

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных технологий

Кафедра проектирования информационных компьютерных систем

Дисциплина "Современные технологии проектирования информационных
систем"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры

_____ А.В.Михалькевич
09.07.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
на тему

Электронное средство обучения по предмету “Белорусский язык”

БГУИР КР 1-40 05 01-10 № 171 ПЗ

Студент (подпись студента)

Курсовая работа
представлена на проверку
09.07.2025

(подпись студента)

Реферат

БГУИР КР 1-40 05 01-10 № 171 ПЗ, гр. 784371

, Электронное средство обучения по предмету “Белорусский язык”, Минск: БГУИР - 2025.

Пояснительная записка 88889 с., 21 рис., 4 табл.

Ключевые слова: Электронное средство обучения, Курсовая работа, Современные технологии проектирования информационных систем, СТПИС

Предмет Современные технологии проектирования информационных систем, А.В.Михалькевич

Аннотация Курсовой проект на тему : Электронное средство обучения по предмету “Белорусский язык” Выполнила – Черехович А.Ю. Руководитель – Михалькевич А.В. Курсовой проект состоит из пояснительной записки, графической части, кода программного продукта. Пояснительная записка включает разделы - введение, описание предметной области, описание основных процессов предметной области, спецификация вариантов использования системы, информационная модель системы и её описание, обоснование выбора компонентов и технологий для реализации, руководство по развёртыванию системы, результаты тестирования разработанной системы. Курсовой проект содержит рисунки, диаграммы и таблицы.

-Abstract Course project on the topic: Electronic learning tool for the subject "Belarusian language" Performed - Cherekhovich A.Yu. Head - Mikhalkevich A.V. The course project consists of an explanatory note, a graphic part, a software product code. The explanatory note includes the following sections - introduction, description of the subject area, description of the main processes of the subject area, specification of the options for using the system, information model of the system and its description, justification for the selection of components and technologies for implementation, guidance on the deployment of the system, test results of the developed system. Course project contains drawings, diagrams and tables.

Содержание

[Введение](#)

[1 Система контроля версий](#)

[2 Описание предметной области](#)

[3 Описание основных процессов предметной области](#)

[4 Спецификация вариантов использования системы](#)

[5 Информационная модель системы и её описание](#)

[6 Обоснование выбора компонентов и технологий для реализации](#)

[7 Руководство по развёртыванию системы](#)

[8 Результаты тестирования разработанной системы, и оценка выполнения задач](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

В настоящее время в век информационных технологий все большую роль в обучении играют различные программы, как при самостоятельном изучении, так и при совместном изучении с учителем. Компьютерные обучающие программы составляют обширный класс средств, относящихся к образовательным. На сегодняшний день о них обеспечивают поддержку учебного процесса наравне с традиционными учебно-методическими средствами. Однако, по сравнению с традиционными учебно-методическими средствами они обеспечивают новые возможности, а многие существующие функции реализуются с более высоким качеством. Они являются базой для развития такой технологии обучения как дистанционное образование. Дистанционное образование невозможно без использования данных программных продуктов, так как именно они позволяют «удаленным» пользователям получать информацию, выполнять практические и лабораторные занятия, проходить тестирование и сдавать экзамены, не покидая места своего основного проживания. При организации дистанционного обучения предусматривается использование всех возможностей Интернет, включая видео и аудиоконференции, а также использование CD-дисков. Такая организация дистанционного обучения несет в себе большие дидактические возможности как для системы вузовского образования, так и для системы повышения квалификации. Основной задачей являлась разработка программного средства, которое поможет в организации учебного процесса по обучению белорусскому языку учеников. Пользователям будет предоставляться удобный и легкоусваиваемый интерфейс. Также в программе будут реализованы функции прохождения заданий для самоконтроля. Появится возможность быстрого и удобного изучения теоретического материала и закрепление его.

1 Система контроля версий

GitHub — крупнейший [веб-сервис](#) для хостинга [IT-проектов](#) и их совместной разработки. Основан на системе контроля версий [Git](#) и разработан на [Ruby on Rails](#) и [Erlang](#) компанией GitHub, Inc.

Сервис абсолютно бесплатен для проектов с [открытым исходным кодом](#) и предоставляет им все возможности (включая SSL), а для частных проектов предлагаются различные платные тарифные планы.

8.1 Установка

В начале необходимо скачать exe-файл инсталлятора со страницы проекта на GitHub'e и запустите его:

<http://msysgit.github.com/>

После установки будет как консольная версия (включающая SSH-клиент), так и стандартная графическая.

После установки Git'a, необходимо указать имя и адрес электронной почты. Это важно, потому что каждый коммит в Git'e содержит эту информацию, и она включена в коммиты, передаваемые вами, и не может быть далее изменена.

Далее необходимо произвести первоначальные настройки. В состав Git'a входит утилита `git config`, которая позволяет просматривать и устанавливать параметры, контролирующие все аспекты работы Git'a и его внешний вид.

8.2 Отправка данных в репозиторий GitHub

Для начала создаем личный кабинет на сайте github.com. Создаем репозиторий в котором, в дальнейшем будет располагаться наш проект (Рисунок 8.1).

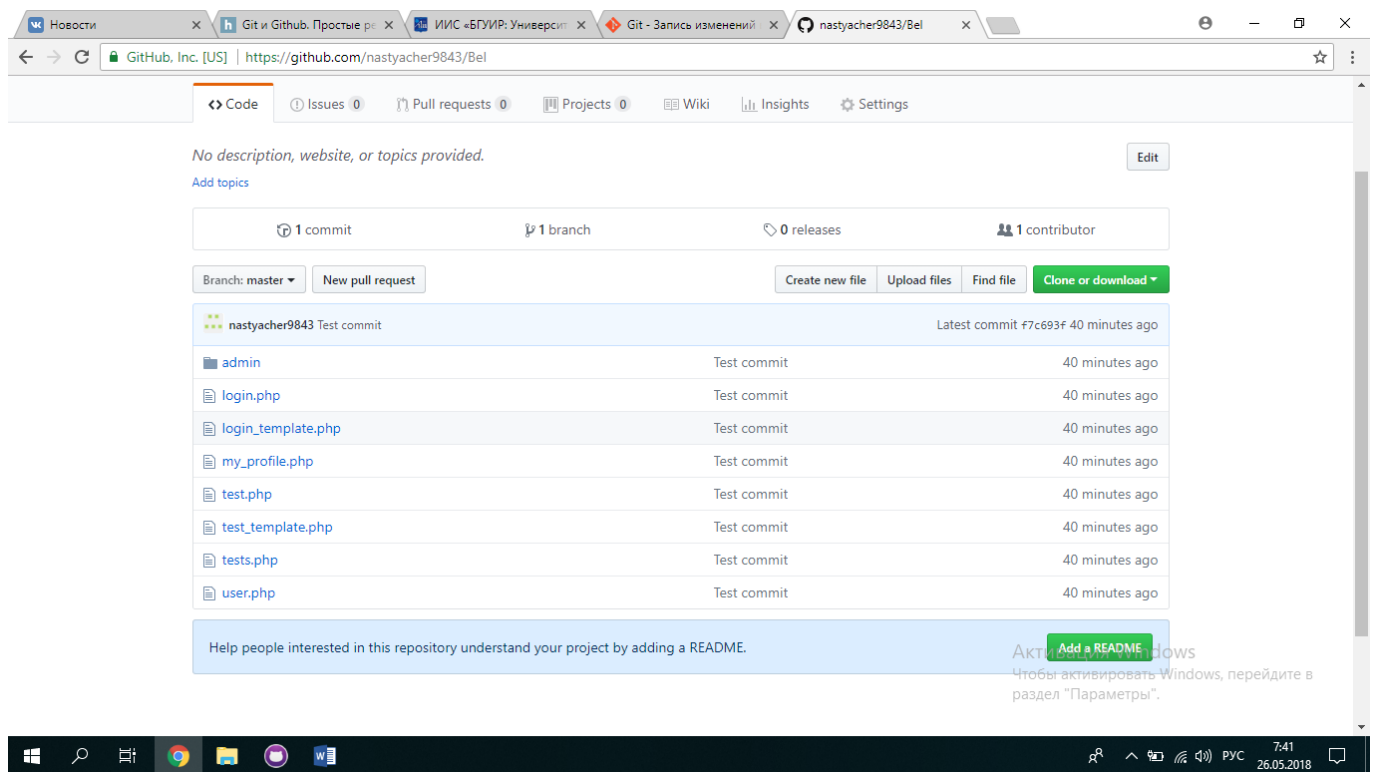


Рис 8.1 – Личный кабинет *GitHub*

Перечень команд для отправки данных на *GitHub*:

1. Переходим в папку с нашим проектом и инициализируем *Git*.

```
git init
```

1. Добавляем все файлы в *Git*.

```
git add .
```

1. Комментируем нашу версию.

```
git commit -m "Init"
```

1. Для связи с *GitHub*'ом следует указать удаленный репозиторий:

```
git remote add origin https://github.com/Maxirus2/oldinclone.git
```

- a. *push -u origin master*

Проект успешно добавлен, для его просмотра переходим по данной ссылке:
<https://github.com/nastyacher9843/Bel>

2 Описание предметной области

В рамках курсового проектирования предлагается разработка программного средства, которое поможет учителям, а также ученикам автоматизировать процесс обучения белорусскому языку.

Одна из наиболее актуальных проблем компьютерного обучения - проблема создания педагогически целесообразных программ различного учебного назначения, в частности, обучающих программ. В этом случае компьютер становится техническим средством обучения.

Процесс обучения весьма сложен по своему содержанию и формам представления информации. Существует множество методов и, соответственно, средств обучения. Состав последних зависит от уровня развития науки и техники. Персональные компьютеры являются универсальными средствами обучения, сочетающими в себе множество преимуществ.

Обучающая программа призвана значительно упрощать работу учителя, давать возможность ученику самостоятельно готовиться к занятиям. Ученику приходится не только изучать теоретический материал, но и учиться применять его на практике [2].

Электронный учебник предназначен для индивидуального обучения и реализует следующие функции:

ознакомительная (информационная) - пользователь может прочесть изучаемый материал;
самоконтроль - пользователь может вернуться к изученному материалу исходя из возможности самостоятельного ответа на контрольные вопросы.

