**Содержание**

|  |  |
| --- | --- |
| Введение………………………………………………………………………. | 5 |
| 1 Анализ предметной области……………………………………………… | 6 |
| 2 Формирование требований к программному продукту………………… | 7 |
| 3.Разработка технического задания……………………………………….. | 8 |
| 4 Проектирование базы данных……………………………………………... | 10 |
| 5 Разработка алгоритмов…………………………………………………….. | 13 |
| 6 Разработка макета интерфейса программы………………………………. | 14 |
| 7 Разработка интерфейса пользователя……………………………………... | 16 |
| 8 Разработка тестов…………………………………………………………... | 19 |
| 9 Экономическая оценка…………………………………………………….. | 20 |
| 10 Контрольный пример……………………………………………………... | 22 |
| 11 Заключение………………………………………………………………... | 25 |
| 12 Список использованных источников……………………………………. | 26 |
| Приложение А………………………………………………………………... | 27 |

**Введение**

**Актуальность темы –** в последнее время такая информационная среда как Интернет развивается очень быстро. Постоянно увеличивается число Интернет-сайтов для молодёжи и студенчества, сетевых изданий вузов. Исследование особенностей подробных изданий актуально, оно помогает понять направления развития данных СМИ, определить их основные характеристики и необходимые свойства.

**Проблематика** состоит в том, что специализация проекта достаточно узкая и разрабатывается для конкретной предметной области, то соответственно аналогов не имеет.

**Цели:**

1. Создать рабочую программу.
2. Разработать алгоритм работы программы.
3. Создать программу с базой данных.
4. Создать интерфейс программы.

**Объектом исследования** является Интернет – приложение «Студенческая интернет – газет».

Для реализации поставленной цели были сформулированы следующие **задачи:**

1. Разработанная программа должна иметь удобный интерфейс.
2. Разработать техническое задание.
3. Разработать макет программы.
4. Программа должна выводить на главную форму запись и автора.

**1 Анализ предметной области**

Основными понятия предметной области являются: Администратор, пользователь, автор, газета.

Администратор – зарегистрированное физическое лицо.

Пользователь - зарегистрированное физическое лицо.

Автор - тот, кто работает с [текстом](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82), а именно составляет, проверяет и исправляет содержание в соответствии с требованиями определённого [жанра](https://ru.wikipedia.org/wiki/%D0%96%D0%B0%D0%BD%D1%80), готовит к печати [издание](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1).

Газета - совокупность систем и элементов, работающих одним целым для достижения общей цели.

Для решения проблемы была выбрана платформа MSSQLServer. База данных будет состоять из следующих таблиц:

- Newspaper (ID\_Newspaper, Newspaper\_Name,Editorial\_Office\_address, Editor\_in\_Chief)

- News\_release (ID\_News\_relaese,Newspaper\_ID, Data\_News\_relaese, Editor)

- News (ID\_News, Author, news\_Bescriptions, News\_relaese\_ID)

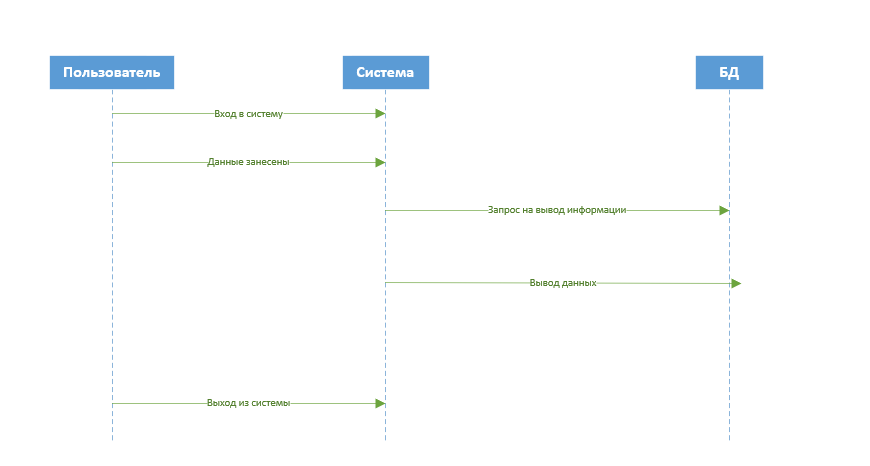


Рисунок 1 – Диаграмма последовательности

**2 Формирование требований к программному продукту**

При разработке приложения необходимо определить некоторые требования к будущему программному продукту, для данного приложения были сформулированы следующие цели и требования:

Цели создания программного продукта:

-автоматизация создания записей;

- обратная связь для работы с пользователем;

- создание единой базы данных для создания записи.

