

Пользователь должен добавить карту, для хранения состояния своего нынешнего бюджета и осуществления транзакций. Для этого необходимо нажать на кнопку «Добавить карту», после заполнения данных, карта отобразится на форме. После добавления карты пользователь получает возможность добавлять новые транзакции. При добавлении транзакции пользователь должен помнить, что бюджет не должен быть отрицательным, иначе транзакция не выполнится. История транзакций сохраняется в файл.

При выборе кнопок «Неделя», «Месяц», «Год» пользователю отображается форма с данными о доходах и расходах за выбранный период в графическом виде. Вид формы за неделю изображён на рисунке 3. При работе с данной формой полученный отчет можно сохранить в файлы разных форматов (.txt, .pdf, .xls), нажав на одноименные кнопки.



Рисунок 3 — Окно отчетов за неделю

При нажатии на кнопку «Сохранить» на форме, показанной на рисунке 3, программа создаёт файлы: «imadedm.bmp» — график доходов за месяц, «imaderm.bmp» — график расходов за месяц, «imaded.bmp» — график доходов за неделю, «imader.bmp» — график расходов за неделю. Они разместятся в папке проекта.

Используя кнопку «Калькулятор», пользователь имеет возможность воспользоваться встроенным калькулятором, позволяющим производить расчёты параллельно с добавлением информации о транзакции.

Заключение. В процессе разработки приложения были изучены основные аспекты программирования на языке C#, а также возможности среды Visual Studio. Результаты тестирования показали, что программа работает корректно и стабильно, выполняя свою задачу. Это приложение предоставляет пользователям возможность эффективно отслеживать свои доходы и расходы, планировать бюджет и анализировать финансовые потоки.

Практическая значимость разработанного приложения по управлению личными финансами заключается в: упрощении финансового учёта, планировании бюджета, анализе финансовых потоков, повышении финансовой грамотности.

Список цитируемых источников

1. *Либерти, Дж.* Изучаем C# / Дж. Либерти, Э. Стеллман. — 5-е изд. — СПб. : Питер, 2021. — 800 с.
2. *Арнольд, Н.* Visual Studio 2017. Руководство разработчика / Н. Арнольд, Б. Наггет. — СПб. : Питер, 2018. — 624 с.

УДК 004.514

Э. А. Веракса, Г. М. Раковцы

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

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ПОСТАВОК ЗАПЧАСТЕЙ И РЕАЛИЗАЦИИ ГОТОВОЙ ПРОДУКЦИИ НА ПРЕДПРИЯТИИ «558 АВИАЦИОННЫЙ РЕМОНТНЫЙ ЗАВОД»

Введение. Предприятия, занимающиеся производством и ремонтом техники, сталкиваются с необходимостью учета большого объема номенклатуры, координации поставок и контроля за движением готовой продукции. Ручное ведение таких процессов часто приводит к ошибкам, задержкам и снижению общей эффективности работы. Автоматизация учета позволяет значительно сократить трудозатраты, повысить точность данных и обеспечить прозрачность всех этапов производственно-логистической цепочки.

В условиях постоянного роста требований к оперативности и надежности поставок внедрение современных информационных систем становится необходимым шагом для повышения эффективности управления и оптимизации производственных процессов.

Основная часть. Целью данной работы является создание автоматизированной системы учета, направленной на повышение эффективности управления внутренними процессами предприятия. Такая система должна обеспечивать интеграцию ключевых функций, охватывающих управление ресурсами, контроль поставок, реализацию продукции и взаимодействие между подразделениями. Основное внимание уделяется упрощению рутинных операций, минимизации человеческого фактора и обеспечению актуальности данных для принятия обоснованных управленческих решений.

При разработке системы ставилась задача создания единой информационной среды с интуитивно понятным интерфейсом, возможностью работы с базой данных, формирования необходимой отчетности, графического представления информации и гибкой системой разграничения прав доступа сотрудников.

В качестве СУБД при проектировании приложения была выбрана MySQL являющейся одной из самых популярных и широко используемых систем управления реляционными базами данных. MySQL — это свободно распространяемая (пользователи имеют право на неограниченную установку, запуск, свободное использование), система управления реляционными базами данных, которая поддерживает SQL-интерфейс. Она разработана для использования в средах с большим количеством пользователей и высокими требованиями к производительности. [1]

Среди преимуществ выбора MySQL можно выделить следующие:

1. Открытый исходный код. Распространяется бесплатно для домашнего применения.
2. Простота. MySQL легко устанавливается, имеет понятный интерфейс, а разнообразие плагинов и дополнительных приложений упрощает работу с БД.
3. Функционал. Включает в себя практически весь необходимый набор инструментов, который может пригодиться при разработке любого проекта.
4. Безопасность. Многие системы безопасности уже встроены и работают по умолчанию.
5. Масштабируемость. Может использоваться в работе как с малым, так и с большим объемом данных.
6. Скорость. Является одной из самых быстрых среди имеющихся на современном рынке [1].

