|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **Исследовательская часть**    1. **Проектирование программного продукта**   **2.1.1 Постановка задачи**  В процессе проектирования индивидуального задания необходимо разработать административную часть интернет-магазина по продаже настольных игр, которая предполагает наличие базы данных и интерфейса для работы с ней.  При разработке программного продукта необходимо учесть следующие требования:  - обеспечение работы в удобной для администратора форме,  - вывод информации на экран,  - добавление информации в базу данных,  - удаление информации из базы данных,  - редактирование информации из базы данных.  **2.1.2 Выбор инструментальных средств разработки**  Для реализации данного продукта были выбраны следующие средства разработки:  - «MySQL» – свободная реляционная система управления базами данных, которая отличается высокой скоростью работы, надежностью, гибкостью.  - HTML – язык гипертекстовой разметки документов, с помощью которого создаются веб-страницы, находящиеся в компьютерной сети Интернет,  - С# – императивный, структурированный, объектно-ориентированный, высокоуровневый язык программирования со строгой статической типизацией | | | | | | | | | | | |
|  |  |  |  |  | *ОПП.ПП.* | *508120* | | *ППП* | |  |  |
|  |  |  |  |  |
|  |  |  |  |  | *Проектирование программного продукта* | *Лит.* | | | *Масса* | | *Масштаб* |
| *Изм.* | *Лист* | *№ докум.* | *Подп.* | *Дата* | *У* |  |  |  | |  |
| *Разраб.* | | *Мощинский А.А.* |  | *06.12.21* |
| *Пров.* | | *Новикова Е.Е.* |  | *10.12.21* |
|  | |  |  |  | *Лист ( )* | | | | *Листов ( )* | |
|  | |  |  |  | *Отделение Информационные системы и технологии* | *Филиал БГТУ Витебский*  *государственный технологический колледж , гр. 32o* | | | | | |
|  | |  |  |  |
|  | |  |  |  |

переменных,

- CSS – свод стилевых описаний, тех или иных HTML тегов, который может быть применён как к отдельному тегу, так и одновременно ко всем идентичным элементам на всех страницах сайта, либо только к какой-то их части.

# 2.1.3 Проектирование программного продукта

На основании изученной предметной области разработана структура базы данных интернет-магазина. Перечень отношений базы данных представлен в таблицах 2.1 – 2.6.

# Таблица 2.1 – Таблица «Users»

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип атрибута** | **Назначение** |
| Id | int | Идентификатор пользователя, первичный ключ |
| Name | nvarchar(MAX) | Имя пользователя |
| Password | nvarchar(MAX) | Пароль |
| Email | nvarchar(MAX) | Электронная почта |
| Role | int | Роль пользователя |

Составлено автором

В таблице 2.1 представлена информация о пользователях и информация необходимая для авторизации на сайте.

# Таблица 2.2 – Таблица «Products»

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип атрибута** | **Назначение** |
| Id | int | Идентификатор товара, первичный ключ |
| Title | nvarchar(50) | Наименование продукта |
| Cost | real | Цена товара |
| Description | int | Описание товара |
| Quantity | int | Количество товара на складе |
| Picture | nvarchar(MAX) | Ссылка на изображение товара |

Составлено автором

В данной таблице 2.2 представлено описание товаров в интернет-магазине.

# Таблица 2.3 – Таблица «Сategories»

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип атрибута** | **Назначение** |
| Id | int | Идентификатор категории, первичный ключ |
| Title | nvarchar(50) | Наименование категории |

Составлено автором

В таблице 2.3 представлены категории интернет-магазина.

# Таблица 2.4 – Таблица «ProductUser»

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип атрибута** | **Назначение** |
| Id | int | Идентификатор заказа, первичный ключ |
| Quantity | int | Количество товара |
| Address | nvarchar(MAX) | Адрес заказчика |
| Data | datetime2(7) | Дата оформления заказа |
| ProductId | int | Идентификатор товара,  внешний ключ |
| UserId | int | Идентификатор пользователя,  внешний ключ |

Составлено автором

В таблице 2.4 представлена данные о заказах пользователей.

