

Рисунок 3 — Сильная подача охлаждающего воздуха

**Заключение.** Моделирование температуры воздуха в помещении с помощью ANSYS Fluent способствует улучшению комфорта, снижению энергопотребления и повышению эффективности зданий, а также обеспечению наилучших трудовых и жилищных условий.

#### Список цитируемых источников

1. Дудич, Л. А. Компьютерное моделирование ветровых нагрузок на здания учебных корпусов Барановичского государственного университета / Л. А. Дудич, А. В. Шах // Наука – практике : материалы III Междунар. науч.-практ. конф., Барановичи, 19 мая 2022 г. : в 3 ч. / М-во образования Респ. Беларусь, Баранович. гос. ун-т; редкол.: В. В. Климук (гл. ред.) [и др.]. — Барановичи : БарГУ, 2022. — Ч. 1. — С. 22—23.
2. Бруйка, В. А. Инженерный анализ в ANSYS Workbench : Учеб. пособ. / В. А. Бруйка [и др.]. — Самара : Самар. гос. техн. ун-т, 2010. — 271 с.
3. Дьяков, И. Ф. Метод конечных элементов в расчетах стержневых систем : учебное пособие / И. Ф. Дьяков, С. А. Чернов, А. Н. Черный. — Ульяновск : УлГТУ, 2010. — 133 с.

УДК 004.4

**О. И. Наранович, В. Д. Астроух**

*Учреждение образования «Барановичский государственный университет»,  
Барановичи, Республика Беларусь*

## РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПЛАНИРОВАНИЯ И УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ ДЕТАЛЕЙ И УЗЛОВ КОМПРЕССОРНОГО ЦЕХА

**Введение.** В ходе прохождения преддипломной практики было выявлено, что на предприятии «Барановичский станкостроительный завод» есть потребность в автоматизации процесса управления производством деталей и узлов для компрессоров. С целью улучшения этой ситуации предлагается создать специальное клиент-серверное приложение.

Разработанное клиент-серверное приложение позволит перенести всю нагрузку на сервер и разгрузить компьютеры пользователей, что сделает работу комфортной и уменьшит риски зависаний. Ещё одна причина заключается в том, что данным программным обеспечением будут пользоваться сотрудники из разных отделов, и за счёт клиент-серверной архитектуры можно сэкономить средства для улучшения технического обеспечения при обновлении только серверного обеспечения.

**Основная часть.** Для разработки клиент-серверного приложения была выбрана среда Microsoft Visual Studio. Она предоставляет широкий набор инструментов для разработки, такие как отладчик, редактор кода, дизайнер форм, а также имеет возможность интегрирования с другими инструментами, такими как Git, что делает её крайне удобной. Данная среда имеет следующие требования к системе [1]:

- операционная система: Windows 7 с пакетом обновления 1 или выше;
- процессор с тактовой частотой не ниже 1,8 ГГц., рекомендуется использовать как минимум двухъядерный процессор;
- минимум 800 Мб свободного места на жёстком диске;
- для повышения производительности рекомендовано устанавливать на SSD-накопители;
- видеоадаптер с минимальным разрешением 1280 на 720 пикселей.

Приложение разработано на языке C#, с использованием фреймворка Entity Framework и визуальной оболочкой WPF, что даёт большую возможность для реализации удобного и понятного интерфейса. Язык C#

выбран из-за возможности разработки многопоточных приложений, удобной работы с СУБД и возможности создания удобного интерфейса для пользователя. Также использование объектно-ориентированного языка позволяет расширять функционал приложения.

Для серверной части предусмотрен следующий функционал:

- взаимодействие с базой данных (создание, чтение, редактирование и удаление записей);
- обработка клиентских запросов;
- регистрация новых пользователей;
- управление аутентификацией и правами доступа пользователей.

В качестве СУБД использована MS SQL Server [2], так как она имеет большую совместимость с языком C# и имеет оптимизированную работу на ОС Windows.

Клиентская часть должна выполнять следующие функции:

- создание запросов и передача их на сервер;
- принятие ответа от сервера;
- отображение пользователю результатов от сервера;
- создание отчётов на основании полученных данных.

В качестве входных данных используются: состав изделий, маршрут изготовления деталей и план работы на месяц.

Разработаны отчёты по маршрутам изготовления и расчетам плана работы за отчётный период.

Также для безопасности данных реализована система авторизации и регистрации с хешированием паролей и временного блокирования возможности входа, если пароль введён неправильно более 3 раз.

