

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

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.01 – Информатика и вычислительная техника

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2024 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка шаблонов интерактивных заданий для образовательного web-сервиса «ActiveTemplate»

Исполнитель _____ Е.О. Яцун
студент группы 0103-об (подпись, дата)

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

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

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

Благовещенск 2024

РЕФЕРАТ

Бакалаврская работа содержит 105 с., 45 рисунков, 18 таблиц, 3 приложения, 25 источников.

ПРОГРАММНЫЙ ПРОДУКТ, ШАБЛОН ИНТЕРАКТИВНОГО ЗАДАНИЯ, ОБРАЗОВАТЕЛЬНЫЙ СЕРВИС, БАЗА ДАННЫХ

В выпускной квалификационной работе выполнено проектирование и разработка шаблонов интерактивных заданий «Кроссворд» и «Угадай слово» при помощи языка разметки HTML, языка программирования JavaScript. Для связи шаблонов с базой данных использованы средства языка программирования C#. Оба шаблона загружены в образовательный web-сервис «ActiveTemplate», проведено тестирование их работоспособности.

Цель работы – разработать и интегрировать в образовательный web-сервис «ActiveTemplate» набор шаблонов интерактивных заданий, обеспечивающих эффективный и увлекательный процесс обучения пользователей.

Задачами выпускной квалификационной работы являются:

- обоснование необходимости и актуальности разработки программного продукта;
- анализ теоретических аспектов и существующих решений исследуемой тематики;
- проектирование базы данных для хранения шаблонов и элементов, необходимых для их работы;
- разработка шаблонов заданий и форм для создания заданий для последующей интеграции в web-сервис;
- интеграция в web-сервис и тестирование разработанных шаблонов.

СОДЕРЖАНИЕ

Введение	6
1 Анализ предметной области	9
1.1 Тестовый контроль знаний в обучении	9
1.2 Обзор существующих методов решения рассматриваемой задачи	11
1.2.1 Конструктор интерактивных заданий «LearningApps»	11
1.2.2 Многофункциональный инструмент «Wordwall»	13
1.2.3 Конструктор учебных тренажёров «eТреники»	14
1.3 Формулировка цели и задач исследования	16
2 Проектирование программного продукта	17
2.1 Описание функционала web-сервиса	17
2.2 Проектирование взаимодействия шаблона с web-сервисом	21
2.3 Обоснование выбора средств реализации	22
2.4 Проектирование базы данных	28
2.4.1 Инфологическое проектирование	28
2.4.1.1 Сущность и их атрибуты	28
2.4.1.2 Связи между сущностями	32
2.4.1.3 Разработка инфологической модели	34
2.4.2 Логическое проектирование	35
2.4.2.1 Отображение концептуальной модели на реляционную	35
2.4.2.2 Нормализация отношений	38
2.4.2.3 Разработка логической модели базы данных	39
2.4.3 Физическое проектирование	40
3 Разработка программного продукта	43
3.1 Этапы разработки шаблонов	43
3.2 Реализация шаблона интерактивного задания «Кроссворд»	44
3.3 Реализация шаблона интерактивного задания «Угадай слово»	60
3.4 Интеграция шаблонов на web-сервис	67

4 Безопасность и экологичность	69
4.1 Безопасность	69
4.1.1 Требования к помещению для работы с ПЭВМ	70
4.1.2 Требования к уровням шума и вибрации при работе с ПЭВМ	71
4.1.3 Требования к освещению на рабочих местах с ПЭВМ	72
4.1.4 Требования к микроклимату рабочего места с ПЭВМ	73
4.1.5 Требования к организации рабочих мест с ПЭВМ	74
4.1.6 Требования к средствам отображения информации индивидуального пользования	77
4.2 Экологичность	78
4.3 Чрезвычайные ситуации	80
Заключение	83
Библиографический список	84
Приложение А	87
Приложение Б	96
Приложение В	105

ВВЕДЕНИЕ

С ростом технологического прогресса компьютерная техника внедряется во всё большее число отраслей повседневной жизни, работы, учёбы. Повышается количество людей, которые используют ЭВМ, глобальную сеть Интернет. Массовое распространение компьютерных технологий и Интернета позволяет внедрять различные информационные технологии, такие как веб-приложения.

Актуальность данной тематики состоит в том, что большинство современных молодых людей с большим желанием проводит свое свободное время в компьютерных сетях, что свидетельствует о значительном потенциале использования этих платформ для образовательных целей. Современные технологии предоставляют возможность не только разнообразить методы обучения, но и сделать их более интерактивными и увлекательными, что особенно важно для молодежи, привыкшей к динамичному и визуально насыщенному контенту.

Внедрение образовательных процессов в онлайн сети позволяет значительно увеличить заинтересованность учеников. Использование игровых элементов мультимедийных материалов, а также интерактивных заданий может сделать обучение более привлекательным и мотивирующим. Это способствует более глубокому усвоению материала и развитию навыков самостоятельного поиска информации.

Кроме того, образование с помощью глобальной сети предоставляет гибкость, которая является неотъемлемым преимуществом для многих учащихся. Возможность обучаться в любое время и в любом месте может позволить обучающимся лучше организовывать своё личное время. Это особенно важно, так как время часто является дефицитным ресурсом

Следует отметить, что использование информационных технологий в обучении постоянно растет. Поэтому с целью обеспечения постоянного до-

ступа к средствам обучения могут использоваться веб-приложения. Они обладают массой преимуществ по сравнению с классическими средствами обучения, что делает их все более популярными и востребованными среди учащихся и преподавателей.

Одним из ключевых преимуществ веб-приложений является возможность мгновенной оценки знаний учащихся. При прохождении онлайн тестирования результат можно увидеть сразу после завершения задания. Это позволяет учащимся моментально получить обратную связь о своих успехах и ошибках, что способствует более эффективному усвоению материала. Преподаватели, в свою очередь, могут оперативно отслеживать прогресс своих учеников и вносить необходимые коррективы в учебный процесс.

Веб-приложения также обладают приятным глазу интерфейсом, который делает процесс обучения более увлекательным и комфортным. Яркие визуальные элементы, анимации и интерактивные компоненты помогают удерживать внимание учащихся и делают обучение более захватывающим.

Синхронизация данных между различными устройствами обеспечивает непрерывность учебного процесса. Учащиеся могут выполнять одни задания на одном устройстве, а продолжать на другом, без потери прогресса. Это позволяет эффективно использовать свободное время и организовать обучение в соответствии с личным расписанием и предпочтениями.

Постоянный рост применения информационных технологий в этой сфере подчеркивает их значимость и потенциал для дальнейшего развития образовательных процессов.

Также следует отметить, что веб-приложения – это экологически чистый инструмент. Переход от бумажных носителей к онлайн процессам сокращает потребление таких товаров, как бумага, что косвенно уменьшает затраты природных ресурсов и снижает негативное воздействие на окружающую среду.

Использование веб-приложений в образовательном процессе способствует значительному снижению количества печатных материалов. Учебные

пособия, конспекты, тесты и другие учебные ресурсы, которые раньше существовали в бумажном формате, теперь доступны в цифровом виде. Это уменьшает потребность в производстве бумаги, что, в свою очередь, ведет к сокращению вырубки лесов.

Помимо сокращения использования бумаги, веб-приложения также уменьшают потребность в транспортировке учебных материалов. Традиционные бумажные учебники и тетради требуют логистики для их доставки в школы, университеты и другие учебные заведения. Цифровые ресурсы, напротив, могут быть мгновенно переданы через Интернет, что значительно снижает выбросы углекислого газа, связанные с транспортировкой.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Тестовый контроль знаний в обучении

Важным элементом в обучении является проверка качества усвоения знаний учащихся, которая дает возможность проверить уровень формирования основных навыков и понятий и своевременно восполнить пробелы. Основным инструментом тестового контроля является педагогическое тестирование.

Педагогическое тестирование играет важную роль в образовательном процессе, помогая учителям оценить уровень понимания материала учениками и выявить пробелы в их знаниях. Это ключевой инструмент для эффективной оценки усвоения учебного материала и адаптации обучения под индивидуальные потребности учащихся.

Подготовка качественных тестов – это первоочередная задача учителя. Тесты должны быть разработаны с учетом уровня сложности материала, целей обучения и индивидуальных особенностей учеников. Они должны быть четко структурированы, сбалансированными по содержанию и обеспечивать возможность проверить все необходимые навыки и знания.

Проведение тестирования требует внимательной организации процесса, чтобы обеспечить справедливость и достоверность результатов. Важно создать подходящие условия для прохождения теста, обеспечить конфиденциальность и надлежащий контроль за его выполнением.

Обработка результатов играет решающую роль в педагогическом тестировании. Это позволяет учителям оценить уровень успеваемости учеников, выявить общие тенденции и индивидуальные потребности, а также принять меры по дальнейшей адаптации обучения.

Тестирование в обучении выполняет три основные взаимосвязанные функции: диагностическую, обучающую и воспитательную.

Диагностическая функция заключается в выявлении уровня знаний, умений, навыков учащегося. Это основная и самая очевидная функция тестирования. По объективности, широте и скорости диагностирования, тестирование превосходит все остальные формы педагогического контроля.

Обучающая функция тестирования состоит в мотивировании учащегося к активизации работы по усвоению учебного материала. Для усиления обучающей функции тестирования могут быть использованы дополнительные меры стимулирования студентов, такие как: раздача преподавателем примерного перечня вопросов для самостоятельной подготовки, наличие в самом тесте направляющих вопросов и подсказок, совместный разбор результатов теста.

Воспитательная функция проявляется в периодичности и неизбежности тестового контроля. Это дисциплинирует, организует и направляет деятельность учащихся, помогает выявить и устранить пробелы в знаниях, формирует стремление развить свои способности.

Тестирование можно проводить как на бумажных носителях, так и при помощи компьютерных технологий, например, таких как веб-ресурсы, а именно веб-приложения.

С помощью веб-приложений в обучении можно вводить различные методы обучения, такие как лекции, интерактивные задания, тесты, делая их интерактивными, в виде игр, чтобы увеличивать вовлеченность обучающихся в процесс обучения, возможность использования мультимедийных элементов, таких как изображения, видео, аудио, в веб-приложениях обогащает обучающий опыт и облегчает восприятие информации. Также ввод современных технологий будет оказывать позитивный эффект на уровень образования.

Поскольку существует большое разнообразие проводимого тестирования, то основной особенностью разрабатываемого веб-приложения являются шаблоны интерактивных заданий. Они представляют собой заранее заготовленный набор функций или структуры, которые могут быть использованы для создания интерактивных обучающих материалов. Сами шаблоны могут яв-

ляться интерактивным модулем программы и позволять преподавателю задавать требуемые значения, которые поддерживает тот или иной шаблон. Сколько можно придумать различных видов заданий, столько можно разработать шаблонов. Можно сделать вывод, что использование шаблонов упрощает процесс создания интерактивных заданий и помогает улучшить эффективность обучения.

1.2 Обзор существующих методов решения рассматриваемой задачи

Основной задачей выпускной квалификационной работы является разработка шаблонов для образовательного веб-приложения. В разрабатываемом веб-приложении необходимо создать информационную систему, в которой возможно при помощи интерактивных шаблонов создавать интерактивные упражнения. Данные упражнения будут создаваться преподавателем, проходиться учениками. Результат пройденного упражнения будет сохраняться и будет доступен преподавателю для просмотра. Данная система будет эффективно способствовать процессу обучения, выполняя диагностическую функцию в обучении. Перед разработкой необходимо рассмотреть существующие методы решения рассматриваемой задачи.

1.2.1 Конструктор интерактивных заданий «LearningApps»

«LearningApps» – это конструктор интерактивных заданий, которые помогут закрепить знания в игровой форме. Сервис содержит Огромную библиотеку готовых упражнений. Она разбита на категории по предметам, конкретным темам и уровням образования, от дошкольного до профессионального.

Сервис позволяет создавать разные типы упражнений: «Хронологическая линейка», «Классификация», «Кроссворд», «Заполнить пропуски», «Викторина с выбором правильного ответа» и другие.

Сервис имеет простой и дружелюбный интерфейс (рисунок 1), полностью бесплатный доступ ко всем модулям и ко всей библиотеке упражнений. Присутствует возможность создания коллекций. Коллекции в LearningApps – это наборы упражнений, которые можно расставить по порядку и сопроводить

подсказками и инструкциями. Коллекции можно назвать своеобразным курсом заданий. В коллекциях присутствует возможность отслеживать кто какие задания выполнил. Также присутствует возможность добавить задание на другой сайт, где предусмотрена такая возможность, например, в систему дистанционного обучения.

К минусам можно отнести то, что в сервисе нет возможности просмотреть кто сколько прошел от курса, а также нет возможности установить, кто действительно прошел курс так как после прохождения можно подписаться любым именем. А также сами курсы реализованы неудобно в виде одной строки, на которой написано название курса и кнопка со списком прошедших.

К минусам можно отнести то, что сами шаблоны на данном сервисе имеют простой, незамысловатый дизайн, однако имеют возможность встраиваться на другие сайты при помощи ссылки, что является плюсом.



Рисунок 1 – Интерфейс веб-сервиса «LearningApps»

Сервис обладает большой широтой возможностей, благодаря богатому набору шаблонов, все из которых бесплатны для использования и возможности встраивания упражнений на другие веб-сайты. Однако он не лишен минусов, которые для системы образования достаточно критичны.

1.2.2 Многофункциональный инструмент «Wordwall»

«Wordwall» представляет собой многофункциональный инструмент для создания как интерактивных, так и печатных материалов. Интерактивные воспроизводятся на любом устройстве с веб-интерфейсом, например, на компьютере, планшете, телефоне или интерактивной доске. Они могут быть воспроизведены самими студентами, или студентами под руководством учителя.

Печатные можно просто распечатать или загрузить в виде файла PDF. Они могут быть использованы как вспомогательный материал к интерактивным или в качестве самостоятельных учебных заданий.

К плюсам можно отнести красивый дизайн и высокую функциональность сервиса, удобное отслеживание результатов по пройденным упражнениям вплоть до каждого ответа и времени прохождения (рисунок 2).

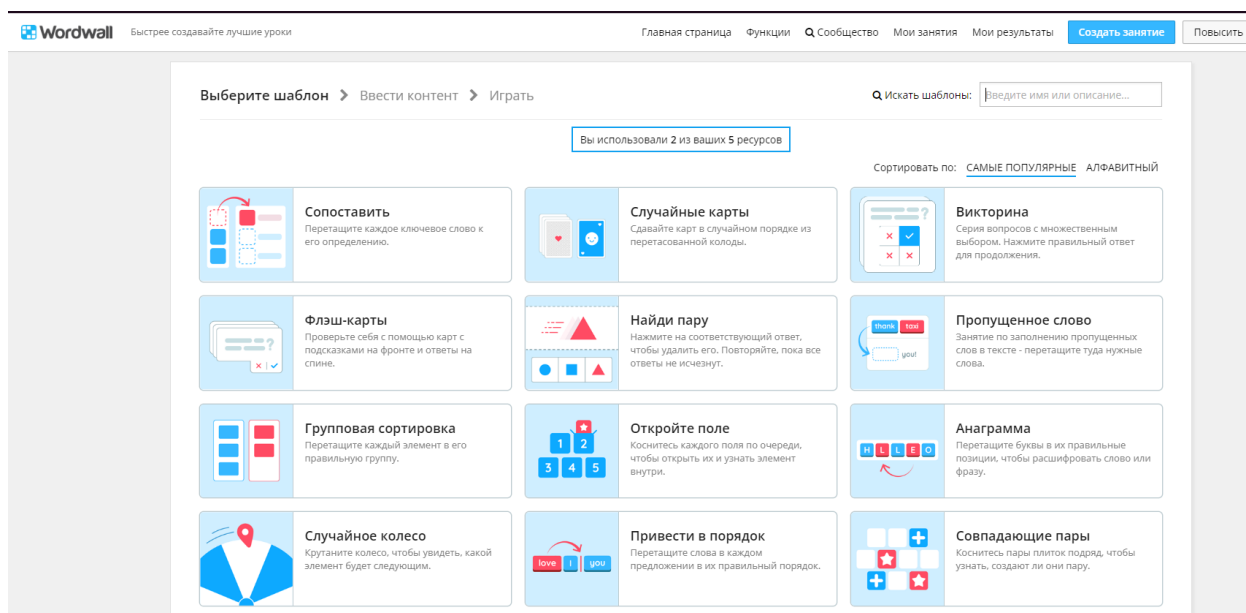


Рисунок 2 – Дизайн веб-сервиса «WordWall»

Возможность создания системы оценок. Сервис содержит 37 интерактивных и 19 печатных шаблонов. Некоторые шаблоны можно переключать на

уже созданных упражнениях, не создавая тем самым новое. Также присутствует возможность разместить упражнение на другом веб-сайте, используя фрагмент HTML-кода.

К минусам можно отнести отсутствие возможности создания курсов, как коллекции в LearningApps. Данный сервис является платным. Существует 3 основных тарифа: «Базовый», «Стандартный» и «Профессиональный». Базовый является бесплатным и выдается всем пользователям, которые только зарегистрировались на сервисе. Он имеет сильное ограничение на доступные шаблоны и количество заданий, которое можно создать. На данном тарифе можно создать лишь пять заданий, после чего необходимо приобретать тариф более высокого уровня. По тарифу «Стандартный» снимается ограничение на количество создаваемых заданий. А по тарифу «Профессиональный» открываются для использования все шаблоны.

Тем не менее, данный сервис обладает намного более широким функционалом, чем LearningApps, в особенности, в области просмотра результатов пройденных обучающимися упражнений.

1.2.3 Конструктор учебных тренажёров «Треники»

«Треники» – это онлайн-конструктор учебных тренажёров. Самих шаблонов для создания тренажеров на сервисе не так много, однако все они имеют красивые пользовательские интерфейсы (рисунок 3).

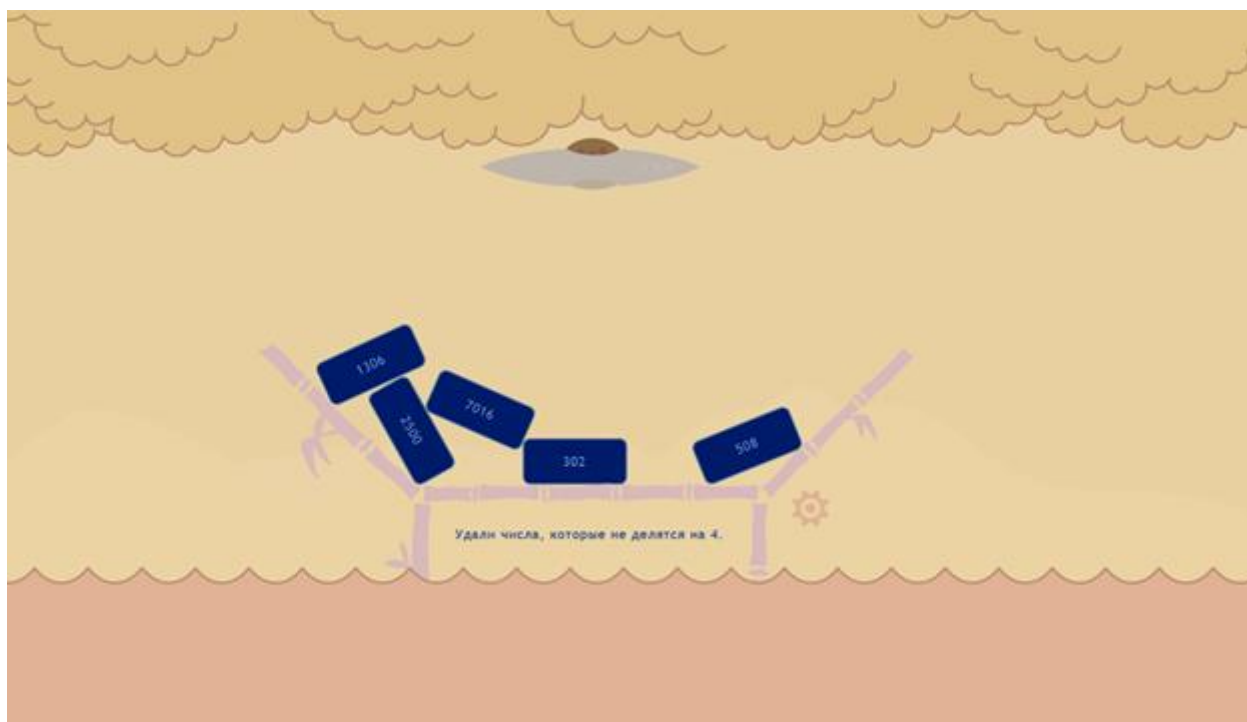


Рисунок 3 – Пример упражнения на сервисе «Треники»

Сервис полностью русскоязычный и бесплатный. Редактор довольно простой. Есть возможность изменять размер шрифта, что позволяет адаптировать тренажёры к воспроизведению на любых устройствах. Все готовые разработки публикуются в общем доступе. Основным плюсом являются сами шаблоны. Например, шаблон «НЛО» представляет собой плот, на котором написано условие задачи. Затем НЛО сбрасывает на плот блоки с ответами. Задача ученика в данном тренажере выбросить неправильные ответы. Или, например, шаблон «Кокла», в котором обучающийся должен построить за время дорожку, до правильного ответа, куда скатится блок с вопросом.

Основные минусы в этом сервисе: отсутствие возможности встраивать тренажёры на другой сайт. А также отсутствие возможности как на предыдущих рассмотренных сервисах отслеживать результаты обучающихся, прошедших тренажеры. На момент выполнения работы автор приостановил поддержку и развитие данного сервиса.

Исходя из рассмотренных примеров было принято решение о разработке веб-сервиса, который будет помогать преподавателям в учебном процессе.