В настоящее время нет единой классификации обучающих программ, хотя многие авторы выделяют среди них следующие пять типов:

программы тренировочного типа предназначены преимущественно для закрепления умений и навыков. Предполагается, что теоретический материал уже усвоен. Они широко используются для обработки математических навыков, упражнений по переводу с иностранного языка. Многие из них составлены в духе бихевиоризма, когда за один из ведущих принципов берется подкрепление правильного ответа. Компьютер в случайной последовательности генерирует учебные задачи, уровень трудности которых определяется учителем. Если ученик дал правильное решение, ему сообщается об этом. Если ответ неправильный, ему либо предъявляется правильный ответ, либо предоставляют возможность запросить помощь. Программы этого типа реализуют обучение, мало чем отличающееся от программированного обучения с помощью простейших технических средств, то есть управление учебным процессом осуществляется по ответу;

программы наставнического типа ориентированы преимущественно на усвоение новых понятий, многие из них также работают в режиме, близком к программированному обучению с разветвленной программой. И хотя в этих программах после предъявления информации обучаемому задаются вопросы, то есть обучение ведется в форме диалога, но при этом диалог построен на основе формального преобразования ответа обучаемого, создавая лишь видимость общения;

программы, направленные на организацию проблемного обучения, построены в основном на идеях и принципах когнитивной психологии, в них осуществляется не прямое управление деятельностью ученика. Это значит, что предъявляются разнообразные задачи и ученики побуждаются решать их путем проб и ошибок. Особенно полезны программы, моделирующие и анализирующие конкретные ситуации, так как они

способствуют формированию умения принимать решения при различных обстоятельствах;

в программах имитационного и моделирующего типа осуществляется предъявление заданий, требующих от обучаемого воспроизведения последовательности рассуждений или «сборки» правильного результата на основе знаний, предоставленных системой, например, программы для построения рисунков на экране компьютера; программы игрового типа способствуют формированию мотивации учения, стимулируют инициативу и творческое мышление, развивают умение совместно действовать, подчинять свои интересы общим целям. Игра позволяет выйти за рамки определенного учебного предмета, побуждая учеников к приобретению знаний в смежных областях и практической деятельности.

3 Описание основных процессов предметной области

Разработка электронного средства обучения — процесс довольно трудоёмкий, но доступность и интерактивность изложения материала, интерес учеников к информационным технологиям приводят к повышению эффективности и качества образовательного процесса.

В сфере обучения существует множество программных средств, помогающих ученикам лучше усваивать материал, а также проверять свои знания. Можно выделить некоторые из них:

интернет-приложение Effor.by;
интернет-приложение GetAClass;
интернет-приложение «Мая родная мова» .

Интернет-приложение Effor.by, главная страница которого представлена на рисунке 2.1 является комплексом информационных ресурсов и сервисов, мобильных приложений для обучения современных школьников.

На данном сайте представлено множество предметов: русский язык, белорусский язык, математика, физика, информатика. Есть возможность прохождения тестов по каждому предмету.

Также существуют различные разделы для каждого пользователя: учеников, родителей или учителей.

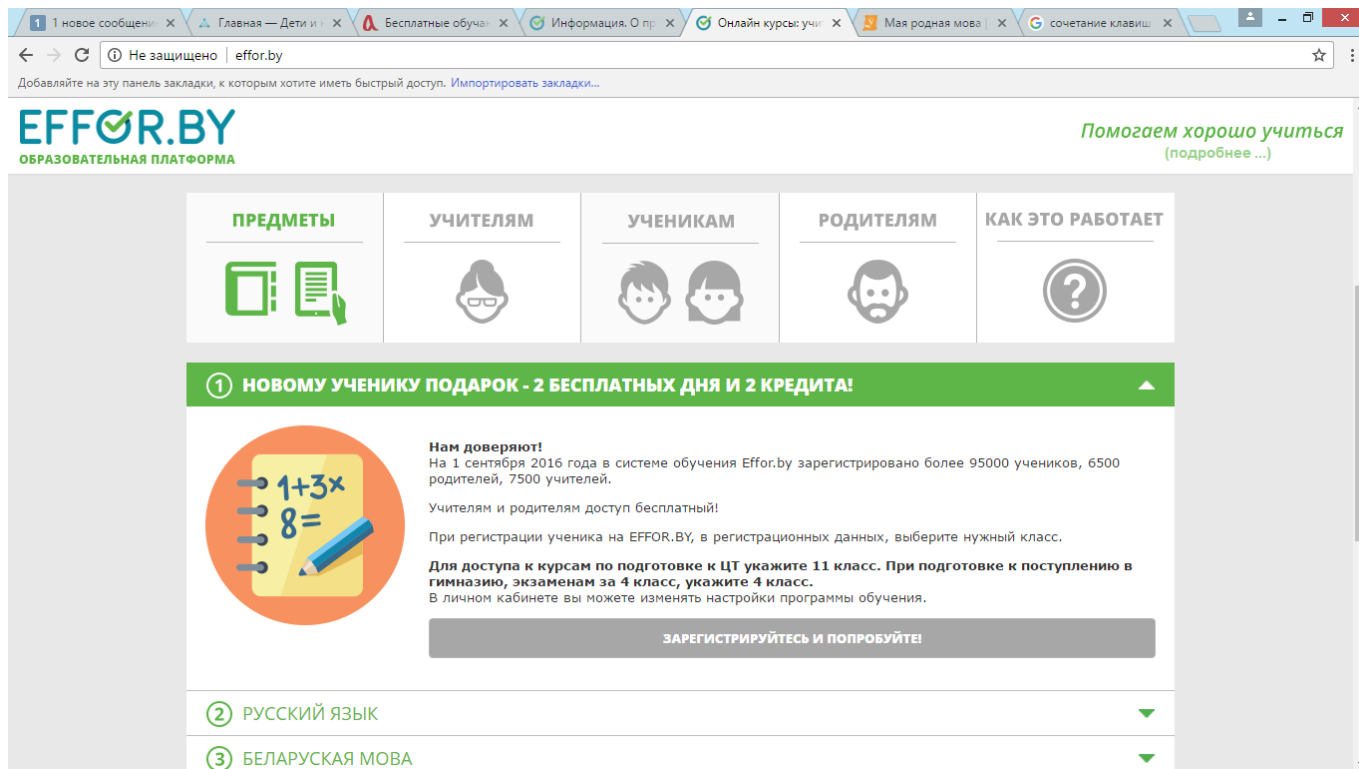


Рисунок 2.1 – Главная страница интернет-приложения Effor.by

Достоинства приложения:

- наличие нескольких предметов;
- возможность работы по категориям;
- удобное тестирование;
- возможность выбора заданий;

Недостатки приложения:

- пользователям полная версия предоставляется только в течении 7 дней, после чего необходима оплата, для дальнейшей работы;
- достаточно непривлекательный интерфейс;
- нет возможности сохранения результатов самоконтроля;
- отсутствует теоретический материал.

Портал GetAClass – это образовательный ресурс по физике и математике для школьников и учителей.

Школьникам GetAClass предлагает видеоуроки по физике и математике. Для учителей на GetAClass есть видеоуроки и другие методические материалы по физике и математике, которые можно использовать для подготовки и проведения уроков в школе.

Достоинства приложения:

- возможность работы нескольких режимах;
- возможность проверки знаний;

- наличие конспектов для учителей;
- наличие видеоуроков.

Недостатки приложения:

работа с приложением платная;
отсутствие теоретической части для школьников;
неудобная навигация;
непривлекательный интерфейс.

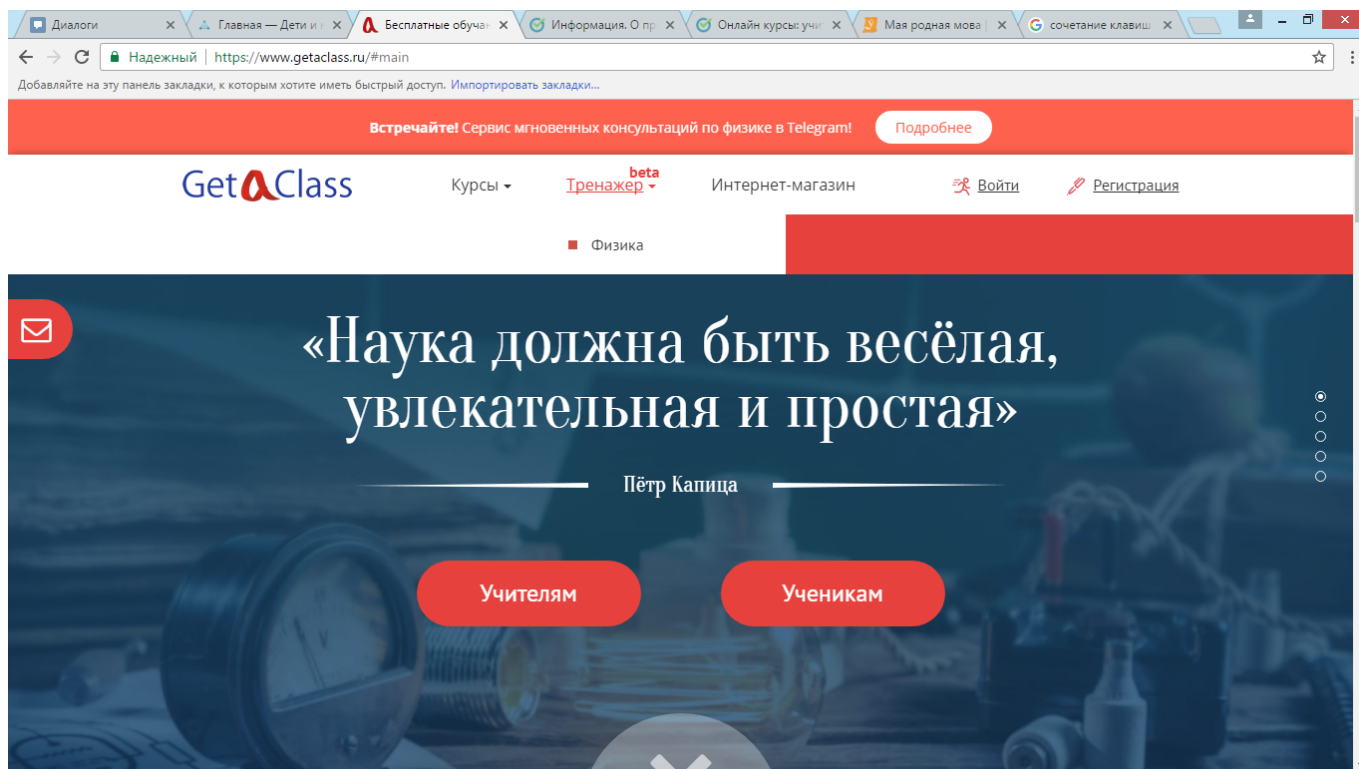


Рисунок 2.2 – Главная страница интернет-приложения “GetAClass”

Интернет-приложение «Мая родная мова», его главная страница представлена на рисунке 2.3, предназначено для контроля знаний пользователей.

В каждом задании нужно выбирать один из трех предложенных вариантов. После ответа на все вопросы, ошибки будут показаны красным цветом, правильный ответ – зеленым.



Рисунок 2.3 – Интернет-приложение «Мая родная мова»

На главной странице задания случайно выбираются из базы. Есть возможность обновлять их снова и снова, отвечая на вопрос или пропуская задание.

До авторизации результаты не сохраняются, а после авторизации пользователь получает доступ ко всем сервисам сайта.

Достоинства приложения:

- существует система самоконтроля;
- возможность проведения турнира;
- наличие сохранения результатов.

Недостатки приложения:

- задания для самоконтроля неструктурированные;
- нет возможности прохождения данного теста второй раз;
- сохранение только последних результатов;
- нет теоретического материала;
- неудобный интерфейс.