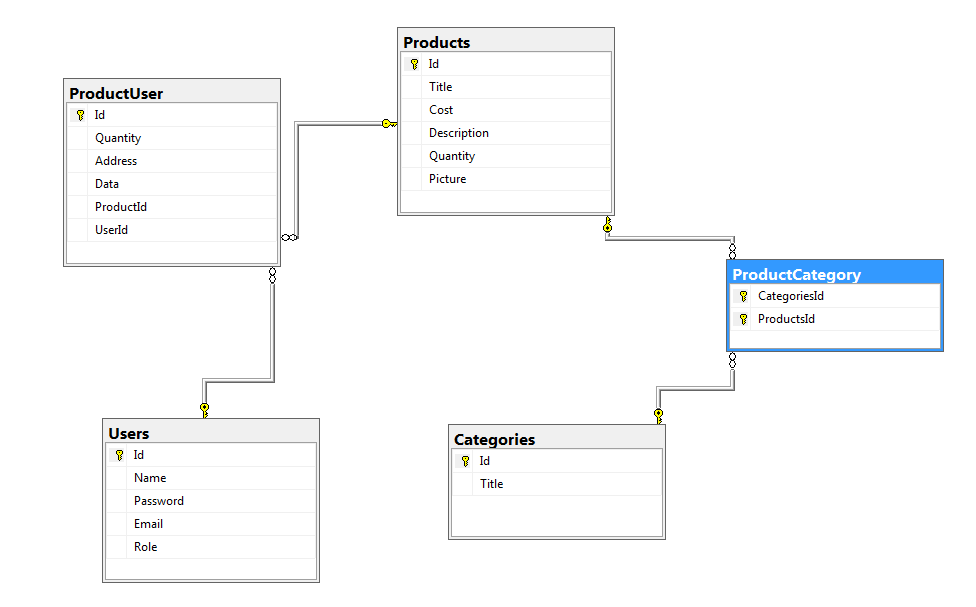
# Таблица 2.5 – Таблица «ProductCategory»

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип атрибута** | **Назначение** |
| CategoriesId | int | Идентификатор категории, первичный и внешний ключ |
| ProductsId | int | Идентификатор товара, первичный и внешний ключ |

Составлено автором

В таблице 2.5 представлены соединения категорий и товаров.

Логическая схема базы данных представлена на рисунке 2.1.



# Рисунок 2.1 – Логическая схема базы данных

Таким образом проведено проектирование программного продукта.

# Реализация программного продукта

Процесс создания программного продукта делится на несколько стадий:

* изучение выбранной предметной области,
* обработка полученной информации,
* выбор средств реализации,
* реализация.

Для получения доступа к админ-панели сайта для начала требуется авторизоваться на сайте «‘хост’/Account/Login», после чего перейти по пути «’хост’/Admin» в адресной строке. При этом происходит проверка согласно данным в БД.

# Листинг 2.1 – Проверка администратора

private bool CheckAdmin()

{

int id = Convert.ToInt32(HttpContext.Request.Cookies["id"]);

User user = \_repositoryManager.User.GetUsers()

.FirstOrDefault(c => c.Id == id);

if(user != null)

