Интерфейс главного окна разделен на 2 части. Слева располагается панель навигации для переключения между разделами, справа находится основное рабочее окно, оно служит для работы с данными, новые заказы поступают по мере заказов клиентов на веб сайте и отмечаются как новые. Таблица снабжена строкой поиска, что позволяет быстро отыскать текст по его тегу, статусу или клиенту, есть фильтры по статусу заказа. Возможность зайти в личный кабинет, некоторый функционал открывается в зависимости от прав аккаунта, который выдаётся при регистрации.

Функционал редактирования записей реализован через модальные окна. При выборе элемента из таблицы поля формы автоматически заполняются текущими значениями, что сокращает время на ручной ввод и снижает риск ошибок. Встроенная валидация данных в режиме реального времени блокирует некорректные изменения до их сохранения.

Организация данных о товарах реализована через иерархическую модель категорий, основанную на таблицах ProductCategory, ProductSubcategory и Product. ProductCategory определяет основные категории товаров, а ProductSubcategory задает подкатегории, связанные с родительскими категориями через внешний ключ. Таблица Product содержит детальную информацию о товарах: уникальный артикул, цену, габариты (вес, длина, ширина) и связь с подкатегорией. При удалении категории все её подкатегории автоматически удаляются, а товары сохраняются с обнуленной связью, что предотвращает потерю данных. Такая структура обеспечивает гибкую классификацию, упрощает фильтрацию товаров в интерфейсе и поддерживает масштабируемость ассортимента. Например, при добавлении новой газонокосилки система автоматически сопоставляет её с подкатегорией «Садовый инвентарь», а интерфейс отображает товар в соответствующем разделе, используя методы *load_products_by_subcategory()* и *validate_product_attributes()* для контроля целостности данных (рисунок 3).

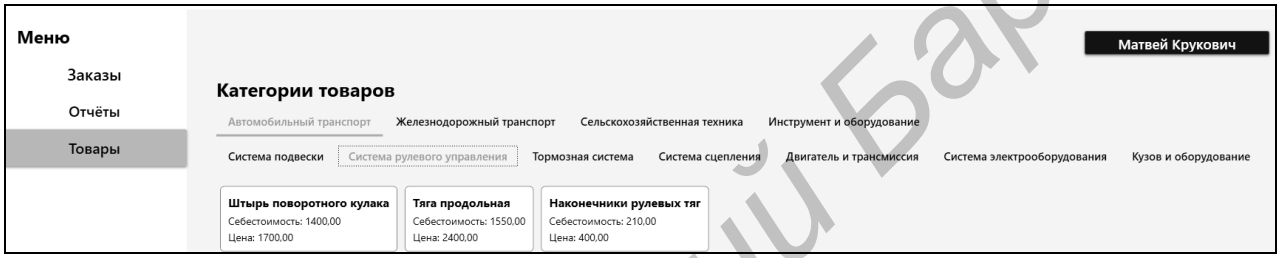


Рисунок 3 — Окно склад товаров

**Заключение.** Разработанное десктопное приложение представляет собой специализированное решение для автоматизации учета и анализа предварительных заказов в сегментах розничной торговли и услуг. Использование технологий C#, .NET WPF и MySQL позволило создать производительную систему с адаптивным интерфейсом, ориентированным на сотрудников без технической подготовки.

Система оптимизирует ключевые бизнес-процессы: от сокращения времени обработки заказов на 30—40 % (за счет автоматического предзаполнения форм и валидации) до минимизации логистических издержек через прогнозирование пикового спроса. Перспективы развития включают внедрение модуля динамического ценообразования, интеграцию с мобильными приложениями клиентов и расширение аналитической платформы с использованием ML-алгоритмов для персонализации предложений.

**Список цитируемых источников**

1. MY SQL: документация по функционалу и преимуществам. — URL: https://dev.mysql.com/doc/ (дата обращения: 05.05.2025).
2. Windows Presentation Foundation (WPF): руководство для разработчиков. — URL: https://learn.microsoft.com/en-us/dotnet/-desktop/wpf/overview/ (дата обращения: 05.05.2025).
3. Хеширование паролей с использованием SHA256 и криптографической соли. — URL: https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/ (дата обращения: 05.05.2025).

УДК 004.514

**К. А. Крутько, О. И. Наранович**
*Учреждение образования «Барановичский государственный университет»,*
*Барановичи, Республика Беларусь*

## РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ СОЗДАНИЯ И УЧЕТА ЗАЯВОК НА РЕМОНТ И ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ ОБОРУДОВАНИЯ НА ПРЕДПРИЯТИИ ЗАО «АТЛАНТ» — БСЗ

**Введение.** На современных промышленных предприятиях своевременное техническое обслуживание и оперативное реагирование на отказ оборудования играют ключевую роль в обеспечении непрерывности производства и снижении простоев. При большом количестве станков и сложных технологических про-

Репозиторий БарГУ

цессов ручное оформление заявок на ремонт и контроль их выполнения нередко сопровождаются потерей информации, длительным согласованием и проблемами с отслеживанием статуса работ. Это приводит к задержкам, увеличению непредвиденных простоев и росту затрат на содержание оборудования.