Формирование функциональных требований к программному продукту:

- авторизация пользователя;

- регистрация пользователя;

- ввод информации о создании записи;

- выбор автора записи;

- выбор даты создания записи;

- вывод информации;

- организация защиты базы данных.

**3Разработка технического задания**

**3.1 Наименование и область применения**

«Разработка Интернет-приложение «Студенческая интернет-газета»». Данная программа будет внедрена в организацию и будет обеспечивать студентов и преподавателей информацией о том, что происходит в колледже.

**3.2 Назначение разработки**

Функциональное назначение

Приложение предоставляет возможность обеспечивать студентов и преподавателей новой информацией о колледже. Так же газета нужна, для самовыражения студентов. Печатать рассказы тех, кто тут учится.И так же газета должна непременно информировать о тех мероприятиях, которые намечаются в колледже в ближайшее время.

Эксплуатационное назначение

Программа должна использоваться для обеспечения студентов и преподавателей новой информацией о колледже.Конечными пользователями программы могут являться как студенты колледжа (полное право доступа к информации), так и преподаватели (частичное предоставление информации).

**3.3 Требования к программе**

- требования к функциональным характеристикам («программа должна позволять сохранять файл проекта» и т.п.);

- требования к надежности («программа должна обеспечивать проверку корректности входных данных» и т.п.);

- требования к составу и параметрам технических средств;

- требования к информационной и программной совместимости.

**3.4 Технико-экономические показатели**

- ориентировочная экономическая эффективность (насколько будет иметь смысл пользоваться вашим продуктом) Данным продуктом будет очень легко и просто пользоваться.

- предполагаемая потребность (кто, зачем и когда будет пользоваться вашим продуктом) Данным продуктом будут пользоваться студенты и преподаватели, которые хотят самовыражаться.

- экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами (чем то, что вы делаете, будет лучше того, что кто-то уже сделал) Данная программа просто в освоении и не кому не составит труда зарегистрироваться в ней.

**3.5 Порядок контроля и приемки**

Виды испытаний

Производится проверка корректного выполнения программой заложенных в нее функций, т.е. осуществляется функциональное тестирование программы. Также осуществляется визуальная проверка интерфейса программы. Функциональное тестирование осуществляется в соответствии с документом “ «Программа построения поверхностей вращения». Программа и методика испытаний (ГОСТ 19.301-79)”, в котором указывают:

1) перечень функций программы, выделенных в программе для испытаний, и перечень требований, которым должны соответствовать эти функции (со ссылкой на пункт 4.1.1. настоящего технического задания);

2) перечень необходимой документации и требования к ней (со ссылкой на пункты 4.9 и 4.10 настоящего технического задания);

3) методы испытаний и обработки информации;

4) технические средства и порядок проведения испытаний

Общие требования к приемке работы

Прием программы будет утвержден при корректной работе программы при различных входных данных, соответствующих условиям и при предоставлении полной документации к продукту.

1. **Проектирование базы данных**

**4.1 Описание процесса проектирования**

I этап – Сущность.

Newspaper (ID\_ Newspaper……)

News\_release (ID\_ News\_relaese……..)

News (ID\_ News……)

II этап – ER-диаграммы.

На рисунке 2 предоставлена ER-диаграмма с учётом всех сущностей и их связей

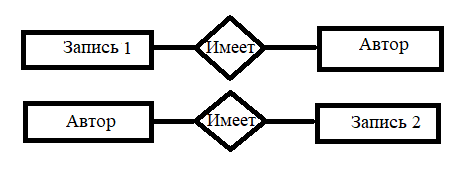


Рисунок 2 – ER-диаграмма

III этап – Формирование набора предварительных отношений с указанием ключевого атрибута показано на рисунке 3.

|  |
| --- |
| News\_release  ID\_News\_relaese  Newspaper\_IDData\_News\_relaese  Editor |

|  |
| --- |
| Newspaper  ID\_NewspaperNewspaper\_Name  Editorial\_Office\_addressEditor\_in\_Chief |

|  |
| --- |
| News  ID\_News Author  news\_Bescriptions  News\_relaese\_ID |

Рисунок 3 - Набор предварительных отношений с указанием ключевого атрибута

**4.2 Описание процесса создания базы данных**

Для создания базы данных используется СУБД: SQL ServerManagementStudio 2012.

После создаем базу данныхgazeta и создаем в ней таблицы, разработанные в пункте 4.1. Ниже представлены скриншоты таблиц 4-6.