{

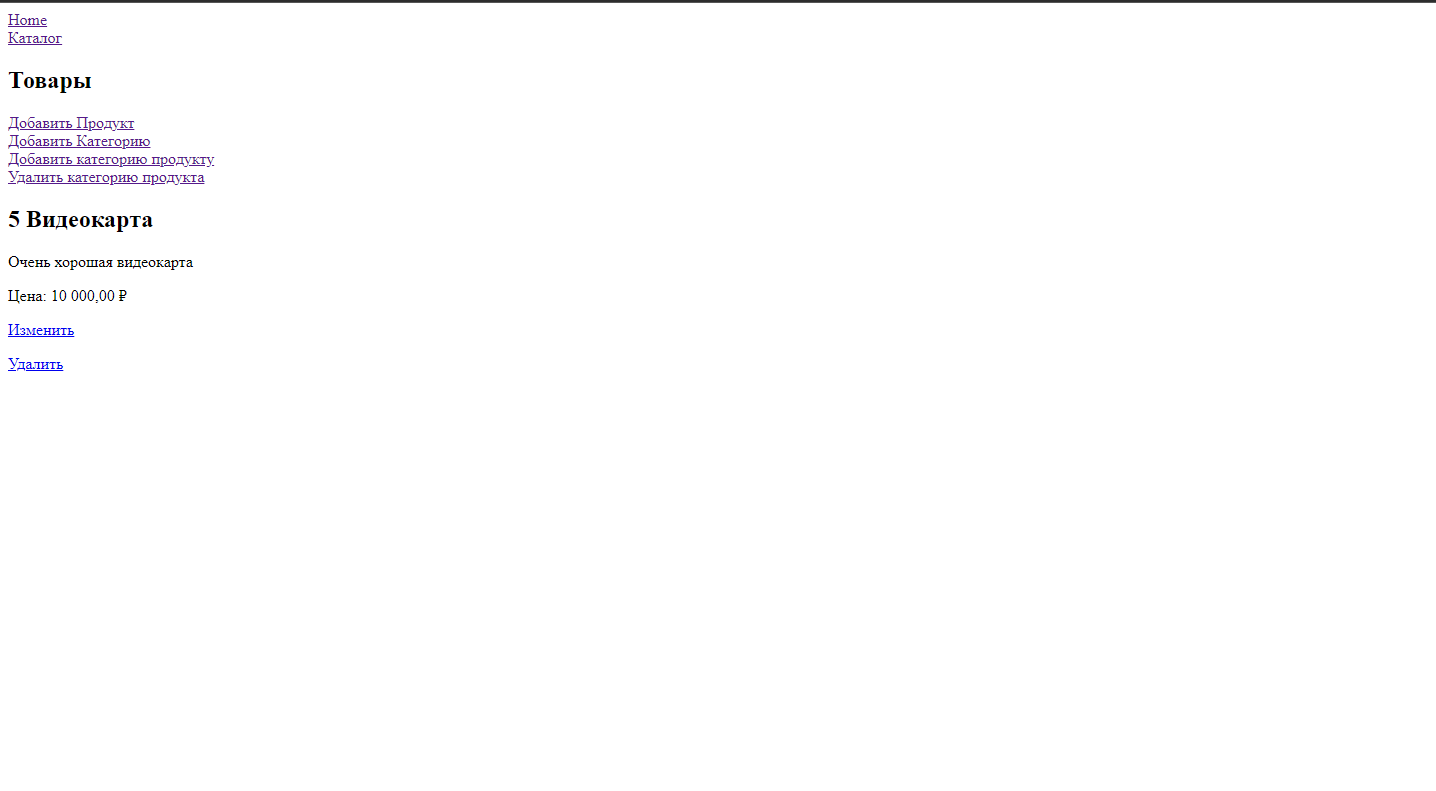
return (int)user.Role == 1 ? true : false;

}

return false;