Таким образом проведя обзор обучающих программ, можно сделать вывод, что несмотря на всё многообразие электронных средств обучения, многие из них обладают существенными недостатками, такими как платность, отсутствие лекционного материала, отсутствие выбора тем, сложный и неудобный интерфейс.

Рассмотрев все достоинства программного средства, можно сделать вывод о том, что программа будет разработана качественно и актуально среди пользователей.

4 Спецификация вариантов использования системы

Визуальное моделирование в UML можно представить, как процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей вначале строится модель в форме так называемой диаграммы вариантов использования, которая описывает функциональное назначение системы.

Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Диаграммы вариантов использования определяют общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы; формулируют общие требования к функциональному поведению проектируемой системы

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик.

В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

В качестве актера данной диаграммы выступает два субъекта: администратор и пользователь. Рассмотрим построение диаграммы со стороны актера-пользователя. Данный актер взаимодействует с рассматриваемым приложением и является его пользователем, то есть он обращается к соответствующему действию «Работать с электронным средством обучения по предмету “Белорусский язык”». Между актером и вариантом использования «Запустить приложение» возникает отношение ассоциации. Графическое представление отношения ассоциации между актером и вариантом использования представлено на рисунке 3.1

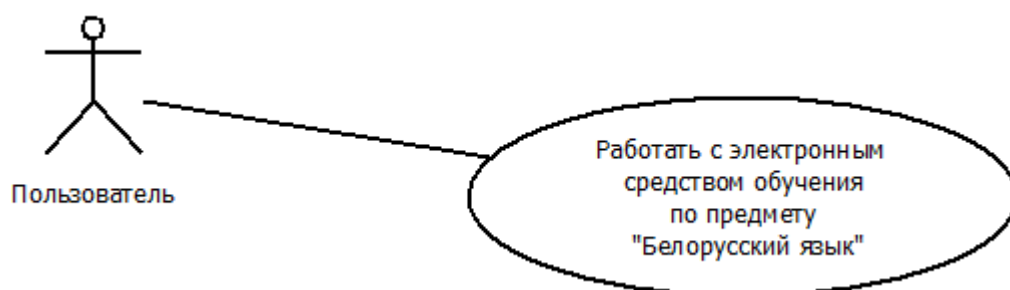


Рисунок 3.1 — Графическое представление отношения между актером и вариантом использования

На следующем этапе разработки данной диаграммы вариант использования «Работать с электронным средством обучения по предмету “Белорусский язык”» может быть уточнен на основе введения в рассмотрение двух дополнительных вариантов использования. Это позволяет выделить такие действия, как «Изучить лекционный материал» и «Пройти задания для самоконтроля». Вполне очевидно, что указанные действия раскрывают поведение исходного варианта использования в смысле его конкретизации, и поэтому между ними будет иметь место отношение включения. Уточнение варианта использования «Запустить приложение» представлено на рисунке 3.2.



Рисунок 3.2 — Уточнение варианта использования

Затем «Пройти задания для самоконтроля» включает в себя дополнительные варианты использования «Отобразить задания», а также «Получить результат» и «Сохранить результат». Вариант использования «Отобразить задания» связан с родительским вариантом использованием отношением включения, а «Получить результат» и «Сохранить результат» связаны отношением расширения. Данный фрагмент диаграммы представлен на рисунке 3.3.

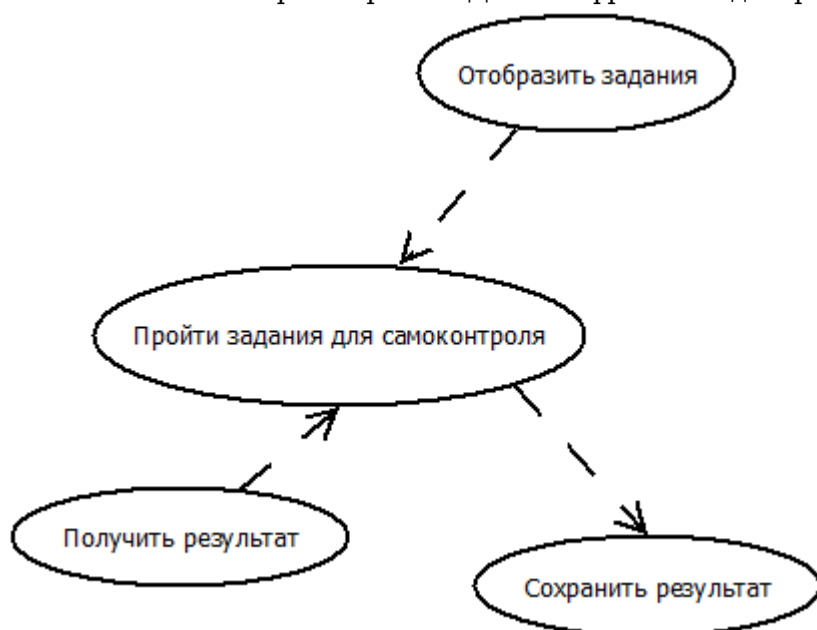


Рисунок 3.3 - Варианты использования раздела
«Пройти задания для самоконтроля»

При разработке данной диаграммы вариант использования «Изучить лекционный материал» может быть расширен на основе введения «Отобразить лекционный материал». Уточнение варианта использования представлено на рисунке 3.4.



Рисунок 3.4 — Графическое изображение отношения расширения

В свою очередь между администратором, который обращается к действию «Работать с

электронным средством обучения по предмету «Белорусский язык», возникает отношение ассоциации. Это же отношение связывает администратора с таким действием, как «Редактировать данные». «Редактировать данные» включает в себя варианты использования «Изменить данные», «Добавить данные» и «Сохранить данные», все они связаны с родительским вариантом использования отношением включения. Данный фрагмент диаграммы представлен на рисунке 3.5

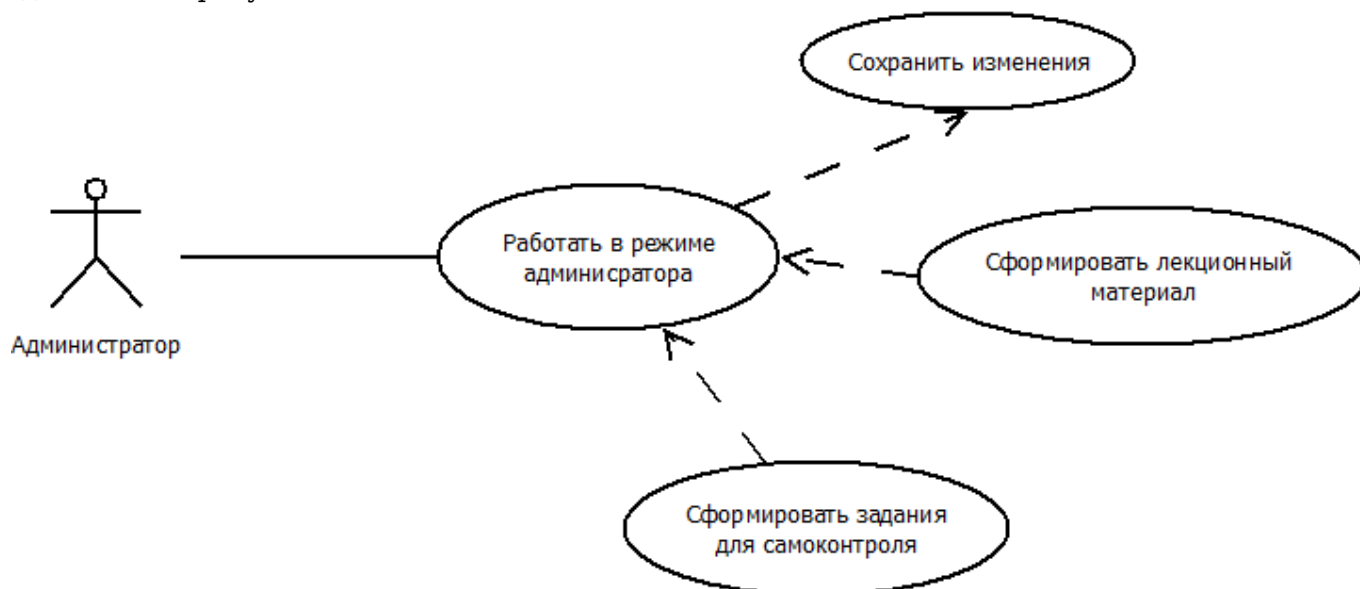


Рисунок 3.5 — Графическое изображение вариантов использования программы в режиме администратора

В диаграмме создано множество вариантов использования, которые расширяют действия над программным продуктом или дополняют предыдущие.

5 Информационная модель системы и её описание

Диаграммы сущность-связь позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей и предлагают некоторый набор стандартных обозначений для определения данных и отношений между ними.

Диаграммы сущность-связь позволяют использовать наглядные графические обозначения для моделирования сущностей и их взаимосвязей.

С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

Первым шагом является извлечение информации из описания предметной области и выделение сущностей. Сущность – это класс однотипных объектов, информация о которых должна быть учтена в модели.

Исходя из описания предметной области, можно выделить некоторые сущности:

- пользователь;
- вопросы;
- ответы;
- тест;
- результаты пользователя;

результат прохождения теста.

Следующим шагом моделирования является идентификация атрибутов. Каждая сущность имеет свои атрибуты.

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности.

Далее идёт выявление ключа сущности. Ключ сущности - это избыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Избыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Сущность может иметь несколько различных ключей.

Ключевые атрибуты изображаются на диаграмме подчеркиванием

Данные о сущностях и их атрибутах представлены на рисунке 4.1.

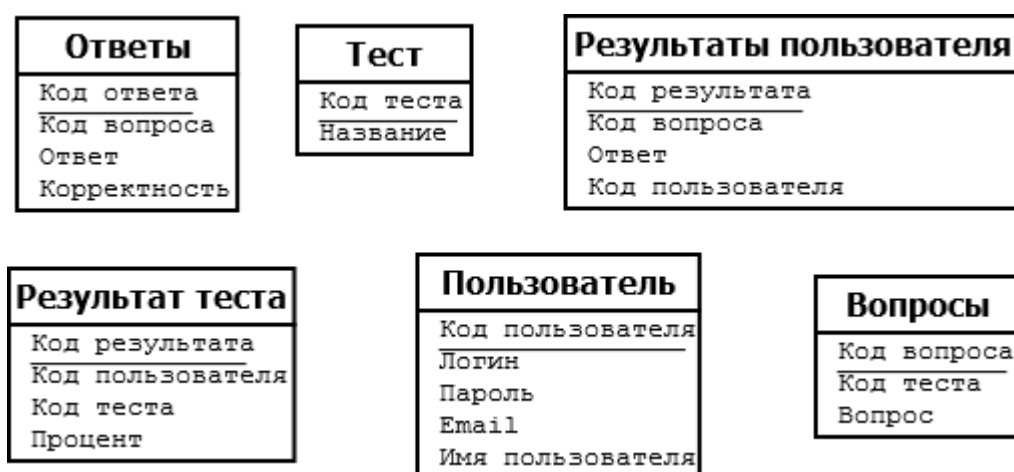


Рисунок 4.1 - Перечень сущностей

Следующим шагом является идентификация связей.

