

ОТЧЕТ ПО ЗАДАНИЮ 4

Дополнительный функционал системы управления производством

Калькулятор сырья и цеха производства

Содержание

1. Описание задания
 2. Алгоритм расчета сырья
 3. Реализация метода
 4. API endpoints
 5. Веб-интерфейс
 6. Интеграция с системой
 7. Тестирование
 8. Примеры использования
 9. Заключение

1. Описание задания

Добавление функционала в систему работы с продукцией компании согласно требованиям заказчика:

- ✓ Реализация вывода списка цехов для производства продукции
 - ✓ Разработка метода расчета количества сырья для производства
 - ✓ Интеграция нового функционала в существующую систему
 - ✓ Обеспечение единого стиля интерфейса

Цель: Обеспечить одинаковый расчет количества сырья для производства продукции с учетом потерь материала и коэффициентов типа продукции.

2. Алгоритм расчета сырья

2.1 Описание алгоритма

Метод рассчитывает количество сырья, необходимого для производства заданного количества продукции с учетом потерь сырья.

Формула расчета:

$$\text{Базовое количество} = \text{param1} \times \text{param2} \times \text{коэффициент_типа_продукции} \times \text{количество}$$
$$\text{Итоговое количество} = \text{Базовое количество} \times (1 + \text{процент_потерь} / 100)$$
$$\text{Результат} = \text{ceil}(\text{Итоговое количество})$$

2.2 Параметры метода

| Параметр | Тип | Описание | Ограничения |
|------------------|-------|------------------------------|--------------------------|
| product_type_id | int | Идентификатор типа продукции | Должен существовать в БД |
| material_type_id | int | Идентификатор типа материала | Должен существовать в БД |
| quantity | int | Количество продукции | Положительное число |
| param1 | float | Первый параметр (длина) | Положительное число |
| param2 | float | Второй параметр (ширина) | Положительное число |

2.3 Возвращаемое значение

- **Положительное число** - количество сырья с учетом потерь
- **-1** - при ошибке или некорректных данных

3. Реализация метода

3.1 Сервис MaterialCalculatorService

```
class MaterialCalculatorService: def __init__(self, db: Session): self.db = db def calculate_raw_material( self, product_type_id: int, material_type_id: int, quantity: int, param1: float, param2: float ) -> int: try: # Валидация входных данных if quantity <= 0 or param1 <= 0 or param2 <= 0: return -1 # Получение типа продукции product_type = self.db.query(ProductType).filter( ProductType.id == product_type_id ).first() # Получение типа материала material_type = self.db.query(MaterialType).filter( MaterialType.id == material_type_id ).first() if not product_type or not material_type: return -1 # Расчет количества сырья base_material_per_unit = param1 * param2 * product_type.coefficient total_base_material = base_material_per_unit * quantity loss_multiplier = 1 + (material_type.loss_percentage / 100) total_material_with_losses = total_base_material * loss_multiplier return math.ceil(total_material_with_losses) except Exception as e: return -1
```

3.2 Структура файлов

```
backend/app/ └── services/ | └── __init__.py | └── material_calculator.py  
# Сервис расчета сырья └── routers/ └── material_calculator.py # API  
роутер для калькулятора
```

4. API endpoints

4.1 Расчет сырья

Endpoint: POST /api/calculator/calculate-material

Запрос:

```
{ "product_type_id": 1, "material_type_id": 2, "quantity": 10, "param1":  
1.5, "param2": 2.0 }
```

Ответ:

```
{ "required_material": 45, "product_type_id": 1, "material_type_id": 2,
  "quantity": 10, "param1": 1.5, "param2": 2.0, "success": true, "message":
  "Для производства 10 единиц продукции потребуется 45 единиц сырья" }
```

4.2 Цеха для продукта

Endpoint: GET /api/calculator/workshops-for-product/{product_id}

Ответ:

```
[ { "workshop_id": 1, "workshop_name": "Цех сборки", "workshop_type":
  "Основной", "staff_count": 15, "production_time_hours": 4.5 } ]
```

