**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по дисциплине «Автоматизация технологического проектирования»

на тему: «Разработка программного обеспечения для задачи «Разузлование изделий» АСТПП»

Выполнил: студент гр. ИТП-41

Боженова А.А.

Столярова Ю.Ю.

Принял: преподаватель

Комраков В.В.

Гомель 2023

**Цель работы:** получение навыков разработки программ для реализации типовых алгоритмов типовых алгоритмов на примере задач АСТПП.

**Задание**

1. Разработать структуру таблиц БД для задачи «Разузлование изделий»
2. Реализовать и отладить программу, реализующую алгоритм разузлования с получением результатов в окне интерфейса.

**Ход работы**

Программа реализована с использованием языка программирования *C#*, фреймворка *Entity Framework* и реляционной базы данных *MS SQL Server*.

Было выделено два вида сущностей:

* деталь (*Detail*);
* детали изделия (*DetailProduct*);
* изделие (*Product*).

На рисунке 1 представлена схема таблиц базы данных.

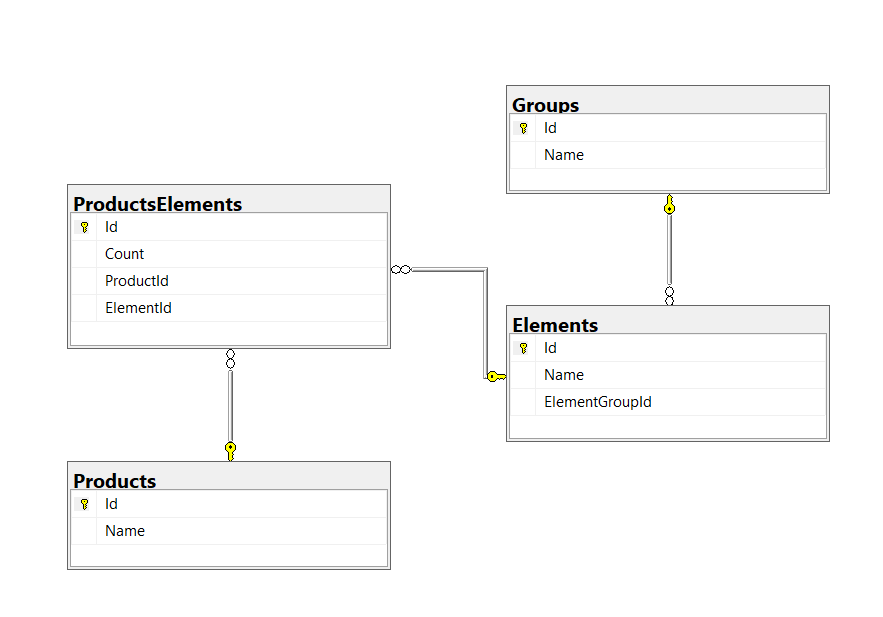


Рисунок 1 – Схема таблиц базы данных

Для пользователя данные отображены в виде таблиц с возможностью фильтрации по любым свойствам. В таблице с изделиями детали, подсвеченные красным цветом являются дефектными. Веб-страницы с таблицами сущностей представлены на рисунках 2–5.

