеличивает время на групповые и индивидуальные занятие со студентами и снижает нагрузку на самих преподавателей, так как не приходится тратить время на рутинную работу. Всё, что понадобится от конечного пользователя: установить систему на любой локальный или глобальный сервер, в модуле администрирования и управления контентом назначить необходимые роли, заполнить в удобной форме все необходимые тесты по разным предметам внести студентов в систему и записать их на прохождение тестов.

В итоге пользователь получит удобную и полнофункциональную систему для тестирования студентов с широкими возможностями по администрированию и составлению тестов, что и является приоритетной возможностью для всей системы в целом.

Итоговая версия программного продукта в настоящее время проходит официальную регистрацию.

ЗАТЕКСТОВЫЕ БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 https://tproger.ru: Веб-фреймворки [Электронный ресурс]. URL: https://tproger.ru/translations/web-frameworks-how-to-get-started/

2 Компьютерное тестирование знаний MyTestXPro [Электронный ресурс]. URL: http://www.mytest. klyaksa.net.

3 Программный комплекс NetTest [Электронный ресурс]. URL: http://www.kpolyakov.spb.ru/prog/nettest

4 Комплекс программ для компьютерного тестирования «АСТ-Тест» [Электронный ресурс]. URL http://www.ast-centre.ru/testirovanie/ast_test

5 Программа тестирования знаний «Айрен» [Электронный ресурс]. URL: http://www.irenproject.ru

6 Орлов, С.А. Технологии разработки программного обеспечения // учеб. СПб.: Питер, 2002. – 464 с.

7 Белокопытов, В.Н., Белокопытов А.В. Безопасность работы с ПЭВМ и копировально-множительной техникой. 2011

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Белокопытов, В.Н., Белокопытов А.В. Безопасность работы с ПЭВМ и копировально-множительной техникой. 2011

2 Орлов, С.А. Технологии разработки программного обеспечения // учеб. СПб.: Питер, 2012. – 464 с.

3 Петров, В.И. Информационные системы. СПб.: Питер, 2002. – 688 с.

4 Комплекс программ для компьютерного тестирования «АСТ-Тест» [Электронный ресурс]. URL: http://www.ast-centre.ru/testirovanie/ast_test (дата обращения: 25.05.2019)

5 Компьютерное тестирование знаний MyTestXPro [Электронный ресурс]. URL: http://www.mytest. klyaksa.net (дата обращения: 25.05.2019)

6 Комплекс Архитектура ИС, типы архитектур. Классификация ИС [Электронный ресурс]. URL: https://studopedia.ru/7_188991_arhitektura-is-tipi-arhitektur-klassifikatsiya-is.html (дата обращения: 03.05.2019)

7 Программный комплекс NetTest [Электронный ресурс]. URL: http://www.kpolyakov.spb.ru/prog/nettest (дата обращения:25.05.2019)

8 Программа тестирования знаний «Айрен» [Электронный ресурс]. URL: http://www.irenproject.ru (дата обращения: 25.05.2019)

9 DEVELOPER.MOZILLA.Org: Документация к языку программирования JavaScript [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/ (дата обращения: 13.03.2019)

10 DIGITALOCEAN.Com: Введение в OAuth 2[Электронный ресурс]. URL: https://www.digitalocean.com/community/tutorials/oauth-2-ru (дата обращения:03.05.2019)

11 HABR.Com: Архитектура REST [Электронный ресурс]. URL: https://habr.com/ru/post/38730/ (дата обращения:21.04.2019)

12 HABR.Com: Что такое API [Электронный ресурс]. URL: https:// habr.com/ru/post/38730/ (дата обращения:20.04.2019)

13 Laravel.com: Официальная документация фреймворка Laravel. Laravel Passport [Электронный ресурс]. URL: https://laravel.com/docs/5.8/passport (дата обращения: 15.04.2019)

14 Laravel.com: Официальная документация фреймворка Laravel. Laravel Dusk [Электронный ресурс]. URL: https://laravel.com/docs/5.8/dusk (дата обращения: 15.04.2019)