В соответствии со спроектированной базой данных создано 3 таблицы:

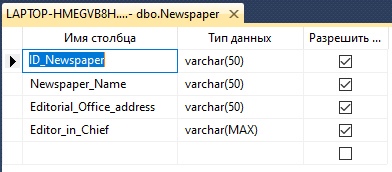


Рисунок 4 – Таблица «Newspaper»

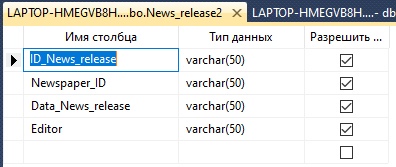


Рисунок 5 –Таблица «News\_release»

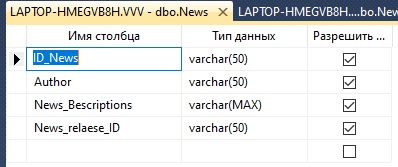


Рисунок 6 – Таблица «News»

**4.3 Установка связи между таблицами**

В меню базы данных настраиваем связи между таблицами, для этого необходимо нажать правой кнопкой мыши и выбрать создать диаграмму базы данных. Выбрать нужные таблицы и связывать их по средствам перетаскивания и соединения элементов с помощью мыши.

На рисунке 7 показана схема данных таблиц:



Рисунок 7 – Диаграмма базы данных

**5 Разработка алгоритмов**

В приложение вводится информация и пользователь нажимает на кнопку, если информация корректная, программа подключается к базе данных и создаётся запрос для внесения новой записи в таблицу.

Кнопка нажата

Нажатие кнопки

Ввод: данные

нет

Данные введены корректно

Вывод: запись добавлена

Вывод : ошибка

**6 Разработка макета интерфейса программы**

**6.1 Форма авторизации**

Главной формой является форма авторизации. Если пользователь не проходит аутентификацию или идентификацию, то доступ к базе данных он не получает. Форма авторизации представлена на рисунке 8.

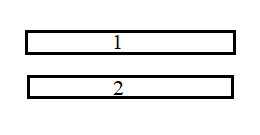


Рисунок 8 – Форма входа

1 – поле логина администратора

2 – поле пароля администратора

**6.2 Форма выбора записи и автора**

Форма выбора записии автора предназначена для добавления новой записи и автора. Форма выбора записи и авторапредставлена на рисунке 9.

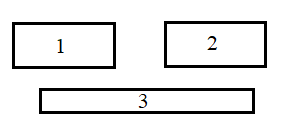


Рисунок 9 – Форма выбора записи и автора

1 –запись 1

2 –запись 2

3 – кнопка выбора автора

**6.3 Форма ввода информации о записи**

Данная форма предназначена для ввода информации о записи. Форма представлена на рисунке 10.

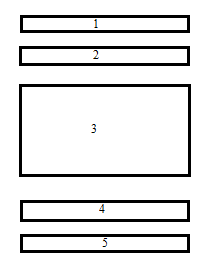
****

Рисунок 10 – Форма вывода информации

1 - тема записи

2 - заголовок записи

3 - содержание записи

4 - автор записи

5 – дата добавления записи

**6.4 Форма вывода информации о записи**

Данная форма предназначена для вывода информации о записи. Форма представлена на рисунке 11.

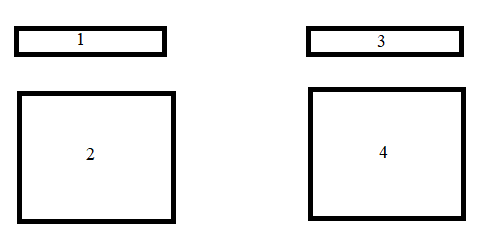
****

Рисунок 11 – Форма вывода информации о записи

**7 Разработка интерфейса пользователя**

Для работы с базой данных были созданы следующие формы:

* Форма авторизации пользователя;
* Форма вывода информации;
* Форма изменения записи;
* Форма удаления записи;

Создание форм происходит с помощью интегрированной среды разработки MicrosoftVisualStudio 2012 на платформе. NetFramework, языка C#.

На рисунке 12 представлена форма входа пользователя. После входа пользователь зайдет на главный экран или получит сообщение об ошибке в противном случае.

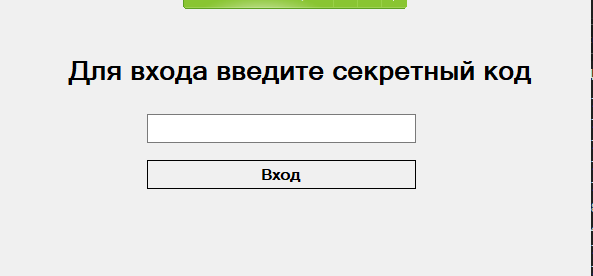


Рисунок 12 – Форма входа пользователя

Затем, после успешной авторизации, пользователь перенаправляется на форму добавления записи и автора. На этой форме предоставлена возможность добавить запись, добавить автора. Выбираете то что вам нужно и переходе на следующую форму.

