

ТЗ: Универсальная воздушная система копирования

Концепция

"Один клик - весь процесс повторен" - фирменная система мгновенного создания документов по шаблонам во всех модулях ERP.

Архитектура системы

Глобальный компонент

```
typescript

// UniversalAirborneButton.tsx
interface AirborneConfig {
  module: 'products' | 'warehouse' | 'sales' | 'purchases' | 'banking' | 'clients' | 'accounts';
  apiEndpoint: string;
  itemName: string; // "товар", "документ", "клиент"
  lastItemsCount: number; // по умолчанию 5
}
```

Размещение

- **Позиция:** fixed, bottom: 24px, right: 24px
- **Z-index:** 9999 (поверх всех элементов)
- **Дизайн:** градиент blue→purple, иконки Copy+Zap
- **Анимация:** hover scale(1.1), pulse на уведомлении

Модули внедрения

1. Products (готов)

- **Endpoint:** `POST /api/company/products/:id/copy`
- **Логика:** копирует товар с новым кодом (+1) и меткой "(Копия)"
- **Результат:** новый товар готов к редактированию

2. Warehouse

- **Endpoint:** `POST /api/company/warehouse/:id/copy`
- **Логика:** копирует последнее движение склада (приход/расход)
- **Поля:** новая дата, те же товары, количества, склад
- **Результат:** новое движение в статусе "Черновик"

3. Sales

- **Endpoint:** `POST /api/company/sales/:id/copy`
- **Логика:** копирует последнюю продажу
- **Поля:** сегодняшняя дата, тот же клиент, товары, цены
- **Результат:** новый заказ готов к подтверждению

4. Purchases

- **Endpoint:** `POST /api/company/purchases/:id/copy`
- **Логика:** копирует последнюю закупку
- **Поля:** сегодняшняя дата, тот же поставщик, товары
- **Результат:** новая закупка в статусе "Черновик"

5. Banking

- **Endpoint:** `POST /api/company/banking/:id/copy`
- **Логика:** копирует последнюю операцию (поступление/списание)
- **Поля:** сегодняшняя дата, тот же счет, сумма, назначение
- **Результат:** новая операция готова к проведению

6. Clients

- **Endpoint:** `POST /api/company/clients/:id/copy`
- **Логика:** копирует клиента
- **Поля:** новый ID, те же контакты, категория, условия
- **Результат:** новый клиент готов к редактированию

7. Chart of Accounts

- **Endpoint:** `POST /api/company/accounts/:id/copy`
- **Логика:** копирует счет
- **Поля:** новый номер, та же структура, тип
- **Результат:** новый счет в плане счетов

Техническая реализация

Frontend компонент

typescript

```
// components/UniversalAirborneButton.tsx
```

```
const UniversalAirborneButton: React.FC<AirborneConfig> = ({
  module,
  apiEndpoint,
  itemName,
  lastItemsCount = 5
}) => {
  const [isOpen, setIsOpen] = useState(false);
  const [items, setItems] = useState([]);
  const [copying, setCopying] = useState<number | null>(null);

  const handleAirborneCopy = async (itemId: number, itemName: string) => {
    setCopying(itemId);
    try {
      const response = await api.post(`${apiEndpoint}/${itemId}/copy`);
      if (response.data.success) {
        showAirborneNotification(itemName);
        onSuccess(response.data.data);
      }
    } catch (error) {
      showError(error.message);
    } finally {
      setCopying(null);
    }
  };

  return (
    <div className="fixed bottom-6 right-6 z-[9999]">
      { /* Плавающая кнопка */ }
      <button
        onClick={() => setIsOpen(true)}
        className="w-16 h-16 bg-gradient-to-r from-blue-500 to-purple-600
          text-white rounded-full shadow-2xl hover:scale-110
          transition-all duration-300 flex items-center justify-center"
        >
        <Copy className="w-8 h-8" />
        <Zap className="w-4 h-4 absolute -top-1 -right-1 animate-pulse text-yellow-300" />
      </button>

      { /* Модальное окно */ }
      {isOpen && (
        <AirborneModal
          module={module}
          items={items}
          itemName={itemName}
          onCopy={handleAirborneCopy}
        >

```

```
      onClose={() => setIsOpen(false)}  
      copying={copying}  
    />  
  })  
</div>  
);  
};
```

Backend API структура

javascript

```
// Универсальный контроллер копирования
```

```
const copyEntity = async (req, res) => {
  try {
    const { id } = req.params;
    const { entityType } = req.body; // 'products', 'sales', etc.
    const companyId = req.companyContext.companyId;
    const userId = req.user?.id;

    const original = await getOriginalEntity(entityType, id, companyId);
    if (!original) {
      return res.status(404).json({
        success: false,
        error: `${entityType} не найден`
      });
    }

    const copy = await createEntityCopy(entityType, original, {
      companyId,
      userId,
      date: new Date()
    });

    res.json({
      success: true,
      message: `${getEntityDisplayName(entityType)} воздушно скопирован!`,
      data: copy,
      airborne: {
        originalId: original.id,
        copyId: copy.id,
        timeSaved: '5 минут',
        method: 'Воздушная система'
      }
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      error: 'Ошибка воздушного копирования'
    });
  }
};
```