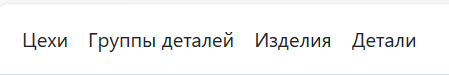


Рисунок 2 – Меню приложения

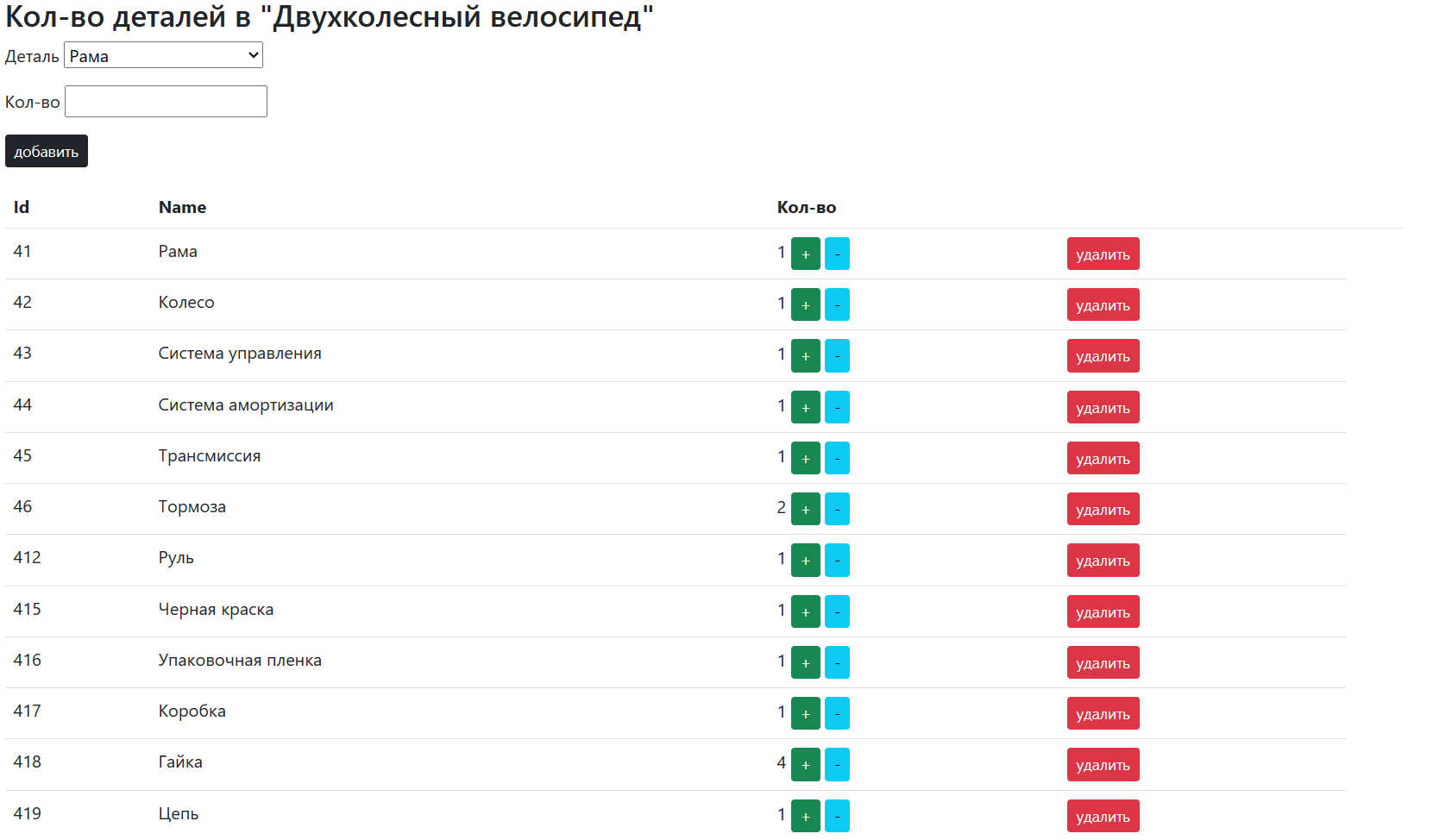


Рисунок 3 – Вывод информации об изделии и их составляющих