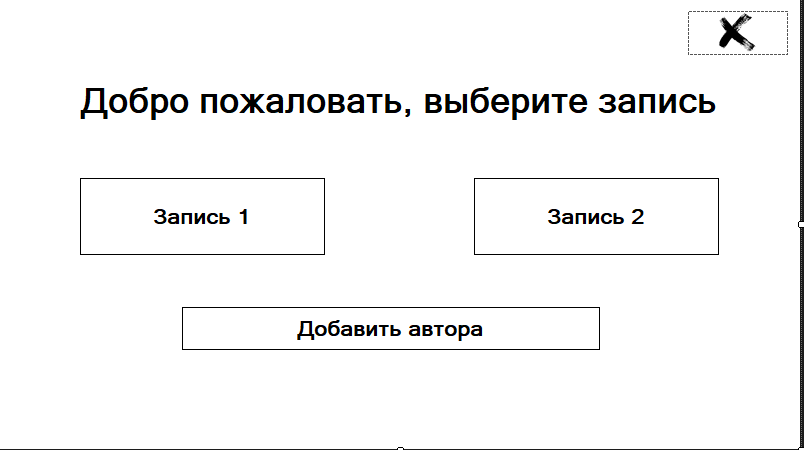
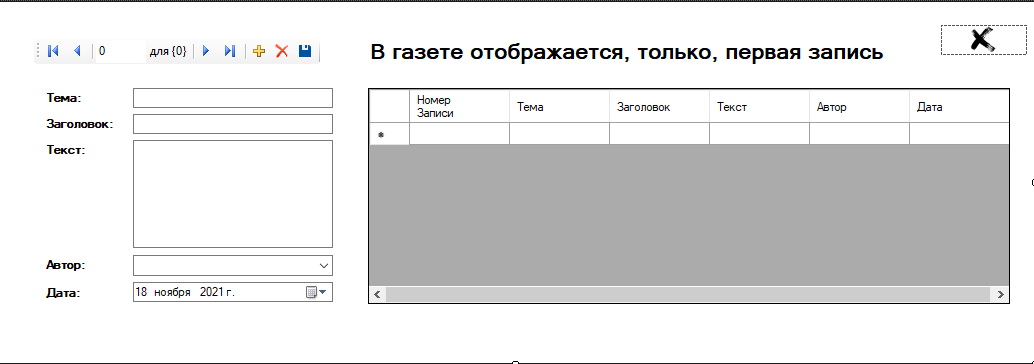
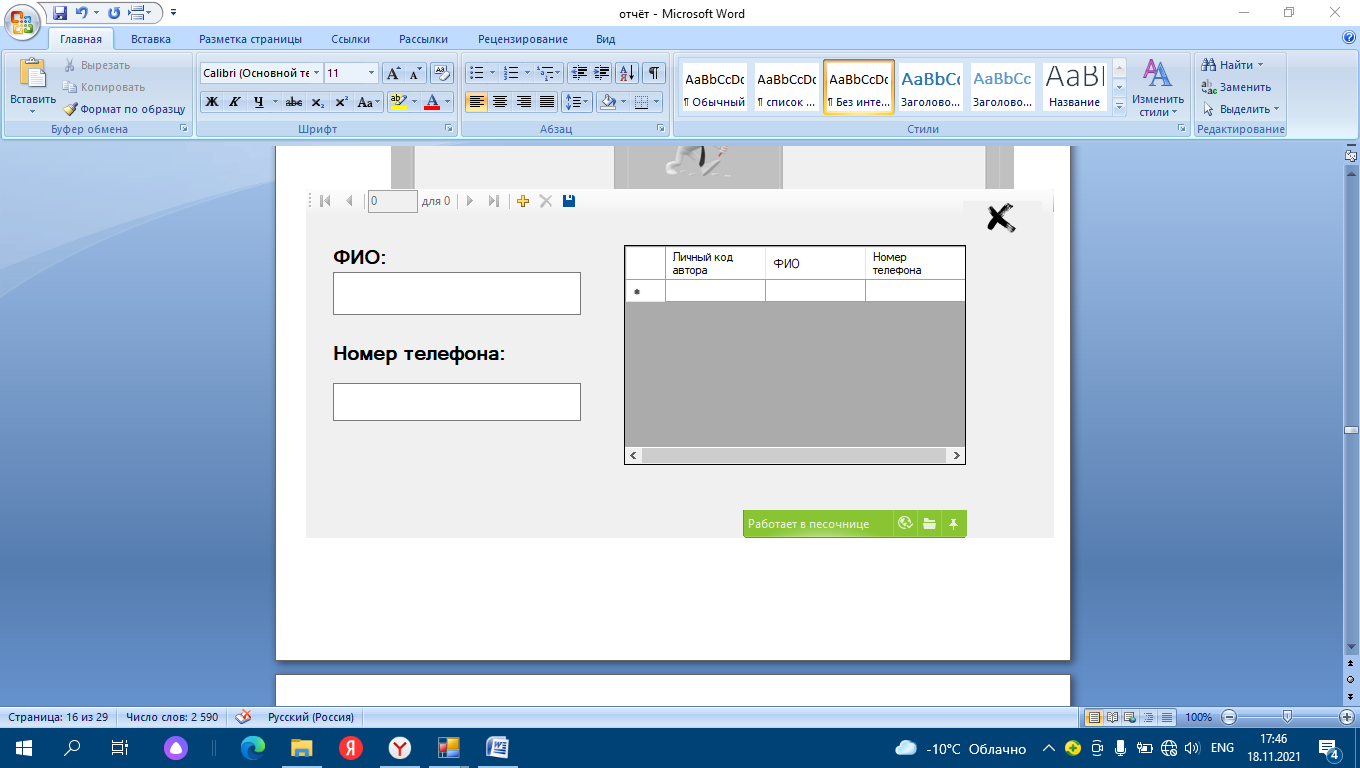