15 Laravel.com: Официальная документация фреймворка Laravel. Контроллеры Laravel [Электронный ресурс]. URL: https://laravel.com/docs/5.8/controllers (дата обращения: 15.04.2019)

16 Laravel.com: Официальная документация фреймворка Laravel. Eloquent ORM [Электронный ресурс]. URL: https://laravel.com/docs/5.8/eloquent-resources (дата обращения: 15.04.2019)

17 OAUTH.Net: Официальная документация протокола авторизации OAuth 2.0 [Электронный ресурс]. URL: https://oauth.net/2/ (дата обращения: 15.04.2019)

18 PHP.Net: Официальная документация языка программирования PHP [Электронный ресурс]. URL: https://www.php.net/manual/ru/ (дата обращения: 13.03.2019)

19 TPROGER.Ru: Веб-фреймворки [Электронный ресурс]. URL: https://tproger.ru/translations/web-frameworks-how-to-get-started/ (дата обращения: 10.03.2019)

20 VUEJS.Org: Официальная документация фреймворка Vue.js [Электронный ресурс]. URL: https://ru.vuejs.org/v2/guide/ (дата обращения: 15.04.2019)

ПРИЛОЖЕНИЕ А

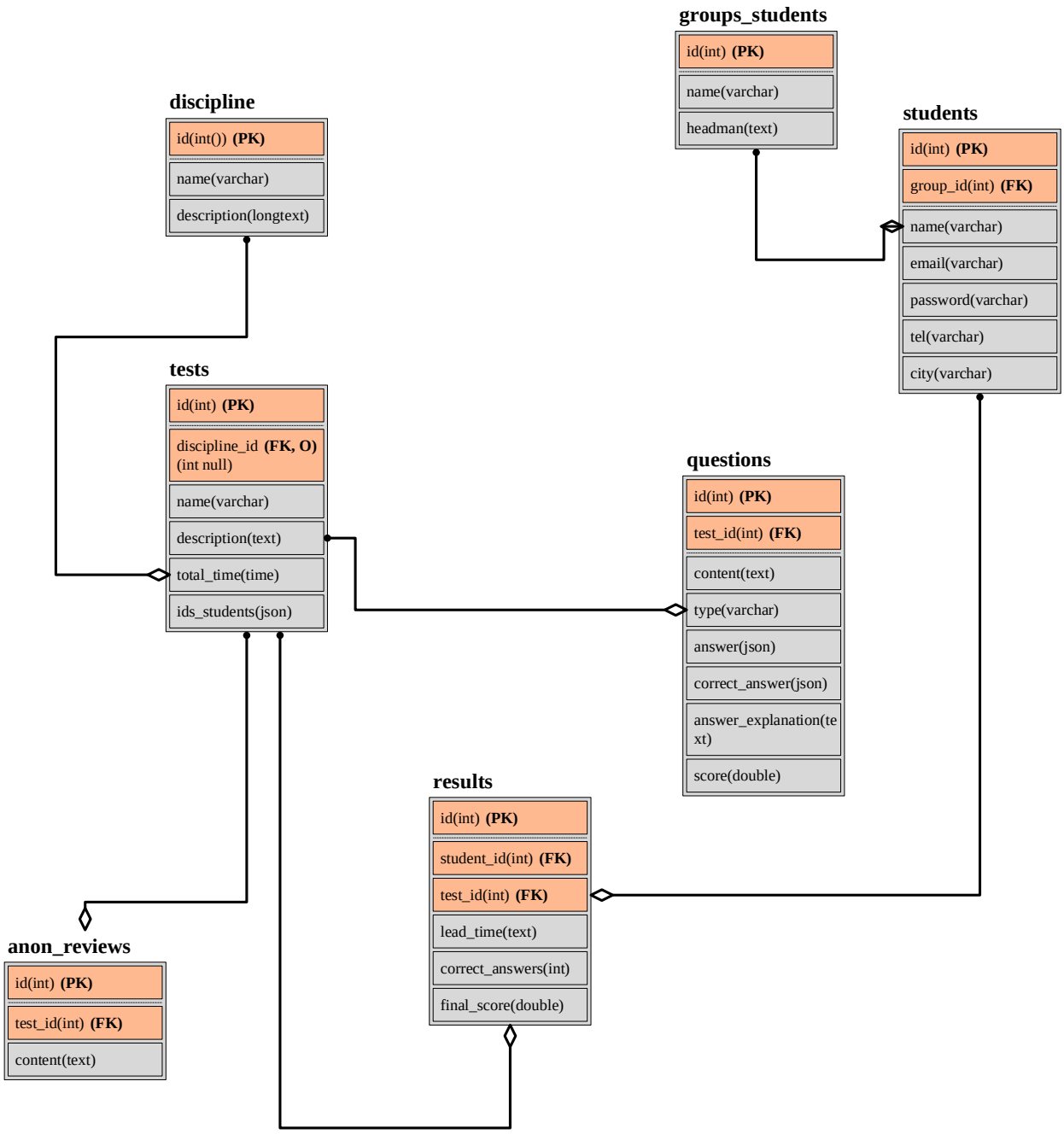

Рисунок А.1 – Физическая модель базы данных модуля панели управления

ПРИЛОЖЕНИЕ Б

Техническое задание

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование информационной системы

Наименование информационной системы: веб-приложение «Система тестирования»

1.2 Перечень документов, на основании которых разрабатывается информационная система

Используемые документы, на основании которых создаётся подсистема:

- Федеральный закон от 27 июля 2006 года № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;

- Федеральный закон от 27 июля 2006 года № 152-ФЗ «О персональных данных».

1.3 Плановые сроки начала и окончания работ

Начало работ: 15.04.2019.

Окончание работ: 03.06.2019.

1.4 Сведения об источниках и порядке финансирования работ

Информационная подсистема является разрабатывается без привлечения каких-либо финансовых средств, так как все компоненты являются свободно распространяемыми.

1.5 Порядок оформления и предъявления заказчику результатов разработки информационной системы

Система представляется заказчику в виде законченного проекта в установленные сроки и работает на базе средств вычислительной техники Заказчика.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Разрабатываемая система предназначена тестирования разных групп людей.