Связь - это некоторая ассоциация между двумя сущностями. Связи позволяют по одной сущности находить другие сущности, связанные с нею. Графически связь изображается линией, соединяющей две сущности.

Каждая связь может иметь один из трёх типов связи:

- один к одному;
- один ко многим;
- многие ко многим.

Связь один-ко-многим — это, когда одной записи в таблице соответствует несколько записей в связанной таблице.

Связь много-к-одному — это, когда несколькими записями в таблице соответствует одна в связанной таблице.

Связь много-ко-многим — когда несколькими записями в таблице соответствует несколько записей в связанной таблице.

В соответствии с диаграммой, приведённой на рисунке 4.2, пользователь может иметь несколько результатов, тест может содержать несколько вопросов, ответов. Ответ, в свою

очередь должен относиться к конкретному тесту.

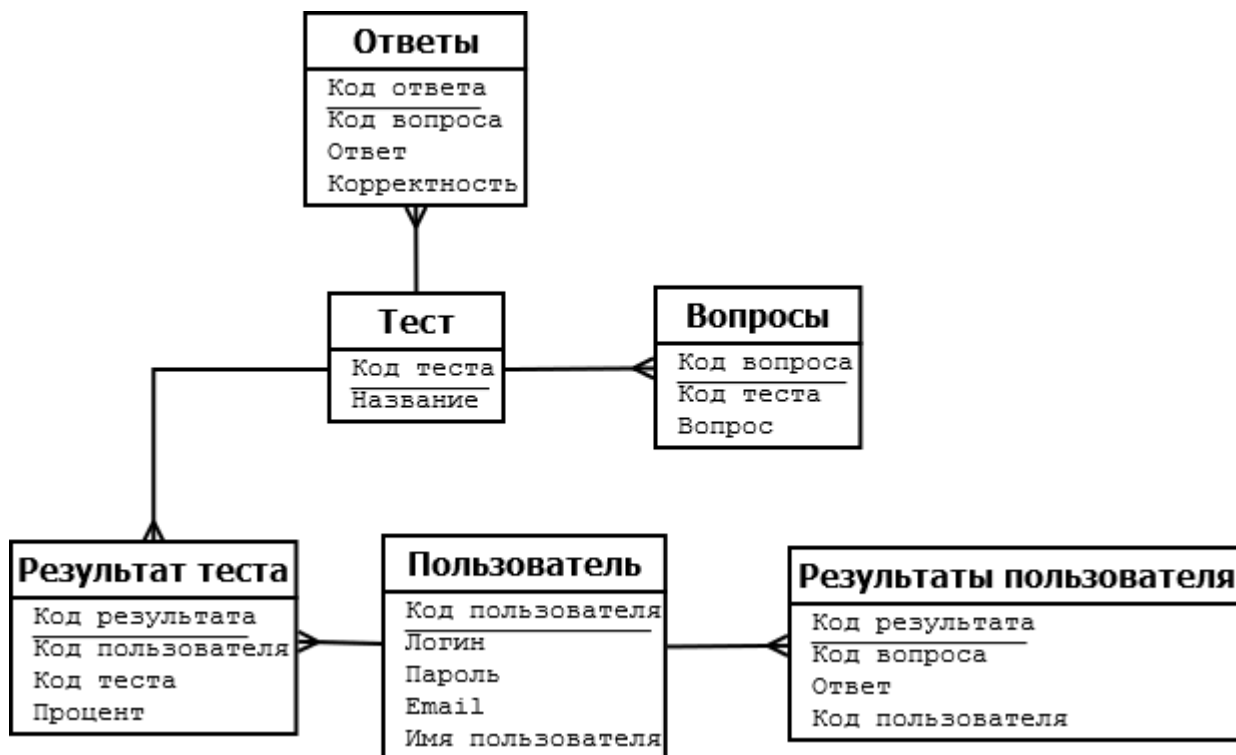


Рисунок 4.2 - Типы связей

Кроме того нужно определить модальность. Каждая связь может иметь одну из двух модальностей:

может;
должен.

Модальность "может" означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность "должен" означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Диаграмма "Сущность-связь" приведена в приложении А.

ER-диаграммы удобны тем, что процесс выделения сущностей, атрибутов и связей является итерационным.

6 Обоснование выбора компонентов и технологий для реализации

Для создания реляционной базы данных был использован такой непроцедурный язык, как SQL.

SQL символизирует собой Структурированный Язык Запросов. Это - язык, который дает возможность создавать и работать в реляционных базах данных, являющихся наборами связанной информации, сохраняемой в таблицах.

Язык был создан в 1970х годах под названием “SEQUEL” для системы управления базами данных (СУБД) System R.

Несмотря на существование стандартов, большинство распространенных реализаций SQL отличаются так сильно, что код редко может быть перенесен из одной СУБД в другую без внесения существенных изменений. Это объясняется большим объемом и сложностью стандарта, а также нехваткой в нем спецификаций в некоторых важных областях реализации.

SQL создавался как простой стандартизированный способ извлечения и управления данными, содержащимися в реляционной базе данных. Позднее он стал сложнее, чем задумывался, и превратился в инструмент разработчика, а не конечного пользователя.

Информационное пространство становится более унифицированным. Это привело к необходимости создания стандартного языка, который мог бы использоваться в большом количестве различных видов компьютерных сред.

Такой язык был выбран исходя из следующих его преимуществ:

- независимость от конкретной СУБД. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД;

- наличие стандартов. Наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка.;

- декларативность. С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Также у SQL есть и свои недостатки: несоответствие реляционной модели данных;

- сложность. Хотя SQL и задумывался как средство работы конечного пользователя, в конце концов он стал настолько сложным, что превратился в инструмент программиста;

- отступления от стандартов. Несмотря на наличие международного стандарта ANSI SQL-92, многие разработчики СУБД вносят изменения в язык SQL, применяемый в разрабатываемой СУБД, тем самым отступая от стандарта;

- сложность работы с иерархическими структурами. Ранее диалекты SQL большинства СУБД не предлагали способа манипуляции древовидными структурами.

SQL реализует декларативную парадигму программирования: каждый оператор описывает только необходимое действие, а СУБД принимает решение о том, как его выполнить, т.е. планирует элементарные операции, необходимые для выполнения действия и выполняет их.

Для администрирования данного проекта выбран PHPMyAdmin. Этот очень популярный, с большими возможностями и в тоже время простой в использовании бесплатный инструмент, написанный на PHP, он был создан для разработки и администратии MySQL. Он может создавать и удалять базы данных, создавать/удалять/изменять таблицы, удалять/редактировать/добавлять поля, выполнять любые SQL-команды, управлять пользователями и разрешениями для них.

PHPMyAdmin позволяет через [браузер](#) и не только осуществлять администрирование сервера MySQL, запускать команды [SQL](#) и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет

управлять [СУБД MySQL](#) без непосредственного ввода SQL команд, предоставляя дружелюбный интерфейс.

На сегодняшний день PHPMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения [СУБД MySQL](#). Подавляющее большинство российских [провайдеров](#) используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

В качестве альтернативы также использовался такой инструмент для работы с SQL, как HeidiSQL.

HeidiSQL - это легкий и быстрый клиент под Windows для управления MySQL базами данных. HeidiSQL позволяет просматривать и редактировать данные, создавать и редактировать таблицы, в общем тотальный контроль за базами данных созданных вами сайтов.

Для людей которые занимаются созданием сайтов программа просто незаменима.

У HeidiSQL множество различных возможностей:

- подключение к нескольким серверам в одном окне;
- подключение к серверам с помощью командной строки;
- подключение через туннель SSH, SSL или передать настройки;
- создание и редактирование таблиц, представлений, хранимых процедур, триггеров и запланированные события;
- создание дампов SQL;
- экспорт с одного сервера или базы данных непосредственно на другой сервер или база данных;
- управление пользователями и привилегиями;
- импорт текстовых файлов;
- экспорт в строковые таблицы: CSV, HTML, XML, SQL, LaTeX, вики-разметки и PHP массивы;
- просмотр и редактирование таблиц, данные с помощью удобной сетки;
- массовое изменение таблиц ;
- пакетная вставка ASCII или бинарных файлов в таблицы;
- онлайн редактор запросов с настраиваемой подсветкой синтаксиса и автозавершением кода;
- мониторинг и закрытие клиентских-процессов.

В качестве языка для разработки приложения выбран язык программирования для создания web-приложений php .

PHP — [скриптовый язык](#) общего назначения, интенсивно применяемый для разработки веб-приложений.

Аббревиатура PHP означает "Hypertext Preprocessor (Препроцессор Гипертекста)". Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц. Подробнее о преимуществах PHP можно узнать [здесь](#).

Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP.

Если сравнивать его с другими языками, применяемыми в веб-разработке, к примеру, с Python, то можно выделить следующие достоинства выбранного языка и его отличия от сравниваемого.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера [3].

Есть возможность сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

PHP прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

Даже если пользователь впервые услышал о PHP, изучить этот язык не составит для Вас большого труда..

Язык PHP постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков web -программирования, по крайней мере, в ближайшее время.

Преимущества PHP:

- является свободным программным обеспечением, распространяемым под особой лицензией;
- легок в освоении на всех этапах;
- поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может использоваться в изолированной среде;
- предлагает альтернативные средства организации веб-сессий, программный интерфейс расширений;
- является довольно полной заменой проприетарной среды ASP;
- может быть развёрнут почти на любом сервере;
- портирован под большое количество аппаратных платформ и операционных систем .

Недостатки PHP:

- не подходит для создания десктопных приложений или системных компонентов;
- глобальные параметры конфигурации влияют на базовый синтаксис языка, что затрудняет настройку сервера и разворачивание приложений;
- веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью .

Также PHP часто сравнивают с не менее популярным ASP.NET.

ASP.NET — технология создания [веб-приложений](#) и [веб-сервисов](#) от компании [Майкрософт](#).

Преимущества ASP.NET:

- реализация объектной модели приложения;
- выполняемый код;

поддержка технологии Microsoft.NET встроена в ОС Microsoft;
встроенная поддержка технологии Ajax – любой элемент формы может выполнен в «классическом» варианте или с помощью Ajax.

Недостатки ASP.NET:

для работы ASP.NET приложение нужно лицензионное программное обеспечение.
шаблонизация заложена в основу;

Изучив все преимущества и недостатки данных языков появилась возможность создания сравнительной характеристики , которая приведена в таблице 5.1

Таблица 5.1 –Сравнительная характеристика

Область сравнения	PHP	ASP.NET
Базы данных	В PHP есть возможность работы с Microsoft SQL Server. Преимуществом MySQL может казаться бесплатность.	Чаще всего ASP.Net-приложения используют Microsoft SQL Server. Продукты подобного на phpMyAdmin для Microsoft SQL
Безопасность	В PHP, в случае обнаружения некорректных данных никаких предупреждений программист не видит.	Большинство случаев взлома веб-сайтов происходит из-за ошибок разработчиков.