}

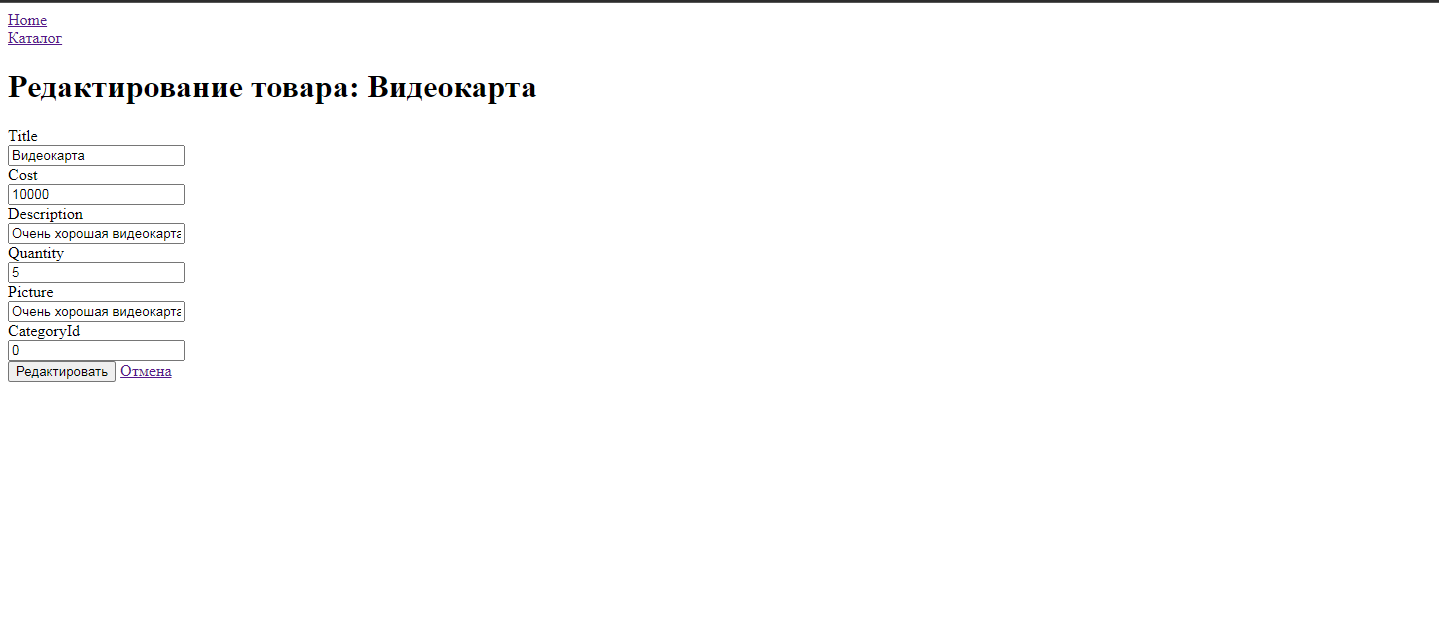
В случае прохождения проверки, администратор может перейти в панель администратора.



# Рисунок 2.2 – Главная страница

На главной странице отображаются все товары и категории, а также возможности администратора.

Существующие товары и категории можно удалить или изменить. Пример редактирования товара представлен на рисунке 2.3.



**Рисунок 2.3 – Редактирование товара**

Реализация метода редактирования товара представлена в листинге 2.2.

# Листинг 2.2 – Реализация редактирования товара

public IActionResult EditProduct(int id)

{

var product = \_repositoryManager.Product.GetProducts().FirstOrDefault(c => c.Id == id);

ProductForCreationDto productDto = new()

{

Id = product.Id,

Title = product.Title,

Quantity = product.Quantity,

Description = product.Description,

Cost = product.Cost,

Picture = product.Picture

};

return View(productDto);

}

[HttpPost]

public IActionResult EditProduct(ProductForCreationDto productDto)

{

var product = \_repositoryManager.Product.GetProducts()

.FirstOrDefault(c => c.Id == productDto.Id);

product.Title = productDto.Title;

product.Quantity = productDto.Quantity;

product.Cost = productDto.Cost;

product.Description = productDto.Description;

product.Picture = productDto.Description;

\_repositoryManager.Product.Update(product);

\_repositoryManager.Save();

return RedirectToAction("Index");

}

Реализация метода удаления товара представлена в листинге 2.3.

# Листинг 2.3 – Реализация удаления товара

public IActionResult DeleteProduct(int prodcutId)

{

if (!CheckAdmin())

return BadRequest("Вы не администратор");

Product product = \_repositoryManager.Product.GetProducts().FirstOrDefault(c => c.Id == prodcutId);

if (product == null)

return BadRequest("Продукт не найден");

\_repositoryManager.Product.Delete(product);

\_repositoryManager.Save();

return RedirectToAction("Index");

}

Также администратор может добавить товар или категорию. Пример добавления товара представлен на рисунке 2.4.



**Рисунок 2.4 – Добавление товара**

Реализация метода добавления товара представлена в листинге 2.4.

