

Д. А. Бовт
Учреждение образования «Барановичский государственный университет»,
Барановичи, Республика Беларусь

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТА ТОВАРОВ НА СКЛАДЕ ООО «ТУЛС БАЙ»

Введение. Организации, занимающиеся оптовой и розничной продажей строительного и садового оборудования, постоянно сталкиваются с необходимостью эффективного управления запасами на складе. Точное отслеживание различных товаров, организация логистических процессов (приемка и отправка) и поддержание актуальной информации о наличии являются ключевыми аспектами для бесперебойной работы.

В условиях использования ручных методов или разрозненных электронных таблиц нередко возникают ошибки в данных и задержки в операциях, что приводит к снижению операционной эффективности. Внедрение цифровых технологий для автоматизации учета на складе способствует значительному улучшению точности данных, сокращению времени, затраченного на рутинные задачи, и обеспечивает полный контроль над движением товаров и материалов. В условиях усиливающейся конкуренции и давления на повышение скорости обработки заказов и точности информации о наличии товаров, применение современных информационных систем для управления запасами становится необходимым инструментом для оптимизации логистики и поддержки процесса принятия управленческих решений.

Основная часть. Ключевой целью настоящей работы является создание и описание интегрированного десктопного решения, разработанного для комплексного управления процессами складского учета на предприятиях, специализирующихся на строительном оборудовании. Система призвана консолидировать основные функции складской логистики, включая ведение детального каталога продукции, мониторинг уровней запасов, регистрацию полного жизненного цикла товаров на складе и администрирование доступа пользователей. Приоритет отдавался созданию инструмента, который бы упрощал ежедневные задачи персонала, снижал риск ошибок, связанных с человеческим фактором, и предоставлял руководству актуальную картину состояния склада для эффективного планирования и контроля.

В ходе разработки приложения ставилась задача формирования единой информационной платформы с интуитивно воспринимаемым графическим интерфейсом. Система должна обеспечивать надежное взаимодействие с централизованной базой данных, предоставлять гибкие возможности для поиска и фильтрации информации, а также включать многоуровневую систему разграничения пользовательских прав доступа.

Для хранения и управления данными была выбрана система управления базами данных Microsoft SQL Server. Данное решение широко применяется в корпоративном секторе и известно своей надежностью, производительностью при работе со значительными объемами информации и развитыми средствами обеспечения безопасности и целостности данных. Масштабируемость MS SQL Server позволяет системе эффективно функционировать как при небольшом, так и при постоянно растущем количестве номенклатуры и операций. Подключение к базе данных из приложения реализовано с использованием стандартного интерфейса ODBC через библиотеку pyodbc, обеспечивающую стабильное и производительное взаимодействие [1].

Ключевые преимущества MS SQL Server для данного проекта:

1. Производительность и масштабируемость. Способность обрабатывать большие объемы транзакций и данных без существенной потери скорости.

2. Надежность. Встроенные механизмы резервного копирования, восстановления и обеспечения отказоустойчивости.

3. Безопасность. Гранулированная система управления доступом на уровне сервера и базы данных.

4. Экосистема. Широкая поддержка и развитые инструменты администрирования и разработки.

Основным языком программирования для разработки приложения был выбран Python, а для создания пользовательского интерфейса использовался фреймворк PyQt5. Эта технологическая связка была выбрана по ряду причин:

1. Эффективность разработки. Python позволяет быстрее реализовывать бизнес-логику благодаря простому синтаксису и богатой экосистеме библиотек.

2. Возможности GUI. PyQt5 предоставляет обширный набор стандартных виджетов, механизмы сигналов и слотов, а также поддержку таблиц стилей Qt (аналог CSS), что позволило реализовать гибкую настройку внешнего вида, включая переключение тем оформления (светлая/темная) [2].

3. Библиотеки для данных. Наличие мощных библиотек, таких как pandas и openpyxl, значительно упрощает реализацию функций импорта/экспорта данных в популярные форматы, например, Microsoft Excel.

4. Кроссплатформенность (потенциал). PyQt5 позволяет создавать приложения, которые с минимальными доработками могут работать на различных операционных системах.

При первом запуске приложения пользователю предлагается пройти аутентификацию через специализированное диалоговое окно (рисунок 1). Введенные логин и пароль сверяются с данными в таблице Users. Для защиты учетных данных пароли хранятся в базе исключительно в виде хеш-сумм, созданных с помощью криптографически стойкого алгоритма bcrypt, который включает использование «соли» и адаптивной сложности, предотвращая простое восстановление пароля даже в случае компрометации хеша. При успешном подтверждении учетных данных открывается главное окно приложения (рисунок 2) [3].