Рисунок 13 – Форма добавление записи и автора

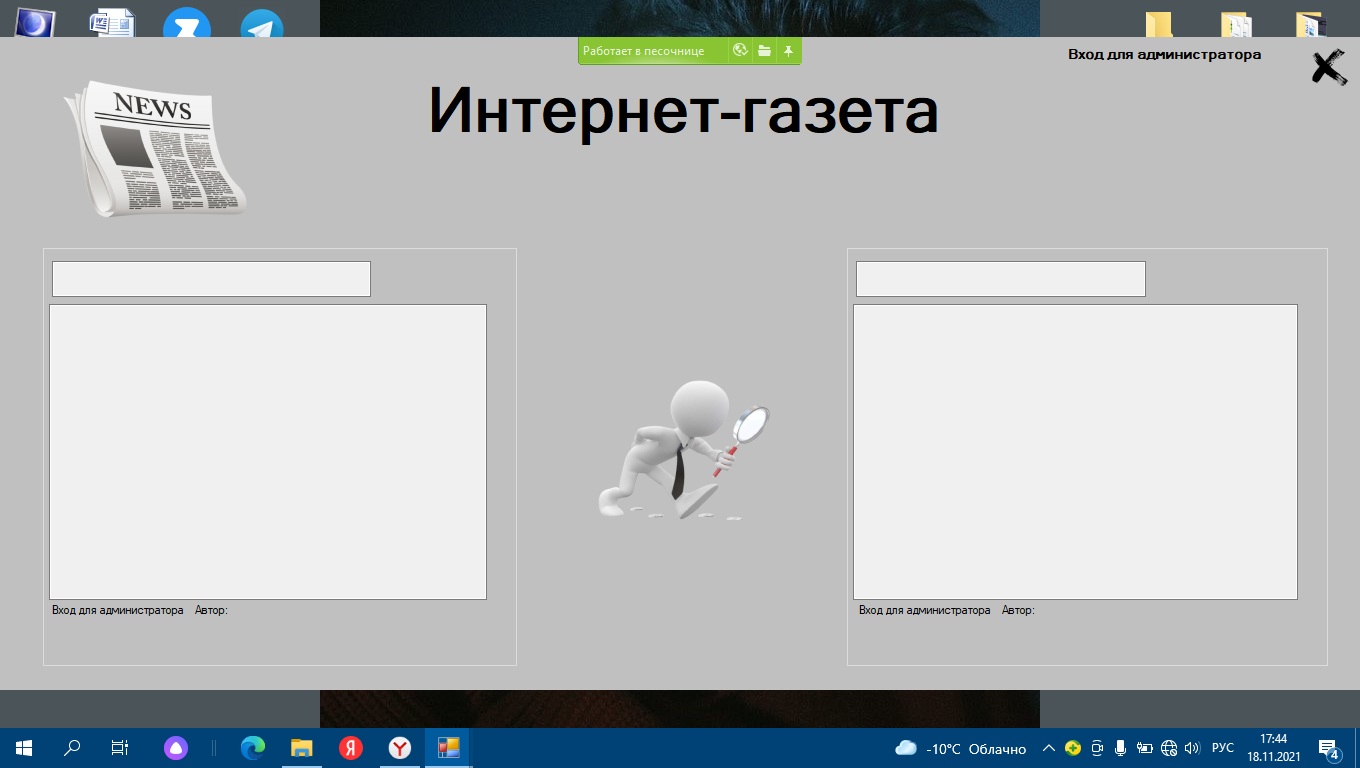
Если вам нужно добавить запись, то нажимаете на запись 1 после это заполняете все поля и нажмите кнопку «Сохранить». Форма добавления записи представлена на рисунке 14.