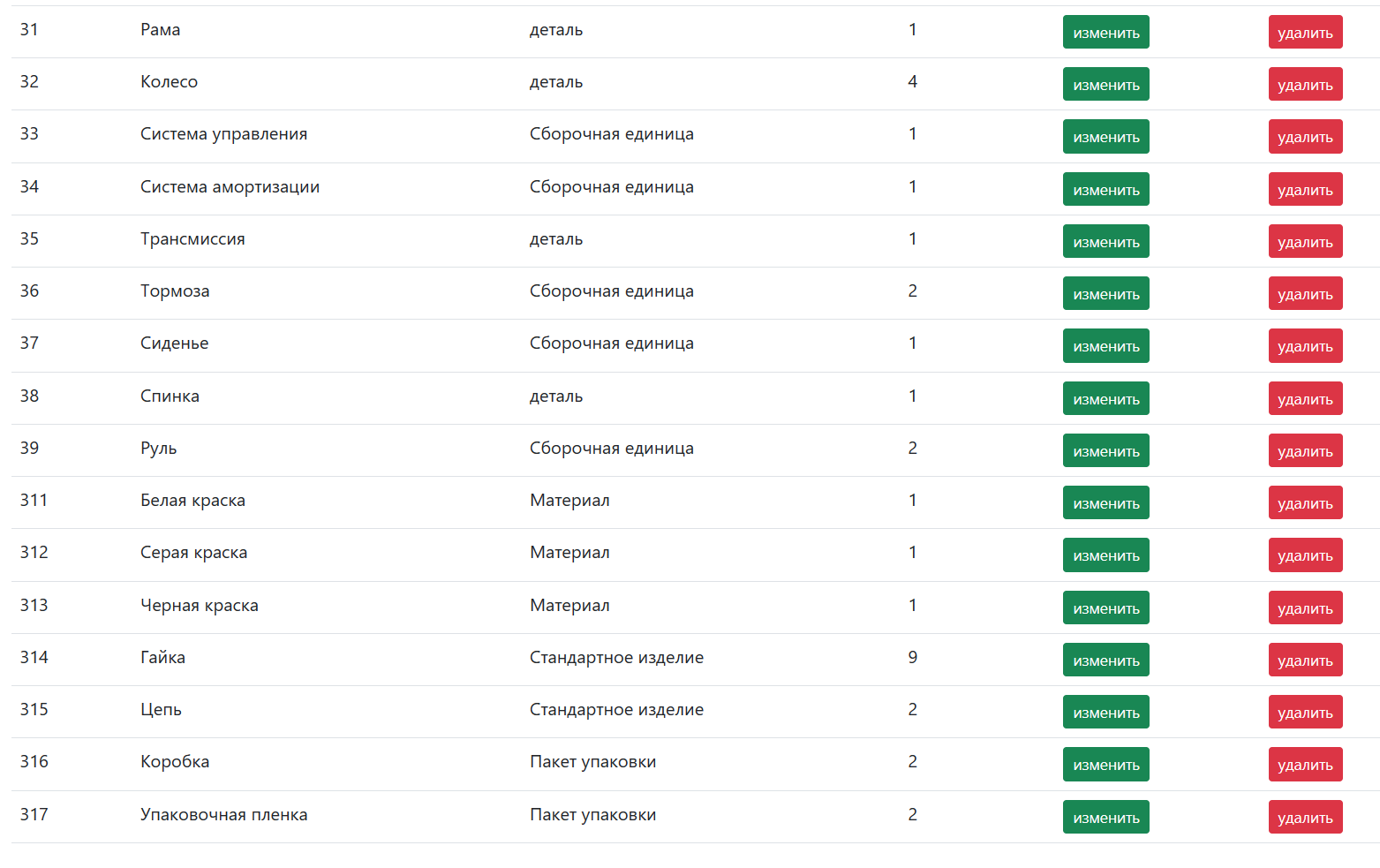


Рисунок 4 – Список деталей

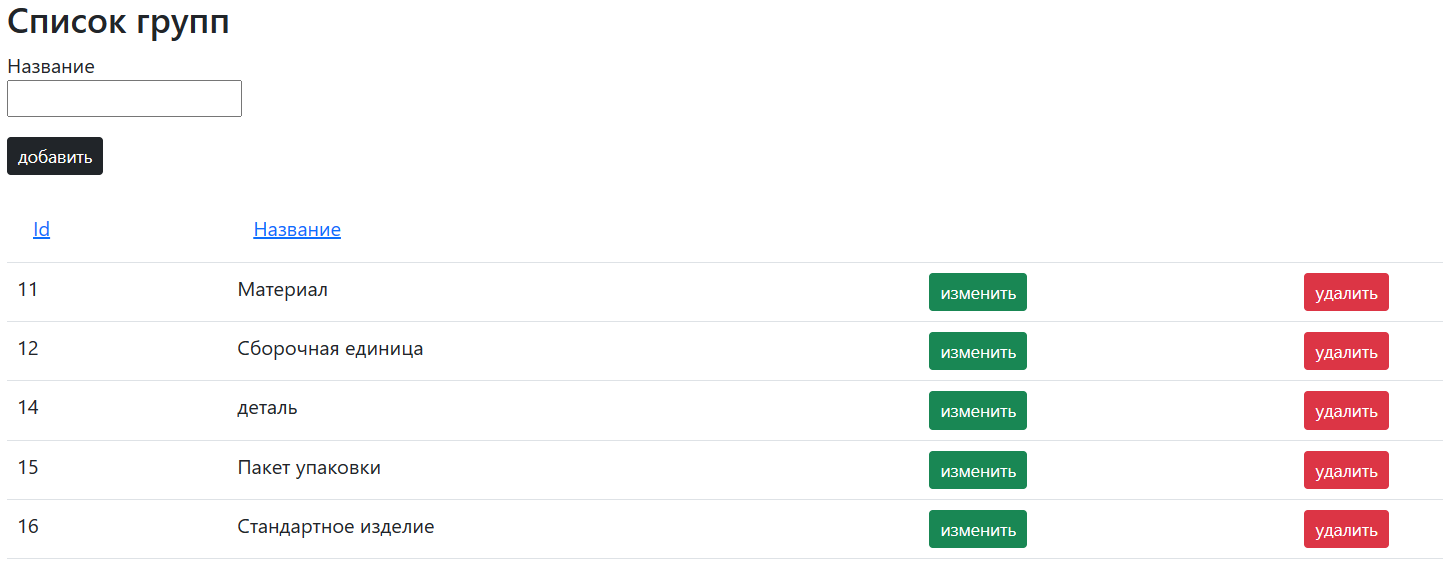


Рисунок 5 – Список групп деталей

Листинг программы представлен в приложении А.