Разработка автоматизированной системы для создания и учёта заявок на ремонт и техническое обслуживание оборудования позволит значительно ускорить процесс регистрации заявок. Внедрение такой информационной платформы способствует сокращению трудозатрат на подготовку и обработку заявок, повышению точности учётных данных, а также оперативному анализу загруженности ремонтных служб и состояния оборудования. В условиях постоянного роста требований к надёжности и эффективности производства автоматизация процессов обслуживания становится важным инструментом повышения общей производительности предприятия.

**Основная часть.** Целью данной работы является разработка автоматизированной системы для создания и учёта заявок на ремонт и техническое обслуживание оборудования на предприятии, направленной на повышение операционной эффективности, снижение простоев и прозрачности внутренних процессов. Такая система должна обеспечивать интеграцию ключевых функций, охватывающих автоматизацию процессов добавления, контроль сроков проведения ремонтных работ и обслуживания, интеграцию данных о состоянии оборудования, истории обслуживания, формирование аналитической отчетности. Основное внимание уделяется упрощению рутинных операций, минимизации человеческого фактора и обеспечению актуальности данных для принятия решений.

При разработке системы была задача по созданию информационной среды с интуитивно понятным интерфейсом, возможностью работы с базой данных, формирования необходимой отчетности.

При проектировании системы в качестве системы управления базами данных была выбрана MySQL — популярная реляционная база данных (далее — БД) с открытым исходным кодом, обеспечивающая высокую производительность и масштабируемость даже при работе с большими объемами данных. Ее ключевые преимущества включают:

1. Бесплатное использование для некоммерческих проектов и гибкие лицензии для предприятий.
2. Простота установки и настройки, а также обширная экосистема инструментов и плагинов.
3. Надежная безопасность за счет встроенных механизмов шифрования и аутентификации.
4. Высокая скорость обработки запросов, что критично для систем с интенсивной нагрузкой.
5. Совместимость с Entity Framework (далее — EF) — ORM-фреймворком, упрощающим взаимодействие приложения с базой данных через объектно-ориентированную модель [1].

Для реализации логики использовался EF Core — современный фреймворк для работы с БД в экосистеме .NET. EF обеспечил:

1. Автоматическую генерацию SQL-запросов и миграций.
2. Удобное маппирование сущностей C# на таблицы MySQL.
3. Поддержку LINQ для написания типобезопасных запросов.
4. Интеграцию с транзакциями и кэшированием [2].

В качестве фреймворка для разработки был использован Windows Presentation Foundation (далее — WPF), представляющий платформу пользовательского интерфейса для создания клиентских приложений для настольных систем. WPF использует расширяемый язык разметки для приложений (XAML), чтобы предоставить декларативную модель для программирования приложений [3].

В качестве среды разработки была использована Microsoft Visual Studio 2022 являющаяся мощным средством разработчика, которое можно использовать для выполнения всего цикла разработки в одном месте. Это комплексная интегрированная среда разработки, которую можно использовать для записи, редактирования, отладки и сборки кода.

Данная система была выбрана по следующим причинам:

– Мощные средства написания кода и функции.

– Поддержка нескольких языков, в состав которых входит используемый для создания приложения C#.

– Интеграция управления версиями — позволяющая контролировать версии приложения и эффективнее справляться с ошибками.

– Разработка с поддержкой искусственного интеллекта — позволяющая значительно увеличить скорость разработки [4].

При запуске приложения пользователь попадает на окно авторизации (рисунок 1), где требуется ввести логин и пароль. Данные проверяются через EF, который выполняет запрос к базе данных MySQL, сопоставляя введенные значения с хешированными записями в таблице пользователей. При успешной проверке данных система определяет роль пользователя. Интерфейс формируется на основе роли, загружая только те элементы управления (кнопки, меню, вкладки), которые разрешены для указанной роли. На рисунке 2 показан интерфейс администратора с доступом к функциям приложения. В случае ошибки ввода появляется уведомление о совпадения логина или пароля.