Продолжение таблицы 5.1

Область сравнения	PHP	ASP.NET
Скорость	С каждым годом этот скрытый язык все быстрее в том числе из-за возможности ОС быстрее читать данные из файлов. Даже неподвинутый пользователь сможет быстро запустить и под любой ОС.	Скорость исполнения зависит от скорости обращения машинного кода к такому же коду компонентов, которые через кодовые функции фреймворка обрабатывается.
Разработка	PHP не требует особых знаний и тем более глубоких знаний ни в одной области .	ASP требует некоторого обязательного компилятора и некоторых способов отладки
Проекты	Считается что PHP является языком только лишь для небольших проетов, однако существует множество шаблонизаторов и фреймворков, которые приближают его к ASP.Net с точки зрения архитектуры приложения.	На ASP.Net. проще писать крупные прокты. Он использует общую платформу .Net.

Рассмотрев все характеристики данных языков, изучив все достоинства и недостатки, был

выбран язык PHP из-за некоторых причин. Во-первых веб-программированием сейчас интересуются многие, и первым средством разработки является именно он. Во-вторых, многие не хотят использовать продукт от Microsoft, полагая, что за это нужно платить немаленькие суммы. В-третьих, порог вхождения в PHP гораздо ниже, чем в ASP.Net, т.к. сам язык проще для изучения. Также PHP может работать как на Windows сервере, так и на UNIX, а вот ASP только на Windows.

В настоящее время PHP поддерживается подавляющим большинством [хостинг-провайдеров](#) и является одним из лидеров среди языков, применяющихся для создания [динамических веб-сайтов](#).

В качестве редактора для написания курсового проекта я выбрала Notepad plus plus.

Notepad plus plus — это редактор файлов HTML, который имеет подсветку синтаксиса. В действительности же у него имеется поддержка многих языков программирования и различной разметки, в том числе C++, php, css и прочее, соответственно это делает его прекрасным инструментом для любого программиста.

Из-за большого количества возможностей он имеет свои плюсы:

- notepad plus plus абсолютно бесплатен, что нельзя сказать про аналоги, такие как Dreamweaver и прочие. Отсутствие необходимости платить за что-либо всегда привлекала российского пользователя.
- имеет свой клиент FTP;
- текст разделяется на блоки;
- существует портативное приложение, оно легко помещается на обычную флешку;
- наличие русифицированной версии, которая делает работу с программой проще. Сегодня скачать notepad русскую версию не проблема;

Из минусов можно выделить:

- отсутствие кроссплатформенности, программа может работать только на Windows;
- программа не умеет отличать разные языки в рамках одного документа.

После изучения и сравнения всех инструментов, были взяты наиболее удобные в использовании. Данные инструменты были выбраны для реализации поставленных задач и дальнейшего использования электронного приложения по предмету «Белорусский язык».

7 Руководство по развертыванию системы

Программное средство «Электронное средство обучения по предмету «Белорусский язык»» предназначен для автоматизации учебного процесса учеников.

Для работы с данным программным средством не нужны практически никакие системные требования, кроме наличия выхода в интернет и соответственно интернет-браузера.

Интерфейс удобен и понятен в использовании. Компоненты страниц хорошо структурированы, нет ничего лишнего, но несмотря на малое количество элементов, они используются правильно и должны вызвать интерес у пользователя.

Цветовая гамма страницы выделяет наиболее важные части страницы, усиливает выразительность сайта, при этом не отвлекая пользователей от контента.

При загрузке сайта, первым делом открывается главная страница. На ней слева располагается меню с различными разделами сайта, также на данной странице есть возможность регистрации пользователя, перехода в личный кабинет и различные социальные сети.

На главной странице представлены основные цели и концепт данного проекта.

Главная страница изображена на рисунках 6.1 и 6.2

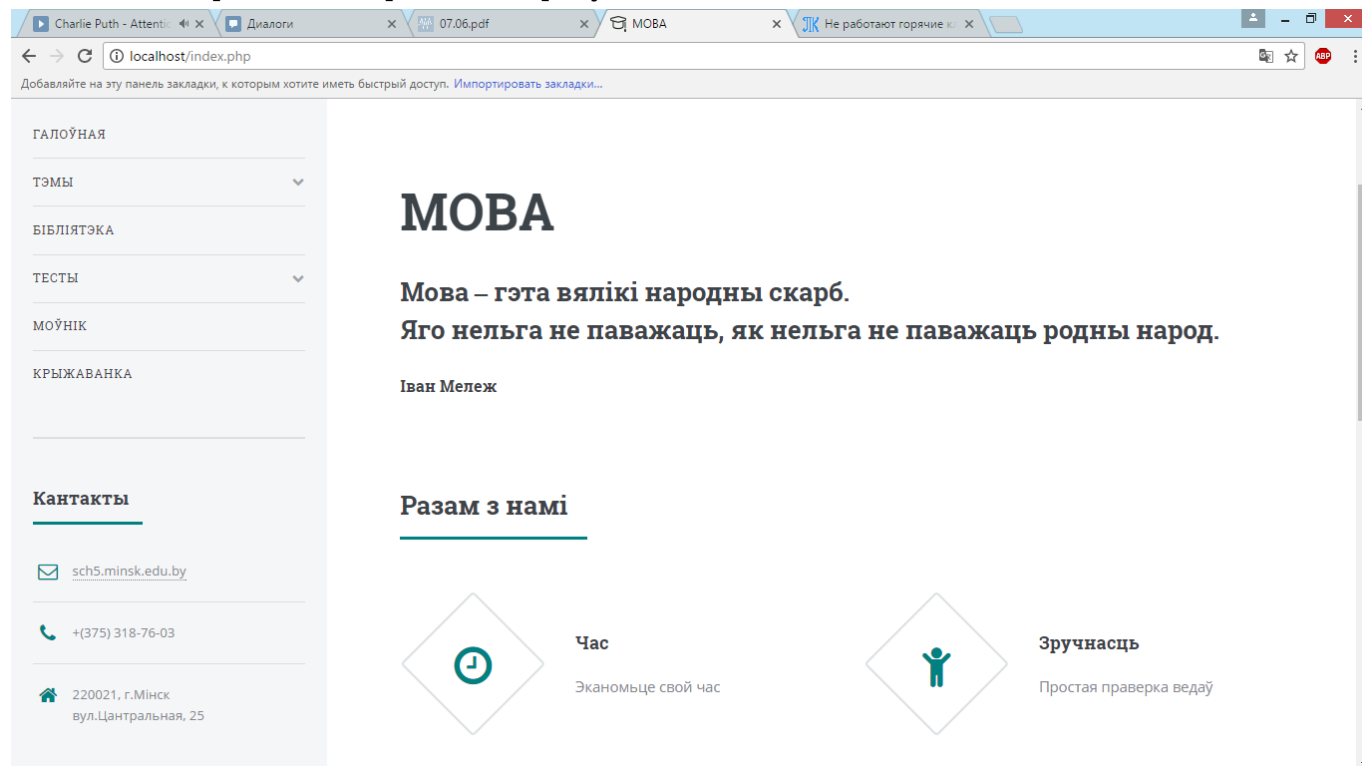


Рисунок 6.1 – Главная страница

Также на странице представлена контактная информация учреждения образования.

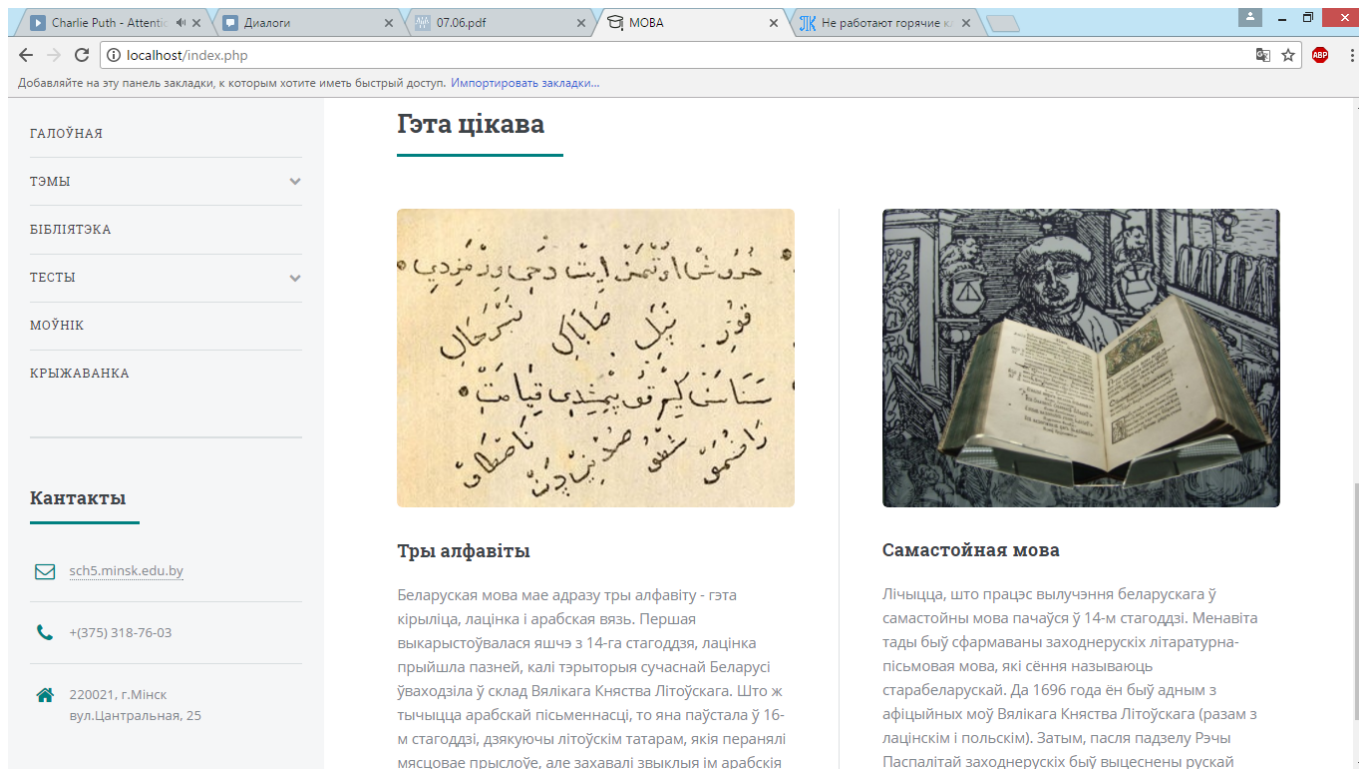


Рисунок 6.2 – Информация на главной странице

Пользователю предоставляется некоторый перечень разделов в пункте “Меню”, одним из которых является “Тэмы”. В данном пункте представлен лекционный материал, для самостоятельного изучения. Лекционный материал разделён на темы. Каждая тема представлена в удобном виде с использованием графического материала, для более быстрого усвоения материала. Данный раздел представлен на рисунке 6.3.