Модальное окно

typescript

// AirborneModal.tsx - универсальное модальное окно

```
const AirborneModal: React.FC = ({ module, items, itemName, onCopy, copying }) => {
  return (
    <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center p-4 z-[9999]">
      <div className="bg-white rounded-xl shadow-2xl max-w-2xl w-full">
        {/* Заголовок */}
        <div className="p-6 border-b bg-gradient-to-r from-blue-50 to-purple-50">
          <h2 className="text-2xl font-bold">Воздушное копирование</h2>
          <p className="text-gray-600">Выберите {itemName} для мгновенного создания копии</p>
        </div>

        {/* Список элементов */}
        <div className="p-6 max-h-96 overflow-y-auto">
          {items.map((item) => (
            <div key={item.id} className="flex items-center justify-between p-4 border rounded-lg hover:bg-blue-50">
              <div className="flex-1">
                <div className="font-medium">{getItemDisplayName(item, module)}</div>
                <div className="text-sm text-gray-500">{getItemDetails(item, module)}</div>
              </div>

              <button
                onClick={() => onCopy(item.id, getItemDisplayName(item, module))}
                disabled={copying === item.id}
                className="px-6 py-2 bg-gradient-to-r from-blue-500 to-purple-600 text-white rounded-lg hover:from-
              >
                {copying === item.id ? (
                  <>
                    <Loader className="w-4 h-4 animate-spin" />
                    <span>Копируем...</span>
                  </>
                ) : (
                  <>
                    <Zap className="w-4 h-4" />
                    <span>Топнуть</span>
                  </>
                )}
              </button>
            </div>
          ))}
        </div>

        {/* Подвал */}
        <div className="px-6 py-4 bg-gray-50 border-t">
          <div className="flex items-center gap-2 text-sm text-gray-600">
            <Zap className="w-4 h-4 text-blue-500" />
            <span>Воздушная система: один клик = готовый документ</span>
          </div>
        </div>
      </div>
    </div>
  )
}
```

```
    </div>
  </div>
</div>
</div>
);
};
```

Интеграция в страницы

Пример для каждого модуля

typescript

// ProductsPage.tsx

```
<UniversalAirborneButton
  module="products"
  apiEndpoint="/api/company/products"
  itemName="товар"
  lastItemsCount={5}
/>
```

// SalesPage.tsx

```
<UniversalAirborneButton
  module="sales"
  apiEndpoint="/api/company/sales"
  itemName="продажу"
  lastItemsCount={3}
/>
```

// WarehousePage.tsx

```
<UniversalAirborneButton
  module="warehouse"
  apiEndpoint="/api/company/warehouse"
  itemName="движение"
  lastItemsCount={5}
/>
```

Уведомления

typescript

```
const showAirborneNotification = (itemName: string) => {
  const notification = document.createElement('div');
  notification.className = 'fixed top-4 right-4 bg-green-500 text-white p-4 rounded-lg shadow-lg z-[9999] anim
  notification.innerHTML = `
    <div class="flex items-center gap-2">
      <span class="text-xl">✂ </span>
      <div>
        <div class="font-bold">Воздушно скопировано!</div>
        <div class="text-sm">${itemName} готов к использованию</div>
      </div>
    </div>
  `;
  document.body.appendChild(notification);

  setTimeout(() => {
    document.body.removeChild(notification);
  }, 3000);
};
```



Этапы внедрения

Этап 1: Ядро системы (1 день)

- Создать UniversalAirborneButton компонент
- Настроить универсальное модальное окно
- Создать систему уведомлений

Этап 2: Backend API (1 день)

- Создать универсальный copyEntity контроллер
- Добавить routes для всех модулей
- Настроить логику копирования по типам

Этап 3: Интеграция (2 дня)

- Внедрить кнопку в Products
- Внедрить в Warehouse, Sales, Purchases
- Внедрить в Banking, Clients, Chart of Accounts

Этап 4: Тестирование (1 день)

- Протестировать копирование в каждом модуле
- Проверить корректность создания документов
- Оптимизировать производительность

Результат

- Единая воздушная кнопка во всех 7 модулях
- Мгновенное создание документов по шаблонам
- Фирменная фишка "Воздушная бухгалтерия"
- Экономия времени пользователей в 10 раз
- Уникальное конкурентное преимущество

Эффект

Пользователь заходит в любой модуль → видит знакомую воздушную кнопку → один клик → документ готов. Система становится предсказуемой и интуитивно понятной во всех разделах.