Рисунок 14 – Форма добавления записи

После этого можно добавить нового автора, для этого заполнить определенные данные и нажать кнопку «Сохранить». Форма добавления автора представлена на рисунке 15.

Рисунок 15 – Форма добавления автора

После добавления записи и автора вся информация выводится на главную форму. Форма интернет – газеты представлена на рисунке 16.

 Рисунок 16 – Форма Интернет - газеты

**8. Разработка тестов**

Тест-Кейс №1 «Добавление записи»

Шаги: 1 Вход в окно

2 Нажмите на кнопку «Добавить запись»

В таблице 1 приведен результат позитивного теста

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат |
| Войти в программу | Открытие программы | Программа открылась |
| Нажать на кнопку «Добавить запись» | Переход в окно «Запись» | Переход в окно «Запись» |

Таблица 1 – Позитивный тест

Тест-Кейс №2 «Добавление автора»

Шаги: 1 Вход в окно «Добавление автора»

2Ввод «ФИО»

3Ввод «Номер телефона»

В таблице 2 приведен результат позитивного теста

|  |  |  |
| --- | --- | --- |
| Действия | Ожидаемый результат | Результат |
| Войти в программу | Открытие программы | Программа открыта |
| Войти в окно «Добавление автора» | Войти в окно «Добавление автора» | Зашли в окно «Добавление автора» |
| Заполнить поле «ФИО» | Заполнение поля«ФИО» | Поле «ФИО» заполнено |
| Заполнить поле «Номер телефона» | Заполнения поля «Номер телефона» | Поле «Номер телефона» заполнено |

Таблица 2 – Позитивный тест

**9. Экономическая оценка**

Конечно, чтобы можно было говорить о работе этого приложения с точки зрения экономики: о прибыли и о затратах. Его нужно дорабатывать и доводить до хотя бы какого-то совершенства.

Но на данный момент для эксплуатации приложения необходимо подключение к базе данных, что достаточно дешево в некоторых случаях может обойтись даже бесплатно. Данное приложение позволяет молодым людям научиться высказывать свои идеи, мысли, помогает лучше познать себя, открыть мир.

Распространить приложение можно по сети интернет, это будет наилучшим решение для рекламы.

В дальнейшем приложение можно дорабатывать, что позволит увеличить его функционал и качество.

А) Расчет стоимости материалов с учетом транспортно-заготовительных расходов (ТЗР).

Исходные данные и результаты расчетов приведены в таблице 3-6.

Таблица 3 – Затраты на расходные материалы и оборудование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| п/п | Наименование материала | Расход  , шт. | Цена, руб. | Сумма, руб. |
| 1 | Ноутбук DELL G3 3590 | 1 | 52000 руб. | 52000 руб. |
| 2 | Мышь компьютерная 985G | 1 | 890 руб. | 890 руб. |
| 3 | Пакет Visual Studio Community | 1 | 0 руб. | 0 руб. |
| 4 | Пакет Office 356 | 1 | 2500 руб. | 2500 руб. |
| 5 | Пакет SQL Server Express | 1 | 0 руб. | 0 руб. |
| Итого | | | | 55390 руб. |

Таблица 4 – Затраты на электроэнергию

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оборудование | Мощность, кВт | Стоимость, 1  кВт/час | Время работы, час | Сумма затрат, руб.  Гр.5=гр.2× гр.3× гр.4 |
| Ноутбук | 0,05 | 3.19 | 42 | 133.98 |
| Итого | | | | 133.98 руб. |

Таблица 5 – Амортизационные отчисления

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид оборудования | Первоначальная стоимость, руб. | Количество рабочих месяцев | Норма амортизации  %, месячных | Сумма амортизационных отчислений, руб.  Гр.5=гр.2× гр.3×  гр.4 |
| Мышь компьютерная | 890 | 1 | 100 | 890 |
| Зарядное устройство ноутбука | 2990 | 1 | 100 | 2990 |
| Итого | | | | 3880 руб. |

**10 Контрольный пример**

На рисунке 17 представлена форма входа пользователя. После входа пользователь зайдет на главный экран или получит сообщение об ошибке в противном случае.

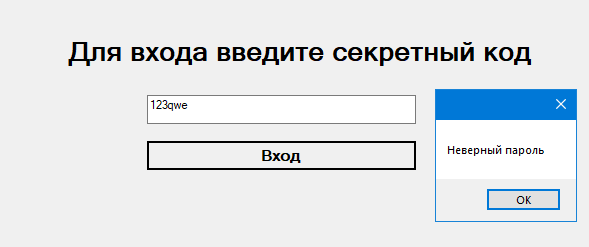


Рисунок 17 – Форма входа пользователя

Затем, после успешной авторизации, пользователь перенаправляется на главную форму. На этой форме предоставлена возможность добавить запись, добавить автора. Выбираете то что вам нужно и переходе на следующую форму.