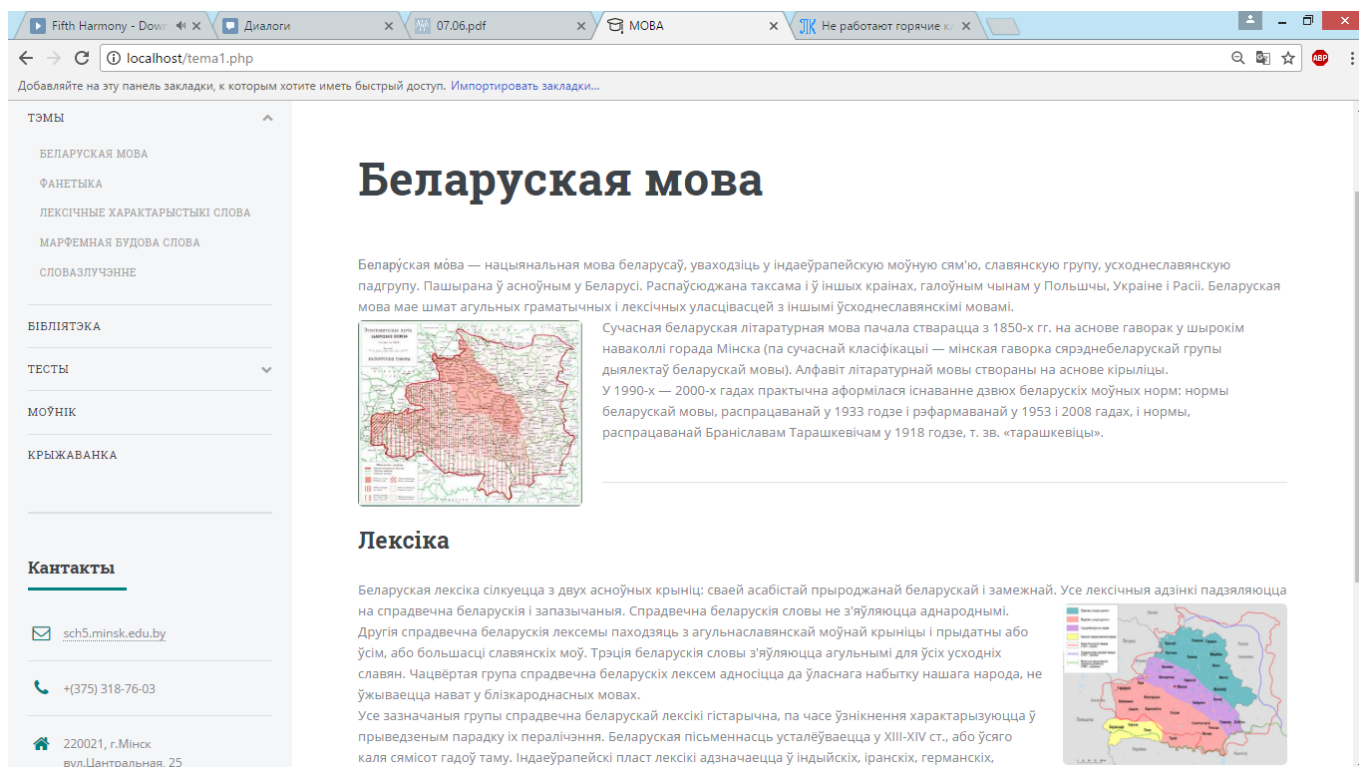


Рисунок 6.3 – Лекционный материал

В электронном приложении также представлен перечень литературы в разделе “Бібліятэка”. При входе на данную страницу отображается перечень книг с обложками. Данная страница представлена на рисунке 6.4.

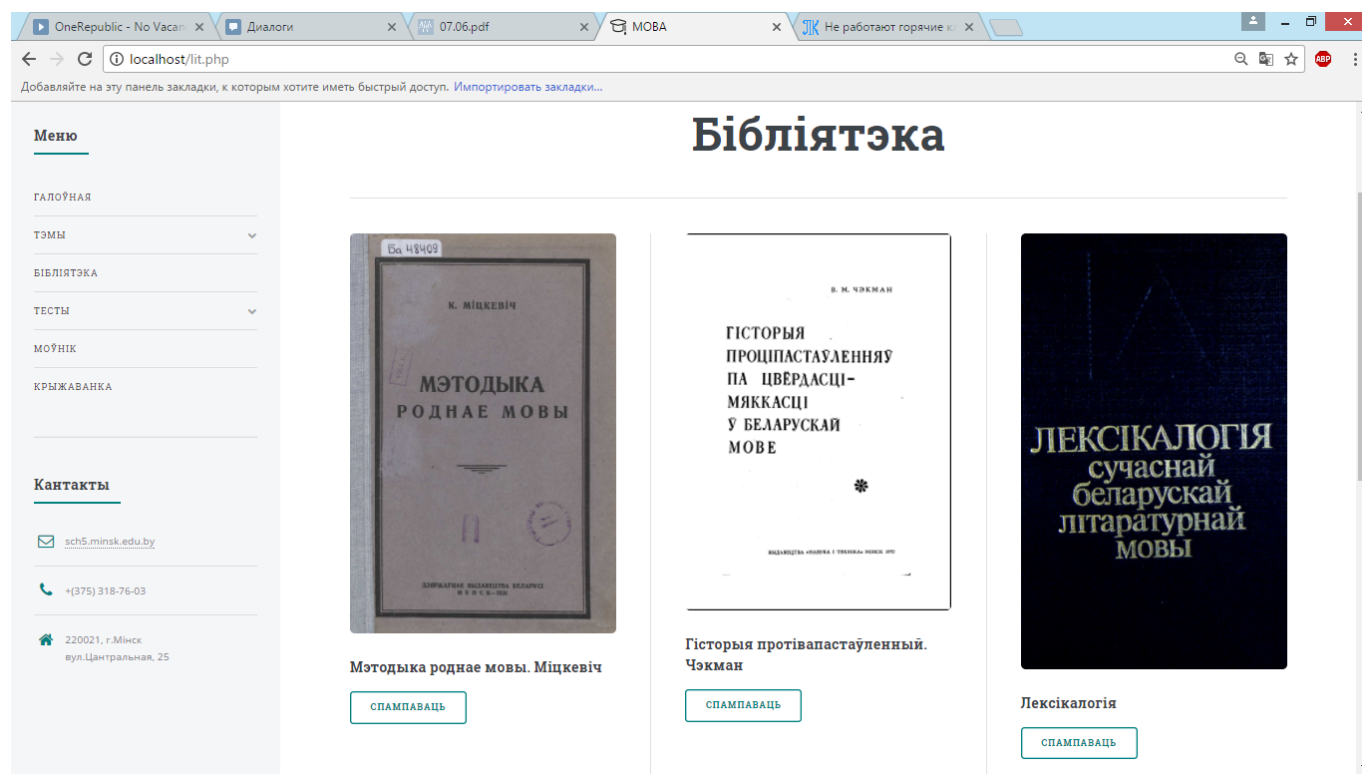


Рисунок 6.4 – Библиотека

Для того, чтобы скачать или просто просмотреть содержание книги, необходимо нажать на кнопку “Спампанаваць”, после чего в новой отдельной вкладке отобразиться книга, как показано на рисунке 6.5. У пользователя появится возможность просмотра книги в полноэкранном режиме и распечатки данного материала. А также будет возможность скачать книгу на свой компьютер, предварительно указав путь сохранения.

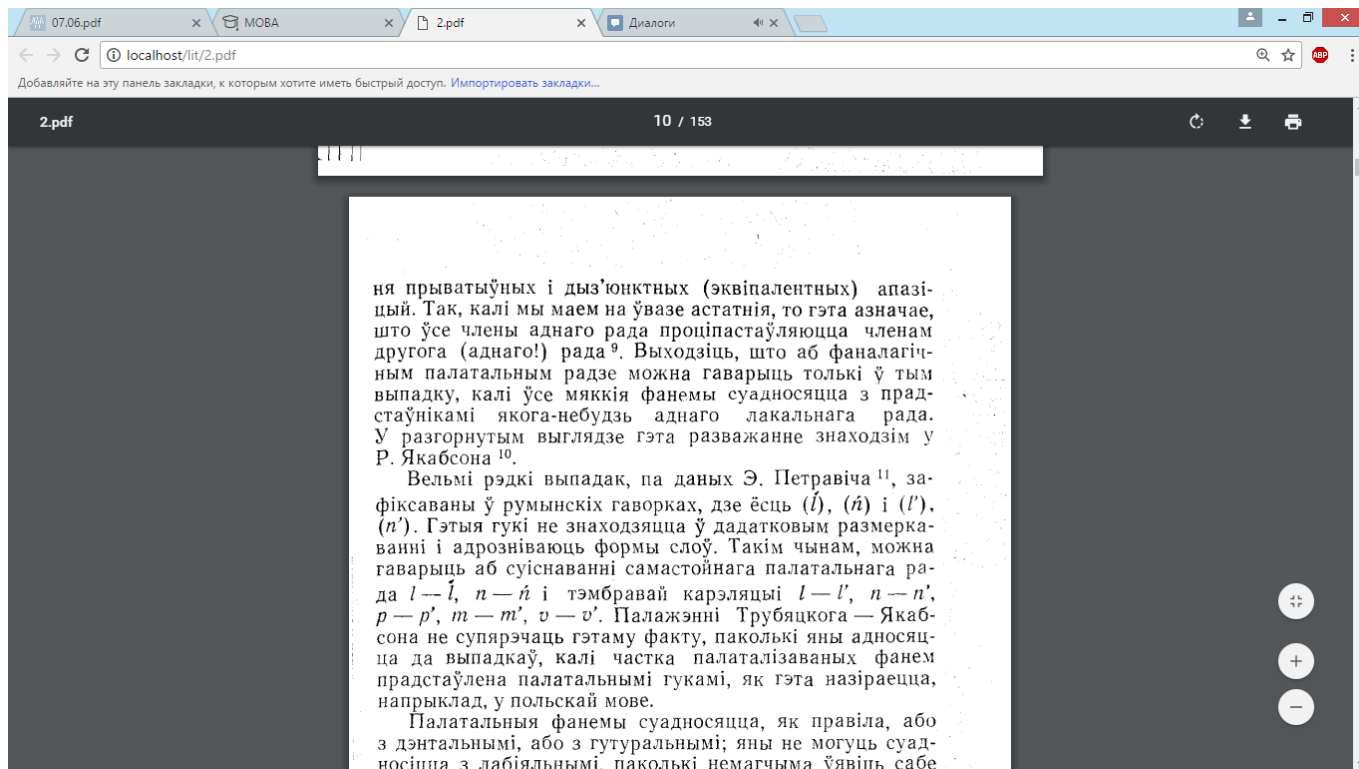


Рисунок 6.5 – Просмотр литературы

Также пользователю предоставляется возможность проверки своих знаний с помощью прохождения кроссворда. Пройти его можно в разделе “Крыжаванка” в меню. Процесс прохождения представлен на рисунке 6.6.