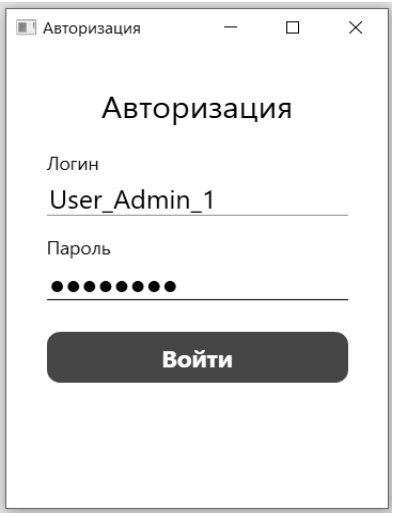


Рисунок 1 — Окно авторизации

Репозиторий БарГУ

Хеширование паролей в системе реализовано с помощью алгоритма Bcrypt, который обеспечивает высокий уровень безопасности и устойчивость к различным видам атак. Bcrypt использует адаптивную функцию хеширования на основе шифра Blowfish и встроенный механизм «соли», что позволяет генерировать уникальный хеш даже для одинаковых паролей. Кроме того, за счёт возможности настройки параметра «cost» (числа раундов обработки) можно гибко регулировать сложность вычислений, повышая устойчивость к перебору при увеличении вычислительных мощностей атакующего. В результате применение Bcrypt гарантирует защиту пользовательских паролей от взлома и утечек. Хеширование выполняется на стороне клиента перед отправкой данных, что исключает хранение паролей в открытом виде и снижает риски взлома.

Окно разделено на навигационную панель, которая предназначена для выбора необходимой вкладки и основную часть, которая предоставляет доступ к данным, а также просмотр, добавление, редактирование, удаление записей. Структура навигационной панели различается в зависимости от роли пользователя. В верхней части находиться строка поиска для быстрого нахождения необходимых записей по содержимому любой ячейки в таблицах, кнопка «Добавить» и кнопка «Фильтры».

В случаи редактирования (двойной клик по записи) или добавления записей (нажатие кнопки «Добавить»), будет открыто новое окно (рисунок 3) со семи необходимыми полями. При редактировании все поля заполняются в соответствии с выбранной записью в таблице.

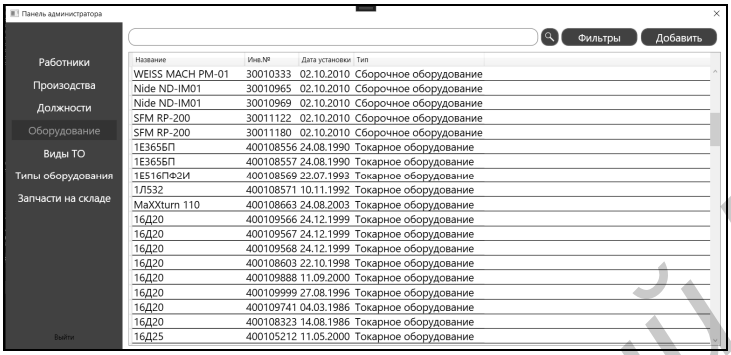

Рисунок 2 — Окно администратора

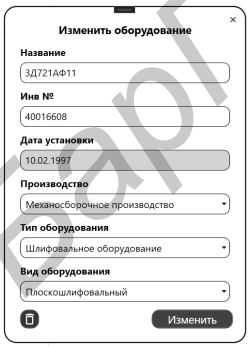

Рисунок 3 — Окно редактирования данных

**Заключение.** В процессе разработки было создано приложение, предназначенное для автоматизации ремонтной деятельности предприятия. Использование технологий современной разработки позволило обеспечить стабильную работу системы, удобство для пользователей и гибкость для дальнейшего расширения функционала.

Благодаря ипользованию WPF был создан интуитивно понятный интерфейс с адаптивным дизайном. Использование архитектурного паттерна MVVM обеспечило строгую разделённость представления и бизнес-логики, что упростило сопровождение и расширение проекта. Интеграция EF Core с MySQL реализовала объектно-реляционное отображение (ORM), что минимизировало ручное написание SQL-запросов и ускорило взаимодействие с данными через LINQ-операторы. В системе реализовано разграничение прав доступа, гарантирующее защиту информации от неавторизованных пользователей.

**Список цитируемых источников**

1. Что такое MySQL. — URL: https://mchost.ru/articles/chto-takoe-mysql (дата обращения: 06.05.2025).
2. Entity Framework Core. — URL: https://learn.microsoft.com/ru-ru/ef/core/ (дата обращения: 06.05.2025).
3. Начало работы (WPF). — URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/getting-started/?view=netframeworkdesktop-4. (дата обращения: 06.05.2025).
4. Что такое Visual Studio? — URL: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022 (дата обращения: 06.05.2025).

УДК 004.45

**М. О. Лазарчук, Г. М. Раковцы**
*Учреждение образования «Барановичский государственный университет», Барановичи, Беларусь*

## РАЗРАБОТКА ОБУЧАЮЩЕЙ ПРОГРАММЫ ДЛЯ ЗАПОМИНАНИЯ ИНОСТРАННЫХ СЛОВ НА ОСНОВЕ C++ BUILDER

**Введение.** В современном мире изучение иностранных языков приобретает всё большую значимость в связи с процессами глобализации и расширением международных коммуникаций. В условиях цифровой трансформации образования особую актуальность приобретают интерактивные методы обучения, позволяющие преодолеть ограничения традиционных подходов.

Репозиторий БарГУ