# Листинг 2.4 – Реализация добавления товара

public IActionResult AddProduct()

{

return View();

}

[HttpPost]

public IActionResult AddProduct(ProductForCreationDto productDto)

{

if (!CheckAdmin())

return BadRequest("Вы не администратор");

var product = new Product

{

Title = productDto.Title,

Quantity = productDto.Quantity,

Cost = productDto.Cost,

Description = productDto.Description,

Picture = productDto.Description

};

Category category = \_repositoryManager.Category.GetCategories()

.FirstOrDefault(c => c.Id == productDto.CategoryId);

if(category != null)

product.Categories.Add(category);

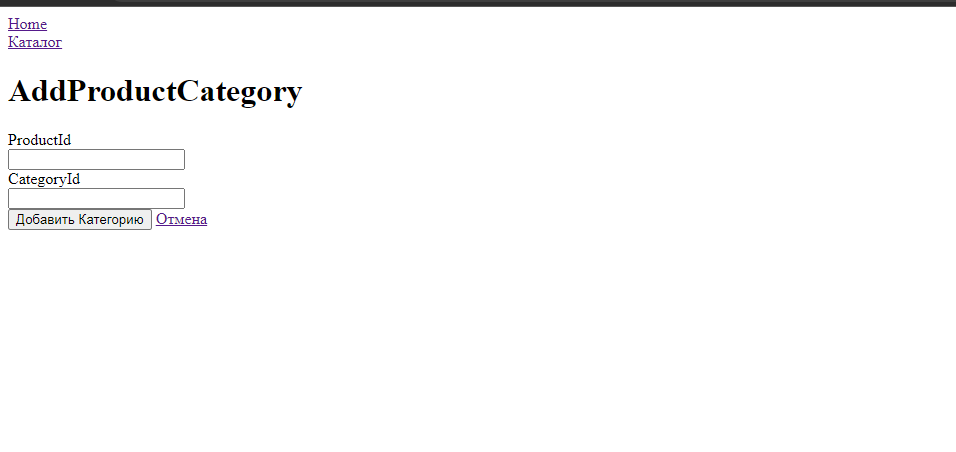
\_repositoryManager.Product.Create(product);

\_repositoryManager.Save();

return RedirectToAction("Index");

}

Администратор может добавить категорию товара или удалить её. Пример добавления категории товару представлен на рисунке 2.5.



**Рисунок 2.5 – Добавление категории товару**

На листинге 2.5 представлена реализация функции добавления категории товару.

# Листинг 2.5 – Реализация добавления категории товару

public IActionResult AddCategoryProduct()

{

return View();

}

[HttpPost]

public IActionResult AddCategoryProduct(ProductCategoryDto categoryDto)

{

if (!CheckAdmin())

return BadRequest("Вы не администратор");

var category = \_repositoryManager.Category.GetCategories()

.FirstOrDefault(c => c.Id == categoryDto.CategoryId);

var product = \_repositoryManager.Product.GetProducts()

.FirstOrDefault(c => c.Id == categoryDto.ProductId);

if (category == null || product == null)

return BadRequest("Продукт или категория не найденны");

product.Categories.Add(category);

\_repositoryManager.Product.Update(product);

return RedirectToAction("Index");

}

На листинге 2.6 предаставлена реализация функции удаления категории у товара.

# Листинг 2.5 – Реализация добавления категории товару

public IActionResult RemoveCategoryProduct()

{

return View();

}

[HttpPost]

public IActionResult RemoveCategoryProduct

(ProductCategoryDto categoryDto)

{

if (!CheckAdmin())

return BadRequest("Вы не администратор");

var category = \_repositoryManager.Category.GetCategories()

.FirstOrDefault(c => c.Id == categoryDto.CategoryId);

var product = \_repositoryManager.Product.GetProducts()

.FirstOrDefault(c => c.Id == categoryDto.ProductId);

if (category == null || product == null)

return BadRequest("Продукт или категория не найденны");

product.Categories.Remove(category);

\_repositoryManager.Product.Update(product);

return RedirectToAction("Index");

}

Остальные методы добавления, редактирования и удаления обработаны аналогичным образом.

Таким образом была реализована админ-панель интернет-магазина.

* 1. **Тестирование и отладка программного продукта**

**2.3.1 Основные концепции тестирования и отладки программного подхода**

Тестирование программного обеспечения – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом.

Отладка программного обеспечения – процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. Локализацией называют процесс определения оператора программы, выполнение которого вызвало нарушение нормального вычислительного процесса. Для исправления ошибки необходимо определить ее причину, то есть найти оператор или фрагмент, содержащие ошибку.

Тестирование и отладка программ различаются тем, что при тестировании устанавливается факт ошибки, а отладка выявляет причину и предлагает способы ее устранения.