В качестве фреймворка для разработки был использован WPF (Windows Presentation Foundation) представляющий платформу пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность. WPF использует расширяемый язык разметки для приложений (XAML), чтобы предоставить декларативную модель для программирования приложений [2].

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 (IDE), которую можно использовать для записи, редактирования, отладки и сборки кода [3].

Данная система была выбрана по следующим причинам:

1. Мощные средства написания кода и функции.
2. Поддержка нескольких языков, в состав которых входит используемый для создания приложения C#.
3. Интеграция управления версиями — позволяющая контролировать версии приложения и эффективнее справляться с ошибками.
4. Разработка с поддержкой искусственного интеллекта — позволяющая значительно увеличить скорость разработки [3].

За основу при проектировании приложения была взята работа предприятия «558 Авиационный ремонтный завод». При запуске системы пользователь первым делом сталкивается с окном авторизации (рисунок 1), где необходимо ввести имя пользователя и пароль. Эти данные сверяются с информацией, хранящейся в базе данных. В случае успешной проверки открывается главное окно приложения (рисунок 2), предоставляющее доступ к функциональности системы.

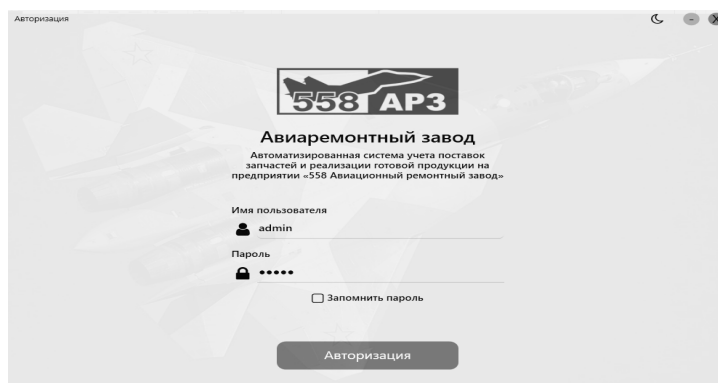


Рисунок 1 — Окно авторизации

Если введённые данные некорректны, над кнопкой «Авторизация» отображается сообщение «Неверный логин или пароль», выделенное красным цветом, информирующее пользователя о неудачной попытке входа. Дополнительно в окне авторизации реализована опция «Запомнить пароль». При её активации логин и пароль сохраняются в зашифрованном виде на локальном устройстве, что позволяет пользователю автоматически входить в систему при следующих запусках приложения без повторного ввода данных, упрощая процесс авторизации и повышая удобство работы.

Перед сохранением данных в базу данных пароль пользователя проходит процедуру хеширования с использованием алгоритма MD5. Это позволяет значительно повысить безопасность системы, предотвращая утечку конфиденциальной информации в случае перехвата сетевых пакетов или получения несанкционированного доступа к базе данных. В результате даже при компрометации хешированных значений злоумышленник не сможет восстановить исходный пароль и получить доступ к учетной записи пользователя. Хеширование выполняется на стороне клиента перед отправкой данных, что исключает хранение паролей в открытом виде и снижает риски взлома.

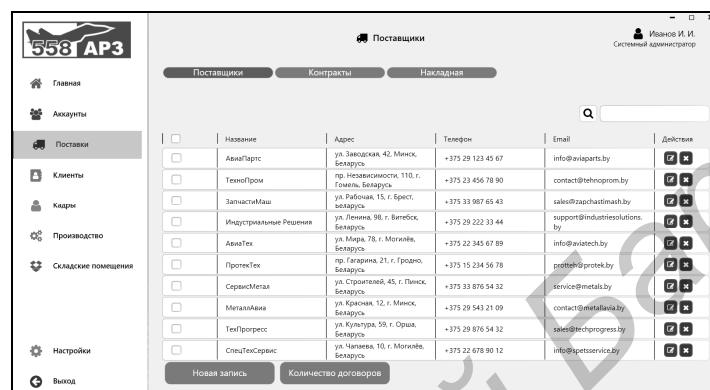


Рисунок 2 — Основное окно приложения

Основное окно приложения логически разделено на две части для удобства взаимодействия с данными. Слева располагается панель навигации, предоставляющая доступ к различным таблицам или их группам, в зависимости от структуры данных и прав пользователя. Справа находится основной рабочий раздел, который занимает большую часть экрана и служит для работы с данными: просмотра, добавления, редактирования и удаления записей. В этом же разделе может быть реализован функционал генерации документации или отображения графической информации — если он предусмотрен для конкретной таблицы.

