**Руководство программиста**

1. Назначение программы

Программное средство «Учет выпуска и реализации продукции ООО «ВакТайм» предназначено для автоматизации процессов учета выпускаемой продукции и её реализации. Система предоставляет пользователям возможность регистрировать новые партии продукции, учитывать их реализацию, отслеживать остатки на складе и генерировать отчетность по продажам. Реализация данного программного средства позволит повысить точность учета, оптимизировать управление складскими запасами и упростить процесс взаимодействия с клиентами.

2. Системные требования

Аппаратные требования:

* Процессор: Intel Celeron N4020 или выше
* Оперативная память: 200 MB или выше
* Жесткий диск: 500 MB или выше
* Монитор: 17” или более
* Ввод: Клавиатура и мышь
* Принтер: Для печати отчетов

Программные требования:

* Операционная система: Windows 10 или выше
* .NET Framework: версия 8.0 или выше

3. Характеристика программ

Программа разработана с использованием языка программирования C# и библиотеки WPF для создания графического интерфейса пользователя. Логика обработки данных реализована через взаимодействие с базой данных SQLite с использованием ORM Entity Framework. Программа использует паттерн MVVM для разделения логики представления и бизнес-логики, что улучшает тестируемость и поддержку кода.

4. Обращение к программам

Пользователи взаимодействуют с программой через графический интерфейс, который включает в себя:

* Окна для ввода информации (материалы, сотрудники, контрагенты, изделия),
* Кнопки для выполнения операций (добавление, удаление, сохранение данных),
* Отображение отчетности по результатам работы.

Каждое действие сопровождается соответствующими сообщениями, которые выводятся в нижней части окна.

5. Входные данные

Программа принимает следующие входные данные:

* Информация о материалах и комплектующих: Название, количество, стоимость, единицы измерения.
* Информация о контрагентах: Название компании, контактные данные, договора.
* Информация о сотрудниках: ФИО, должность, подразделение.
* Информация о изделиях: Наименование изделия, связанные материалы, стоимость.

6. Выходные данные

Программа генерирует следующие отчетные данные:

* Отчет по использованию материалов на изделия: Показывает, какие материалы использовались для производства определенных изделий, их количество и стоимость.
* Отчет по сотрудникам и их подразделениям: Список всех сотрудников с указанием их подразделений, должностей, даты приема и даты увольнения (если есть).
* Отчет по остаткам материалов и комплектующих: Информация о текущих остатках материалов и комплектуюших с указанием мест хранения.
* Отчет по договорам с контрагентами: Информация о заключенных договорах с контрагентами, их суммах и изделиях.
* Отчет по реализации продукции: Детализация продаж продукции: что было продано, кому, за какую сумму и в каком количестве.

7. Сообщения пользователю

Программа выводит следующие типы сообщений:

* Успешное добавление записи: После добавления новой записи в базе данных отображается сообщение об успешном добавлении.
* Успешное удаление записи: Сообщение при успешном удалении записи.
* Предупреждение о невыбранной записи: Если перед удалением не была выбрана запись, программа выведет сообщение.
* Успешное сохранение: При сохранении данных появляется сообщение.
* Предупреждение: нечего отменять: Если не было изменений в таблице, а была нажата кнопка «Отмена», выводится предупреждение.
* Предупреждение: нечего сохранять: При отсутствии изменений, программа предупреждает, что нечего сохранять.
* Ошибка при сохранении: В случае ошибки при сохранении данных отображается сообщение об ошибке.

8. Взаимодействие с базой данных

Логика взаимодействия с базой данных реализована с использованием Entity Framework. База данных SQLite хранит информацию о материалах, сотрудниках, контрагентах и произведенной продукции.

* Использование Entity Framework позволяет автоматически синхронизировать объекты программы с таблицами базы данных, минимизируя необходимость написания SQL-запросов вручную.

9. Архитектура программы

Программа использует паттерн MVVM (Model-View-ViewModel), который разделяет логику представления и бизнес-логику. Это облегчает поддержку программы, улучшает тестируемость и ускоряет развитие новых функциональностей.

* Model: Представляет данные программы (например, сущности для материалов, сотрудников, контрагентов и т. д.).
* View: Графический интерфейс, отображающий данные пользователю.
* ViewModel: Логика, связывающая модель данных и представление, обеспечивая необходимое взаимодействие.