1.3 Формулировка цели и задач исследования

Основная цель исследования – разработать и интегрировать в веб-приложение «ActiveTemplate» шаблоны интерактивных заданий, обеспечивающих эффективный и увлекательный процесс обучения пользователей.

Задачи выпускной квалификационной работы: обоснование необходимости и актуальности разработки программного продукта; анализ теоретических аспектов и существующих решений исследуемой тематики; проектирование базы данных для хранения шаблонов и элементов, необходимых для их работы; разработка шаблонов заданий и форм для создания заданий для последующей интеграции в web-сервис; интеграция в web-сервис и тестирование разработанных шаблонов.

Основные требования к работе заключаются в создании программного продукта, обладающего следующими функциями: обеспечение возможности выбора количества вопросов; обеспечение возможности добавления или удаления вопроса в задании; обеспечение возможности добавления или удаления вариантов ответа в задании; обеспечение возможность менять фон у макета задания; обеспечение возможности указывать время на прохождение упражнения; автоматическая генерация полей упражнения; генерация упражнения по заданным параметрам.

В качестве входных данных для программы будут использоваться информация, введенная на формах, содержащихся на HTML страницах. На странице для создания задания, для пользователя преподавателя. И на форме самого задания для пользователя ученика.

Выходной информацией будет информация в виде HTML страниц, на которых будут отображаться вопросы, при необходимости возможные варианты ответа и тому подобное.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1 Описание функционала web-сервиса

Создание шаблонов происходит в рамках веб-приложения, функционал которого представлен на рисунке 4 в виде диаграммы вариантов использования.

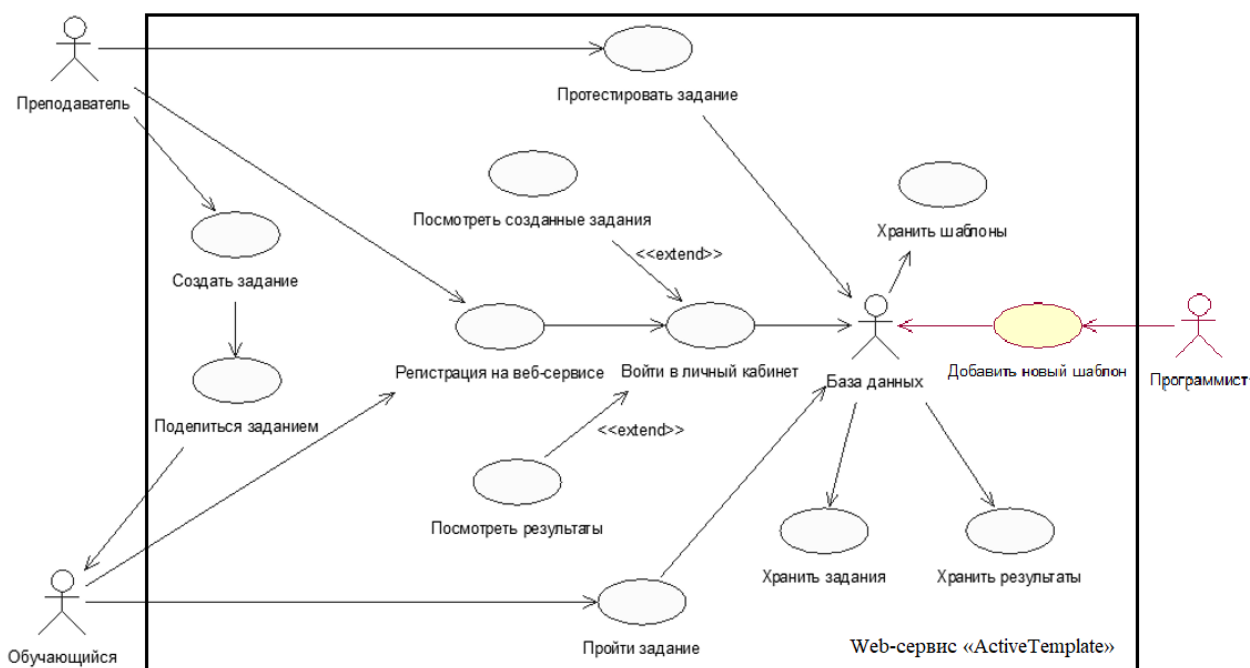


Рисунок 4 – Диаграмма вариантов использования веб-сервиса

Основными действующими лицами являются:

- преподаватель;
- обучающийся;
- база данных;
- программист.

Описания вариантов использования диаграммы прецедентов описаны в таблице 1.

Таблица 1 – Описания вариантов использования диаграммы прецедентов

Вариант использования	Краткое описание
Регистрация на веб-сервисе	Пользователи должны иметь возможность зарегистрироваться на веб-сервисе для получения доступа к функциям сервиса.

Войти в личный кабинет	Пользователи должны иметь возможность авторизоваться на веб-сервисе для получения доступа к функциям сервиса.
Создать задание	Преподаватель должен иметь возможность, используя свои данные, создать задание для прохождения учениками с помощью шаблона.
Протестировать задание	Преподаватель должен иметь возможность перед завершением создания задания просмотреть предварительную версию по введенным им данным.
Посмотреть созданные задания	Преподаватель должен иметь возможность посмотреть какие задания он создал.
Поделиться заданием	Преподаватель должен иметь возможность получить и скопировать ссылку на прохождение созданного им задания.
Посмотреть результаты	Преподаватель должен иметь возможность посмотреть результаты учеников, прошедших созданное им задание. Ученик должен иметь возможность посмотреть результаты пройденного им задания.
Пройти задание	Ученик должен иметь возможность пройти задание по ссылке, предоставленной ему преподавателем.
Хранить задания	База данных должна хранить созданные пользователем задания.
Хранить шаблоны	База данных должна хранить созданные программистами шаблоны заданий.
Хранить результаты	База данных должна хранить результаты прохождения заданий.
Добавить новый шаблон	Программист должен иметь возможность добавлять новые шаблоны.

Для данной работы ключевыми являются действующие лица «Преподаватель», «Обучающийся», поскольку с их стороны рассматривается процесс взаимодействия через шаблоны с web-сервисом. А также действующее лицо «Программист», так как с его стороны рассматривается процесс добавления шаблонов на web-сервис.

После разбора действующих лиц и их вариантов взаимодействия с веб-сервисом необходимо рассмотреть какие действия будут совершать пользователи при взаимодействии. Для этого подходят диаграммы последовательностей UML – это диаграммы взаимодействия, в которых подробно описывается, как выполняются операции. Они фиксируют взаимодействие между объектами в контексте сотрудничества. Диаграммы последовательности ориентированы на время и визуально показывают порядок взаимодействия.

На рисунке 5 изображена диаграмма последовательности варианта использования «Создать задание».

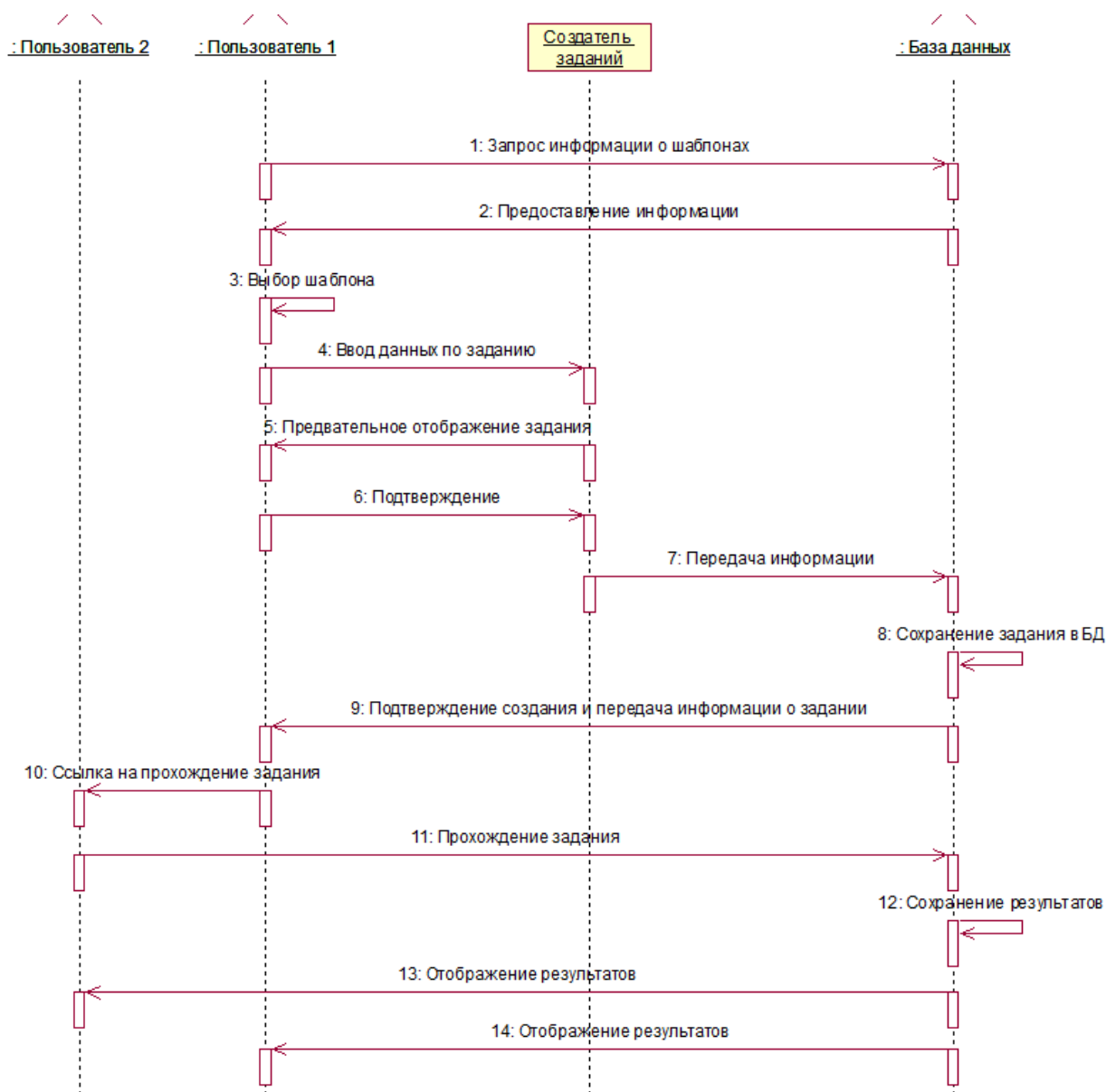


Рисунок 5 – Диаграмма последовательности

Данная диаграмма отображает последовательность действий, совершаемые преподавателем и обучающимся при их взаимодействии с веб-сервисом. Она отображает действия, которые происходят во время вариантов использования между данными действующими лицами.

Для системы была создана диаграмма состояний (рисунок 6). Она определяет все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий.

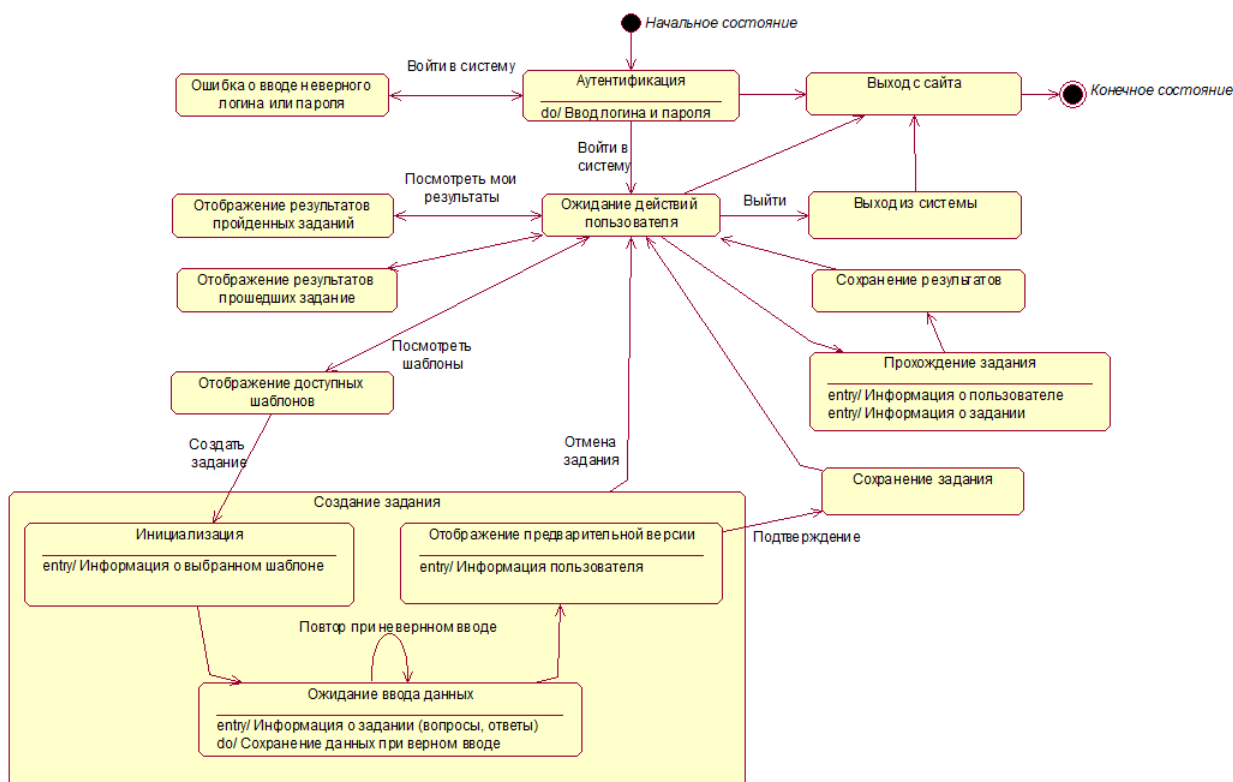


Рисунок 6 – Диаграмма состояний веб-сервиса

Основными состояниями являются:

- аутентификация;
- ошибка о вводе неверного логина или пароля;
- ожидание действий пользователя;
- отображение результатов пройденных заданий;
- отображение результатов, прошедших задание;
- отображение доступных шаблонов;

- создание задания, которое включает в себя состояния «Инициализация», «Ожидание ввода данных» и «Отображение предварительной версии»;
- сохранение задания;
- прохождение задания;
- сохранение результатов;
- выход из системы;
- выход из сайта;
- а также начальное и конечное состояния.

Для данной работы важным являются состояние «Создание задания» с вложенными в него состояниями.

Данные диаграммы помогают описать различные группы пользователей и их взаимодействие. Также это помогает определить, какая функциональность должна быть включена в систему и выявить потенциальные проблемы и ограничения, связанные с дизайном и разработкой.

2.2 Проектирование взаимодействия шаблона с web-сервисом

Поскольку шаблон задания является самостоятельной программой, то он должен взаимодействовать с web-сервисом. В процессе интеграции шаблона на web-сервис ему должны быть добавлены функции, которые обеспечивают взаимодействие с базой данных, а именно для сохранения в базе данных информации о созданном на основе шаблона задании. Также генерация на сервисе ссылки для прохождения данного задания. При переходе по ссылке шаблон должен понимать какие данные ему нужно брать из базы, чтобы сгенерировать созданное пользователем задание.

Для шаблонов входными данными при создании задания являются: шаблон, по которому создается задание; название темы, по которой создано задание, например «О природе», может содержать как символьные знаки, так и числовые; вопрос, на который необходимо дать ответ, может содержать символьные и числовые знаки; ответ на вопрос, должен содержать только буквы, написанные на одном языке.

Выходными данными будут записи в таблицах базы данных о задании.

Также при прохождении задания входными данными будут данные из таблиц базы данных, а также вводимые пользователем данные при прохождении заданий. Выходными данными будет представление, отображаемое через браузер пользователя при открытии задания, а также реакция программы на действия пользователя.

2.3 Обоснование выбора средств реализации

Поскольку создаваемые шаблоны будут встраиваться в веб-сервис, то для их разработки были выбраны следующие языки:

HTML. Основная цель языка – структурировать и оформлять контент на сайте. Он создаёт иерархическую структуру веб-страницы, используя заголовки, абзацы, списки и таблицы. Такая структура помогает пользователю легче ориентироваться на сайте.

С помощью HTML можно:

- делать текстовую разметку,
- встраивать медиа,
- создавать ссылки и навигацию,
- создавать таблицы,
- создавать формы.

CSS формальный язык декодирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки, чаще всего HTML или XHTML.

Основная цель разработки CSS – ограждение и отделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом;

JavaScript – это язык программирования, который в первую очередь применяют в веб-разработке.

С его помощью сайты делают интерактивными: добавляют всплывающие окна, анимацию, кнопки лайков и формы для отправки информации.

JavaScript позволяет:

- делать сайт интерактивным;
- передавать информацию между пользователем и сервером;
- анимировать объекты;
- производить вычисления;
- писать серверную часть сайта или приложения.

Области применения JavaScript:

- веб-сайты и веб-приложения;
- расширения для браузера;
- мобильные приложения;
- серверная часть сайтов и программ;
- игры.

Поскольку веб-сервис реализован на технологии ASP.NET CORE, то также необходимы знания языка программирования C#. ASP.NET Core – свободно распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET. Он используется для связывания базы данных с шаблоном заданий.

Выбор среды программирования очень важен, поскольку от этого зависит комфорт и эффективность работы. Каждая среда программирования имеет свои особенности, инструменты и функциональность, которые могут повлиять на производительность кода, его качество и скорость разработки.

Также при выборе необходимо помнить о выбранных языках программирования и фреймворках, поскольку не все среды имеют возможность работать с выбранными технологиями.

При выборе среды разработки необходимо учитывать следующие факторы:

- подсветка синтаксиса или цветовое кодирование. Выделение синтаксических конструкций текста с использованием различных цветов, шрифтов и начертаний. Обычно применяется для облегчения чтения исходного текста программ, улучшения визуального восприятия;

– контроль версий. Если вы когда-нибудь захотите откатить изменение или просмотреть предыдущую версию своего кода, поищите функции контроля версий, которые хранят старые копии. Это практически необходимо при работе с другими разработчиками;

– автосохранение. Защита от потери продленной работы при возникновении непредвиденной ошибки является важной составляющей при выборе среды разработки, поэтому необходимо выбирать среду, которая включает функцию автосохранения на случай непредвиденного сбоя;

– автозаполнение и предложения. В редакторе кода автозаполнение позволяет быстро заполнять более длинный код нажатием кнопки при появлении предложения. Что позволяет сократить время написания кода и уменьшить вероятность ошибки при ручном написании;

– сворачивание кода. Данная функция помогает при работе над большим документом, сворачивание кода позволяет закрыть ненужные части документа и сосредоточиться на определенных областях;

– возможность использования нескольких курсоров или одновременного редактирования. Функциональность с несколькими курсорами позволяет писать код в нескольких местах одновременно. Данная функция полезна при добавлении дубликатов одного и того же фрагментов кода;

– поиск и замена. Данная функция позволяет находить определенные строки и заменять их на другие;

– обнаружение ошибок. Данная функция позволяет определять правильность написания кода, например уведомлять если не был поставлен специальный символ;

– стоимость. Многие среды разработки являются платными и стоимость их может достигать больших сумм.

Наиболее популярными редакторами кода являются: Visual Studio, Notepad++ и Sublime Text.

Notepad++ – простой и удобный редактор с открытым исходным кодом. В нём есть подсветка синтаксиса нескольких языков, в том числе и JS, автоматическое форматирование и автодополнение. Присутствует навигация в виде вкладок, файлового менеджера и карты кода.

Доступны работа с разными кодировками, подключение компиляторов, использование плагинов и другие полезные функции. Например, добавив QuickText, можно улучшить встроенное автодополнение.

Notepad++ регулярно обновляется, поэтому в нём быстро исправляют баги и добавляют новые возможности. Работает только на Windows.

Visual Studio 2022 – это интегрированная среда разработки (IDE) от компании Microsoft, предназначенная для создания различных типов приложений на различных платформах. Она предоставляет широкий набор инструментов, библиотек и ресурсов для разработки, отладки, тестирования и развертывания программного обеспечения.

Visual Studio включает в себя редактор кода, отладчик, дизайнер форм, поддержку различных языков программирования (C++, C#, F#, Visual Basic, Python и другие), инструменты для управления версиями кода, создания пользовательских интерфейсов, а также инструменты для управления проектами и сборкой приложений.

Однако весит программа достаточно много и систему загружает сильно, поэтому не очень подойдёт для работы на слабых компьютерах.

Sublime Text – это простой кроссплатформенный редактор. Его интерфейс настраивается, а выполнять некоторые действия можно с помощью горячих клавиш.

Также в нём есть поддержка автодополнений, автосохранений, подсветки синтаксиса и навигация.

Главный минус Sublime Text в том, что полная версия стоит 80 долларов. Лицензия выдаётся на одного пользователя, который может использовать редактор на любом компьютере. В остальном он хорошо подходит JS-разработчикам.

Для того, чтобы определиться с программным обеспечением был проведен анализ и сравнение возможностей по пунктам, описанным выше (таблица 2).