Далее представлены UML диаграммы, демонстрирующие работу описываемого приложения. UML (с английского аббревиатура расшифровывается как Unified Modeling Language — унифицированный язык моделирования) — это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем. Код типичного приложения включает в себя тысячи строк, за связями и иерархиями которых очень непросто уследить. С помощью диаграмм UML структуру программы можно разделить на компоненты и подкомпоненты [3]. Use Case диаграммы — это визуальные представления того, как пользователи взаимодействуют с системой. Они показывают, что пользователи могут делать с системой, и как система будет реагировать на их действия. Use Case диаграммы помогают определить основные требования к системе и убедиться, что система отвечает потребностям пользователей [4]. Use Case диаграмма представлена на рисунке 1.

В качестве сущности «Администратор» на предприятии может выступать системный администратор или другой сотрудник, имеющий доступ к серверу и имеющий полномочия управления правами доступа других сотрудников.

В качестве сущности «Работник» может выступать мастер или начальник цеха, который руководит его работой и на котором лежит ответственность за предоставляемые материалы и результаты работы, выраженные в отчётах.

В качестве сущности «Конструктор» на предприятии может выступать сотрудник, ответственный за разработку и утверждение в производство изделий.

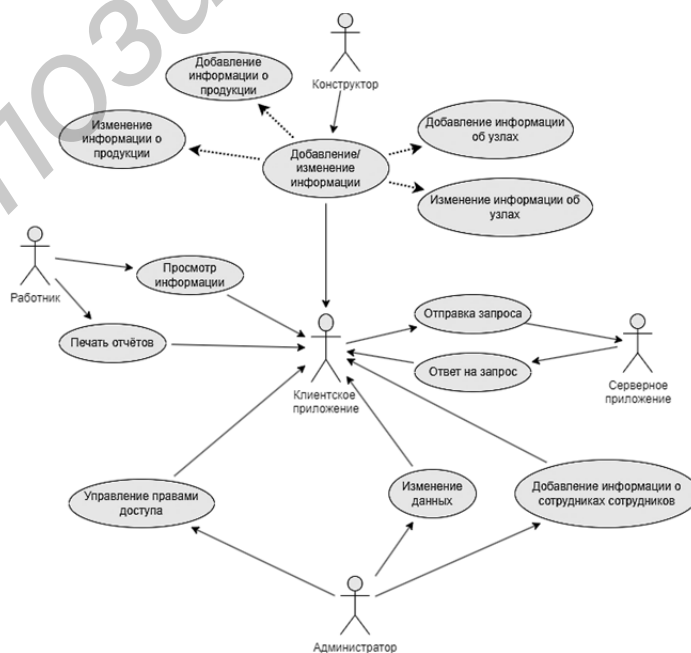


Рисунок 1 — Диаграмма Use Case

Sequence диаграммы — это визуальные представления того, как объекты взаимодействуют друг с другом во времени. Они показывают, какие сообщения отправляют и получают объекты, и когда это происходит. Sequence диаграммы помогают разработчикам понять порядок операций в системе и тем самым помогают в проектировании и реализации системы [5]. Sequence диаграмма представлена на рисунке 2.

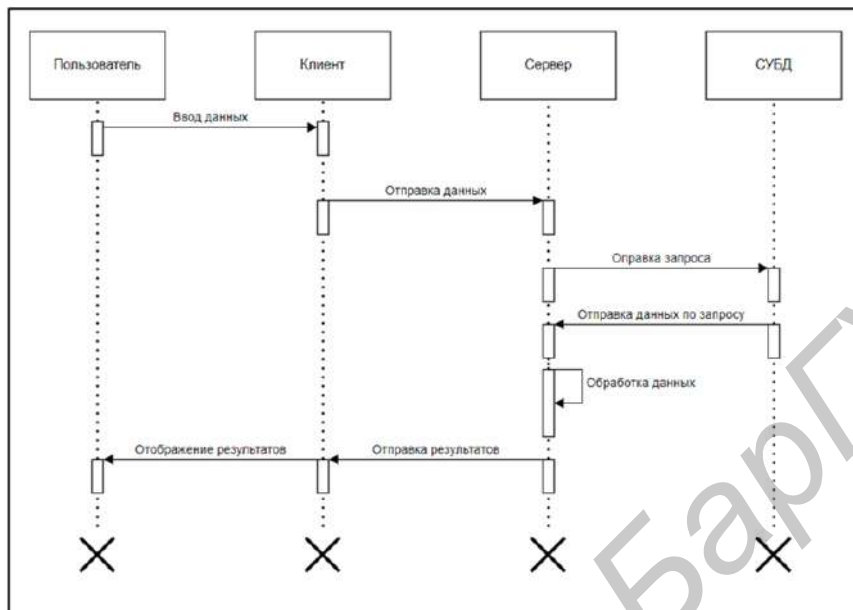


Рисунок 2 — Диаграмма Sequence

В этой диаграмме отображается взаимодействие между клиентом, сервером и СУБД.