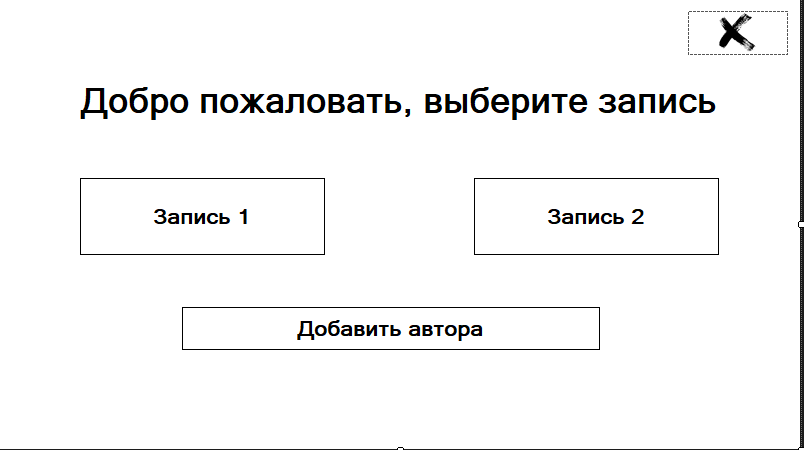
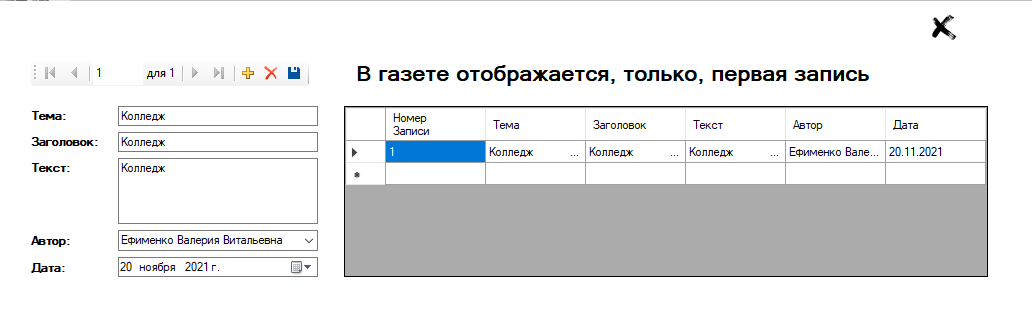


Рисунок 18 – Форма добавление записи и автора

Если вам нужно добавить запись, то нажимаете на запись 1 после это заполняете все поля и нажмите кнопку «Сохранить». Форма добавления записи представлена на рисунке 19.

Рисунок 19 – Форма добавления записи

После этого можно добавить нового автора, для этого заполнить определенные данные и нажать кнопку «Сохранить». Форма добавления автора представлена на рисунке 20.

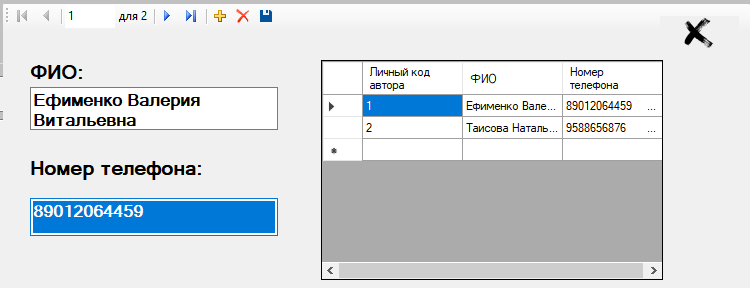


Рисунок 20 – Форма добавления автора

После добавления записи и автора на главную форму выводится информация, которую вы добавили. Главная форма представлена на рисунке 21.

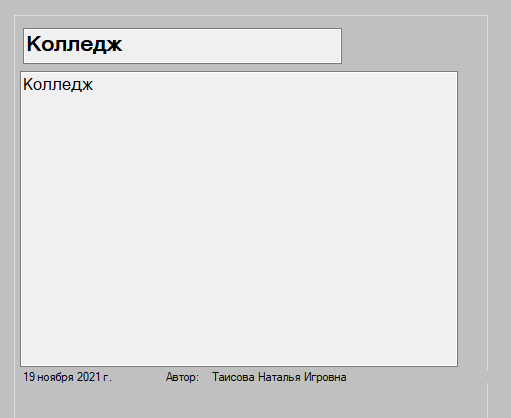


Рисунок 21 – Главная форма

**Заключение**

Благодаря проведенной работе были четко определены цели и задачи разработки программного продукта «Интернет-приложение «Студенческая интернет-газета»». Так же были выявлены недостатки программного продукта и определены способы их устранения.