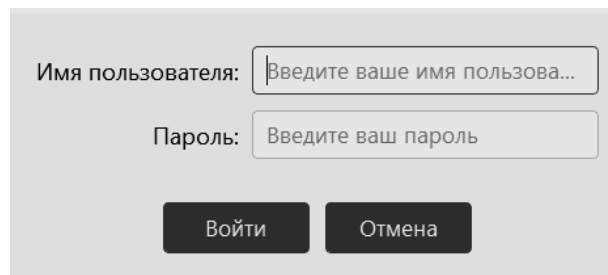


Рисунок 1 — Окно авторизации

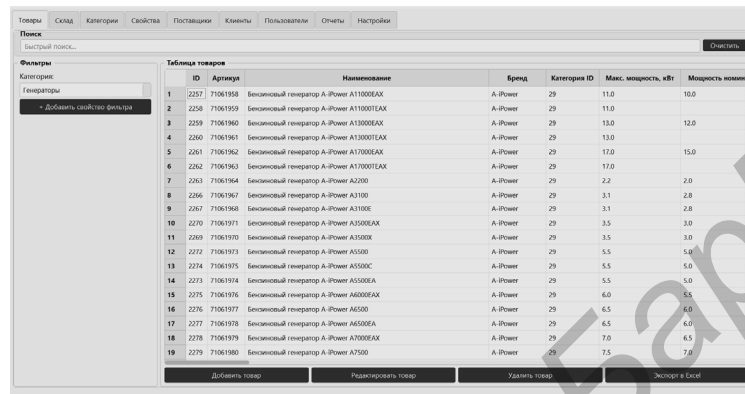


Рисунок 2 — Основное окно приложения

Интерфейс главного окна построен на основе вкладок (QTabWidget), предоставляя пользователю быстрый доступ к основным разделам: управлению каталогом товаров («Товары»), отслеживанию остатков и операций («Склад»), ведению справочников («Категории», «Свойства», «Поставщики», «Клиенты»), администрированию учетных записей («Пользователи»), генерации отчетов («Отчеты» — на этапе планирования) и конфигурации приложения («Настройки»).

Каждая основная вкладка содержит интерактивную таблицу (QTableWidget) для визуализации данных, поле для быстрого поиска по содержанию, а также панель с кнопками для выполнения стандартных CRUD-операций (создание, чтение, обновление, удаление) и специфических действий (например, складские операции, экспорт). Для детализированного отбора данных предусмотрены панели фильтров, позволяющие выбирать записи по категории, бренду или значениям определенных свойств товара. Функционал сортировки по любому столбцу активируется кликом на его заголовок.

Процессы добавления новых или модификации существующих записей вынесены в модальные диалоговые окна (QDialog). При редактировании поля диалога автоматически заполняются значениями выбранной строки таблицы, что минимизирует ошибки ввода и ускоряет работу. В диалогах предусмотрены механизмы валидации вводимой информации.

Контроль доступа реализован через ролевую модель (RBAC), опирающуюся на структуру таблиц Permissions, RolePermissions и UserPermissions в базе данных. Permissions определяет список всех возможных действий в системе (например, PRODUCT_VIEW, PRODUCT_EDIT, STOCK_RECEIPT_ADD, USER_DELETE). RolePermissions назначает стандартный набор этих прав для каждой определенной роли («Admin», «Manager», «WarehouseWorker»). Таблица UserPermissions позволяет администратору предоставлять или отзывать конкретные права отдельным пользователям, перекрывая настройки их роли. Перед отображением элементов интерфейса или выполнением критических операций приложение вызывает метод check_permission для проверки полномочий текущего пользователя. Недоступные функции или разделы скрываются или блокируются (элементы становятся неактивными) с помощью метода apply_role_permissions, адаптируя интерфейс под конкретного сотрудника (рисунок 3).

Заключение. Разработанное десктопное приложение предоставляет комплексное решение для автоматизации учета и управления запасами строительного оборудования и материалов. Выбранный технологический стек — Python, PyQt5, MS SQL Server — позволил создать гибкую, функциональную и достаточно производительную систему с удобным графическим интерфейсом.