Каждая таблица снабжена строкой поиска, которая позволяет находить данные по содержимому любой ячейки, проверяя наличие введённого текста. Дополнительно реализована функция сортировки, что упрощает структурирование и анализ информации. В каждой таблице последним столбцом размещён столбец «Действия», содержащий кнопки для управления строками — редактирования и удаления. В некоторых таблицах функциональность расширена: например, в таблице «Контракты» из раздела «Логистика» добавлена третья кнопка в виде иконки Word-документа, которая позволяет автоматически сформировать договор поставки на основе текущих данных.

При добавлении или редактировании записи открывается отдельное окно (см. рисунок 3), где необходимо заполнить все поля и подтвердить изменения кнопкой «Сохранить». В режиме редактирования все поля предварительно заполняются в соответствии с выбранной строкой, что ускоряет процесс внесения изменений и снижает вероятность ошибок.

Система ограничения доступа к данным реализована с использованием ролевой модели на основе таблицы «Permissions» в базе данных, которая позволяет гибко управлять правами пользователей. В данной таблице содержатся следующие поля:

- IdPermission — уникальный идентификатор записи, являющийся первичным ключом таблицы.
- PermissionType — определяет тип доступа: View (только просмотр) или Edit (просмотр и изменение), что позволяет точно регулировать, какие действия разрешены пользователю.
- Tables — указывает название таблицы, к которой применяется соответствующее разрешение.

Для привязки прав доступа к конкретным пользователям используется промежуточная таблица Users_Has_Permissions, реализующая связь типа «многие ко многим» между пользователями и разрешениями. Это позволяет одному пользователю иметь доступ к нескольким таблицам с разными уровнями прав, и одновременно — разрешения могут быть назначены сразу нескольким пользователям.

Система безопасности встроена в интерфейс приложения: если у пользователя отсутствует право View к определённой таблице, соответствующий пункт меню не отображается в навигации, скрывая недоступные разделы. В случае отсутствия права Edit, пользователь может просматривать данные, но элементы управления для добавления, редактирования и удаления (такие как кнопки и контекстные команды) становятся неактивными или скрываются.

Кроме того, данная система легко расширяема — при необходимости можно добавить новые уровни прав (например, только чтение с возможностью экспорта, утверждение изменений и т.д.).

Форма для добавления прав доступа для пользователя представлена на рисунке 4.

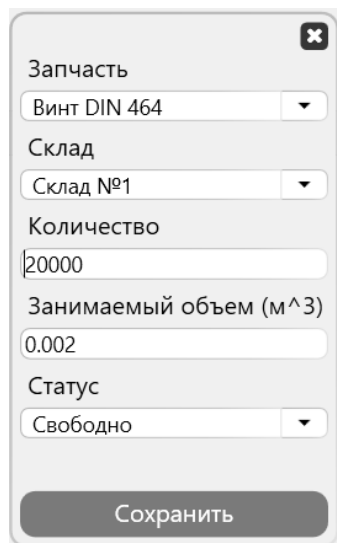


Рисунок 3 — Окно редактирования данных



Рисунок 4 — Окно разграничения доступа

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

Фреймворк WPF обеспечил создание визуально понятного и удобного интерфейса, а применение архитектурного паттерна MVVM позволило четко разделить логику представления и бизнес-логику, упростив поддержку и масштабирование проекта. В качестве системы управления базами данных была выбрана MySQL, а для взаимодействия с ней использовался Entity Framework Core, что позволило работать с таблицами в виде объектов и упростило разработку. В приложении реализована система разграничения прав доступа, обеспечивающая защиту данных от несанкционированного доступа. Дополнительно была внедрена система логирования действий пользователей, позволяющая отслеживать ключевые события в системе и повышающая уровень контроля за безопасностью и работоспособностью приложения.

Список цитируемых источников

1. Что такое MySQL. — URL: <https://mchost.ru/articles/chto-takoe-mysql/> (дата обращения: 02.05.2025).
2. Начало работы (WPF). — URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/getting-started/?view=netframeworkdesktop-4.8> — (дата обращения: 02.05.2025).
3. Что такое Visual Studio? — URL: <https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022> — (дата обращения: 02.05.2025).

УДК 003.26.09

Р. Д. Дедулько, С. В. Лещинский, Ю. Е. Горбач

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

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СТЕГАНОГРАФИИ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕК PYTHON

Введение. В эпоху стремительного развития цифровых технологий и информационного обмена защита данных становится одной из важнейших задач. Один из эффективных способов обеспечения конфиденциальности информации является стеганография – это наука о тайной передаче информации путем сокрытия самого факта передачи [1].

Развитие вычислительной техники и новых каналов передачи информации привело к появлению новых методов стеганографии, в основе которых лежат особенности представления информации в компьютерных файлах, вычислительных сетях и т.п. Этот вид стеганографии получил название компьютерной стеганографии.