**Вывод:** в результате выполнения лабораторной работы получены навыки разработки программ для реализации типовых алгоритмов на примере задач АСТПП.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Листинг программы**

using lab\_2.EF;

using lab\_2.EF.Context;

using lab\_2.Models.ProductElementsModels;

using lab\_2.Models.ProductModels;

using Microsoft.AspNetCore.Mvc;

namespace lab\_2.Controllers

{

public class ProductController : Controller

{

private WorkshopContext \_context;

public ProductController(WorkshopContext context)

{

\_context = context;

}

public IActionResult Index(string sortType = "id\_asc")

{

var products = \_context.Products.AsQueryable();

switch (sortType)

{

case "id\_asc":

products = products.OrderBy(x => x.Id);

break;

case "id\_desc":

products = products.OrderByDescending(x => x.Id);

break;

case "name\_asc":

products = products.OrderBy(x => x.Name);

break;

case "name\_desc":

products = products.OrderByDescending(x => x.Name);

break;

}

var filterModel = GetFilterModel();

var viewModel = new ProductIndexModel()

{

SortType = sortType,

Products = products.Where(p => p.Name.Contains(filterModel.ProductName)),

FilterModel = filterModel

};

return View("Index", viewModel);

}

public IActionResult Info(int id)

{

var viewModel = new ProductElementsIndexModel()

{

Elements = \_context.Elements.AsEnumerable(),

Products = \_context.Products.AsEnumerable(),

Product = \_context.Products.FirstOrDefault(x => x.Id == id),

ProductsElements = \_context.ProductsElements.Where(x => x.ProductId == id)

};

return View("Info", viewModel);

}

public IActionResult CreateProductElement(ProductElements productElements)

{

\_context.ProductsElements.Add(productElements);

\_context.SaveChanges();

return Info(productElements.ProductId);

}

public IActionResult AddElement(int id)

{

var productElements = \_context.ProductsElements.FirstOrDefault(x => x.Id == id);

productElements.Count++;

\_context.ProductsElements.Update(productElements);

\_context.SaveChanges();

return Info(productElements.ProductId);

}

public IActionResult RemoveElement(int id)

{

var productElements = \_context.ProductsElements.FirstOrDefault(x => x.Id == id);

productElements.Count--;

\_context.ProductsElements.Update(productElements);

\_context.SaveChanges();

return Info(productElements.ProductId);

}

public IActionResult DeleteProductElement(int id)

{

var productElements = \_context.ProductsElements.FirstOrDefault(x => x.Id == id);

\_context.ProductsElements.Remove(productElements);

\_context.SaveChanges();

return Info(productElements.ProductId);

}

private ProductFilterModel GetFilterModel()

{

string workshopName = GetStringFromSession(HttpContext, "workshopname", "workshopName");

HttpContext.Session.SetString("workshopname", workshopName);

string productName = GetStringFromSession(HttpContext, "productname", "productName");

HttpContext.Session.SetString("productname", productName);

return new ProductFilterModel()

{

WorkshopName = workshopName,

ProductName = productName

};

}

private string GetStringFromSession(HttpContext context, string key, string queryName, string defaultValue = "")

{

if (context.Request.Query[queryName].Count() > 0)

{

return context.Request.Query[queryName][0];

}

else if (context.Session.GetString(key) != null)

{

return context.Session.GetString(key);

}

else

{

return defaultValue;

}

}

public IActionResult Create(Product product)

{

\_context.Products.Add(product);

\_context.SaveChanges();

return RedirectToAction("Index");

}

public IActionResult UpdateView(int id)

{

var product = \_context.Products.FirstOrDefault(w => w.Id == id);

var viewModel = new ProductUpdateModel()

{

Product = product

};

return View("Update", viewModel);

}

public IActionResult Update(Product updatedProduct)

{

\_context.Products.Update(updatedProduct);

\_context.SaveChanges();

return RedirectToAction("Index");

}

public IActionResult Delete(int id)

{

var product = \_context.Products.FirstOrDefault(x => x.Id == id);

\_context.Products.Remove(product);

\_context.SaveChanges();

return RedirectToAction("Index");

}

}

}