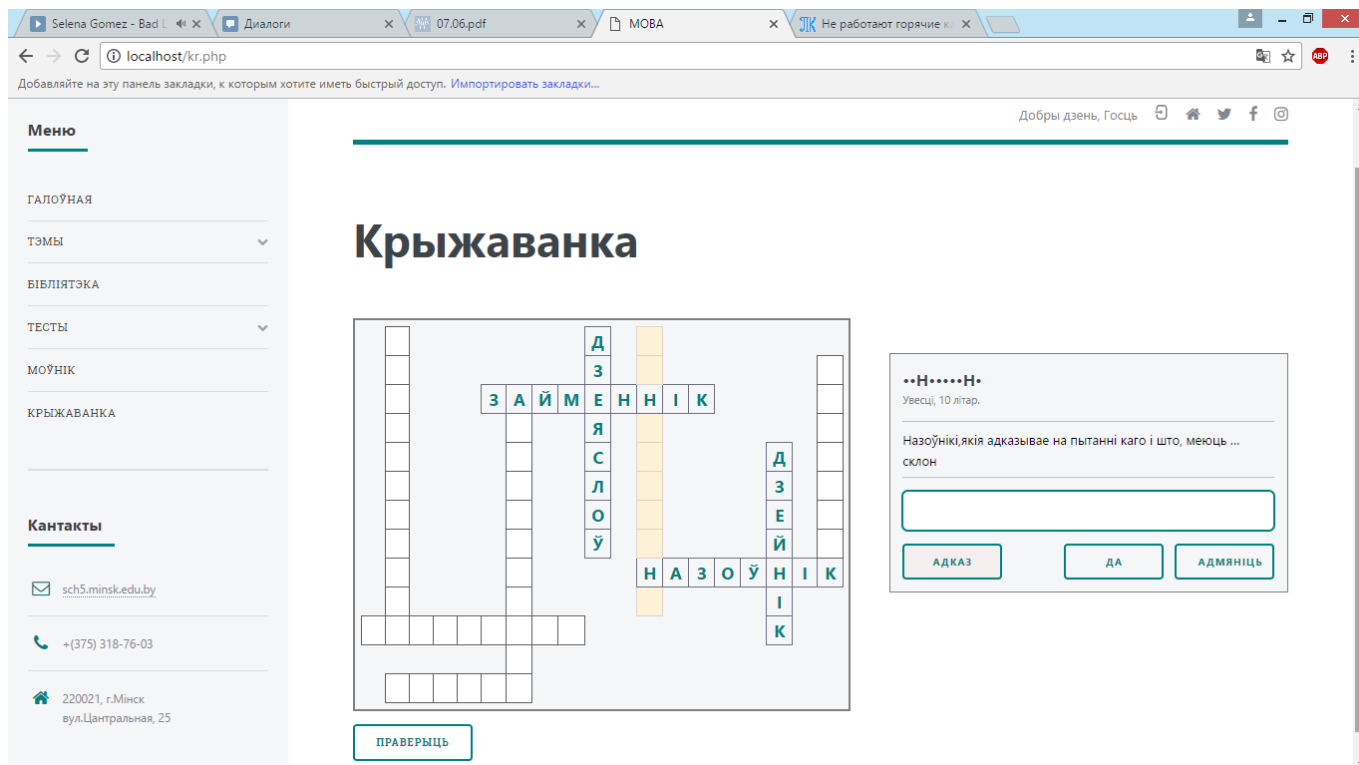


Рисунок 6.6 – Прохождение кроссворда

Главной проверкой знаний пользователя является прохождение теста, но оно доступно

только после регистрации. Чтобы зарегистрироваться, пользователь должен нажать на определённый значок в правом углу экрана либо зайти в раздел тестов, после этого ему необходимо ввести несколько параметров. Если пользователь уже был зарегистрирован, то ему необходимо авторизоваться, введя свое имя и пароль. На рисунке 6.7 изображено окно авторизации и регистрации.

The screenshot shows a web browser window with the address bar displaying 'localhost/login.php'. The page has a header with the text 'Добры дзень, Госць' and social media icons. The main content area is titled 'Рэгістрацыя або аўтарызацыя' (Registration or Authorization). Below the title, there are two columns of forms. The left column is for 'Аўтарызацыя' (Authorization) and contains fields for 'Логін' (Login) and 'Пароль' (Password), along with buttons 'УВАЙСЦІ' (Login) and 'СКІД' (Forgot). The right column is for 'Рэгістрацыя' (Registration) and contains fields for 'Логін' (Login), 'Імя' (Name), 'Пароль' (Password), 'Паўтарыце пароль' (Repeat password), and 'Email'. There is a checkbox labeled 'Я чалавек' (I am human) and buttons 'РЭГІСТРАЦЫЯ' (Register) and 'СКІД' (Forgot).

Рисунок 6.7 – Авторизация пользователя

После авторизации пользователь имеет возможность прохождения тестовых заданий. Для этого во вкладке “Тэсты” в меню необходимо выбрать нужную тему. После чего на экране появится первый вопрос с вариантами ответов. Данный процесс представлен на рисунке 6.8.

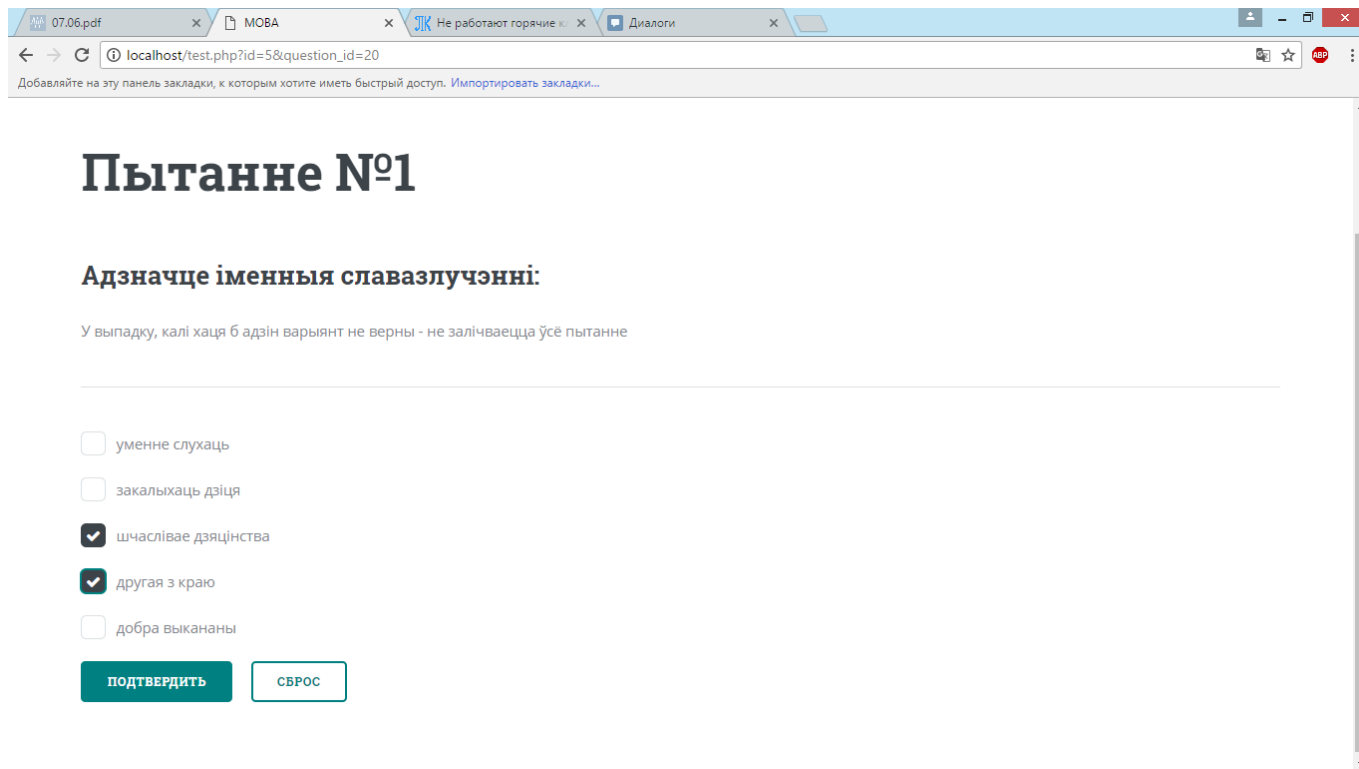


Рисунок 6.8 – Прохождение теста

После прохождения теста результаты появляются в личном кабинете пользователя. Также там отображаются результаты всех пройденных тестов. Окно личного профиля пользователя изображено на рисунке 6.9.

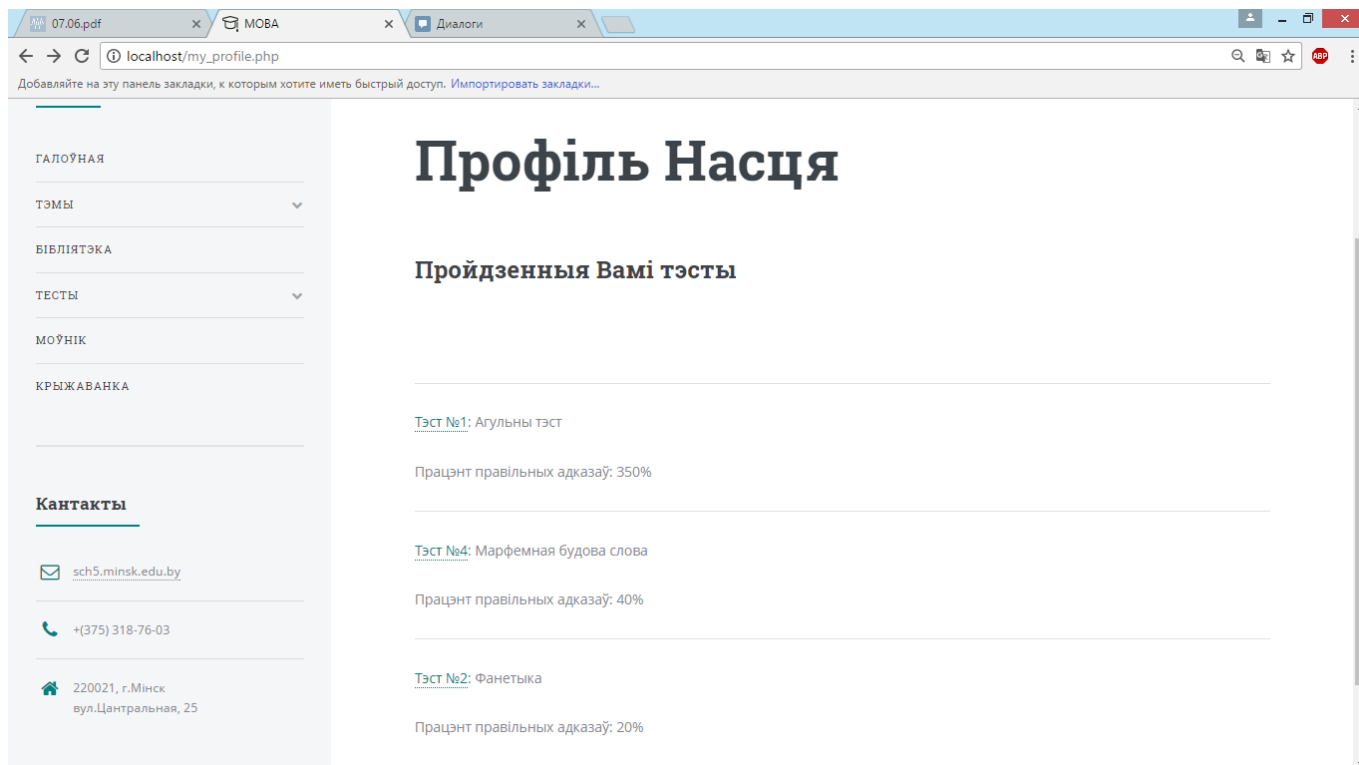


Рисунок 6.9 – Личный кабинет пользователя

Далее пользователь имеет возможность просмотреть ответы на пройденный тест, чтобы в

дальнейшем проанализировать свои ошибки. На рисунке 6.10 представлен просмотр ответов.