**2.3.2 Методы испытаний**

Для тестирования программного продукта использовался метод «Тестовых случаев».

Тестовый случай (тест-кейс) – это артефакт, описывающий совокупность шагов, конкурентных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части. Каждый тестовый случай составляется после разработки общего плана тестирования. Набор из нескольких тест-кейсов называется тестовым набором.

Тестирование проводится путем сравнения ожидаемого и фактического результата. Если ожидаемый и фактический результаты совпадают, то можно считать тест успешно пройденным. Если ожидаемый и фактический результаты отличаются, то тест считается пройденным неуспешно и говорит о необходимости доработки программного средства.

# 2.3.3 Тестирование сайта

План тестирования:

* тестирование запуска приложения,
* тестирование перехода на другие формы,
* тестирование добавления данных,
* тестирование удаления данных.

Тест-кейсы представлены в таблицах 2.5-2.8.

# Таблица 2.5 - Тест-кейс «Запуск сайта»

|  |  |  |
| --- | --- | --- |
| **TC ID/Priority** | **32о001** | **Крайне высокий** |
| **IDEA**: Запуск приложения и авторизация. | | |
| **Revision History** | | |
| **Created on**: / /2021 by А. А. Мощинский | | Новый тест-кейс |
| **Execution part** | | |
| **PROCEDURE** | | **EXPECTED RESULT** |
| 1) Запустить сайт и авторизоваться под администратора | | Успешная авторизация |

Составлено автором

Фактический результат совпадает с ожидаемым. После запуска приложения открылась главная страница. Тест пройден успешно.

# Таблица 2.6 – Тест-кейс «Переход на другие страницы»

|  |  |  |
| --- | --- | --- |
| **TC ID/Priority** | **32о002** | **Средний** |
| **IDEA**: Проверка перехода на другие страницы. | | |
| **Revision History** | | |
| **Created on**: / /2021 by А. А. Мощинский | | Новый тест-кейс |
| **Execution part** | | |
| **PROCEDURE** | | **EXPECTED RESULT** |
| 1. Запустить сайт и авторизоваться, 2. Нажать на ссылку для перехода на другую форму. | | Выполнен переход на другую страницу. |

Составлено автором

Фактический результат совпадает с ожидаемым. Переход на другие страницы происходит без ошибок. Тест пройден успешно.

# Таблица 2.7 – Тест-кейс «Добавление нового товара»

|  |  |  |
| --- | --- | --- |
| **TC ID/Priority** | **32о003** | **Средний** |
| **IDEA**: Добавление товара. | | |
| **Revision History** | | |
| **Created on**: / /2021 by А. А. Мощинский | | Новый тест-кейс |
| **Execution part** | | |
| **PROCEDURE** | | **EXPECTED RESULT** |
| 1. Запустить сайт и авторизация, 2. Перейти к админ-панели, 3. Нажать на кнопку «Добавление товара», 4. Заполнить поля формы соответствующими данными, 5. Нажать кнопку «Добавить». | | Новый товар добавлен в базу данных. |

Составлено автором

Фактический результат совпадает с ожидаемым. Товар успешно добавлен в систему и отображается в таблице. Тест пройден успешно.

**Таблица 2.8– Тест-кейс «Удаление товара»**

|  |  |  |
| --- | --- | --- |
| **TC ID/Priority** | **32о004** | **Средний** |
| **IDEA**: Удаление товара. | | |
| **Revision History** | | |
| **Created on**: / /2021 by А. А. Мощинский | | Новый тест-кейс |
| **Execution part** | | |
| **PROCEDURE** | | **EXPECTED RESULT** |
| 1. Запустить сайт и авторизация, 2. Перейти к админ-панели, 3. Выбрать товар, 4. Нажать кнопку «Удалить» | | Товар удален. |

Составлено автором

Фактический результат совпадает с ожидаемым. Товар успешно удалён из системы. Тест пройден успешно.

В результате проведенных испытаний было установлено, что разработанный программный продукт отвечает предъявленным требованиям. При проведении тестирования на первом этапе были выявлены незначительные ошибки. После исправления программный продукт показал стабильность и надежность при исполнении. Проект готов к эксплуатации.