Таблица 2 – Сравнительная характеристика программных сред

Особенность	Название программной среды		
	Visual Studio	Notepad++	Sublime Text
Подсветка синтаксиса	Есть	Есть	Есть
Сворачивание кода	Есть	Есть	Есть
Контроль версий	Встроена система git	Только с сторонним ПО	Только в платной версии
Автосохранение	Есть	Нет	Есть
Автозаполнения и предложения	Есть, со встроенной нейросетью	Есть без предложений	Есть, без предложений
Одновременное редактирование	Есть	Нет	Есть
Поиск и замена	Есть	Есть	Есть
Обнаружение ошибок	Есть	Частично	Есть
Стоимость	Есть бесплатная версия и платная с расширенным функционалом	Бесплатно	Платная

На основании сравнительной таблицы можно сделать вывод, что Visual Studio обладает наибольшим функционалом и бесплатной версией. Наиболее полезными функциями данной среды являются: контроль версий благодаря встроенной системе git, которая позволяет сохранять как локальные, так и удаленные репозитории система позволяет отслеживать любые изменения в файлах, хранить их версии и оперативно возвращаться в любое сохранённое состояние. А также автозаполнения и предложения со встроенным расширением IntelliCode, благодаря которому среда предлагает наиболее подходящие элементы кода для добавления.

Также выбор пал на данную среду, потому что, Visual Studio имеет богатую экосистему расширений, пакетов и инструментов для разработки от сторонних производителей, что позволяет расширять функциональность среды разработки в соответствии с конкретными потребностями. Например, используются пакеты для работы Entity framework, необходимо для связывания шаблонов с базой данных.

Среда для создания баз данных – это программное обеспечение, которое позволяет разработчикам создавать, изменять и управлять базами данных. Она включает в себя инструменты для проектирования баз данных, создания таблиц, добавления данных, изменения структуры базы данных и многое другое. Среды для создания баз данных обычно используются разработчиками приложений для работы с данными, а также администраторами баз данных для управления и оптимизации производительности.

В процессе выбора среды для создания базы данных выбор пал на SQL Server Management Studio.

SQL Server Management Studio (SSMS) – это удобная и мощная интегрированная среда для управления и разработки SQL Server. SSMS позволяет создавать новые базы данных, изменять их структуру, а также удалять и восстанавливать базы данных. Предоставляет возможности для управления объектами баз данных, такими как таблицы, индексы, представления, хранимые процедуры, триггеры и другие элементы.

В SQL Server Management Studio разработчики и администраторы могут выполнять свою работу в одной удобной среде, предоставляющей доступ ко всем инструментам и ресурсам, необходимым для работы с базами данных SQL Server. SSMS включает в себя редакторы, позволяющие разрабатывать и отлаживать SQL-запросы, хранимые процедуры, триггеры и другие скрипты базы данных.

Пользователи могут управлять настройками и конфигурациями SQL Server, включая безопасность, резервное копирование, планирование задач и

другие аспекты управления базами данных. Интегрированная среда разработки в SSMS позволяет пользователям эффективно выполнять различные задачи, связанные с разработкой, тестированием, администрированием и оптимизацией баз данных, не переключаясь между различными инструментами и окружениями. SQL Server Management Studio (SSMS) предоставляет возможности для анализа данных и создания разнообразных отчетов, что позволяет пользователям осуществлять детальный анализ данных и предоставлять информационные отчеты для принятия решений.

В целом, выбор SQL Server Management Studio основан на его мощных функциях, интеграции с SQL Server, надежности и производительности, мощных инструментах разработки и удобном пользовательском интерфейсе. Это делает SSMS превосходным выбором для создания, разработки и администрирования баз данных SQL Server.

2.4 Проектирование базы данных

2.4.1 Инфологическое проектирование

Инфологическое (концептуальное) проектирование – процесс создания внешней (инфологической) модели данных о предметной области, не зависящее от любых физических аспектов ее представления. На этапе инфологического проектирования решаются вопросы:

- о каких объектах или явлениях требуется накапливать или использовать информацию;
- какие их основные характеристики и взаимосвязи должны при этом учитываться.

На данном этапе необходимо выявить необходимые сущности и атрибуты будущей базы данных.

2.4.1.1 Сущность и их атрибуты

На основании описания предметной области и поставленной задачи можно выделить следующие сущности, которые должны содержаться в базе данных web-сервиса:

– сущность «Шаблон», в которой будет содержаться информация о созданных шаблонах, которые в свою очередь будут отображаться на сайте. Атрибуты сущности можно увидеть в таблице 3;

Таблица 3 – Атрибуты сущности «Шаблон»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Код Шаблона</u>	Код шаблона.	Числовой	>0	11
Название шаблона	Имя, которое отображается на сайте.	Текст	-	Кроссворд
Изображение	Изображение, которое отображается в маске на сайте.	Изображение	-	-
Описание шаблона	Описание, которое будет отображаться, при наведении мыши на шаблон на сайте.	Дата	-	2024-06-28

– сущность «Задание», в которой будет содержаться информация о заданиях, которые будут создавать пользователи сайта из шаблонов. Атрибуты сущности можно увидеть в таблице 4;

Таблица 4 – Атрибуты сущности «Задание»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Код Задания</u>	Код задания.	Числовой	>0	88
Тема задания	Тема, задаваемая для задания.	Текст	-	Программирование
Описание задания	Описание, в личном кабинете, создавшего задание.	Текст	-	Для группы 0103-об
Изображение	Задний фон для задания.	Изображение	-	-

Дата создания	Дата, когда задание было создано.	Дата	-	08-03-2024
Время	Время, которое дается на выполнение задания	Числовой	>0	5000 (в секундах)

– сущность «Вопрос», в которой будет содержаться информация о вопросах, которые содержатся в созданном задании. Атрибуты сущности можно увидеть в таблице 5;

Таблица 5 – Атрибуты сущности «Вопрос»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Id Вопроса</u>	Код вопроса.	Числовой	>0	512
Тело вопроса	Текст, который будет содержаться в вопросе	Текст	-	Зеленое животное, обитающее в мелких водоемах.
Порядковый номер	Порядковый номер вопроса	Числовой	>0	2

– сущность «Ответ», в которой будет содержаться информация об ответах, которые можно дать на вопрос задания. Атрибуты сущности можно увидеть в таблице 6;

Таблица 6 – Атрибуты сущности «Ответ»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Код Ответа</u>	Код ответа.	Числовой	>0	2
Тело ответа	Текст, который является вариантом ответа.	Текст	-	Крокодил
Правильность	Параметр, показывающий правильный ответ или нет	Бинарный	0 или 1	1

– сущность «Результат», в которой будет содержаться информация результатах пройденных заданий. Атрибуты сущности описаны в таблице 7.

Таблица 7 – Атрибуты сущности «Результат»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Код результата</u>	Код результата.	Числовой	>0	111
Дата прохождения	Дата, когда было пройдено задание.	Дата	-	02-03-2024
Длительность выполнения	Время прохождения задания.	Числовой	>0	1200
Процент	Процент правильного решения задания.	Числовой	От 0 до 100	91

– сущность «Пользователь», в которой будет содержаться информация о пользователях, зарегистрировавшихся на сайте. Атрибуты сущности описаны в таблице 8;

Таблица 8 – Атрибуты сущности «Пользователь»

Название	Описание	Тип данных	Диапазон значений	Пример
<u>Код Пользователя</u>	Код пользователя.	Числовой	>0	5
Логин	Имя пользователя, которое будет использоваться для авторизации на сайте	Текст	-	Кроссворд
Пароль	Пароль, который будет использоваться для авторизации на сайте	Текст	-	-
Эл. почта	Электронная почта пользователя.	Текст	-	mybest-mail@mail.ru

Подтверждение эл. почты	Статус подтверждения электронной почты.	Бинарный	0 или 1	1
Телефон	Телефон пользователя.	Текст	-	+79123456789

2.4.1.2 Связи между сущностями

Следующим этапом при построении инфологической модели является определение связей. Важной характеристикой является степень этой связи. Степень связи характеризует число экземпляров сущности, которое может участвовать с каждой стороны связи. Для бинарных связей по степени связи с каждой ее стороны различают связи типов:

- связь 1:1 (один-к-одному);
- связь 1:m (один-ко-многим);
- связь m:n (многие-ко-многим).

На основе описания предметной области и сущностей были разработаны связи между сущностями.

Связь между сущностями «Шаблон» и «Задание» имеет вид (рисунок 7) один-ко-многим, так как по шаблону может создавать многое количество задания, однако задание может создаваться только по одному шаблону.

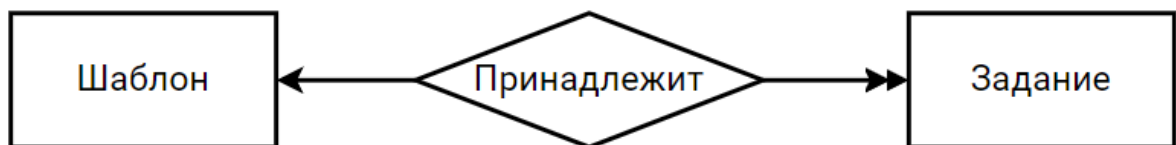


Рисунок 7 – Связь «Шаблон – Задание»

Связь «Задание – Вопрос» (рисунок 8) имеет вид один-ко-многим, так как в задании может содержаться несколько вопросов, но вопрос может принадлежать только одному заданию.

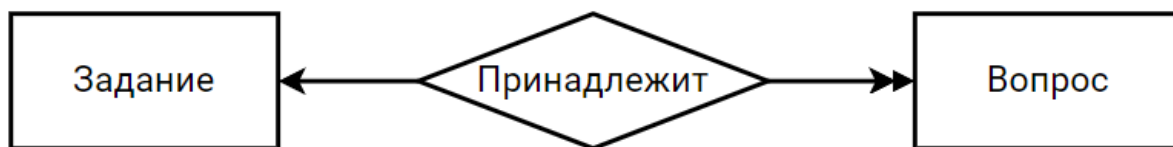


Рисунок 8 – Связь «Задание – Вопрос»

Связь между сущностями «Вопрос» и «Ответ» имеет вид (рисунок 9) один-ко-многим, поскольку на вопрос может быть несколько вариантов ответа, но вариант ответа может принадлежать только к одному вопросу.

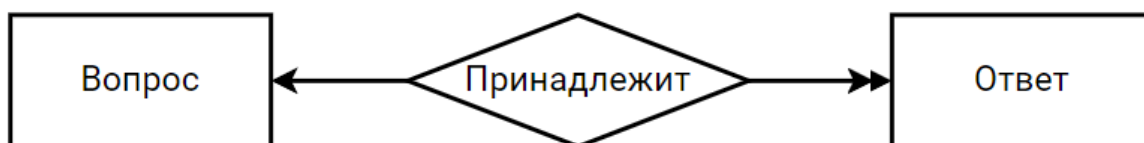


Рисунок 9 – Связь «Вопрос – Ответ»

Связь «Задание – Пользователь» (рисунок 10) имеет вид один-ко-многим, так как задание может создать только один пользователь, но пользователь может создавать множество заданий.

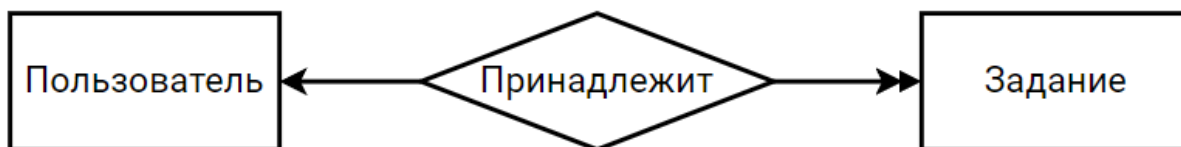


Рисунок 10 – Связь «Пользователь – Задание»

Связь «Пользователь – Результаты» (рисунок 11) имеет вид один-ко-многим, так как пользователю может принадлежать несколько результатов, но результат может принадлежать только одному пользователю.

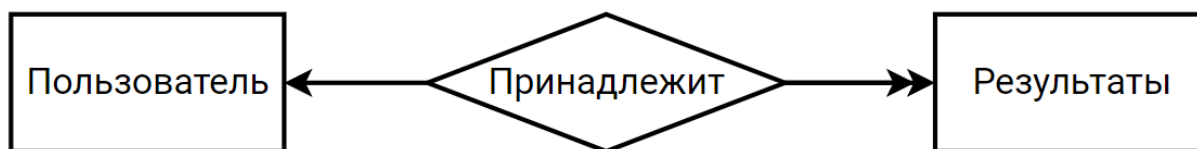


Рисунок 11 – связь «Пользователь – Результаты»

Связь «Задание – Результаты» (рисунок 12) имеет вид один-ко-многим, так как по заданию может быть много результатов, но конкретный результат может быть только по одному заданию.

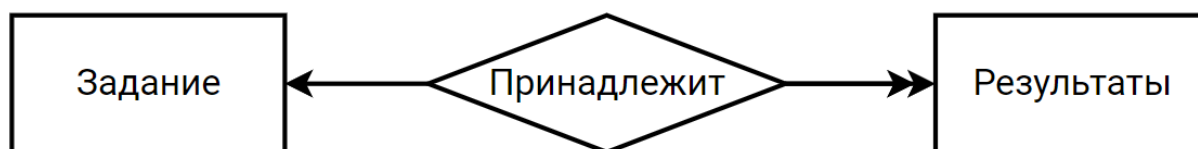


Рисунок 12 – связь «Задание – Результаты»

Таким образом, каждая сущность связана тем или иным образом с другой сущностью, образуя единую цепочку сущностей. Ни одна из сущностей не осталась без связи с другой сущностью.

2.4.1.3 Разработка инфологической модели

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Наибольшую популярность из-за доступности, наглядности и компактности приобрел подход моделирования «сущность-связь», разработанный Ченом в 1976 году. Инфологическую модель базы данных можно увидеть на рисунке 13, на котором можно увидеть все сущности, их атрибуты и связи между ними. Красным цветом выделены сущности и связи между ними, играющие важную роль для данной работы.

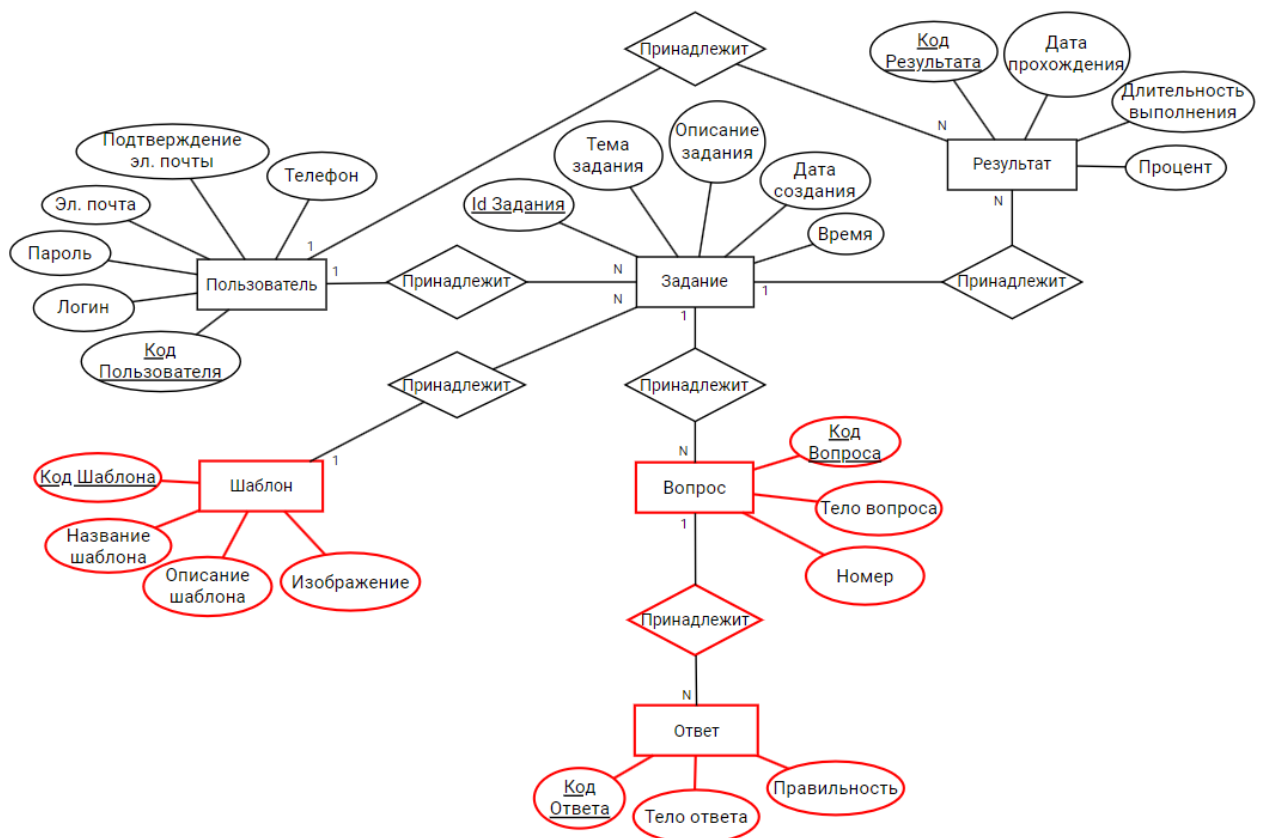


Рисунок 13 – Инфологическая модель в нотации Чена

В результате была спроектирована инфологическая модель базы данных веб-сервиса в нотации Чена, созданы сущности, их атрибуты и связи между ними.

2.4.2 Логическое проектирование

Логическое проектирование – это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

2.4.2.1 Отображение концептуальной модели на реляционную

Цель данного этапа состоит в построении реляционной логической модели. Реляционная логическая модель представляет собой совокупность нормализованных отношений, в которых реализованы связи между объектами предметной области и выполнены все преобразования, необходимые для ее эффективной реализации в среде конкретной СУБД.

Рассмотрим двунаправленную связь «Шаблон – Задание», изображенную на рисунке 14.



Рисунок 14 – Связь «Шаблон – Задание»

Связь «Шаблон – Задание» имеет тип «один-ко-многим». Исходной будет сущность «Задание», так как из нее исходит простая связь, сущность «Шаблон» является порожденной, т.к. простая связь направлена к ней, поэтому, ключ «Код Шаблона» порожденной сущности добавим в исходную сущность, что показано на рисунке 15.



Рисунок 15 – Результат анализа связи «Шаблон – Задание»

Рассмотрим двунаправленную связь «Задание – Вопрос», изображенную на рисунке 16.



Рисунок 16 – Связь «Задание – Вопрос»

Связь «Задание – Вопрос» имеет тип «один-ко-многим». Поэтому исходной будет сущность «Вопрос», так как из нее исходит простая связь, сущность «Задание» является порожденной, т.к. простая связь направлена к ней, поэтому, ключ «Код Задания» порожденной сущности добавим в исходную сущность, что показано на рисунке 17.



Рисунок 17 – Результат анализа связи «Задание – Вопрос»

Рассмотрим двунаправленную связь «Вопрос – Ответ», изображенную на рисунке 18.

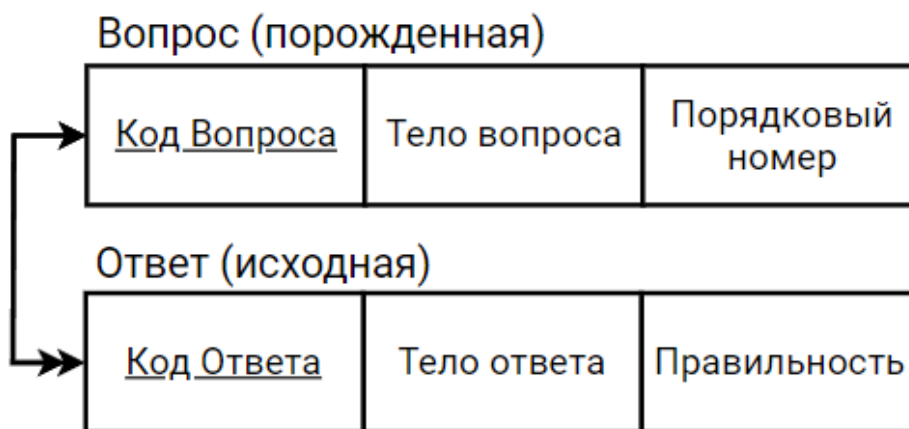


Рисунок 18 – Связь «Вопрос – Ответ»

Связь «Вопрос – Ответ» имеет тип «один-ко-многим». Поэтому исходной будет сущность «Ответ», так как из нее исходит простая связь, сущность «Вопрос» является порожденной, т.к. простая связь направлена к ней, поэтому, ключ «Код Вопросы» порожденной сущности добавим в исходную сущность, что показано на рисунке 19.



Рисунок 19 – Результат анализа связи «Вопрос – Ответ»

Далее для каждой сущности модели создаётся таблица. Имя сущности – имя таблицы. Осуществляется формирование структуры таблицы, устанавливаются связи между таблицами посредством механизма первичных и внешних ключей.

После разбора всех связей, были построены отношения, которые можно наблюдать на рисунке 20.



Рисунок 20 – Результирующие отношения

Таким образом были построены результирующие отношения по их связям.

2.4.2.2 Нормализация отношений

Важной составляющей реляционной модели является теория нормальных форм. Нормальные формы представляют собой определенные состояния

отношений, характеризующие их избыточность. Существуют нормальные формы от первой до шестой включительно, с некоторыми промежуточными состояниями, не имеющими номера. Каждая последующая нормальная форма дополняет требования к предыдущей.

Процесс приведения реляционной базы данных в соответствие с нормальными формами называется нормализацией. Этот процесс является рекомендуемым этапом при проектировании структур реляционных баз данных. Применение нормализации позволяет исключить избыточность, построить чёткую и логичную взаимосвязь данных, заложить фундамент их целостности. На начальном этапе достаточной глубиной нормализации считается доведение до третьей нормальной формы включительно.

Первая нормальная форма требует, чтобы каждый атрибут каждого кортежа содержал единственное значение.

Вторая нормальная форма требует неприводимой зависимости каждого не ключевого атрибута от потенциального ключа отношения.

Третья нормальная форма требует отсутствия транзитивной зависимости не ключевых атрибутов от потенциального ключа.