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

Данная подсистема позволит:

1) проводить тестирование студентов или работников организаций;

2) облегчить работу преподавателей.

2.2 Цели создания системы

- быстрое и удобное тестирование;

- увеличение эффективности работы преподавателей.

3 ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

3.1 Краткие сведения об объекте

Объектом автоматизации является проведение тестирований в учебных заведениях и организациях.

Автоматизации подлежат процессы:

- создания тестов и вопросов;

- проведение тестирования;

- подведение результатов.

3.2 Сведения об условиях эксплуатации объекта автоматизации и характеристика окружающей среды

Помещения, в которых предполагается размещение рабочего места, а также технических средств, должны соответствовать согласованным показателям температуры, влажности и освещённости.

Условия эксплуатации должны соответствовать нормальным климатическим условиям, определённым в ГОСТ 27201-87.

4 ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию системы

Разрабатываемая система должна состоять из следующих компонент:

1) ИС – Веб-приложение для тестирования.

1.1) Панель администратора.

1.2) Модуль заполнения вопросов к тесту.

1.3) Модуль записи студентов на курс.

2) Модуль хранения данных – База данных.

4.1.2 Требования к пользователям

Пользователями системы будут являться клиенты интернет-магазина и его администраторы.

Для работы с системой пользователю необходимо иметь базовые навыки работы с персональным компьютером.

Для работы с системой администратору сайта необходимо ознакомиться с эксплуатационной документацией.

4.1.3 Требования к надёжности

4.1.3.1 Состав и количественные значения показателей надёжности для системы в целом

Надёжность системы в целом определяется надёжностью функционирования её компонентов, а также надёжностью обеспечивающих технических и программных средств:

- технические средства;

- серверы, рабочие станции, сетевое аппаратное обеспечение;

- сетевые кабельные соединения, устройства бесперебойного питания;

- программные средства;

- системное программное обеспечение, установленное на серверах и рабочих станциях;

Прикладное программное обеспечение, установленное на серверах и рабочих станциях.