В приложении используется база данных, которая служит основой для хранения и обработки информации. Концептуальная и физическая модели базы данных изображена на рисунках 3 и 4.

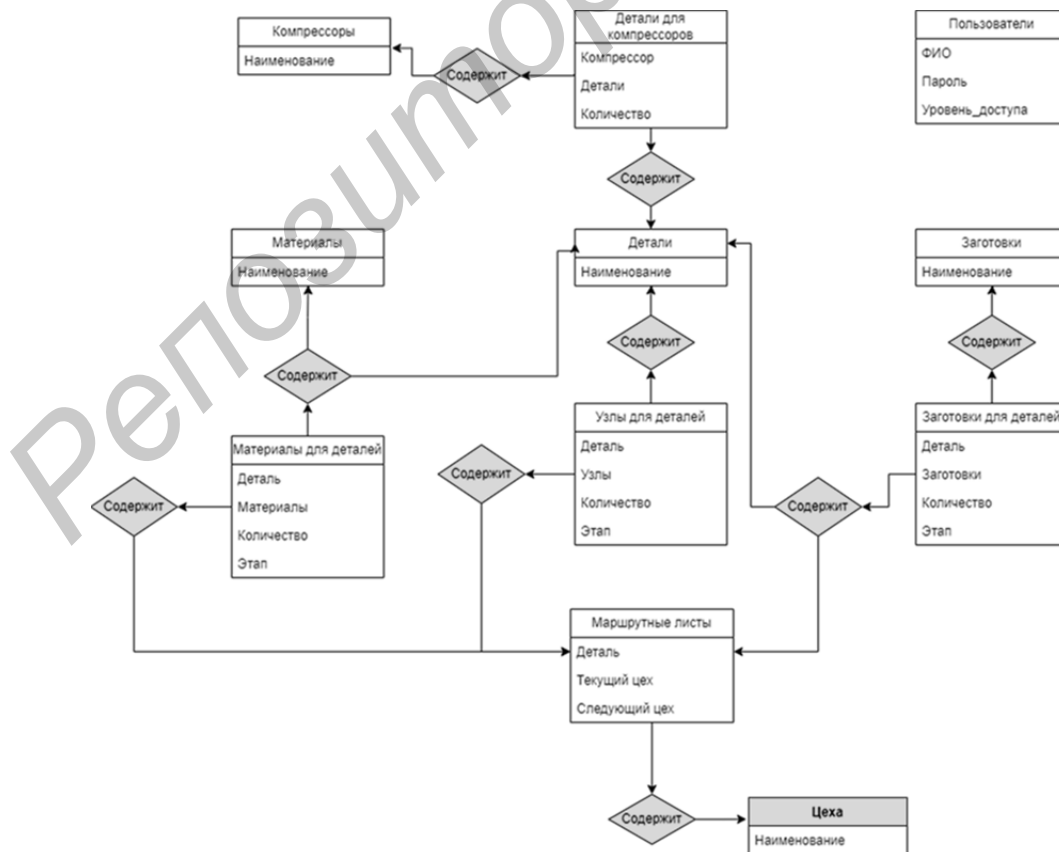


Рисунок 3 — Концептуальная модель базы данных

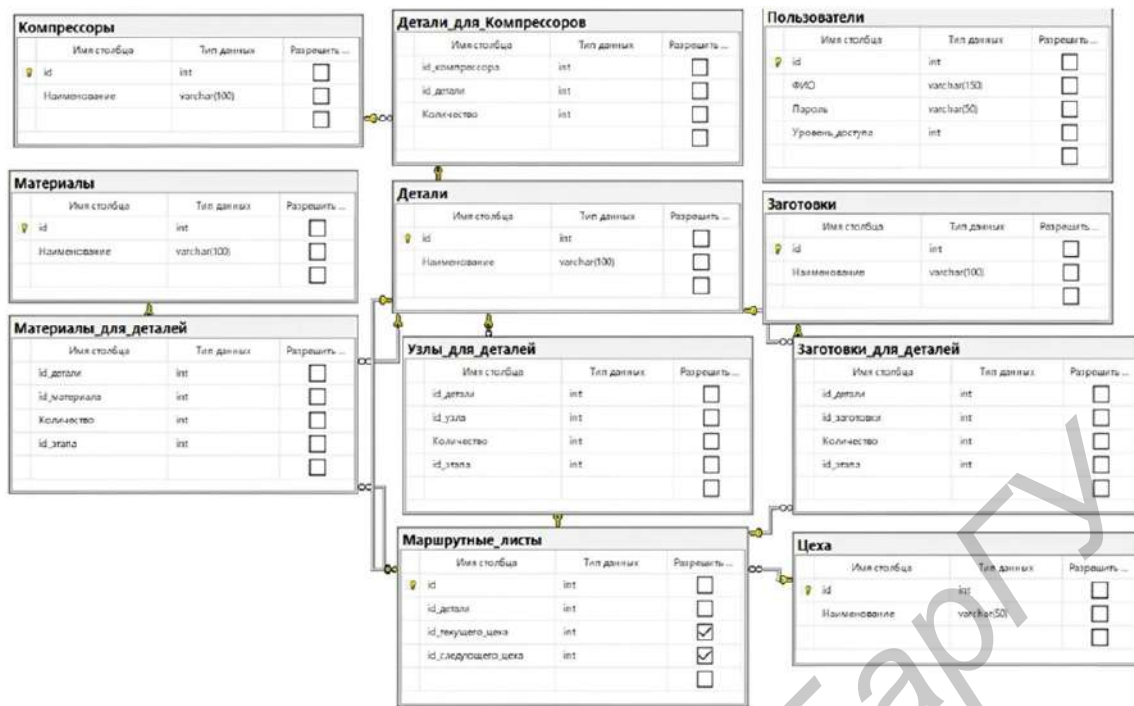


Рисунок 4 — Физическая модель базы данных

Таблица «Пользователи» хранит информацию о пользователях, включая ФИО, пароль и уровень доступа пользователя.

Таблица «Цеха» хранит информацию о цехах, которые задействованы в производстве компрессоров.

Таблица «Компрессоры» хранит информацию о производимых компрессорах.

Таблица «Детали» хранит информацию о необходимых для компрессоров деталях.

Таблицы «Материалы для деталей», «Узлы для деталей» и «Заготовки для деталей» хранят информацию о материалах, узлах, и заготовках, требуемых для производства различных деталей.

Таблица «Маршрутные листы» хранит информацию о порядке передачи между цехами детали в ходе её производства.

Таблица «Детали для компрессоров» хранит информацию о том, какие детали входят в состав определённого компрессора.

**Заключение.** Представленное приложение позволит автоматизировать процесс управления производством деталей и узлов для компрессоров на предприятии. Разработанная база данных обеспечит эффективное управление данными и предоставит быстрый доступ к информации в ходе производства компрессоров и деталей к ним.

#### Список цитируемых источников

1. Требования к системе семейства продуктов Visual Studio [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/visualstudio/releases/2019/system-requirements>. — Дата доступа: 29.04.2024.