Все полученные отношения соответствуют первой нормальной форме, поскольку значения всех атрибутов являются неделимыми или атомарными. Также все отношения являются отношениями во второй нормальной форме, так как они соответствуют первой нормальной форме, и все не ключевые атрибуты полно зависят от первичного ключа в каждом своём отношении. И в конце концов, все отношения соответствуют третьей нормальной форме, так как они находятся во второй нормальной форме, и не ключевые атрибуты не взаимозависимы.

2.4.2.3 Разработка логической модели базы данных

Таким образом, за период выполнения стадии логического проектирования была создана логическая реляционная модель базы данных, изображенная на рисунке 21.

Созданная логическая модель данных является источником информации для этапа физического проектирования и обеспечивает разработчика физической базы данных средствами поиска компромиссов, необходимых для достижения поставленных целей, что очень важно для эффективного проектирования.

Для реализации логической модели было использовано case-средство в виде программного продукта Erwin data modeler, который помогает визуализировать структуру данных.

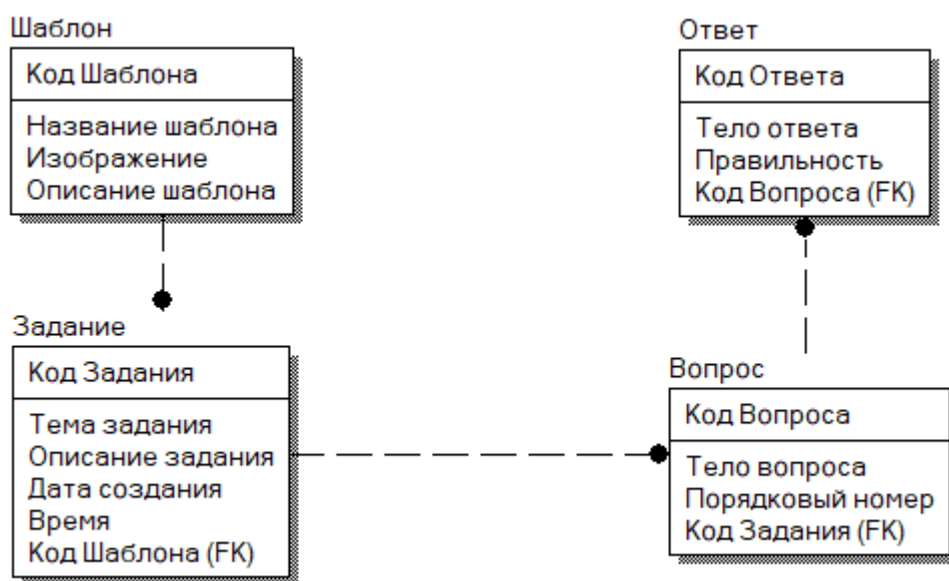


Рисунок 21 – Логическая модель базы данных

Логическая модель данных играет также важную роль на этапе эксплуатации и сопровождения уже готовой системы. При правильно организованном сопровождении поддерживаемая в актуальном состоянии модель данных позволяет точно и наглядно представить любые вносимые в базу данных изменения, а также оценить их влияние на прикладные программы и использование данных, уже имеющихся в базе.

2.4.3 Физическое проектирование

Цель данного этапа – создание базовой функциональной схемы реляционной базы данных на основе глобальной логической модели данных, которая может быть реализована в целевой СУБД.

В первую очередь были спроектированы таблицы базы данных средствами выбранной СУБД. Для создания базы данных была использована программа SQL Server Management Studio. В данной СУБД была реализована полная база данных, которая создавалась на протяжении всех прошедших стадий.

Физическое определение атрибутов сущностей приведено в таблицах 9 – 12.

Таблица 9 – Физическая структура данных отношения «Шаблон»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
<u>Id Шаблона</u>	Числовой	>0	Integer	Primary key
Название шаблона	Текстовый	Not Null	Nvarchar	-
Изображение	Битовый	-	Byte	-
Описание шаблона	Текстовый	Not Null	Nvarchar	-

Таблица 10 – Физическая структура данных отношения «Задание»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
<u>Id Задания</u>	Числовой	>0	Integer	Primary key
Название задания	Текстовый	Not Null	Nvarchar	-
Описание задания	Текстовый	-	Nvarchar	-
Дата создания	Текстовый	Not Null	Nvarchar	-
Время	Дата	-	Date	-
Id Шаблона	Числовой	>0	Integer	Foreign key
Id Пользователя	Числовой	>0	Integer	Foreign key

Таблица 11 – Физическая структура данных отношения «Вопрос»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
<u>Id Вопроса</u>	Числовой	>0	Integer	Primary key
Тело вопроса	Текстовый	Not Null	Nvarchar	-
Порядковый номер	Числовой	Not Null, >0	Integer	-
Id Задания	Числовой	>0	Integer	Foreign key

Таблица 12 – Физическая структура данных отношения «Ответ»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
<u>Id Ответа</u>	Числовой	>0	Integer	Primary key
Тело ответа	Текстовый	Not Null	Nvarchar	-
Правильность	Двоичный	Not Null, 0 или 1	Binary	-
Id Вопроса	Числовой	>0	Integer	Foreign key

После построения таблиц была построена физическая модель базы данных, которую можно увидеть на рисунке 22. Отличие физической модели от логической в том, что физическая отображает типы данных.

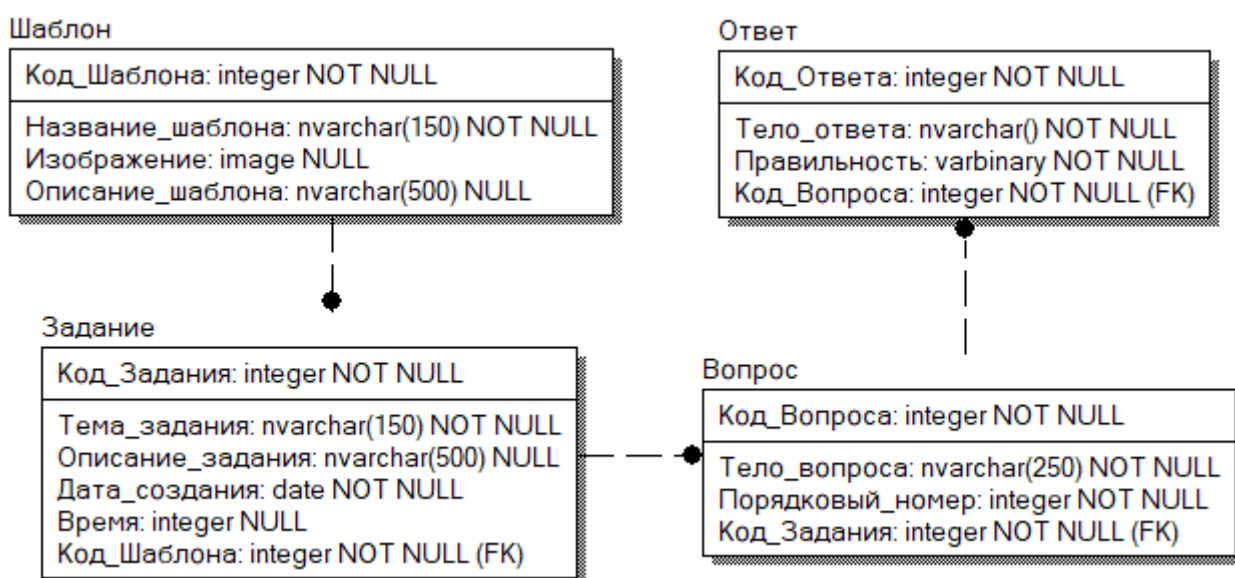


Рисунок 22 – Физическая модель базы данных

Созданная физическая модель позволяет в полной мере создать спроектированную базу данных, используя целевую СУБД.

3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

3.1 Этапы разработки шаблонов

Основных этапов разработки шаблонов 6:

- определение функционала шаблона;
- вёрстка веб-страницы шаблона;
- функциональное наполнение страницы шаблона;
- создание страницы для входных данных;
- тестирование шаблона на различных входных данных;
- интеграция шаблона на web-сервис.

Данные этапы и их порядок следования в виде схемы изображены на рисунке 23.



Рисунок 23 – Этапы разработки шаблона

На первом этапе прорабатывалась концептуальная модель как оно должно выглядеть, какие функции должно выполнять, как пользователь будет взаимодействовать с заданием.

На этапе вёрстки должна быть создана HTML страница шаблона со всеми используемыми элементами на экране.

На третьем этапе производится написание кода, обеспечивающего выполнение поставленных функций.

Затем необходимо создать страницу для ввода входных данных на веб-сервисе, таких как вопросы, ответы и название темы задания.

Затем производится тестирование шаблона на различных входных данных и, если проведенное тестирование доказало работоспособность всех функций, заложенных в шаблон, производится интеграция шаблона на веб-сервис.

3.2 Реализация шаблона интерактивного задания «Кроссворд»

Функции, выполнение которых должен обеспечивать программный продукт описаны в пункте 4.1.1 технического задания в приложении А.

На этапе вёрстки создавался шаблон рабочего задания. При помощи HTML и CSS создавалась обёртка, элементы на странице, которые будут использоваться при генерации задания с пользовательскими данными.

На рисунке 24 изображен внешний вид шаблона без функциональных возможностей.



Рисунок 24 – Финальный внешний вид (вёрстка) шаблона «Кроссворд»

Были добавлены поля – пустые ячейки как область, где будет генерироваться кроссворд, после его генерации они должны исчезать. Область для отображения вопросов с полями для слов по горизонтали, по вертикали и для

названия темы. Область для таймера и область для основных клавиш как пробел и `backspace` для подтверждения и отмены выбора введенного слова. Добавлен задний фон для потенциальной возможности его замены. Также были прописаны при помощи языка стилей некоторые анимации как появление букв на полях.

При помощи JavaScript прописываются переменные, функции, обеспечивающие функционал шаблона. В данном шаблоне используется 3 файла со скриптами JavaScript:

- `globalCross`. В данном файле хранятся общие переменные для всего проекта, которые хранят в себе слова, используемые для генерации кроссворда, переменные отвечающие за начало игры, доступность ввода с клавиатуры, время игры и тому подобное;

- `scriptCross`, в котором содержатся основные функции, генерирующие кроссворд, размещающие слова на сетке и другие;

- `eventListenersCross`, в котором содержатся функции, отвечающие за взаимодействие пользователя с интерфейсом, такие как старт игры, печать слов, ответные реакции на действия пользователя.

Общий алгоритм работы шаблона изображен на рисунке 25.

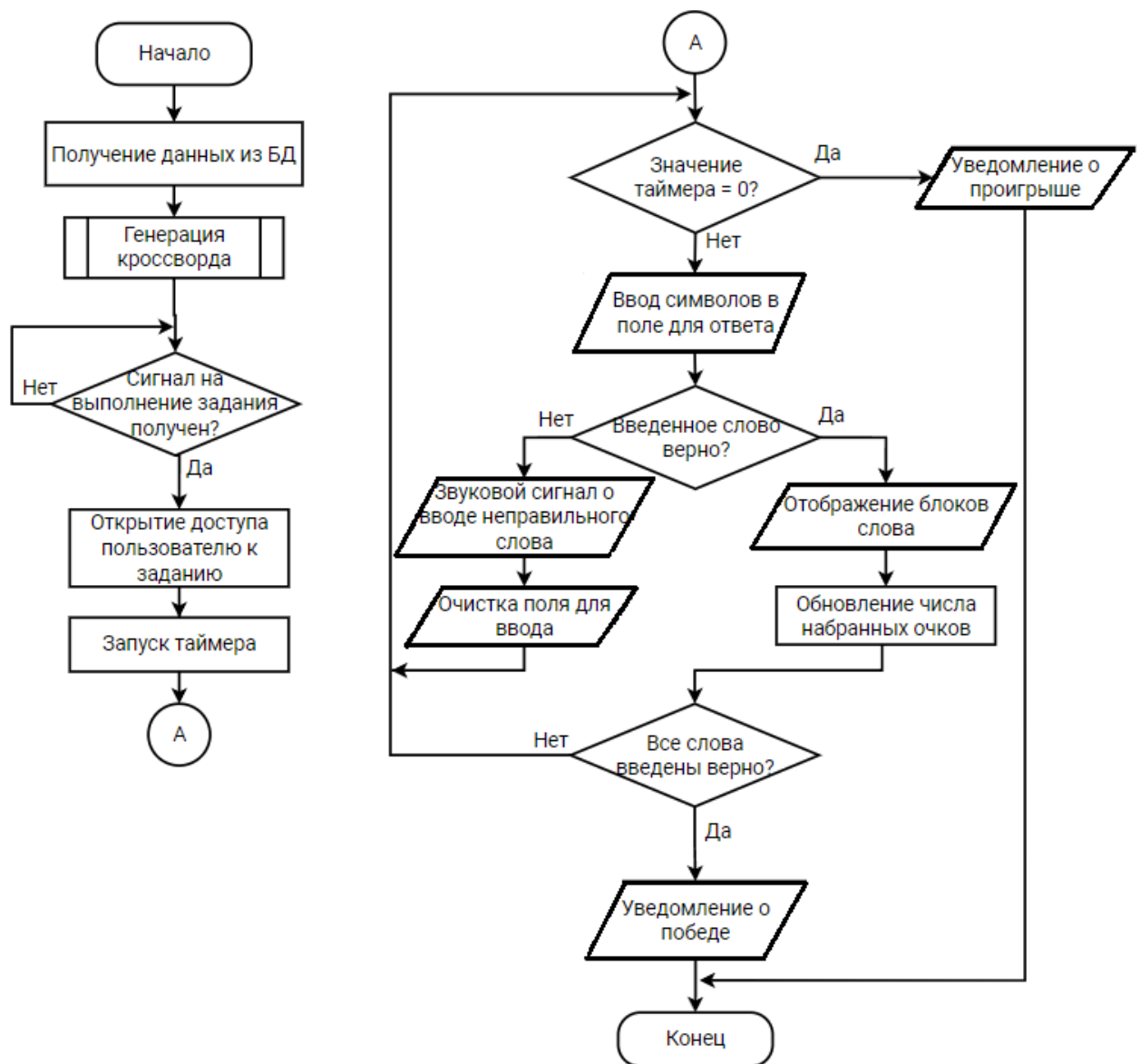


Рисунок 25 – Общий алгоритм работы шаблона «Кроссворд»

Основную часть программы составляет процесс «Генерация кроссворда». Поэтому необходимо рассмотреть работу данного этапа программы.

В первую очередь в файле `globalCross.js` задается словарь, в котором должны храниться вопросы и ответы на них. Создаются переменные для работы с HTML тегами.

Следующим шагом создавались рабочие функции программы, содержащиеся в файле `scriptCross.js`. Список всех функций и их краткое описание можно увидеть в таблице 13.

Таблица 13 – Основные функции шаблона «Кроссворд» в файле scriptCross.js

Название функции	Краткое описание
coordsToCellNo	Принимает значения координат ячейки и возвращает на основании этих координат её индекс.
cellNoToX	Принимает в качестве параметра индекс ячейки и возвращает её координаты по оси X.
cellNoToY	Принимает в качестве параметра индекс ячейки и возвращает её координаты по оси Y.
marginLeft	Принимает в качестве параметра HTML контейнер и возвращает его параметр стиля margin-left.
marginTop	Принимает в качестве параметра HTML контейнер и возвращает его параметр стиля margin-top.
invertDirection	Предназначена для изменения параметра направления слова, для его размещения на поля.
placeResult	Создает и размещает HTML теги букв слова кроссворда.
placeFirstResult	Выбирает место для размещения первого слова и размещает его.
placeResults	Создает размещения для всех слов на основе первого, также проверяет правильность этих размещений на основе списка из функции getGridWords, содержит в себе вызов всех остальных функций.
getGridWords	Перебирает ячейки кроссвордной сетки по вертикали и горизонтали, тем самым получает список размещенных слов по горизонтали и вертикали.
arrangeBlocks	Функция, обеспечивающая модификацию массива расположенных слов, для отцентровки созданного кроссворда.
shuffleDict	Перемешивание случайным порядком массива слов из словаря.

В первую очередь задаются вспомогательные функции:

– «coordsToCellNo» – функция, принимающая значения координат ячейки и возвращающая на основании этих координат её индекс;

– «cellNoToX», «cellNoToY» – функции, принимающие в качестве параметра индекс ячейки и возвращающие её координаты по оси X и Y соответственно;

– «marginLeft», «marginTop» – функции, принимающие в качестве параметра HTML контейнер и возвращающие его параметр стиля margin-left и margin-top соответственно;

– «invertDirection» – функция, предназначенная для изменения параметра направления слова, для его размещения на поля.

Данные функции будут использоваться в других функциях для определения заложенных в них характеристик.

Для размещения блоков с буквами в ячейки на поле реализована функция «placeResult». Она получает в качестве параметров слово для размещения, направление размещения и координаты ячейки откуда начинается размещение. Алгоритм функции изображен на рисунке 26.

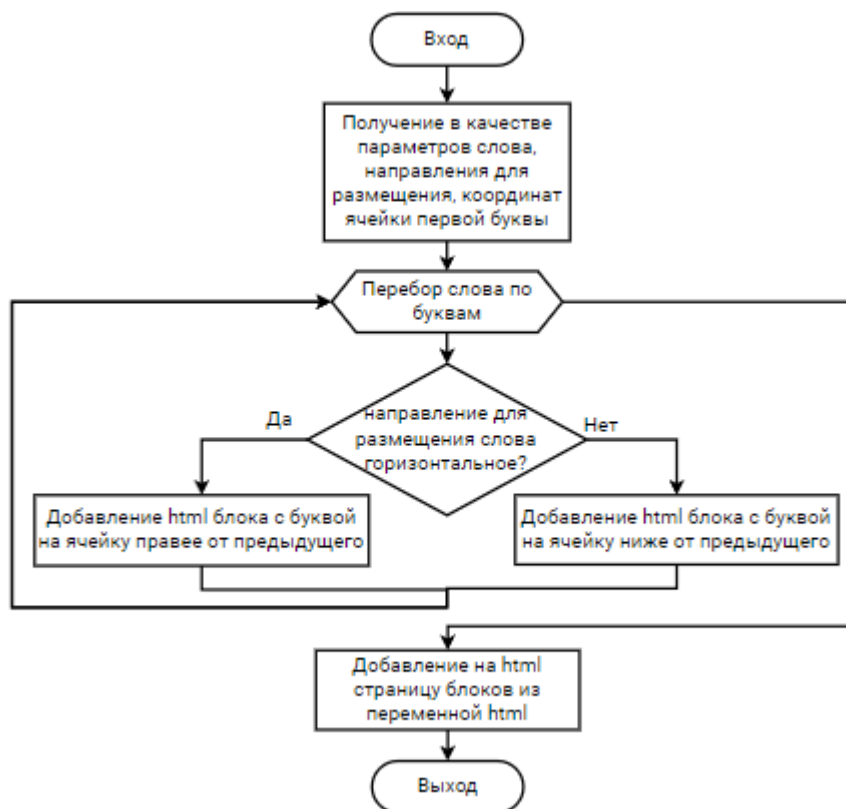
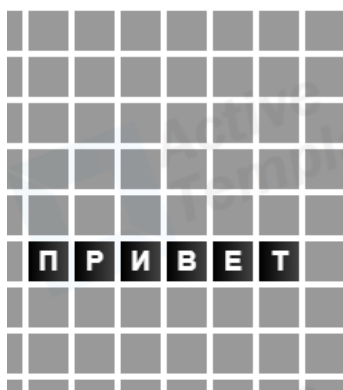


Рисунок 26 – Алгоритм функции «placeResult»

Результатом выполнения функции является созданные и расположенные блоки с буквами слова, которое функция принимает в качестве параметра. Работа функции была проверена с использованием различных входных данных, на рисунке 27 изображен пример с использованием слова «привет», расположение по горизонтали и координатами начальной ячейки 390 по оси X и 150 по оси Y.



```
Элементы Консоль Источники Сеть
top Фильтр
> placeResult("привет", "horizontal", 390, 150)
< (6) [113, 114, 115, 116, 117, 118]
  0: 113
  1: 114
  2: 115
  3: 116
  4: 117
  5: 118
  length: 6
  [[Prototype]]: Array(0)
```

Рисунок 27 – Проверка работоспособности функции «placeResult»

Функция «placeFirstResult» необходима для размещения первого слова, потому что данный процесс немного отличается от размещения остальных слов. Первое слово необходимо разместить приблизительно в центре поля, чтобы последующие слова могли располагаться как выше, так и ниже по отношению к нему. Для этого необходимо получить значения по осям X и Y откуда будет начинаться слово, а также выбрать направление размещения. Направление выбирается случайным образом либо горизонтальное, либо вертикальное. Так как количество строк и столбцов ячеек одинаковое, то для центрального расположения слова достаточно поделить с округлением количество строк на два с половиной и помножить на размер ячеек таким образом будет получено примерно центральное расположение первого слова.

Результатом выполнения функции является добавление в массив верно расположенных слов «data» одного слова в середине поля.

Функция «placeResults» отвечает за заполнение массива верно расположенных слов «data».

Поскольку первое слово уже размещено, то осталось разместить оставшиеся. Для этого был создан массив для еще не размещенных слов «remaining». Затем создается массив «placements», который будет заполняться размещениями слов, как правильными, так и неправильными. Данные размещения создаются путем перебора массивов «remaining» и «data». Из первого слова массива неразмещенных слов «remaining» берется буква и её порядковый номер и сравнивается с буквами уже размещенных слов. При совпадении

букв слов из двух данных массивов неразмещенное слово сохраняется в массиве размещений с обратным направлением расположения, по сравнению с тем словом, с которым совпали их буквы. Алгоритм данного участка кода можно увидеть на рисунке 28.