Для системы устанавливаются следующие количественные значения показателей надёжности:

- режим работы системы в целом – по необходимости;

- время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) серверов, не должно превышать 12-ти часов.

Перечень аварийных ситуаций

4.1.4 Требования к надёжности технических средств и программного обеспечения

Надёжность системы должна обеспечиваться:

- использованием качественных технических средств;

- профилактикой программного обеспечения;

- использованием бесперебойного интернета.

Назначенные сроки службы, среднее время наработки на отказ не устанавливаются, а определяются в соответствии с заявленными производителями характеристиками выбранных технических средств.

4.1.5 Требования к безопасности

Программно-аппаратные средства системы должны обеспечивать безопасность обслуживающего персонала при эксплуатации, техническом обслуживании и ремонте с учётом требований ГОСТ 21552-84, ГОСТ 25861-83.

Электробезопасность должна соответствовать требованиям ГОСТ 12.1.030-81, ГОСТ 12.2.003, ГОСТ 12.2.007.0-75.

В перечень задач, выполняемых системным администратором, должны входить:

а) задача поддержания работоспособности технических средств;

б) задача инсталляции и поддержания работоспособности системных программных средств – операционной системы;

в) задача инсталляции программ;

4.1.6 Требования к эксплуатации системы

Требования к эксплуатации системы включает в себя предоставление инструкций, методических и нормативных материалов по использованию и эксплуатации информационной системы.

4.1.7 Требования к защите информации от несанкционированного доступа

Информационная система должна соответствовать требованиями к защите информации от несанкционированного доступа. Система должна иметь

разграничение прав доступа к данным и использование ролевой системы, контроль правильной работы и разграничение прав должен осуществляться директором магазина. Должна быть предусмотрена защита базы данных.

4.1.8 Требования по сохранности информации при авариях

После аварии на сервере следует средствами СУБД проверить базу данных на наличие ошибок, и в случае их обнаружения по возможности исправить. Ущерб программному обеспечению в случае аварии мало вероятен, но при возникновении проблем рекомендуется переустановить систему.

4.2 Требования к функциям, выполняемым системой

4.2.1 Перечень подлежащих автоматизации задач

Разрабатываемая система должна автоматизировать следующие задачи:

- проведение тестирований;
- авторизация студентов;
- администрирование и управление контентом сайта;
- подведение итогов тестирования.

4.2.2 Временной регламент реализации каждой функции

Допускается естественная временная задержка в обработке данных при выполнении функции, связанной с загрузкой сети интернет или сервера.

4.2.3 Требования к качеству реализации каждой функции, формы выходной информации

Качество реализации функций должно обеспечивать полное выполнение выходящих в их состав операций и задач.

4.3 Требования к видам обеспечения

4.3.1 Требования к математическому обеспечению

Требования к математическому обеспечению не предъявляются

4.3.2 Требования к информационному обеспечению

Информационная система должна содержать данные зарегистрированных пользователей

4.3.3 Требования к лингвистическому обеспечению

Требования к лингвистическому обеспечению также предполагают использование единого логически понятного интерфейса для пользователей. Ввод и вывод данных должен производиться в удобном формате на русском языке.

4.3.4 Требования к программному обеспечению

Сайт располагается на хостинге (серверное ПО), а клиентам для работы с сайтом необходим только браузер.

4.3.5 Требования к техническому обеспечению

Требования к техническим средствам – минимальны. Разработанная архитектура системы обеспечит работоспособность системы на любой клиентской платформе, необходимо только наличие интернет-соединения.

4.3.6 Требования к методическому обеспечению

Требования к методическому обеспечению не предъявляются

5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

5.1 Перечень стадий и этапов работ по созданию системы

1 этап – исследование предметной области и сравнение с другими продуктами на рынке;

2 этап – составление технического задания, выявление целей и функций системы, составление требований;

3 этап – проектирование информационной системы;

4 этап – программная реализация информационной системы;

5 этап – тестирование продукта;

5.2 Состав организации исполнителя работ

Все виды работ выполняются студентом группы 555-об Манвеляном А.Ю.