2. Возможности SQL Server 2019 [Электронный ресурс]. — Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019-features>. — Дата доступа: 29.04.2024.
3. Знакомство с разными видами диаграмм UML [Электронный ресурс]. — Режим доступа: <https://www.lucidchart.com/blog/ru/types-of-UML-diagrams>. — Дата доступа: 30.04.2024.
4. Проектирование Use Case диаграммы. Определение функциональных возможностей системы [Электронный ресурс]. — Режим доступа: <https://nationalteam.worldskills.ru/skills/proektirovanie-use-case-diagrammy-opredelenie-funktionalnykh-vozmozhnostey-sistemy/>. — Дата доступа: 30.04.2024/
5. Диаграмма последовательности (Sequence diagram) [Электронный ресурс]. — Режим доступа: [https://flexberry.github.io/ru/fd\\_sequence-diagram.html](https://flexberry.github.io/ru/fd_sequence-diagram.html). — Дата доступа: 30.04.2024/

УДК 004.4

О. И. Наранович, П. А. Сенько

Учреждение образования «Барановичский государственный университет»,  
Барановичи, Республика Беларусь

## ПРИМЕНЕНИЕ МНОГОПОТОЧНОСТИ В ПРИЛОЖЕНИИ КЛИЕНТ-СЕРВЕРНОЙ АРХИТЕКТУРЫ ДЛЯ РЕШЕНИЯ ПРОИЗВОДСТВЕННОЙ ЗАДАЧИ

**Введение.** Целью работы является разработка приложения клиент-серверной архитектуры для решения производственной задачи с применением многопоточности. Производственная задача представлена задачей