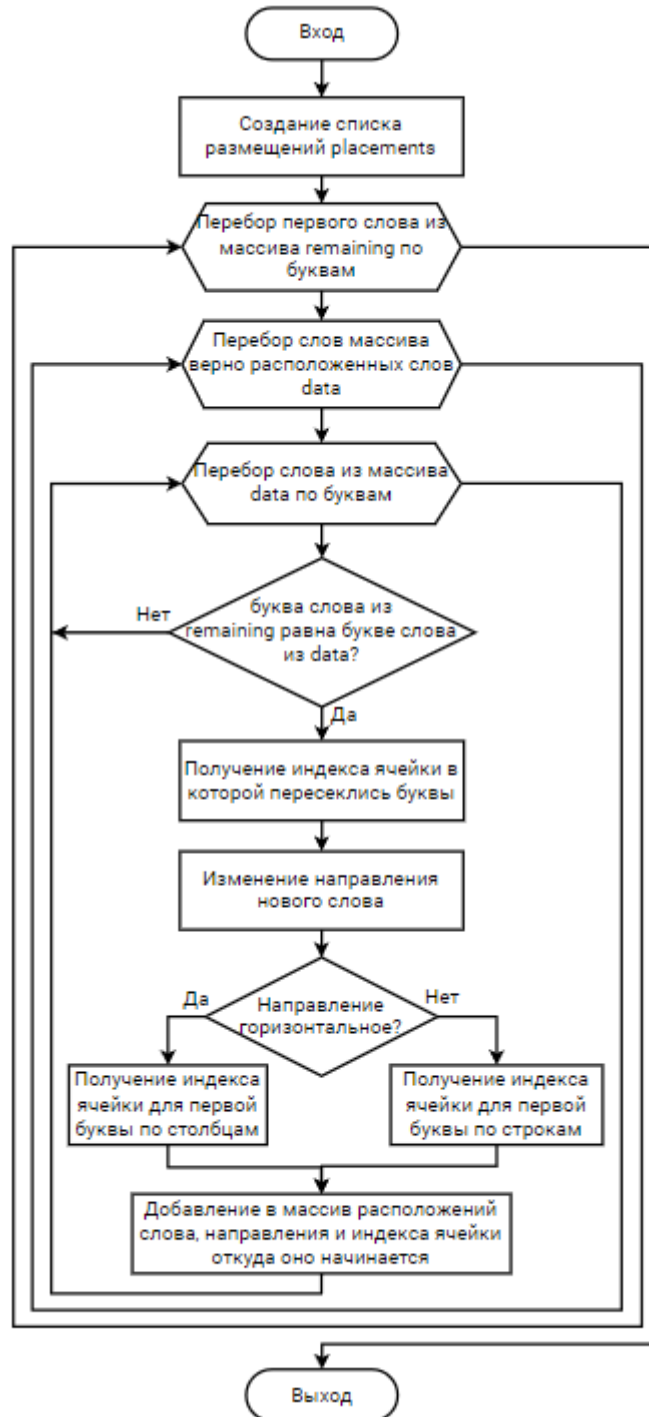


Рисунок 28 – Алгоритм добавления размещений слова

Например, горизонтально расположенное слово «привет» из массива «data» и слово «пока» из массива «remaining» совпадают по букве П. Это означает, что в массив размещений «placements» попадет слово пока с совпадением по индексу буквы П и вертикальным расположением. Пример как это работает можно увидеть на рисунке 29. Цифрами обозначены индексы ячеек.



Рисунок 29 – Пример заполнения массива «placements»

Но поскольку массив «placements» содержит информацию о всех расположениях, то необходимо выделить правильное. Рассмотрим следующую ситуацию, изображенную на рисунке 30.

В данном случае слово «повар» получило размещение через вторую букву о в слове «водяной», и его буква П вышла за границы сетки ячеек, получив из-за этого отрицательное значение своего индекса, в данном случае значение минус четырнадцать. И при получении значения координаты для размещения оно получается с отрицательным знаком. Из-за чего слово при размещении выходит за границы сетки ячеек и данное расположение будет являться всегда неправильным. Поэтому было введено условие, что данное расположение является неправильным при отрицательных значениях, а также при значениях, превышающих максимальный размер сетки, который равняется значению количества столбцов или строк умноженному на размер ячейки.

П
О
В
А
Р

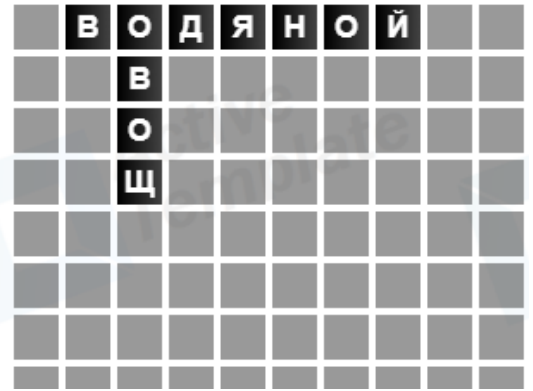


Рисунок 30 – Неправильный вывод слова при отрицательных значениях

Алгоритм данной проверки можно увидеть на рисунке 31.

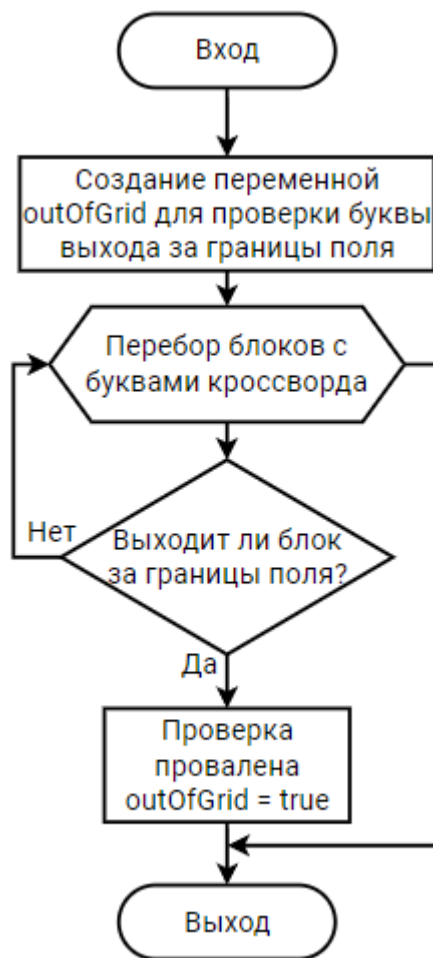


Рисунок 31 – Алгоритм проверки выхода за границы поля

Следующая недопустимая ситуация при размещении изображена на рисунке 32.



Рисунок 32 – Пример неправильного размещения

В данном случае неверным является размещение слов «овощ» и «дом» в соседних ячейках, поскольку они образуют горизонтальные сочетания букв, которые не являются словами, загаданными пользователем. Также возможна ситуация, когда такие сочетания образуются и горизонтально расположенными словами.

Для решения данной проблемы создана функция «getGridWords», которая перебирает ячейки отдельно по строкам и столбцам и, если на в следующих друг за другом ячейках есть буквы они записываются, образуя слово, которое записывается в массив «gridWords». Алгоритм данной функции для проверки по строкам можно увидеть на рисунке 33. Аналогичный алгоритм используется для получения слов по столбцам.

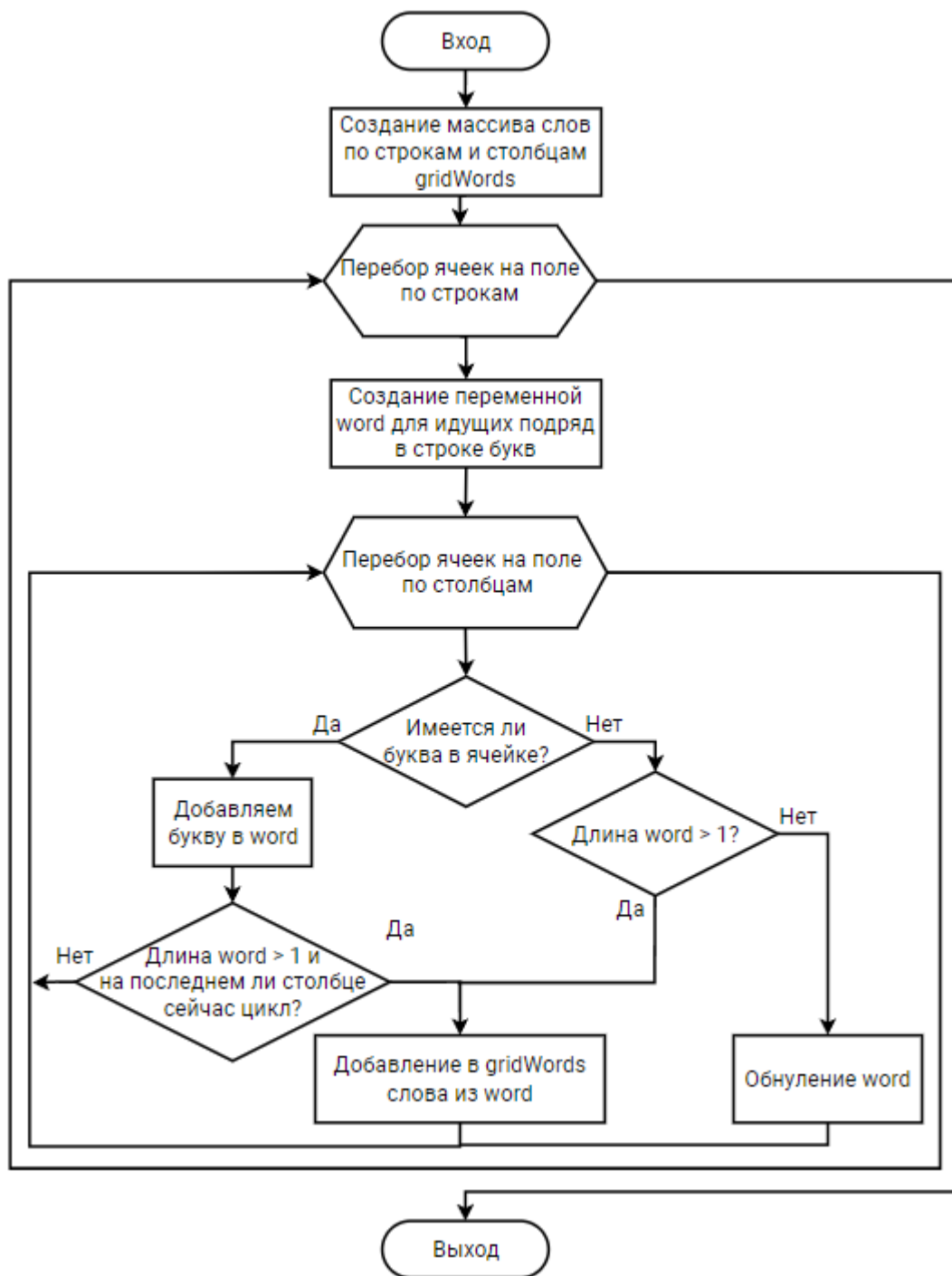


Рисунок 33 – Алгоритм функции «getGridWords» для проверки строк

Далее выполняется проверка, слова из массива «gridWords» сравниваются со словами из массива слов, которые необходимо разместить. Если при проверке обнаружено несоответствие слов в массивах, то данная проверка провалена. Алгоритм данной проверки изображен на рисунке 34.

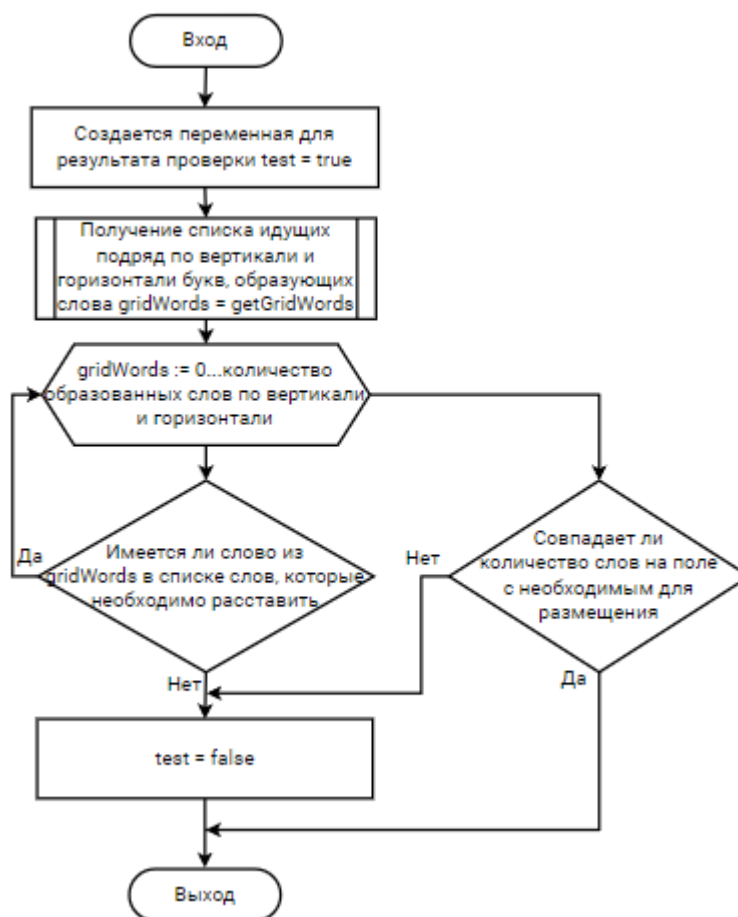


Рисунок 34 – Алгоритм проверки на правильность расстановки слов

Если обе проверки выход за границы поля и отсутствие плотно прилегающих слов успешно пройдены, то данное расположение добавляется в массив верно расположенных слов «data», а из массива еще не размещенных слов «remaining» удаляется.

Если же одна из проверок не была пройдена, то данное слово отправляется в конец массива «remaining», чтобы провести повторную попытку разместиться после добавления на поле других слов.

Весь данный процесс помещается в цикл, в котором количество итераций равно удвоенному числу слов, которые необходимо разместить.

После размещения всех слов необходимо отцентрировать полученный кроссворд. Данную работу выполняет функция «arrangeBlocks», она получает координаты крайних по вертикали и горизонтали блоков HTML тегов, содержащих буквы выставленных слов, затем высчитывает количество пустых

ячеек от крайних блоков до краев поля. И изменяет значения координат блоков с буквами в центр относительно количества пустых ячеек сверху, снизу, справа и слева. В конце производится добавление цифр на ячейки, где располагается первая буква размещенного слова и добавление подсказки в список вопросов. А также скрытие ячеек, где не располагаются буквы. Алгоритм данного участка программы изображен на рисунке 35.

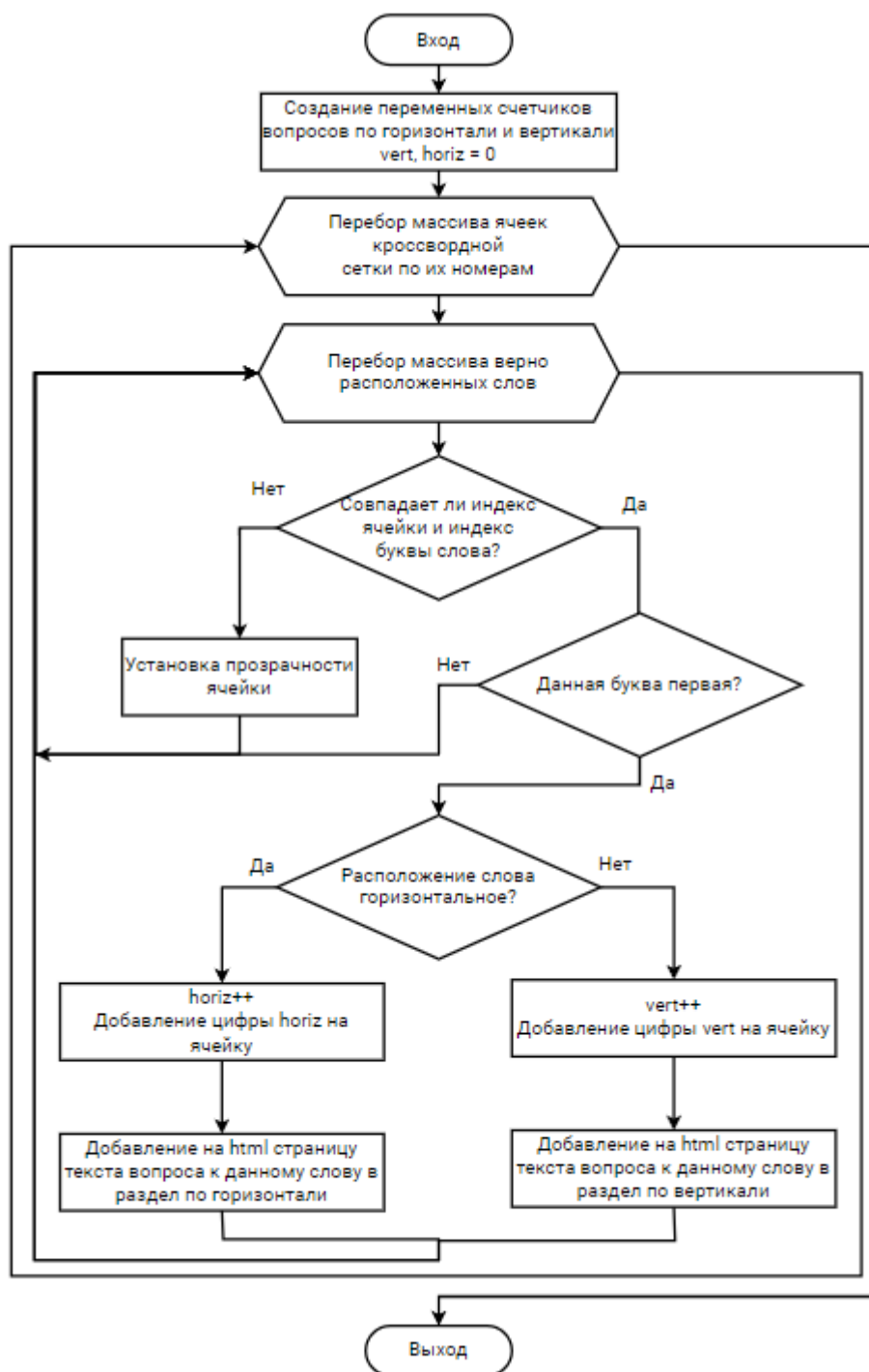


Рисунок 35 – Алгоритм размещения цифр на ячейки и вопросов.

Последний файл «eventListenersCross» содержит функции, реагирующие на действия пользователей, а также запуск генерации кроссворда. Общий алгоритм действий из данного файла можно увидеть после процесса «Генерация кроссворда» на рисунке 25.

При открытии веб-страницы запускается функция «placeResults» в цикле пока количество верно размещенных слов не будет равно количеству слов, введенных пользователем, а также таймер, отсчитывающий 120 секунд, по истечению которого будет считаться, что кроссворд не может сгенерироваться. Поскольку в данном случае может происходить многократный вызов функции, то в начало функции была помещена очистка массива правильно расположенных слов «data» и случайная сортировка словаря слов, введенных пользователем. Это необходимо для того, чтобы порядок слов при генерации различался при повторном вызове функции.

В данном файле реализованы функции старта игры по клику, реакция на ввод клавиш пользователем, например при нажатии на клавишу «space» программа отправляет проверку введенного пользователем слова, и если оно совпадает со словом загаданным создателем кроссворда, то оно появляется на экране, в случае, когда слово не совпадает, воспроизводится звук неправильного ввода и очищение поля для ввода.

Также реализована функция конца игры по достижению конца таймера ввод запрещается, все скрытые буквы отображаются.

Таким образом проводилось создание всех функций шаблона и их тестирование. После создания функциональности шаблона создается форма для создания задания. Она представляет собой страницу для веб-сайта с полями для ввода данных, которые в последствии используются для генерации задания. Её внешний вид можно увидеть на рисунке 36. Так как предполагается, что количество слов, которые будет задавать пользователь больше одного, то был добавлена кнопка для добавления дополнительных полей для ввода слов и подсказок. А также кнопки в виде корзины для удаления этих полей.

Кроссворд

Название темы:

Природа

Подсказка:

Первый цвет радуги

Ответ:

красный

Подсказка:

Животное с длинной шеей

Ответ:

жираф

Подсказка:

Ночью воет в лесу

Ответ:

волк

Добавить элемент

Добавить слово

Загрузите задний фон (необязательно):

Выберите файл

Файл не выбран

Предпросмотр

Создать задание

Рисунок 36 – Форма для создания задания по шаблону «Кроссворд»

В результате проект содержит в себе файлы, описанные в таблице 14.

Таблица 14 – Файлы, относящиеся к шаблону «Кроссворд»

Название файла	Формат файла	Краткое описание
crossword	.html	HTML страница шаблона, с которой взаимодействует пользователь.
crosswordStyles	.css	Файл стилей для HTML страницы шаблона.
globalCross	.js	Файл скриптов, содержащий глобальные переменные.
scriptCross	.js	Файл скриптов с основным функционалом.
eventListenersCross	.js	Файл скриптов с функциями, реагирующими на действие пользователя.
logo	.png	Изображение стандартного заднего фона.
space key	.jpg	Изображение кнопки пробел.
backspace key	.jpg	Изображение кнопки backspace.
correct	.mp3	Звук, издаваемый при вводе правильного слова.
wrong	.mp3	Звук, издаваемый при вводе неправильного слова.
keyPress	.mp3	Звук, издаваемый при вводе текста.
createFormCross	.html	HTML страница для создания задания кроссворда.
crFormCrossStyles	.css	Файл стилей для HTML страницы для создания задания.