4.3 Общее время производства

Endpoint: GET /api/calculator/total-production-time/{product_id}

Ответ:

```
{ "product_id": 1, "product_name": "Стул офисный",
  "total_production_time_hours": 12.5, "workshops_count": 3 }
```

5. Веб-интерфейс

5.1 Новый раздел "Калькулятор"

- ✓ Добавлена ссылка в главное меню навигации
- ✓ Иконка калькулятора для лучшего восприятия
- ✓ Активная подсветка при выборе раздела
- ✓ Плавные анимации переходов

5.2 Форма расчета сырья

- ✓ Выпадающие списки для типов продукции и материалов
- ✓ Поля для ввода количества и параметров
- ✓ Валидация на клиентской стороне
- ✓ Отображение детального результата
- ✓ Обработка ошибок с понятными сообщениями

5.3 Просмотр цехов для продукта

- ✓ Выпадающий список продуктов
- ✓ Таблица с информацией о цехах
- ✓ Отображение времени производства
- ✓ Информация о персонале цехов
- ✓ Расчет общего времени производства

6. Интеграция с системой

6.1 Единый стиль интерфейса

- ✓ Корпоративные цвета и шрифты
- ✓ Соответствие общему дизайну
- ✓ Адаптивная верстка
- ✓ Анимации в едином стиле

6.2 API интеграция

- ✓ Подключение нового роутера к FastAPI
- ✓ Обновление документации Swagger
- ✓ Добавление методов в JavaScript API
- ✓ Обработка CORS для фронтенда

7. Тестирование

7.1 Тестирование API



7.2 Тестирование интерфейса

- ✓ Загрузка данных в формы
- ✓ Валидация полей
- ✓ Отображение результатов
- ✓ Адаптивность на мобильных
- ✓ Совместимость с браузерами

8. Примеры использования

8.1 Пример расчета сырья

Исходные данные:

- Тип продукции: Мебель (коэффициент 1.2)
- Тип материала: Дерево (потери 5%)
- Количество: 10 штук
- Параметр 1 (длина): 1.5 м

- Параметр 2 (ширина): 2.0 м

Расчет:

1. Базовое количество на единицу: $1.5 \times 2.0 \times 1.2 = 3.6$
2. Общее базовое количество: $3.6 \times 10 = 36$
3. С учетом потерь: $36 \times (1 + 5/100) = 36 \times 1.05 = 37.8$
4. Результат: $\text{ceil}(37.8) = 38$ единиц сырья

8.2 Пример цехов для продукта

| Цех | Время (ч) | Персонал | Тип |
|--------------|------------|----------------|------------------|
| Цех раскряя | 2.0 | 5 чел. | Подготовительный |
| Цех сборки | 4.5 | 8 чел. | Основной |
| Цех отделки | 3.0 | 3 чел. | Финишный |
| Итого | 9.5 | 16 чел. | - |

9. Заключение

9.1 Достигнутые цели

- ✓ Разработан метод расчета сырья с учетом потерь
 - ✓ Реализован вывод цехов для производства продукции
 - ✓ Создан веб-интерфейс для нового функционала
 - ✓ Обеспечена интеграция с существующей системой
 - ✓ Добавлена валидация и обработка ошибок

9.2 Технические характеристики



9.3 Функциональная полнота

Система теперь обеспечивает:

- ✓ Полный CRUD для всех сущностей
 - ✓ Расчет необходимого сырья для производства
 - ✓ Просмотр производственных процессов по цехам
 - ✓ Расчет времени изготовления продукции
 - ✓ Интуитивный веб-интерфейс для всех операций

Результат: Система готова к промышленному использованию и может быть легко расширена дополнительными модулями расчетов.

© 2025 Система управления производством

Отчет по заданию 4: Дополнительный функционал

Разработано с использованием FastAPI, SQLite, HTML5, CSS3, JavaScript