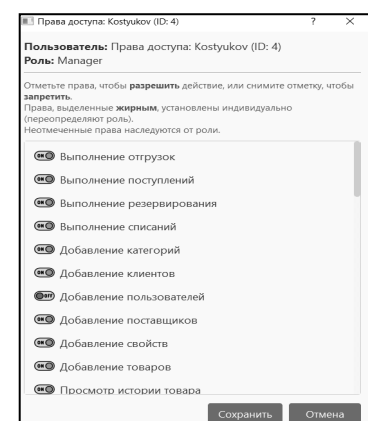


Рисунок 3 — Окно разграничения доступа

Централизация данных, структурированное ведение каталога товаров со специфичными свойствами, отслеживание полного цикла складских операций и внедрение ролевой системы разграничения доступа являются ключевыми преимуществами разработанного решения. Функции поиска, фильтрации, сортировки, экспорта данных и настройки внешнего вида повышают удобство повседневного использования. Система спроектирована с учетом возможности дальнейшего масштабирования и добавления нового функционала, в первую очередь — модуля формирования аналитических отчетов.

Список цитируемых источников

1. База данных MS SQL Server : что это, зачем нужна, как появилась и чем хороша. — URL: <https://skillbox.ru/media/code/baza-dannykh-ms-sql-server-cto-eto-zachem-nuzhna-kak-poyavilas-i-chem-khorosha> (дата обращения: 05.05.2025).
2. PyQt. — URL: <https://blog.skillfactory.ru/glossary/pyqt/> — (дата обращения: 05.05.2025).
3. Сохранение случайной соли при хешировании паролей bcrypt. — URL: <https://sky.pro/wiki/java/sokhranenie-sluchaynoy-soli-pri-kheshirovanii-paroley-bcrypt/> — (дата обращения: 05.05.2025).

УДК 004.438

М. А. Вареник, А. А. Новогран

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

РАЗРАБОТКА ИГРЫ «СОКОБАН» С ПРИМЕНЕНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Введение. Игра «Сокобан», хотя и имеет довольно долгую историю, продолжает оставаться популярной среди людей различного возраста. Впервые выпущенная в 1981 году, она представляет собой головоломку, в которой игрок управляет персонажем, перемещая ящики по лабиринту с целью разместить их на заданных местах. Суть игры заключается не только в математической логике, но и в пространственном мышлении, что делает её отличным инструментом для тренировки ума.

Разработка игры «Сокобан» в рамках современного исследования представляет собой актуальную тему по нескольким причинам. Во-первых, с точки зрения педагогики, это отличный способ вовлечения студентов в изучение объектно-ориентированных концепций и принципов программирования. Во-вторых, игра служит примером доступного для понимания, но в то же время требующего глубокого анализа приложения, что делает её идеальной кандидатурой для изучения в академических кругах. В-третьих, учитывая растущий интерес к играм, разработка нового проекта в жанре головоломок может привнести свежий взгляд на классическую механику.

Объектно-ориентированный подход разрабатывает архитектуру игры, где каждый элемент рассматривается как объект с определенными свойствами и методами. Такой подход не только оптимизирует процессы разработки, но и облегчает дальнейшую модификацию и масштабирование проекта. Его основные преимущества следующие [1]:

1. Инкапсуляция позволяет скрыть внутренние детали реализации и защищает данные от несанкционированного доступа. Это упрощает поддержку и модификацию кода.
2. Наследование помогает создавать новые классы на основе существующих, что способствует вторичному использованию кода и уменьшает дублирование.
3. Полиморфизм позволяет объектам быть обработанными по общему интерфейсу, что увеличивает гибкость и расширяемость кода.
4. Более высокое качество кода, благодаря чему код становится легче читать, тестировать и отлаживать.
5. Объектно-ориентированный подход хорошо подходит для проектирования сложных программных систем, где важно управление состоянием объектов.

Использование языка C++ позволяет создавать производительные и эффективные решения, давая разработчикам гибкость в реализации сложных игровых механик и взаимодействий. Одним из основных преимуществ C++ является его способность работать на низком уровне, что способствует оптимизации кода и снижению затрат на ресурсы во время выполнения программы.

В качестве среды разработки был выбран Qt Creator с фреймворком Qt, который обладает рядом преимуществ по сравнению с другими фреймворками и графическими библиотеками на C++ [2].

Основная часть. Диаграмма классов, разработанного приложения игры, представлена на рисунке 1.

Класс Game является основным элементом графического пользовательского интерфейса игры. Его задача — обеспечить пользователю интуитивно понятный и компактный интерфейс для взаимодействия с игрой: главное меню, меню выбора встроенных или пользовательских уровней, доступ к редактору уровней, окно «О программе» и т. д.

Класс Editor обеспечивает создание, редактирование и сохранение уровней в файл.

Класс PlayObject описывает объекты, присутствующие на уровне. Классы Box, Finish, Player и Wall наследуются от класса PlayObject.