В конце проводится тестирование шаблона на случайных наборах данных, которые вписываются в словарь, который будет заполняться данными, которые будет вводить пользователь после интеграции шаблона на веб-сервис. На рисунке 37 изображен рабочий вариант шаблона с ответами и вопросами на русском языке и фоном с символом темы.

Тема: C++

По горизонтали:

- 1: Явление, при котором функция вызывает саму себя
- 2: Именованная последовательность описаний и операторов, выполняющая какое-либо законченное действие
- 3: Заголовок функции, оканчивающийся точкой с запятой

По вертикали:

- 1: Переменная, объявленная в теле функции
- 2: Параметр, указанный в вызове функции
- 3: Переменная, стоящая в заголовке в круглых скобках после имени функции
- 4: Описание функции включает в себя заголовок и ...

Рисунок 37 – Рабочий вариант шаблона «Кроссворд»

На рисунке 38 изображен рабочий вариант шаблона с ответами на английском языке и вопросами на русском языке и фоном с символом объектно-ориентированного программирования.



Тема: Объектно-ориентированное программирование

По горизонтали:

- 1: Объявление перегруженной операции
- 2: Функция с общим именем, но различным телом в иерархии классов

По вертикали:

- 1: Переопределение виртуальных функций
- 2: Элемент класса с единственным значением во всей иерархии
- 3: Элемент, доступный вне класса
- 4: Элемент, не доступный вне класса
- 5: Внутренний указатель класса

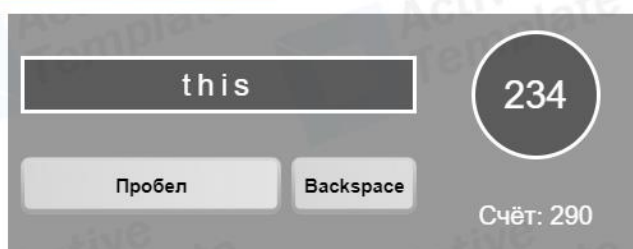


Рисунок 38 – Рабочий вариант шаблона «Кроссворд»

Данный шаблон был протестирован на правильность отображения и работы контента на пяти популярных браузерах: Google Chrome, Яндекс браузер, Microsoft Edge, Mozilla Firefox и Opera.

Таким образом были созданы и протестированы шаблон интерактивного задания «Кроссворд» и форма для создания заданий по данному шаблону.

3.3 Реализация шаблона интерактивного задания «Угадай слово»

Функции, выполнение которых должен обеспечивать данный шаблон описаны в пункте 4.1.1 технического задания в приложении Б.

На этапе вёрстки при помощи HTML и CSS был создан внешний вид шаблона без функций. На рисунке 39 можно увидеть данный внешний вид шаблона.

Тема: Название темы

Вопрос: Текст вопроса

Время выполнения: 0 сек.

Неправильных ответов: 0 из 9



Слово для отгадывания:

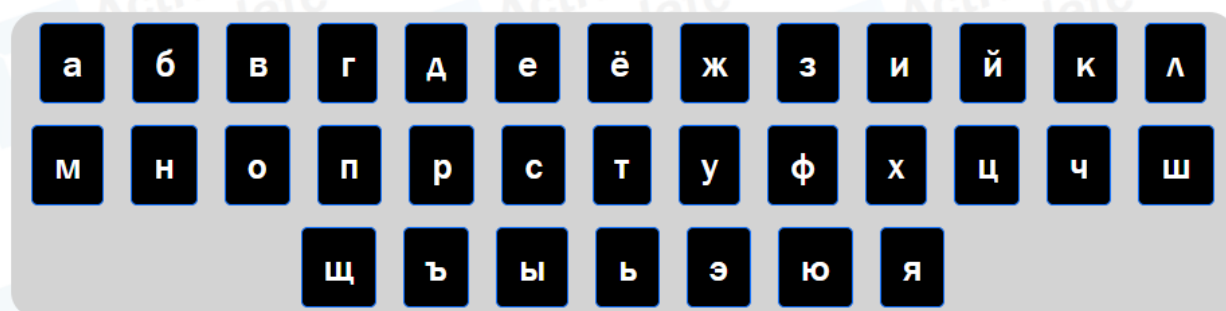


Рисунок 39 – Финальный внешний вид (вёрстка) шаблона «Угадай слово»

В верхней области экрана были размещены поля для названия темы и вопроса. По бокам, а именно с левой стороны размещен таймер выполнения задания, а с правой счётчик неправильных ответов. По середине область, в котором отображается в зависимости от количества неправильных ответов тематический рисунок, который также свидетельствует о количестве неправильных ответов. Ниже области для рисунка располагается место для загаданного слова, которое при вводе правильных букв будет заполняться этими буквами. И в самом низу находится клавиатура с буквами в алфавитном порядке, по которой можно нажимать при помощи мыши, что будет являться вариантом ответа в данном задании. Также был добавлен стандартный задний фон с водным

знаком веб-сервиса, для будущей возможности его смены по желанию пользователя.

Данный дизайн выполнен для экранов с минимальным разрешением 1024 на 768 пикселей. Он является адаптивным и при изменении размеров экрана элементы будут смещаться, оставаясь на своих местах и в своем порядке относительно изменений размеров экрана.

Добавлен задний фон для потенциальной возможности его замены. Также были прописаны при помощи языка стилей некоторые анимации как появление букв на полях.

При помощи JavaScript были прописаны переменные, функции, которые обеспечивают рабочий функционал шаблона. В данном шаблоне создано 2 файла со скриптами JavaScript:

- globalGuess. В данном файле хранятся общие переменные для всего проекта, которые в будущем будут заполняться информацией, которую ввел пользователь из формы;

- scriptGuess, в котором содержатся основные функции программы.

Общий алгоритм работы шаблона «Угадай слово» можно увидеть на рисунке 40.

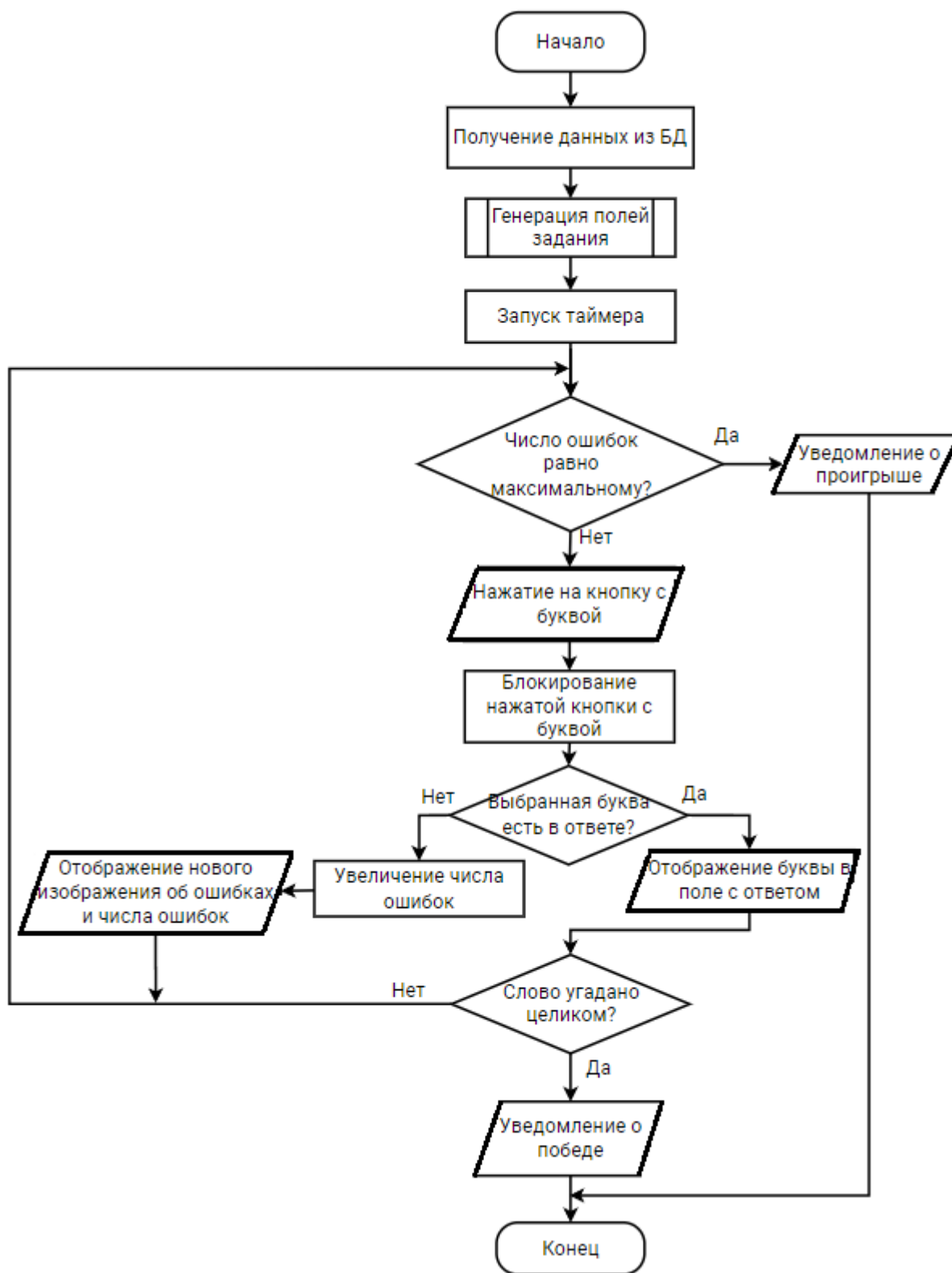


Рисунок 40 – Общий алгоритм работы шаблона «Угадай слово»

В файле `globalGuess.js` были заданы следующие основные переменные, которые описаны в таблице 15.

Таблица 15 – Основные переменные, заданные в файле `globalGuess.js`

Название переменной	Краткое описание
<code>answer</code>	Содержит в себе ответ на задание.
<code>maxWrong</code>	Максимально допустимое количество ошибок при решении задания.
<code>mistakes</code>	Текущее количество ошибок.
<code>guessed</code>	Массив, содержащий в себе буквы, которые были использованы во время решения задания.
<code>wordStatus</code>	Набор символов, который отображается на месте ответа, в соответствии с использованными буквами.
<code>rusAlphabet</code>	Массив, содержащий в себе все буквы русского алфавита.
<code>engAlphabet</code>	Массив, содержащий в себе все буквы английского алфавита.

Следующим шагом создавались рабочие функции программы, содержащиеся в файле `scriptGuess.js`. Список всех функций и их краткое описание описаны в таблице 16.

Таблица 16 – Основные функции, заданные в файле `globalGuess.js`

Название функции	Краткое описание
<code>generateButtons</code>	Создает кнопки для ввода по выбранному алфавиту.
<code>handleGuess</code>	Выполняет набор действий, при нажатии на кнопку с буквой.
<code>checkIfGameWon</code>	Проверка выиграна ли игра.
<code>checkIfGameLost</code>	Проверка проиграна ли игра.

В первую очередь было принято решение, что кнопки на экране, которые должен нажимать обучающийся пользователь должны генерироваться при открытии страницы, поскольку заранее не известно русский или английский алфавит используется для задания. За данное действие отвечает функция «`generateButtons`». Для этого проверяется первая буква из ответа с алфавитами, в каком она содержится, кнопки с буквами того алфавита будут созданы на экране. Алгоритм данной функции изображен на рисунке 41.

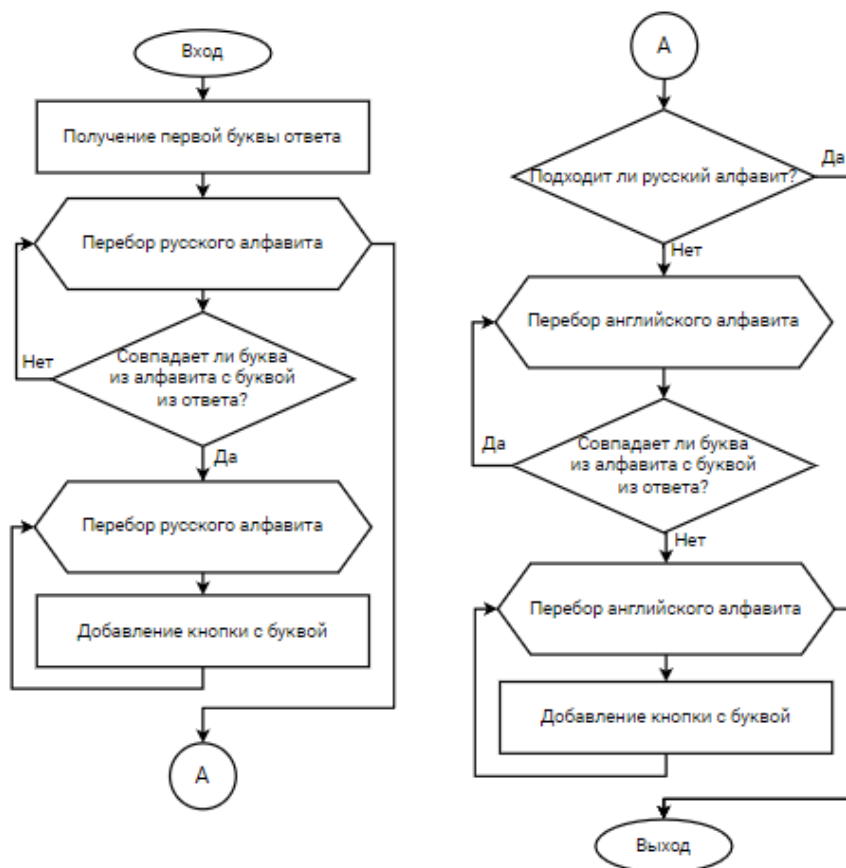


Рисунок 41 – Алгоритм функции «generateButtons»

Функция «handleGuess» обрабатывает нажатия на кнопки с буквами. При нажатии на букву, которая есть в ответе она отображается на поле для ответа, затем проверяется все буквы были добавлены или нет. При нажатии на неправильную букву, добавляется количество ошибок и проверяется проиграна ли игра или еще есть попытки для ответа и также обновляется изображение по центру. Алгоритм функции можно увидеть на рисунке с общим алгоритмом шаблона после запуска таймера.

После этапа функционального наполнения создается форма для создания задания. Её дизайн можно увидеть на рисунке 42. Предполагается, что в задании будет один вопрос и один ответ, поэтому поля для ввода расположены одно под другим и нет возможности добавить дополнительных вопросов с ответов.

Угадай слово

Название темы:

Вопрос:

Слово для отгадывания:

Загрузите задний фон (необязательно):

Рисунок 42 – Форма для создания задания по шаблону «Угадай слово»

В результате данный продукт содержит файлы, описанные в таблице 17.

Таблица 17 – Файлы, относящиеся к шаблону «Угадай слово»

Название файла	Формат файла	Краткое описание
guessWord	.html	HTML страница шаблона «Угадай слово», с которой взаимодействует пользователь.
guessStyles	.css	Файл стилей для HTML страницы шаблона.
globalGuess	.js	Файл скриптов, содержащий глобальные переменные
scriptGuess	.js	Файл скриптов с основным функционалом.
logo	.png	Изображение стандартного заднего фона.
Файлы с порядковыми именами от 0 до 9	.jpg	Изображения о соответствующем количестве ошибок.
createFormGuess	.html	HTML страница формы для создания задания для шаблона «Угадай слово».
crFormGuessStyles	.css	Файл стилей для страницы с формой для создания задания.

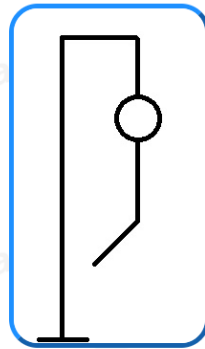
Работа шаблона была протестирована путём ввода различных комбинаций слов в поля «Название темы», «Вопрос», «Слово для отгадывания». Наблюдалось как реагирует вёрстка страницы, проверялась правильность работы функций при нажатии на кнопки с буквами для ответа. На рисунке 43 можно увидеть тестирование шаблона.

Название темы: Принципы ООП

Вопрос: Один из принципов ООП

Время выполнения - 41 сек.

Неправильных ответов: 6 из 9



Слово для отгадывания:
инкапс __ _ я _ ия



Рисунок 43 – Рабочий вариант шаблона «Угадай слово»

Шаблон был протестирован на работоспособность в пяти часто используемых браузерах: Google Chrome, Яндекс браузер, Microsoft Edge, Mozilla Firefox и Opera.

Таким образом были созданы и протестированы шаблон интерактивного задания «Угадай слово», а также форма для создания заданий по данному шаблону.

3.4 Интеграция шаблонов на web-сервис

Для интеграции шаблонов на web-сервис необходимо добавить файлы шаблонов на web-сервис. Но так как web-сервис реализован на технологии asp.net core, то необходимо изменить расширение файлов html на cshtml, это позволит использовать возможности языка C# в представлении HTML.

В данном случае C# используется для связи базы данных и web-сервиса при помощи технологии «Entity framework». На странице, где отображаются доступные для использования шаблоны, используется создание элементов на экране посредством перебора циклом всех значений из таблицы «Шаблон» базы данных. Оттуда берется информация из атрибутов и встраивается на

страницу сайта. Поэтому достаточно ввести данные о шаблонах в базу данных, и они сразу отобразятся на сайте (рисунок 44).

Выберите один из представленных шаблонов:

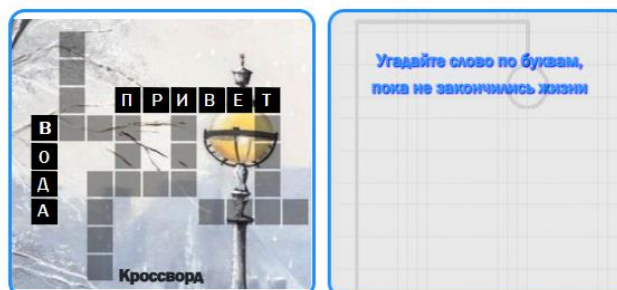


Рисунок 44 – Добавленные элементы страницы по записям из БД

По полученному коду шаблона система понимает, что ей дальше делать. По нажатию на шаблон на сайте открывается модальное окно с тем же кодом, что и у выбранного шаблона. По этому коду на модальном окне выбирается то окно для создания задания, что соответствует шаблону. Для полей для ввода данных в форме создания задания задаются параметры, которые затем будут переданы в базу данных.

В самих же шаблонах добавляются функции, ответственные за получение данных из базы данных, чтобы заполнить переменные с ответами и вопросами для генерации кроссворда и полей в угадай слово.

Для работоспособности страницы с заданием шаблонов, в контроллере, отвечающем за страницы заданий «ExerciseController» были добавлены функции, которые обеспечивают работоспособность представлений шаблонов, а также параметры из базы данных, которые передаются в представления для последующей их работы.

Таким образом были интегрированы на web-сервис шаблоны интерактивных заданий «Кроссворд» и «Угадай слово».

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1 Безопасность

В сфере работы с компьютерной техникой, знание правил техники безопасности имеет огромное значение для операторов. Эти знания помогают предотвратить несчастные случаи на рабочем месте, снизить риски воздействия вредных факторов, связанных с трудовой деятельностью, и предупредить развитие профессиональных заболеваний, которые могут возникнуть в процессе работы. Одним из ключевых аспектов обеспечения безопасности и здоровья при работе с компьютерами является соблюдение требований к рабочим местам, оборудованным персональными компьютерами.

Точное соблюдение правил безопасности на рабочем месте с ПК является необходимым условием для снижения риска несчастных случаев, предотвращения травм и создания комфортных условий для эффективной работы. Это включает в себя соблюдение правил эргономики, правильную настройку рабочего места с учетом индивидуальных особенностей оператора и установленных требований, а также регулярное обслуживание и проверку оборудования.

Такой подход способствует поддержанию здоровья оператора, повышению его продуктивности и минимизации возможных негативных последствий от работы с компьютером.

Для веб-сервиса, разработанного в рамках данной ВКР, используется оборудование для стандартного автоматизированного рабочего места:

- компьютер;
- программное обеспечение.

ГОСТ 12.0.003-2015 «Система стандартов по безопасности труда. Опасные и вредные производственные факторы. Классификация.» устанавливает понятийную и терминологическую систему опасных и вредных факторов производственной среды и трудового процесса, которые могут вызвать негативное воздействие на организм работающих. Поэтому для создания комфортных

условий труда на рабочем месте с ЭВМ необходимо обеспечить выполнение санитарно-гигиенических, технических, эргономических и медико-профилактических требований.

4.1.1 Требования к помещению для работы с ПЭВМ

Сервер веб-сервиса это в первую очередь достаточно мощный компьютер, который должен располагаться в отдельном помещении с рабочим местом для системного администратора. Поэтому важную роль играет обеспечение безопасных, комфортных условий для работы, пресечение возникновения профессиональных заболеваний и несчастных случаев.

Регулярная работа за компьютером сопровождается постоянным влиянием множества вредных для здоровья факторов. Среди них несоответствие эмиссионных характеристик дисплея, системного блока, клавиатуры и мыши установленным стандартам, некорректные визуальные параметры дисплеев, чрезмерное освещение, нерациональная организация рабочего пространства, высокий уровень шума и увеличенные показатели электрического и магнитного полей.

Зачастую специалисты, проводящие больше 12 часов в день за экраном монитора, со временем начинают страдать от профессиональных заболеваний. Поэтому для всех без исключения офисных работников очень важна правильная организация рабочего места за компьютером.