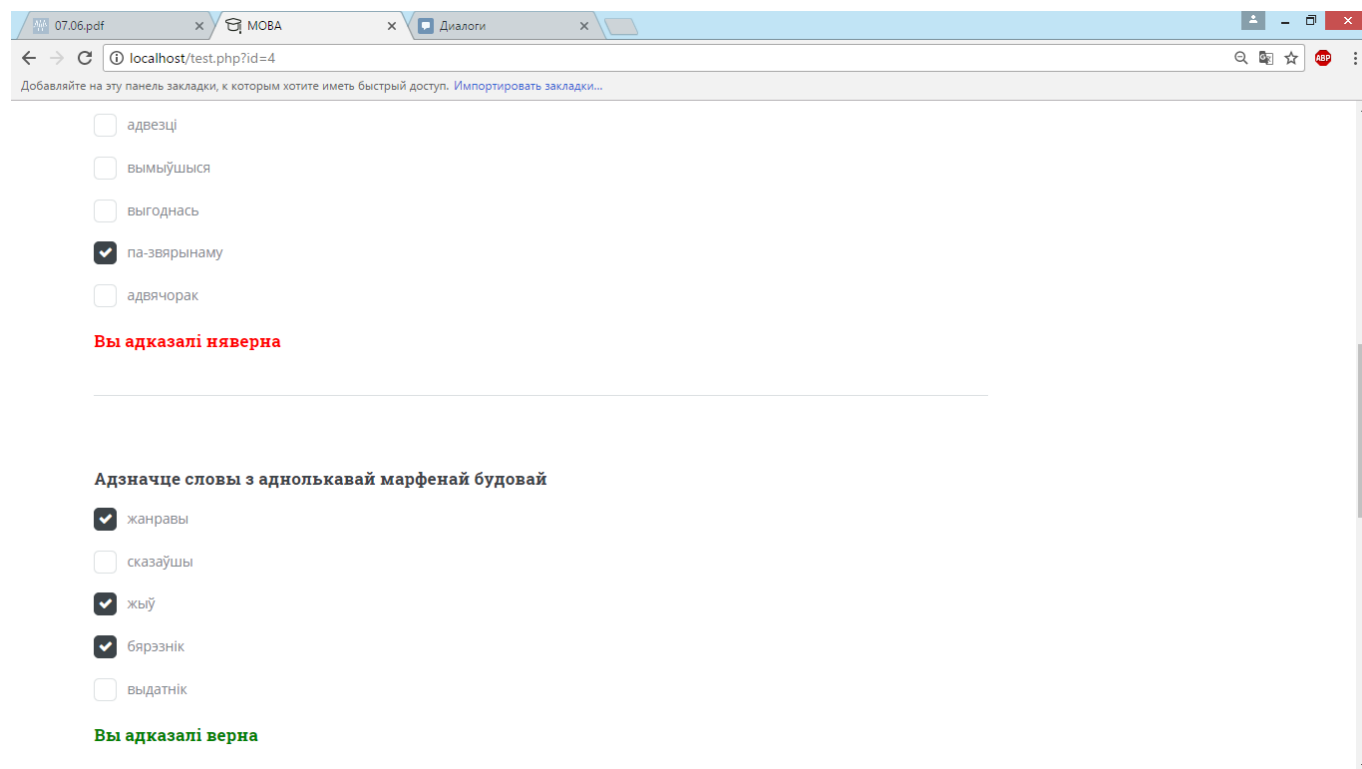


Рисунок 6.10 – Проверка ответов

Интерфейс данного электронного средства обучения был разработан в минималистичном стиле, для лучшего усвоения информации.

8 Результаты тестирования разработанной системы, и оценка выполнения задач

Тестирование ПС - это процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ, это любая деятельность, направленная на обнаружение ошибок в программном продукте.

Указанный набор данных называется тестовым или просто тестом. Таким образом, отладку можно представить в виде многократного повторения трех процессов: тестирования, в результате которого может быть констатировано наличие в ПС ошибки, поиска места ошибки в программах и документации ПС и редактирования программ и документации с целью устранения обнаруженной ошибки.

Тестирование проводится для того, чтобы найти ошибки в программе и тем самым повысить ее надежность. Делается это путём исправления ошибок, внесенных в процессе разработки.

Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

- функциональные;
- нефункциональные;
- связанные с изменениями.

Функциональные тесты базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном или модульном, интеграционном, системном и приемочном.

Функциональные требования включают в себя:

- функциональная пригодность;
- точность;
- способность к взаимодействию;
- соответствие стандартам и правилам;
- защищённость.

Функциональные виды тестирования рассматривают внешнее поведение системы. Одни из самых распространенных видов функциональных тестов:

- функциональное тестирование;
- тестирование безопасности;
- тестирование взаимодействия.

Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. Основные виды нефункциональных тестов:

- все виды тестирования производительности (нагрузочное тестирование, стрессовое тестирование, тестирование стабильности или надежности, объемное тестирование);
- тестирование установки;
- тестирование удобства пользования;
- тестирование на отказ и восстановление;
- конфигурационное тестирование.

Связанные с изменениями виды тестирования проводятся после исправления ошибок для подтверждения того факта, что проблема была действительно решена. Виды тестирования, которые необходимо проводить после установки программного обеспечения, для подтверждения работоспособности приложения или правильности осуществленного исправления дефекта:

- дымовое тестирование;
- регрессионное тестирование;
- тестирование сборки;
- санитарное тестирование или проверка согласованности и исправности.

В процессе разработки данного программного средства также было проведено функциональное тестирование. Результаты тестирования представлены в таблице 7.1

Таблица 7.1 – Результаты тестирования

Действие	Ожидаемый результат	Полученный результат	Принятые меры
Регистрация пользователя	Ввод данных о пользователе	Ошибка при регистрации	Введены некорректные данные

Изучение лекционного материала	Просмотр лекционного материала	Лекционный материал загружен успешно	
Открытие тестов	Отображение заданий на экране	Задания отображены	
Получение результата прохождения теста	Отображение результата теста в процентах	Результат прохождения теста выведен на экран	

Продолжение таблицы 7.1

Действие	Ожидаемый результат	Полученный результат	Принятые меры
Сохранение результата	Данные о прохождении теста отображаются в личном кабинете	Данные о прохождении теста отображены	
Добавление новых данных	Добавлены новые данные	Новые данные отображены	
Удаление данных	Данные удалены	Данные не удалены	Изменение названия таблицы с данными
Редактирование данных	Отображение внесенных изменений	Изменения не отображены	Проверка правильности ввода

В ходе тестирования были выявлены следующие ошибки:

- присвоение некорректных данных;
- некоторые поля не были защищены от ввода некорректных данных;
- неправильная обработка команд;

Тестирование оказывается довольно необычным процессом (вот почему оно и считается трудным), так как этот процесс разрушительный.

Очевидно, что процесс управления тестированием ПО затрагивает все этапы жизненного цикла разработки. Он подразумевает сравнение действительного состояния продукта и того состояния, которое было запланировано и задокументировано в плане тестирования продукта. Процесс тестирования, анализа и мониторинга помогает спланировать и изменить последующие задачи наилучшим путем.

В ходе разработки, конечно же, возникали ошибки, которые были обнаружены и исправлены. После этого повторно было проверено функционирование всех кнопок и всех возможных комбинаций входных данных. Полученные результаты были сравнены с необходимыми.

Заключение

По итогам проделанной работы были пройдены следующие этапы разработки электронного средства обучения по предмету “Белорусский язык”: □ анализ требований и предметной области; □ выбор инструментов разработки; □ проектирование архитектуры решения; □ разработка структуры приложения и модели данных; □ непосредственная реализация; □ отладка и тестирование. Кроме того, была написана документация к пользованию разработанным приложением, содержащая информацию о методах веб-интерфейса, их

параметрах, назначении и виде возвращаемых данных. Подробно были рассмотрены задачи программы, их решение, основные функции, гарантирующие удобство работы с данным программным средством и алгоритм работы программы. В качестве языка программирования был выбран PHP. В разделе охрана труда был проведен анализ обеспечения работающих средствами коллективной защиты при изготовлении проектируемого объекта. Программное средство реализовано в полном объеме и в соответствии с заданными требованиями. Полностью отлажено и протестировано. Поставленные задачи выполнены. Нагрузочное тестирование показало, что приложение справляется с большим количеством запросов в единицу времени без явной потери производительности. В процессе разработки программы использовался в большом объеме материал по программированию, что способствовало закреплению наработанных навыков и умений в этих интересных областях знаний.

Список использованных источников

1. [печатное издание] **1. Зервас, К. Web 2.0: создание приложений на PHP / К. Зервас - М.: «Вильямс», 2009. - С. 544.**
2. [печатное издание] **Клащук В.И. Обучающие программы / В. Клащук. - Солон пресс, 2009. - 18 с.**
3. [печатное издание] **Кузнецов, М. Р., Симдянов, И. В. PHP. Практика создания Web-сайтов. / М. Р. Кузнецов, И. В. Симдянов - СПб.: «БХВ-Петербург», 2008**
4. [url] **Git и GitHub. - [Электронный ресурс]. - Режим доступа: <https://habr.com/post/273897/>- Дата доступа: 01.05.2017. <https://habr.com/post/273897>**
5. [url] **GitHub. - [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/GitHub>- Дата доступа: 01.05.2017 <https://ru.wikipedia.org/wiki/GitHub>**

Приложения

1. [Приложение] **Приложение [5ed39bb07a245_Приложение А \(Диаграмма «Сущность-связь»\).docx](#)**
2. [Приложение] **Приложение [5ed39bb0d7ad8_Приложение Б \(Текст программы\).docx](#)**
3. [Задание] **Задание [5ed3c92679657_List_zadania.docx](#)**