6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

6.1 Виды, состав, объём и методы испытания

В процессе приёмки проекта информационной системы должны быть проведены следующие действия:

- анализ выполненной работы;

- проверка соответствий проекта поставленной задаче и обеспечения выполнения поставленных требований;

- определение достоинств и недостатков разработанной системы.

7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

8.1 Перечень подлежащих разработке документов

Состав и содержание документации должны соответствовать требованиям ГОСТ 34.201-89 и нормативно-технических документов.

Документация на проектируемую систему должна включать:

- рабочую документацию (на систему в целом, достаточную для ввода в действие, функционирования и обеспечения работоспособности системы);

- техническое задание.

9 ИСТОЧНИКИ РАЗРАБОТКИ

9.1 Документы и информационные материалы, на основании которых разрабатывается техническое задание

- ГОСТ 34.201-89 «Виды, комплексность и обозначение документов при создании автоматизированных систем»;

- ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания»;

- РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов»;

- ГОСТ 34.603-92 «Виды испытаний автоматизированных систем»;

- ГОСТ Р ИСО/МЭК 12207-99 «Процессы жизненного цикла программных средств».

ПРИЛОЖЕНИЕ В

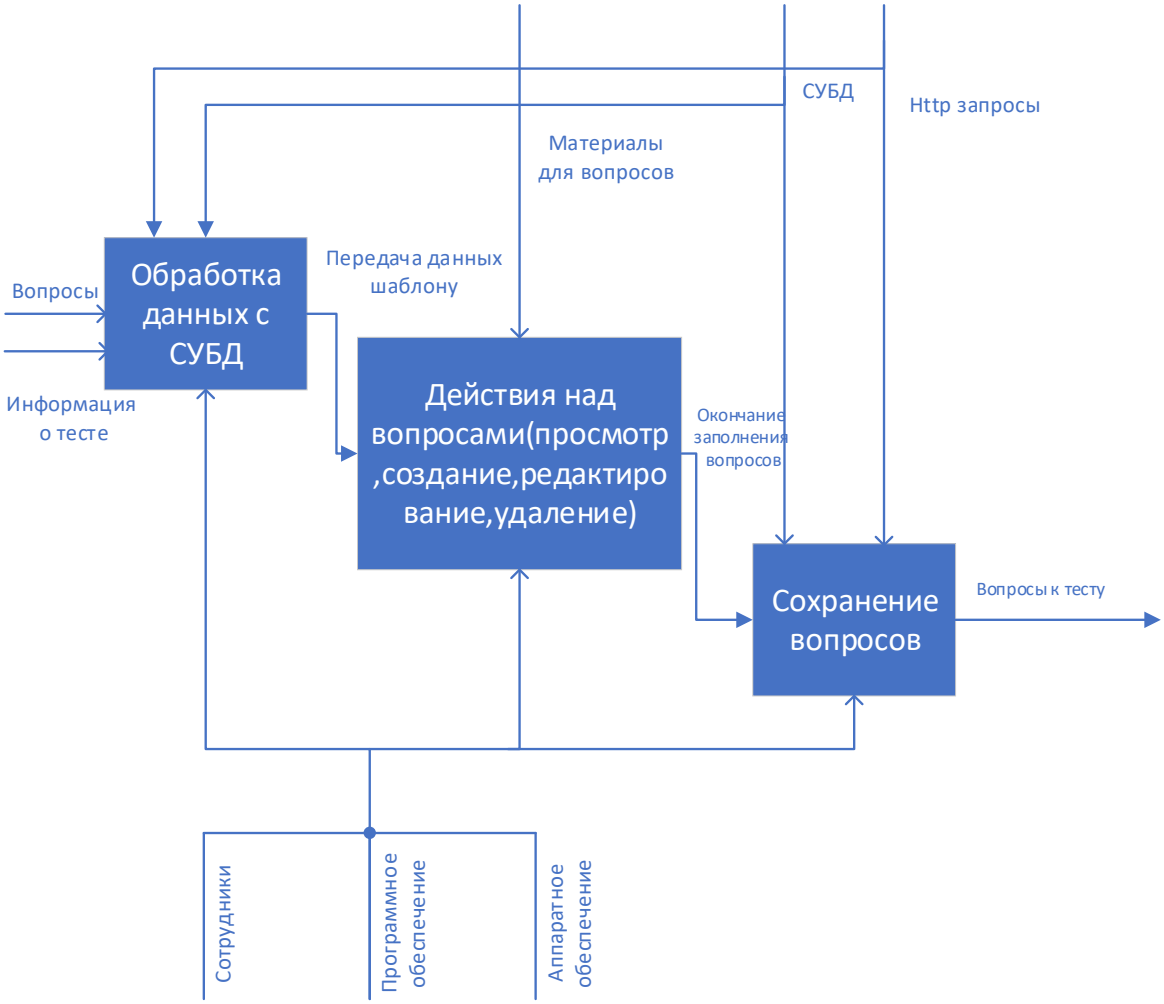

Рисунок В.1 – декомпозиция функциональной схемы IDEF0 конструктора вопросов

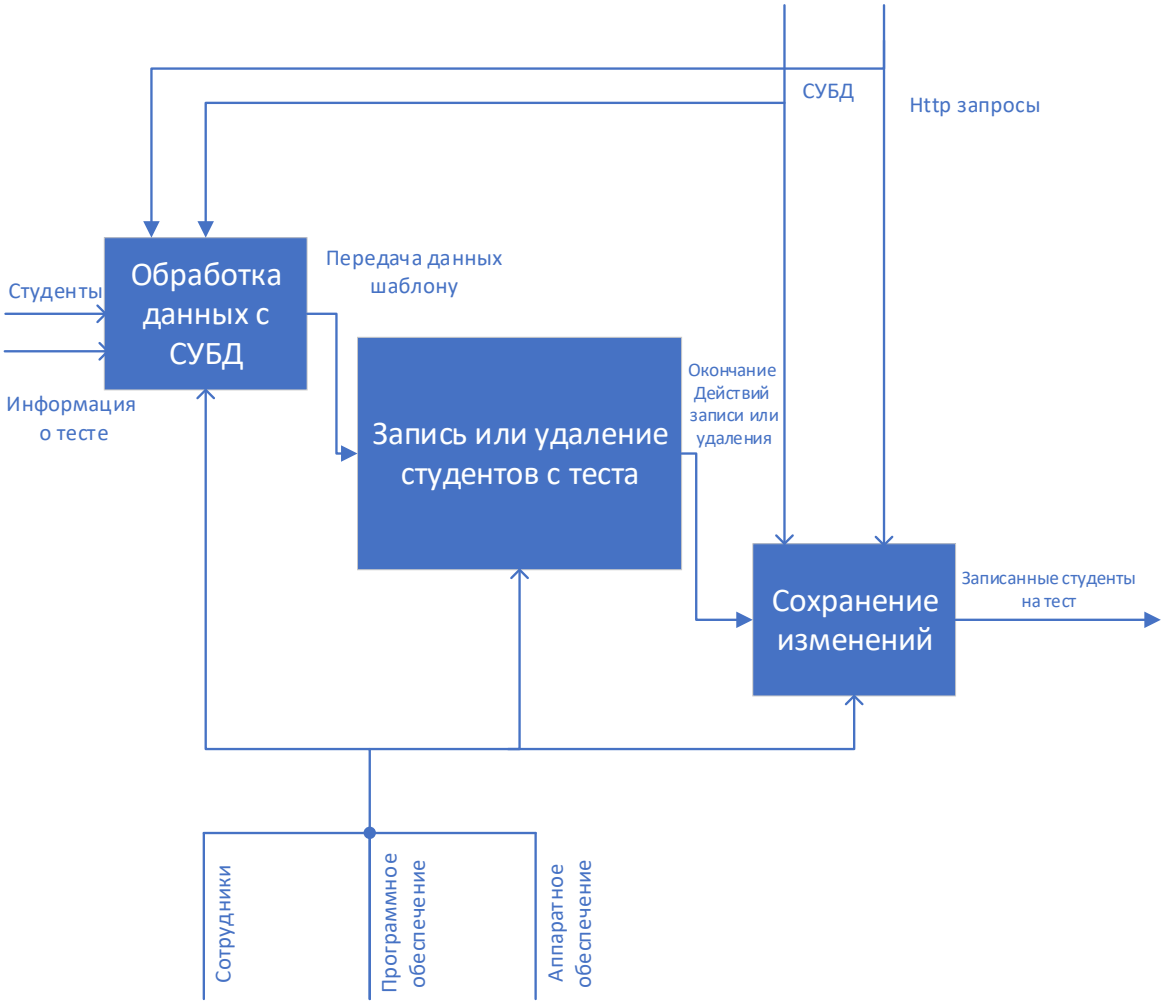

Рисунок В.2 – декомпозиция функциональной схемы IDEF0 модуля записи студентов на прохождение тестов

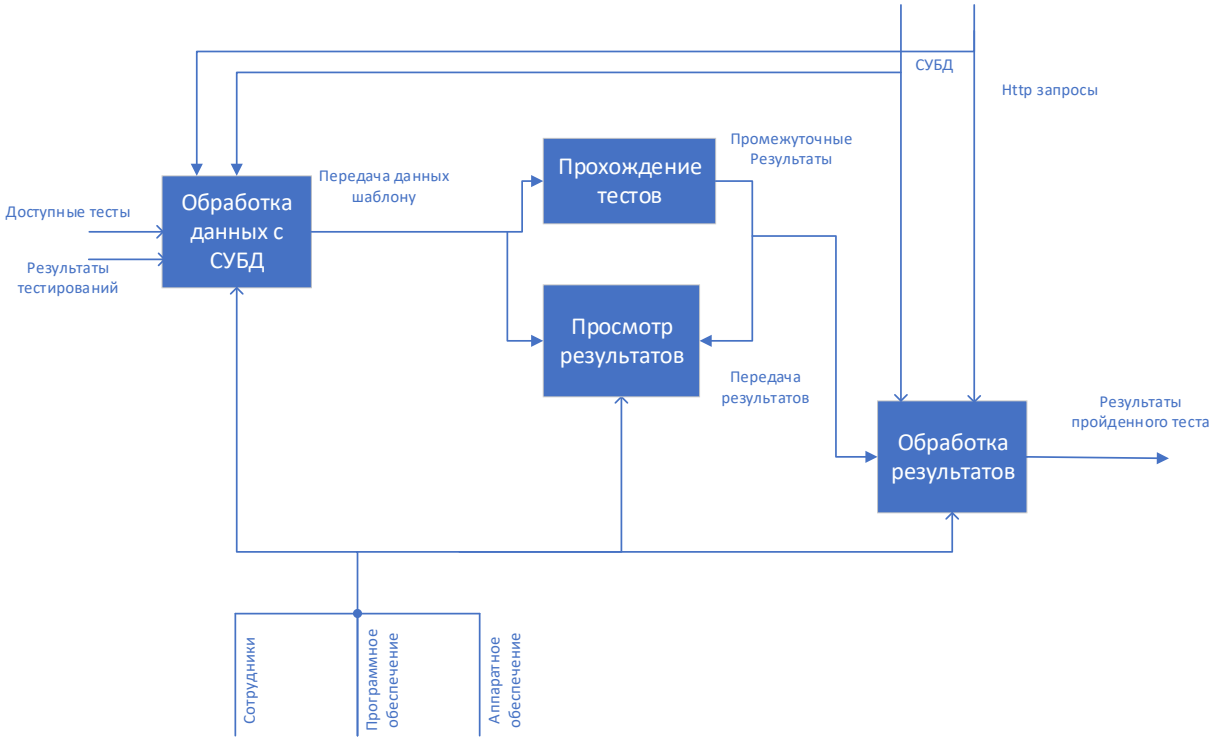

Рисунок В.3 – декомпозиция функциональной схемы IDEF0 личного кабинета пользователя