Поэтому необходимо соблюдать требования к производственным зданиям, помещениям и сооружениям. Они включают следующие пункты:

- окна в помещении, используемом для эксплуатации ПЭВМ, должны быть ориентированы на север или северо-восток, если они служат источником естественного освещения в дневное время;

- если рабочее место в помещении оборудовано экраном с использованием жидкокристаллических или плазменных технологий, минимальная площадь такого рабочего места должна составлять не менее 4,5 квадратных метров;

– при размещении нескольких промышленных установок, генерирующих электромагнитное излучение (ЭМИ), в одном помещении, их размещение должно исключать возможность превышения гигиенических нормативов на рабочих местах в результате суммирования энергии излучения;

– помещение, где происходит эксплуатация ПЭВМ, должно обладать как искусственным, так и естественным освещением. В случае невозможности обеспечить естественное освещение, его использование допускается лишь при наличии достаточного обоснования и положительного заключения от санитарно-эпидемиологических служб;

– в случае наличия рядом силовых кабелей, высоковольтных линий или трансформаторов, а также технологического оборудования, способного вызывать помехи в работе вычислительной техники, необходимо переместить рабочие места как можно дальше от указанных элементов, чтобы избежать ошибок как программного, так и аппаратного характера при работе с ПЭВМ.

4.1.2 Требования к уровням шума и вибрации при работе с ПЭВМ

Уровень шума на рабочем месте также играет важную роль при работе с ЭВМ. Высокий уровень шума может вызывать стресс, утомляемость и снижать концентрацию, что отрицательно сказывается на производительности и благополучии сотрудников. Поэтому контроль уровня шума является необходимым аспектом обустройства рабочих мест.

Для того, чтобы избежать негативных последствий необходимо следовать рекомендациям:

– уровень звукового давления на рабочем месте оператора ПЭВМ не должен превышать 50 дБА;

– если в помещении используется оборудование, создающее шумовую нагрузку, превышающую установленные нормативы, такое оборудование должно быть размещено вне помещения, где находится персональная электронно-вычислительная машина;

– при использовании оборудования, превышающего установленные нормативы по уровню шума, необходимо применять средства и методы, направленные на снижение шума или препятствующие его распространению. Возможным решением является замена шумного оборудования на аналогичные модели с более низким уровнем шума;

– для снижения негативного влияния шума и вибрации необходимо соблюдать режим труда и отдыха, предусматривающий регулярные перерывы и смену деятельности;

– в случае превышения допустимых уровней шума или вибрации, операторы ПЭВМ должны обеспечиваться средствами индивидуальной защиты, такими как наушники или противошумные вкладыши для защиты от шума.

4.1.3 Требования к освещению на рабочих местах с ПЭВМ

Освещение является ключевым аспектом обустройства рабочих мест, включая рабочие места с использованием ЭВМ. Правильное освещение играет важную роль в комфорте, здоровье и производительности операторов ЭВМ.

Для того чтобы правильно обустроить освещение на рабочем месте необходимо соблюдать следующие нормы:

– естественный свет должен падать преимущественно слева на рабочее место, для чего рабочие места необходимо размещать так, чтобы мониторы стояли боком к световым проемам;

– искусственное освещение должно осуществляться с помощью систем равномерного освещения, а в производственных помещениях с помощью систем комбинированного освещения. На поверхности экранов освещение не должно создавать бликов, а их освещенность должна быть не более 300 лк. Освещенность стола в рабочей зоне должна быть в пределах от 300 до 500 лк;

– яркость светящихся поверхностей, находящихся в поле зрения, не должна превышать 200 кд/м², при этом необходимо ограничить прямую блескость;

– яркость бликов на экране не должна превышать 40 кд/м², а от потолка 200 кд/м², при этом необходимо ограничить отраженную блёскость;

– в зоне углов освещения от 50 до 90 градусов яркость светильников общего освещения не должна составлять более 200 кд/м², при этом защитный угол светильников должен быть не менее 40 градусов, так же, как и светильники местного освещения;

– неравномерность распределения яркости рабочей зоны следует ограничить;

– при использовании люминесцентных светильников общее освещение необходимо строить в виде продольных линий сплошных или прерывистых, расположенных сбоку от рабочего места параллельно линии зрения пользователя;

– коэффициент пульсации осветительных установок не должен превышать 5 %;

– следует своевременно проводить замену перегоревших ламп, а также проводить чистку стекол светильников и окон не реже чем два раза в год.

Требования к освещению рабочих мест также описаны в СанПиН 1.2.3685-21.

4.1.4 Требования к микроклимату рабочего места с ПЭВМ

Микроклиматические условия играют ключевую роль при работе с электронно-вычислительными машинами, непосредственно влияя на комфорт и производительность сотрудников. Поддержание оптимальных параметров микроклимата помогает снизить физическое и психологическое напряжение, связанное с работой на компьютере, и способствует общему благополучию.

Согласно санитарно-эпидемиологическим нормам СанПиН 1.2.3685-21, установлены стандарты и показатели микроклимата на рабочих местах.

Оптимальные условия микроклимата определяются исходя из требований к комфортной температуре для человека, одетого в одежду с теплоизоляцией 1 кло в холодное время года и 0,7–0,8 кло в теплый период. Они обеспечивают удовлетворительное тепловое состояние в течение рабочего дня при

минимальной нагрузке на механизмы терморегуляции, не влияют на здоровье и способствуют высокой эффективности труда.

Допустимые микроклиматические условия также определяются требованиями к тепловому комфорту для человека, одетого в одежду с теплоизоляцией 1 кло в холодный период и 0,7–0,8 кло в теплый период в течение восьмичасовой рабочей смены. Они не представляют угрозы для здоровья, но могут вызывать некоторое чувство дискомфорта или напряжение механизмов терморегуляции, снижая общее самочувствие и эффективность труда.

Допустимые величины параметров микроклимата на рабочих местах применительно к выполнению работ различных категорий в холодный и теплый периоды года приведены в таблице 18.

Таблица 18 – Допустимые величины параметров микроклимата на рабочих местах

Период года	Категория работ по уровням энергозатрат, Вт	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с, не более	
		диапазон ниже оптимальных величин	диапазон выше оптимальных величин			ниже оптимальных величин, не более	выше оптимальных величин, не более
1	2	3	4	5	6	7	8
Холодный	Ia (до 139)	20,9-21,9	24,1-25,0	19-26	15-75	0,1	0,1
	Iб (140-174)	19,0-20,9	23,1-24,0	18-25	15-75	0,1	0,2
Теплый	Ia (до 139)	21,0-22,9	25,1-28,0	20-29	15-75	0,1	0,2
	Iб (140-174)	20,0-21,9	24,1-28,0	19-29	15-75	0,1	0,3

4.1.5 Требования к организации рабочих мест с ПЭВМ

Исследования и практический опыт показывают, что правильная организация рабочего пространства играет значительную роль в работоспособности, комфорте и благополучии сотрудников. Важность этого аспекта связана с несколькими факторами.

Рабочее место представляет собой целостную единицу производства, все его элементы должны соответствовать антропометрическим, гигиеническим, физическим и психологическим требованиям.

К антропометрическим требованиям, показатели для соответствия строения оборудования характеристикам человека: возможность свободного движения; поза трудящегося. К физическим и психологическим относится восприятие и переработка информации.

Под организацией рабочего места понимается проведение системы мероприятий по его оборудованию способами, предметом труда и размещения их в определенном порядке с целью достижения:

- оптимизации условий деятельности;
- безопасности труда;
- максимальной эффективности;
- комфортности работы человека.

Эргономическими аспектами проектирования рабочих мест также являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте, характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места.

Правильная организация рабочих мест обеспечивает безопасность персонала: свободный доступ к аварийным выходам, безопасное размещение оборудования и материалов. Неправильная организация рабочего пространства может привести к нарушениям опорно-двигательного аппарата, возникновению сколиоза, нарушениям обмена веществ и образованию тромбов, особенно в нижних конечностях, что может увеличить риск инсульта. Для предотвращения подобных проблем существуют рекомендации по расположению пользователя за компьютером, которые представлены на рисунке 45.

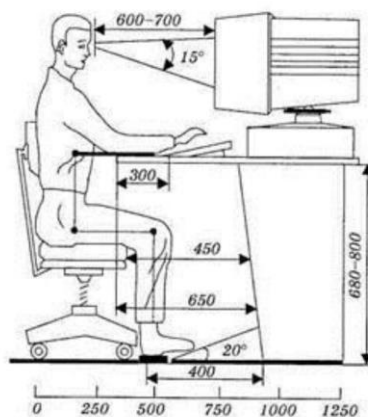


Рисунок 45 – Рекомендуемое размещение пользователя за ПЭВМ

Также в данной области существуют определенные рекомендации, направленные на организацию рабочего места пользователя вычислительной техники:

- рабочий стол должен быть регулируемым по высоте в диапазоне от 680 до 800 миллиметров или иметь стандартную высоту в 725 миллиметров. Рекомендуемые размеры: ширина от 800 до 1400 миллиметров, глубина от 800 до 1000 миллиметров;

- под столом должно быть свободное пространство для ног высотой не менее 600 миллиметров, шириной 500 миллиметров, глубиной на уровне колен не менее 450 миллиметров, а при вытянутых ногах не менее 650 миллиметров;

- рабочий стул должен иметь сиденье шириной и глубиной не менее 400 миллиметров, регулируемую высоту от 400 до 550 миллиметров, наклон вперед до 15 градусов, назад до 5 градусов. Спинка должна быть высотой от 280 до 320 миллиметров, шириной 380 миллиметров, радиусом кривизны 400 миллиметров, с наклоном 30 градусов. Регулируемые подлокотники должны быть высотой от 200 до 260 миллиметров над сиденьем, расстоянием между ними от 350 до 500 миллиметров и длиной не менее 250 миллиметров;

- клавиатура должна быть расположена на расстоянии от края стола от 100 до 300 миллиметров;

– необходимо иметь подставку для ног размером шириной 300 миллиметров, глубиной 400 миллиметров, высотой 150 миллиметров, с наклоном до 20 градусов;

– расстояние между мониторами спереди должно быть не менее 2 метров, а сбоку не менее 1,2 метров;

– экран монитора должен располагаться на расстоянии от глаз оператора от 600 до 700 миллиметров.

4.1.6 Требования к средствам отображения информации индивидуального пользования

Соблюдение требований к средствам отображения информации индивидуального пользования с точки зрения здоровья пользователя помогает предотвратить множество потенциальных проблем и улучшить общее качество жизни, обеспечивая более безопасное и комфортное взаимодействие с технологиями.

Данные требования описываются стандартом ГОСТ Р 50948-2001 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности.». Поэтому необходимо соблюдать следующие требования для дисплея:

– яркость знака должна быть не менее 35 кд/м для дисплеев на ЭЛТ и не менее 20 кд/м для плоских дискретных экранов;

– электростатический потенциал экрана дисплеев на ЭЛТ должен быть в пределах от минус 500 В до 500 В;

– яркостный контраст изображения должен быть не менее 3:1 (для плоских дискретных экранов при угле наблюдения от минус 40° до плюс 40°). Яркостный контраст внутри знака и между знаками должен быть не менее 3:1;

– Конструкция дисплея должна обеспечивать возможность фронтального наблюдения изображения на экране путем поворота корпуса дисплея вокруг вертикальной оси в пределах от плюс 30° до минус 30° и вокруг горизонтальной оси в пределах от минус 15° до плюс 30° с фиксированием дисплея в заданном положении;

– Корпус дисплея должен быть окрашен в мягкие тона. Корпус дисплея, клавиатура и другие устройства персонального компьютера должны иметь матовую (диффузно отражающую) поверхность одного цвета и не иметь блестящих деталей, способных создавать блики;

– конструкция дисплея должна предусматривать наличие органов регулирования яркости и контраста;

– конструкция дисплея должна обеспечивать максимально возможное снижение уровней электростатического и электромагнитного полей.

Также при разработке программного продукта были соблюдены следующие требования:

– для точного распознавания и идентификации цветов должны применяться цветное изображение переднего плана на ахроматическом фоне или ахроматическое изображение переднего плана на цветном фоне;

– число цветов, одновременно отображаемых на экране дисплея, должно быть минимальным. Для точной идентификации цвета каждый заданный по умолчанию набор цветов должен включать не более 11 цветов;

– для чтения текстов, буквенно-цифровых знаков и символов при положительной полярности изображения не следует применять синий цвет спектра на красном фоне;

– если цветовой параметр может быть изменен пользователем, то должна быть предусмотрена возможность восстановления назначенного по умолчанию набора цветов.

Соблюдение всех вышеописанных требований позволит создать благоприятные условия труда для операторов ПЭВМ, снизить риск развития профессиональных заболеваний и повысить производительность труда.

4.2 Экологичность

Воздействие активного производства электроники на окружающую среду и здоровье человека является серьезной проблемой, несмотря на его положительное влияние на повседневную жизнь.

Процессы производства электроники включают химические процессы, которые приводят к образованию и накоплению опасных отходов, содержащих значительное количество негативных элементов. При производстве компьютерной техники используются различные материалы, включая бериллий, алюминий, титан, ванадий, хром, марганец, железо, кобальт, никель, медь, цинк, галлий, германий, мышьяк, селен, ниобий, рутений, родий, палладий, серебро, кадмий, индий, олово, барий, европий, тантал, платина, золото, ртуть, свинец, висмут и актиний.

В связи с использованием данных элементов в производстве персональных электронно-вычислительных машин (ПЭВМ), требуются специальные методы утилизации для устранения негативного воздействия на окружающую среду.

Процесс утилизации вычислительной техники включает следующие шаги. Прежде всего, на предприятии создается специальная комиссия, которая занимается выбором техники, подлежащей утилизации. Перед самой утилизацией проводится экспертная оценка, которая подтверждает необходимость утилизации выбранной техники. Экспертом может выступать как сотрудник предприятия, так и независимый специалист, обладающий соответствующей компетенцией.

Далее составляется акт списания, который отражает причины и количество техники, подлежащей списанию, а также отражается в бухгалтерском учете предприятия. Утилизация техники производится на специализированном предприятии, обладающем разрешением на переработку вычислительной техники.

По завершении утилизации, предприятие предоставляет официальный документ, подтверждающий соответствие утилизации требуемым технологиям и отсутствие загрязнения окружающей среды отходами техники.

Также утилизации подлежат отходы осветительных устройств, ртутьсодержащих электрических ламп поскольку они могут являться опасными для жизни из-за паров ртути.

Накопление в организации для последующей утилизации неповрежденных отработанных ртутьсодержащих ламп должно производиться в соответствии с требованиями безопасности, предусмотренными производителем ртутьсодержащих ламп, указанных в правилах эксплуатации таких товаров. Накопление неповрежденных отработанных ртутьсодержащих ламп должно производиться в индивидуальной и транспортной упаковке, обеспечивающих сохранность отработанных ртутьсодержащих ламп. Допускается использовать для накопления отработанных ртутьсодержащих ламп упаковку от новых ламп в целях исключения возможности повреждения таких ламп.

Накопление отработанных ртутьсодержащих ламп должно производиться отдельно от других видов отходов. Не допускается совместное накопление поврежденных и неповрежденных ртутьсодержащих ламп.

Для транспортирования поврежденных отработанных ртутьсодержащих ламп должна использоваться герметичная тара, исключающая возможность загрязнения окружающей среды и причинение вреда жизни и здоровью человека.

Операторы, осуществляющие сбор, транспортирование, обработку, утилизацию, обезвреживание, хранение отработанных ртутьсодержащих ламп, должны вести учет принятых, транспортированных, обработанных, утилизированных, обезвреженных, находящихся на хранении отходов. Захоронение отработанных ртутьсодержащих ламп запрещено.

Остальные отходы, на предприятии эксплуатирующем ЭВМ можно отнести к неопасным отходам пятого класса. Утилизация таких отходов аналогична утилизации обычного бытового мусора.

4.3 Чрезвычайные ситуации

Безопасность при эксплуатации персональных электронно-вычислительных машин (ПЭВМ) является важным аспектом, поскольку помещения, где они используются, подвержены возможным пожарным рискам. Пожары могут возникать из-за короткого замыкания между компонентами ПЭВМ или перегрева ее элементов.

Для предотвращения пожароопасных ситуаций в помещениях, где осуществляется работа с ЭВМ и оргтехникой, необходимо проверять целостность токопроводящих кабелей, вилки и розетки, отсутствие повреждений аппаратуры. Компоненты ЭВМ должны иметь функцию самоотключения при повышении температуры в ходе неисправности систем охлаждения и кондиционирования для предотвращения перегрева. Технические средства должны проходить проверки и техническое обслуживание, а именно, необходимо проверять работоспособность, целостность и другие рабочие характеристики. Необходимо проводить уборку и очистку этих устройств. Для удаления пыли и пятен с ЭВМ должны применяться негорючие жидкости и материалы.

Должен быть определен порядок и сроки прохождения противопожарного инструктажа и занятий по пожарно-техническому минимуму.

В случае пожара следует немедленно позвонить в службу спасения по номерам 01 или 112, обесточить всё электронное оборудование, провести оперативную эвакуацию людей и принять меры для тушения возгорания.

Для борьбы с пожаром на начальном этапе используются огнетушители. Поскольку в помещениях с компьютерной техникой наиболее вероятными классами пожаров являются «А» и «Е» (то есть могут гореть в основном твёрдые вещества, сопровождающиеся тлением – класс А, или возможны пожары, вызванные возгоранием электроустановок – класс Е), следует применять углекислотные и порошковые огнетушители.

Важно обеспечить наличие огнетушителей вблизи возможных очагов возгорания, не превышая расстояние 20 метров в общественных зданиях и сооружениях.

В замкнутых помещениях объемом до 50 м³ также можно использовать подвесные самосрабатывающие порошковые огнетушители в дополнение к переносным огнетушителям.

Для обеспечения безопасной эвакуации персонала в случае задымления помещений и коридоров рекомендуется размещать фотолюминесцентные эвакуационные знаки у дверных проемов, выключателей, рубильников и на пути

эвакуации. Важно иметь не менее двух эвакуационных выходов в комплексе помещений, путь до которых должны указывать настенные эвакуационные знаки.

Для автоматического обнаружения пожаров могут использоваться различные извещатели, которые должны быть установлены согласно требованиям и реагировать на определенные параметры среды.

Ручные извещатели предназначены для передачи информации о пожаре с помощью человека и должны быть размещены на высоте 1,5 м от уровня пола. Автоматические пожарные извещатели, за исключением световых, устанавливаются на потолке помещения.

ЗАКЛЮЧЕНИЕ

В рамках выполнения данной выпускной квалификационной работы было создано два программных продукта. Представляющих собой шаблоны интерактивных заданий «Кроссворд» и «Угадай слово».

На первом этапе был проведен анализ предметной области исследования. В рамках этого анализа рассматривались современные методики обучения и тестирования обучающихся, проведен обзор существующих средств решения, рассмотрены их плюсы и минусы.

На втором этапе работы было детально описано функциональное наполнение исследуемого веб-сервиса. Уделялось внимание взаимодействию разработанных шаблонов с веб-сервисом, что включало рассмотрение механизма интеграции шаблонов и их взаимосвязи с основными модулями сервиса. Было обосновано использование конкретных средств реализации и спроектирована база данных, необходимая для корректной работы шаблонов.

Третий этап посвящен разработке программных продуктов. Выполнен проект двух шаблонов: «Кроссворд» и «Угадай слово»; разработаны алгоритмы их работы, реализован интерфейс пользователя. Разработка шаблонов выполнена на языке программирования JavaScript и языке разметки HTML. Для связи шаблонов с БД использованы средства языка программирования C#.

В настоящее время оба шаблона загружены в образовательный сервис, проведено тестирование их работоспособности. В дальнейшем предполагается увеличение количества шаблонов.

Образовательный web-сервис «ActiveTemplate» получил официальную регистрацию, свидетельство об официальной регистрации программы для ЭВМ приведено в приложении В.

Последний этап выпускной квалификационной работы был посвящен безопасности и экологичности рассматриваемого веб-сервиса и разработанных программных продуктов в соответствии с требованиями государственных стандартов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Адамс, Д. Р. Основы работы с XHTML и CSS : учебник / Д. Р. Адамс, К. С. Флойд. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 567 с.
- 2 Адам, Фримен ASP.NET Core MVC с примерами на C# для профессионалов. Руководство / Ф. Адам. – М.: Диалектика-Вильямс, 2017. – 483 с.
- 3 Арустамян, Д. В. Использование интерактивных методов обучения в учебном процессе / Д. В. Арустамян, Е. А. Дроздова // Евразийский союз ученых. – 2018. – № 7-4(52). – С. 7-8.
- 4 Беликова, С. А. Основы HTML и CSS: проектирование и дизайн веб-сайтов : учебное пособие по курсу «Web-разработка» / С. А. Беликова, А. Н. Беликов. – Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2020. – 174 с.
- 5 Гончаров, А. Ю. Web-дизайн: HTML, JavaScript и CSS: карманный справочник / А.Ю. Гончаров. – М.: КУДИЦ-Образ, 2021. – 367 с.
- 6 ГОСТ 12.0.003-2015. Система стандартов по безопасности труда. Опасные и вредные производственные факторы. Классификация. – Взамен ГОСТ 12.0.003-74 ; введ. 2017-03-01. – Межгос. совет по стандартизации, метрологии и сертификации ; М. : Стандартиформ, 2019. – 16 с.
- 7 ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. – Взамен ГОСТ Р 50948-96 ; введ. 2002-07-01. – Постановлением Госстандарта России ; М.: Стандартиформ, 2008. – 11 с.
- 8 Дебольт, Вирджиния HTML и CSS. Совместное использование / Дебольт В. – М.: ИТ Пресс, 2020. – 512 с.
- 9 Ильин, И.В. Базы данных : учебное пособие / Ильин И.В., Ильяшенко О.Ю.. – Санкт-Петербург : Санкт-Петербургский политехнический университет Петра Великого, 2020. – 96 с.