При разработке программного продукта были достигнуты следующие цели:

1. Созданарабочая программа.
2. Разработан алгоритм работы программы.
3. Созданапрограмма с базой данных.
4. Создан интерфейс программы.

Так же были выполнены следующие задачи:

1. Разработана программа с удобным интерфейсом.
2. Разработано техническое задание.
3. Разработан макет программы.
4. Программа выводит на главную форму запись и автора.

База данных была успешна, разработана и поставленные задачи были выполнены, автоматизация доступа к информации, данные систематизированы и приведены к одному виду.

При помощи C# были разработаны удобные формы для работы с информацией, удобным и простым интерфейсом.

**Список использованных источников**

1. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 272 c.
2. Казанский, А. А. Программирование на visual c# 2013 : учебное пособие для прикладного бакалавриата / А. А. Казанский. — М. : Издательство Юрайт, 2018. — 191 с.
3. Орлов, С. А. Технологии разработки программного обеспечения [Текст] / С. А. Орлов, Б. Я. Цилькер. — СПб.: Питер, 2013. — 608 с.: ил.; 70 × 100/16 (167 × 236 мм) Учебник для ВУЗов. — Библиогр.: с. 596–600. — Предм. указ.: с. 601–608. — 2000 экз. — ISBN 978-5-459-01101-2.
4. Редмонд, Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL [Текст]: [пер. с англ.] / Э. Редмонд, Д. Р. Уилсон. — М.: ДМК Пресс, 2013. — 384 с.
5. Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М.: Юрайт, 2013. - 463 c.
6. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: ИЦ Академия, 2012. - 320 c.
7. ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению»

**ПриложениеА**

**Form 1**

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet1.Table3". При необходимости она может быть перемещена или удалена.

this.table3TableAdapter.Fill(this.database1DataSet1.Table3);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table2". При необходимости она может быть перемещена или удалена.

this.table2TableAdapter.Fill(this.database1DataSet.Table2);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table2". При необходимости она может быть перемещена или удалена.

this.table2TableAdapter.Fill(this.database1DataSet.Table2);

textBox1.Text = текстTextBox.Text;

textBox2.Text = заголовокTextBox.Text;

label4.Text = авторTextBox.Text;

label3.Text = датаDateTimePicker.Text;

textBox4.Text = текстTextBox1.Text;

textBox3.Text = заголовокTextBox1.Text;

label6.Text = авторTextBox1.Text;

label5.Text = датаDateTimePicker1.Text;

}

private void label1\_Click(object sender, EventArgs e)

{

Form2 h = new Form2();

h.Show();

}

private void table2BindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.table2BindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void table2BindingNavigatorSaveItem\_Click\_1(object sender, EventArgs e)

{

this.Validate();

this.table2BindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

**Form 2**

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "123qwe")

{

Form3 h = new Form3();

h.Show();

this.Hide();

}

else

{

MessageBox.Show("Неверныйпароль");

this.Hide();

}

}

}

}

**Form 3**

usingSystem;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form2 h = new Form2();

h.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

Form4 h = new Form4();

h.Show();

this.Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

Form5 h = new Form5();

h.Show();

this.Hide();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Form6 h = new Form6();

h.Show();

this.Hide();

}

}

**Form 4**

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void table2BindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.table2BindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void Form4\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table". При необходимости она может быть перемещена или удалена.

this.tableTableAdapter.Fill(this.database1DataSet.Table);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table2". При необходимости она может быть перемещена или удалена.

this.table2TableAdapter.Fill(this.database1DataSet.Table2);

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

авторTextBox.Text = comboBox1.Text;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form1 h = new Form1();

h.Show();

this.Hide();

}

}

}

**Form 5**

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void table3BindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.table3BindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet1);

}

private void Form5\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table". При необходимости она может быть перемещена или удалена.

this.tableTableAdapter.Fill(this.database1DataSet.Table);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table2". При необходимости она может быть перемещена или удалена.

this.table2TableAdapter.Fill(this.database1DataSet.Table2);

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet1.Table3". При необходимости она может быть перемещена или удалена.

this.table3TableAdapter.Fill(this.database1DataSet1.Table3);

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

авторTextBox.Text = comboBox1.Text;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form1 h = new Form1();

h.Show();

this.Hide();

}

}

}

**Form 6**

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespacegazeta

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

private void tableBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.tableBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void Form6\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "database1DataSet.Table". При необходимости она может быть перемещена или удалена.

this.tableTableAdapter.Fill(this.database1DataSet.Table);

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Form1 h = new Form1();

h.Show();

this.Hide();

}

}

}