10 Кардаш, Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ : учеб. пособие / Т. А. Кардаш ; АмГУ, ИФФ. – Благовещенск : Изд-во Амур. гос. ун-та, 2002. – 60 с.

11 Космачева, И. М. Проектирование защищенных баз данных : учебное пособие / И. М. Космачева, Н. В. Давидюк. – СПб. : Интермедия, 2020. – 144 с.

12 Крокфорд, Д. JavaScript. Сильные стороны / Д. Крокфорд. – М.: Питер, 2018. – 199 с.

13 Кумскова, И. А. Базы данных / И.А. Кумскова. – М.: КноРус, 2021. – 488 с.

14 Лок, Э. ASP.Net Core в действии / Э. Лок ; перевод Д. А. Беликов. – Москва : ДМК Пресс, 2021. – 906 с.

15 Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 245 с.

16 Петюшкин, А. HTML в Web-дизайне / А. Петюшкин. – М.: БХВ-Петербург, 2019. – 509 с.

17 Рындин, Н. А. Технологии разработки клиентских WEB-приложений на языке JavaScript : учебное пособие / Н. А. Рындин. – Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. – 54 с.

18 Саблина, В. А. Основы программирования на JavaScript : учебное пособие / В. А. Саблина, Е. А. Трушина. – Рязань : Рязанский государственный радиотехнический университет, 2022. – 96 с.

19 СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. – Взамен СанПиН 2.2.2/2.4.1340-03 ; введ. 2021-03-01. – Постановлением главного государственного санитарного врача Российской Федерации ; М.: Стандартинформ, 2021. – 988 с.

20 СП 1.13130.2020 Система противопожарной защиты. Эвакуационные пути и выходы. – Взамен СП 1.13130.2009 ; введ. 2020-09-19. – Приказом Министерства Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России) ; М. : Стандартинформ, 2020. – 65 с.

21 Столбовский, Д. Н. Разработка Web-приложений ASP.NET с использованием Visual Studio .NET : учебное пособие / Д. Н. Столбовский. – 3-е изд. – Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 375 с.

22 Титтел, Эд HTML, XHTML и CSS для чайников / Э. Титтел, Д. Ноубл. – М.: Диалектика, 2020. – 400 с.

23 Туманов, В. Е. Основы проектирования реляционных баз данных : учебное пособие / В. Е. Туманов. – 3-е изд. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 502 с.

24 Хантер, Томас. Многопоточный JavaScript / Т. Хантер, Б. Инглиш ; перевод А. А. Слинкин. – Москва : ДМК Пресс, 2022. – 190 с.

25 Чамберс, Джеймс ASP.NET Core. Разработка приложений. Руководство / Д. Чамберс, Д. Пэккетт, С. Тиммс. – М.: Питер, 2018. – 464 с.

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программного продукта «Шаблон интерактивного задания «Кроссворд» для образовательного web-сервиса «ActiveTemplate»»

1 ВВЕДЕНИЕ

1.1 Наименование

Наименование программы «Шаблон интерактивного задания «Кроссворд» для образовательного web-сервиса «ActiveTemplate»»

1.2 Краткая область применения

Программное средство «Шаблон интерактивного задания «Кроссворд» для образовательного web-сервиса «ActiveTemplate»» предназначено для размещения на web-сервисе для обеспечения возможности создания и последующего прохождения интерактивного задания «Кроссворд».

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки является задание к выпускной квалификационной работе от 02.10.2023 г. Задание утверждено ректором Амурского государственного университета, именуемым в дальнейшем Заказчиком, и студентом 4 курса ФГБОУ ВО АмГУ Яцуном Евгением Олеговичем, именуемым в дальнейшем Исполнителем, 03.04.2024 г.

Согласно заданию, Исполнитель обязан разработать программный продукт не позднее 17.06.2024.

Наименование темы работы – «Разработка шаблонов интерактивных заданий для образовательного web-сервиса «ActiveTemplate»». Условное обозначение темы работы (шифр темы) – «Шаблон – 001».

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Программа может использоваться преподавателями любого образовательного учреждения для создания интерактивных упражнений вида «Кроссворд» с целью проверки знаний обучающихся.

Продолжение Приложения А

3.1 Функциональное назначение

Обеспечение возможности создания на web-сервисе на основе шаблона интерактивного задания, в виде кроссворда.

Максимально возможное количество слов для создания задания не должно превышать двадцати, максимальная длина слова не должна превышать двадцати символов на одном языке, кириллице или латинице.

Первый пользователь, именуемый Преподаватель должен иметь возможность создавать задания, указывать количество вопросов, количество ответов, второй пользователь, именуемый Обучающийся должен иметь возможность пройти созданное первым пользователем задание. Для прохождения задания обучающийся должен быть зарегистрирован и иметь доступ к личному кабинету образовательного сервиса. После прохождения задания в личных кабинетах автора и прошедшего упражнение должен появиться результат – оценка пройденного задания.

3.2 Эксплуатационное назначение

Программный продукт является компонентом веб-сервиса для создания интерактивных заданий, который позволяет облегчить процесс проверки знаний обучающихся путем упрощения создания заданий для обучающихся. Каждый пользователь, создавший задание, сможет увидеть результаты тех, кто прошел данное задание, что упрощает механизм оценивания и проверки результатов работы.

4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1 Требования к функциональным характеристикам

Программа после загрузки на web-сервис является компонентом серверной части образовательного web-сервиса.

На серверной части должна быть реализована база данных для хранения созданных преподавателем упражнений с разграничением на шаблон, задание, вопрос, ответ с доступом к этим данным.

Продолжение Приложения А

Пользователи взаимодействуют с web-сервисом посредством браузера. Взаимодействие должно осуществляться посредством HTTP-запросов. При получении GET-запроса от клиента, сервер должен отправить запрашиваемые данные, страницу для генерации шаблона или страницу для прохождения созданного упражнения при наличии доступа у пользователя.

4.1.1 Требования к составу выполняемых функций

Программный продукт должен обеспечивать возможность выполнения следующих функций:

- выбор количества вопросов при создании задания;
- добавление вопроса в задание при его создании;
- возможность удаления вопроса из задания при его создании;
- возможность добавления вариантов ответа в задание при его создании;
- возможность удаления вариантов ответа из задания при его создании;
- обеспечение возможности менять фон у макета задания;
- обеспечение возможности указывать время на прохождение задания;
- автоматическая генерация полей кроссворда с пересекающимися словами;
- добавление слов в ячейки кроссвордной сетки при верном вводе;
- блокирование недопустимых символов;
- подсчёт очков по количеству правильно введенных букв;
- функция запуска начала задания и его окончания;
- генерация упражнения по заданным параметрам.

4.1.2 Требования к организации входных и выходных данных

Ввод данных осуществляет пользователь.

Входные данные представляют собой вводимые через интерфейс программы в специальных формах символьные и численные значения, которые после обработки сохраняются в базе данных системы.

Продолжение Приложения А

Выходными данными являются представления, отображаемые через браузер пользователя, которые будут содержать в себе данные, которые запрашивает пользователь при наличии к ним доступа.

4.1.3 Требования к временным характеристикам

После изменения пользователем данных, находящихся в базе данных, новая информация должна отображаться не позднее, чем через 1 минуту.

4.2 Требования к надежности

4.2.1 Требования к обеспечению надежного функционирования

Пользователю, работающему с программой через веб-браузер, должен быть предоставлен непрерывный доступ к веб-приложению, расположенному по определённой url-адресу. Веб-сервис не должен непредвиденно прерывать свою работу.

Должна быть обеспечена отказоустойчивость: исключение потерь данных при нарушении интерфейсов, сбоях (отказах) технических или программных средств серверной части.

Надежное функционирование должно быть обеспечено Заказчиком, выполнением организационно-технических мероприятий:

- организацией бесперебойного питания технических средств.
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Продолжение Приложения А

4.2.2 Время восстановления после отказа

В случае отказа работы серверной части и последующей недоступности веб-приложения, время восстановления не должно превышать одни рабочие сутки.

4.2.3 Отказы из-за некорректных действий оператора

После запуска программы на сервере отказ программы вследствие некорректных действий оператора должен быть исключён. В том числе должна быть исключена возможность непреднамеренного выключения программы, не связанного с техническими неполадками сервера.

4.3 Условия эксплуатации

4.3.1 Климатические условия эксплуатации

ЭВМ должна эксплуатироваться согласно требованиям стандарта ГОСТ 21552-84 п. 3.2, а именно:

- температура окружающего воздуха 15-35 °С;
- относительная влажность воздуха при температуре 25 °С 45-80 %;
- атмосферное давление 84-107 кПа (630-800 мм рт. ст.).

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

Установку и настройку программы должен проводить системный администратор. Пользователем программы может являться любой посетитель веб-сайта.

Системный администратор должен обладать навыками для выполнения следующего перечня задач:

- установка клиентских приложений;
- настройка СУБД;
- настройка сети между клиентами и СУБД;
- настройка серверной операционной системы;
- установка и запуск серверного веб-приложения.

Продолжение Приложения А

Пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

Компьютер сервера, включающий в себя:

- 4 ядерный процессор x86 с тактовой частотой, не менее 2.5 ГГц;
- оперативную память объемом, не менее 10 Гб;
- периферийные устройства;
- операционная система Windows Server 2008 Service Pack 2 и выше;
- поддерживаемые протоколы передачи данных: HTTP / HTTPS;
- при одновременной работе 100 пользователей минимальная необходи-

мая ширина канала сервера веб-приложения 15 Мбит/сек.

4.5 Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам и методам решения

Пользовательский интерфейс должен быть интуитивно понятным и приятным глазу, не содержать различных ярко и быстро мерцающих объектов.

4.5.2 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть написаны на языках программирования и разметки HTML, CSS, JavaScript, C# с использованием технологии asp.net core. В качестве интегрированной среды разработки должна быть использована среда Microsoft Visual Studio 2022 и новее.

4.5.3 Требования к программным средствам, используемым образовательным web-сервисом

На сервере должны быть установлены:

- SQL Server 2017 и новее;
- SQL Server Management Studio версии 17.9.1 и выше.

Клиентская часть должна содержать следующие элементы:

- операционная система с поддержкой программного обеспечения доступа в Интернет;

Продолжение Приложения А

– браузер (за исключением консольных браузеров).

4.5.4 Требования к защите информации и программ

Функции контроля доступа обеспечиваются программными средствами серверной части.

4.6 Требования маркировке и упаковке

Программное изделие поставляется на дистрибутивном носителе.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Специальных требований не предъявляется.

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Предварительный состав программной документации:

- техническое задание (ГОСТ 19.201-78);
- пояснительная записка (ГОСТ 19.404-79);
- руководство системного программиста (ГОСТ 19.503-79);
- руководство оператора (ГОСТ 19.505-79);
- программа и методика испытаний (ГОСТ 19.301-79);
- текст программы. (ГОСТ 19.401-78).

6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Использование разрабатываемого продукта сократит трудозатраты преподавателей на составление упражнений для проверки знаний обучающихся. А также на проверку результатов. Также это сократит расходы на канцелярские товары такие как бумага, ручки и тому подобное поскольку тестирование будет проводиться посредством технических средств.

Предполагаемое число использования программного продукта при одновременном доступе – 100 пользователей.

7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка должна быть проведена в три стадии:

- техническое задание;

Продолжение Приложения А

- технический (и рабочий) проекты;
- функциональное тестирование программы.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации.

На стадии «Функциональное тестирование программы» необходимо провести испытания программы.

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;

Продолжение Приложения А

– корректировка программы и программной документации по результатам испытаний.

8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1 Виды испытаний

Производится проверка корректного выполнения программой заложенных в нее функций, указанных в пункте 4.1.1 настоящего документа, т.е. осуществляется функциональное тестирование программы. Также осуществляется визуальная проверка интерфейса программы.

8.2 Общее требование к приемке работы

Прием программы будет утвержден при корректной работе программы.

ПРИЛОЖЕНИЕ Б

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программного продукта «Шаблон интерактивного задания «Угадай слово» для образовательного web-сервиса «ActiveTemplate»»

1 ВВЕДЕНИЕ

1.1 Наименование

Наименование программы «Шаблон интерактивного задания «Угадай слово» для образовательного web-сервиса «ActiveTemplate»»

1.2 Краткая область применения

Программное средство «Шаблон интерактивного задания «Угадай слово» для образовательного web-сервиса «ActiveTemplate»» предназначено для размещения на web-сервисе для обеспечения возможности создания и последующего прохождения интерактивного задания «Угадай слово».

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки является задание к выпускной квалификационной работе от 02.10.2023 г. Задание утверждено ректором Амурского государственного университета, именуемым в дальнейшем Заказчиком, и студентом 4 курса ФГБОУ ВО АмГУ Яцуном Евгением Олеговичем, именуемым в дальнейшем Исполнителем, 03.04.2024 г.

Согласно заданию, Исполнитель обязан разработать программный продукт не позднее 17.06.2024.

Наименование темы работы – «Разработка шаблонов интерактивных заданий образовательного web-сервиса «ActiveTemplate»». Условное обозначение темы работы (шифр темы) – «Шаблон – 002».

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Программа может использоваться преподавателями любого образовательного учреждения для создания интерактивных упражнений вида «Угадай слово» с целью тестирования обучающихся.

Продолжение Приложения Б

3.1 Функциональное назначение

Обеспечение возможности создания на web-сервисе на основе шаблона интерактивного задания.

Максимальная длина слова не должна превышать двадцати символов на одном языке, кириллице или латинице.

Первый пользователь, именуемый Преподаватель должен иметь возможность создавать задания, указывать количество вопросов, количество ответов, второй пользователь, именуемый Обучающийся должен иметь возможность пройти созданное первым пользователем задание. Для прохождения задания обучающийся должен быть зарегистрирован и иметь доступ к личному кабинету образовательного сервиса. После прохождения задания в личных кабинетах автора и прошедшего упражнение должен появиться результат – оценка пройденного задания.

3.2 Эксплуатационное назначение

Программный продукт является компонентом web-сервиса для создания интерактивных заданий, который позволяет облегчить процесс тестирования обучающихся путем упрощения создания тестов для обучающихся. Каждый пользователь, создавший задание сможет увидеть результаты тех, кто прошел данное задание, что упрощает механизм оценивания и проверки результатов работы.

4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1 Требования к функциональным характеристикам

Программа после загрузки на web-сервис является компонентом серверной части образовательного web-сервиса.

На серверной части должна быть реализована база данных для хранения созданных преподавателем упражнений с разграничением на шаблон, задание, вопрос, ответ с доступом к этим данным.

Пользователи взаимодействуют с web-сервисом посредством браузера. Взаимодействие должно осуществляться посредством HTTP-запросов. При

Продолжение Приложения Б

получении GET-запроса от клиента, сервер должен отправить запрашиваемые данные, страницу для генерации шаблона или страницу для прохождения созданного упражнения при наличии доступа у пользователя.

4.1.1 Требования к составу выполняемых функций

Программный продукт должен обеспечивать возможность выполнения следующих функций:

- обеспечение возможности задать вопрос для прохождения задания при его создании;
- обеспечение возможности задать ответ на вопрос для прохождения задания при его создании;
- обеспечение возможности менять фон у макета задания;
- добавление букв в поле для ответа при вводе верной буквы;
- блокирование ввода только что введенной буквы;
- таймер, отсчитывающий время прохождения задания;
- обновление (увеличение) количества ошибок при неверном варианте ответа при решении задания;
- обновление изображения, свидетельствующем о количестве ошибок, при неверном варианте ответа при решении задания;
- уведомление о победе при вводе всех правильных букв;
- уведомление о поражении при вводе неправильных букв сверх допустимого количества попыток.
- генерация задания по заданным параметрам.

4.1.2 Требования к организации входных и выходных данных

Ввод данных осуществляет пользователь.

Входные данные представляют собой вводимые через интерфейс программы в специальных формах символьные и численные значения, которые после обработки сохраняются в базе данных системы.

Продолжение Приложения Б

Выходными данными являются представления, отображаемые через браузер пользователя, которые будут содержать в себе данные, которые запрашивает пользователь при наличии к ним доступа.

4.1.3 Требования к временным характеристикам

После изменения пользователем данных, находящихся в базе данных, новая информация должна отображаться не позднее, чем через 1 минуту.

4.2 Требования к надежности

4.2.1 Требования к обеспечению надежного функционирования

Пользователю, работающему с программой через веб-браузер, должен быть предоставлен непрерывный доступ к веб-приложению, расположенному по определённому url-адресу. Веб-сервис не должен непредвиденно прерывать свою работу.

Должна быть обеспечена отказоустойчивость: исключение потерь данных при нарушении интерфейсов, сбоях (отказах) технических или программных средств серверной части.

Надежное функционирование должно быть обеспечено Заказчиком, выполнением организационно-технических мероприятий:

- организацией бесперебойного питания технических средств.
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Продолжение Приложения Б

4.2.2 Время восстановления после отказа

В случае отказа работы серверной части и последующей недоступности веб-приложения, время восстановления не должно превышать одни рабочие сутки.

4.2.3 Отказы из-за некорректных действий оператора

После запуска программы на сервере отказ программы вследствие некорректных действий оператора должен быть исключён. В том числе должна быть исключена возможность непреднамеренного выключения программы, не связанного с техническими неполадками сервера.

4.3 Условия эксплуатации

4.3.1 Климатические условия эксплуатации

ЭВМ должна эксплуатироваться согласно требованиям стандарта ГОСТ 21552-84 п. 3.2, а именно:

- температура окружающего воздуха 15-35 °С;
- относительная влажность воздуха при температуре 25 °С 45-80 %;
- атмосферное давление 84-107 кПа (630-800 мм рт. ст.).

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

Установку и настройку программы должен проводить системный администратор. Пользователем программы может являться любой посетитель веб-сайта.

Системный администратор должен обладать навыками для выполнения следующего перечня задач:

- установка клиентских приложений;
- настройка СУБД;
- настройка сети между клиентами и СУБД;
- настройка серверной операционной системы;
- установка и запуск серверного веб-приложения.

Продолжение Приложения Б

Пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

Компьютер сервера, включающий в себя:

- 4 ядерный процессор x86 с тактовой частотой, не менее 2.5 ГГц;
- оперативную память объемом, не менее 10 Гб;
- периферийные устройства;
- операционная система Windows Server 2008 Service Pack 2 и выше;
- поддерживаемые протоколы передачи данных: HTTP / HTTPS;
- при одновременной работе 100 пользователей минимальная необходи-

мая ширина канала сервера веб-приложения 15 Мбит/сек.

4.5 Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам и методам решения

Пользовательский интерфейс должен быть интуитивно понятным и приятным глазу, не содержать различных ярко и быстро мерцающих объектов.

4.5.2 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть написаны на языках программирования и разметки HTML, CSS, JavaScript, C# с использованием технологии asp.net core. В качестве интегрированной среды разработки должна быть использована среда Microsoft Visual Studio 2022 и новее.

4.5.3 Требования к программным средствам, используемым образовательным web-сервисом

На сервере должны быть установлены:

- SQL Server 2017 и новее;
- SQL Server Management Studio версии 17.9.1 и выше.

Клиентская часть должна содержать следующие элементы:

- операционная система с поддержкой программного обеспечения доступа в Интернет;

Продолжение Приложения Б

– браузер (за исключением консольных браузеров).

4.5.4 Требования к защите информации и программ

Функции контроля доступа обеспечиваются программными средствами серверной части.

4.6 Требования маркировке и упаковке

Программное изделие поставляется на дистрибутивном носителе.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Специальных требований не предъявляется.

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Предварительный состав программной документации:

- техническое задание (ГОСТ 19.201-78);
- пояснительная записка (ГОСТ 19.404-79);
- руководство системного программиста (ГОСТ 19.503-79);
- руководство оператора (ГОСТ 19.505-79);
- программа и методика испытаний (ГОСТ 19.301-79);
- текст программы. (ГОСТ 19.401-78).

6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Использование разрабатываемого продукта сократит трудозатраты преподавателей на составление упражнений для тестирования учеников или студентов. А также на проверку результатов. Также это сократит расходы на канцелярские товары такие как бумага, ручки и тому подобное поскольку тестирование будет проводиться посредством технических средств.

Предполагаемое число использования программного продукта при одновременном доступе – 200 пользователей.

7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка должна быть проведена в три стадии:

- техническое задание;

Продолжение Приложения Б

- технический (и рабочий) проекты;
- функциональное тестирование программы.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации.

На стадии «Функциональное тестирование программы» необходимо провести испытания программы.

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;

Продолжение Приложения Б

– корректировка программы и программной документации по результатам испытаний.

8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1 Виды испытаний

Производится проверка корректного выполнения программой заложенных в нее функций, указанных в пункте 4.1.1 настоящего документа, т.е. осуществляется функциональное тестирование программы. Также осуществляется визуальная проверка интерфейса программы.

8.2 Общее требование к приемке работы

Прием программы будет утвержден при корректной работе программы.

ПРИЛОЖЕНИЕ В

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024661102

Образовательный веб-сервис «ActiveTemplate»

Правообладатель: **Федеральное государственное бюджетное образовательное учреждение высшего образования "Амурский государственный университет" (RU)**

Авторы: **Галаган Татьяна Алексеевна (RU), Марков Никита Евгеньевич (RU), Яцун Евгений Олегович (RU)**

Заявка № **2024619923**

Дата поступления **07 мая 2024 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **15 мая 2024 г.**

*Руководитель Федеральной службы
по интеллектуальной собственности*

документ подписан электронной подписью
Сертификат 429b6a0673953164ba196f83b73b4aa7
Владелец **Зубов Юрий Сергеевич**
Действителен с 19.05.2023 по 02.08.2024

Ю.С. Зубов



Рисунок В.1 – Свидетельство о государственной регистрации веб-сервиса