Pandashop.md - Technical Specification for Test Auto-generation (Safe Mode)

> **For GitHub Copilot Agent**: Create tests using Playwright/TypeScript + Gherkin for urgent scenario (3-4 weeks, 300-400 test cases)
>
> **SAFETY MODE**: No SMS testing, E2E tests stop before final order placement

🏗️ Project Structure

```
pandashop-tests/
├── tests/
│   ├── e2e/                    # E2E tests (Playwright) - SAFE MODE
│   │   ├── features/           # Gherkin feature files
│   │   ├── step-definitions/   # Step definitions
│   │   └── specs/              # Playwright spec files
│   ├── api/                    # API tests (if available)
│   │   ├── endpoints/          # API endpoint tests
│   │   └── schemas/            # JSON schemas for validation
│   └── ui/                     # UI component tests
│       ├── components/         # Component tests
│       └── pages/              # Page Object Model
├── fixtures/                   # Test data
├── utils/                      # Utilities and helpers
├── config/                     # Configuration files
└── reports/                    # Test reports
```

📦 Dependencies and Configuration

package.json dependencies

```json
```

```
{
  "devDependencies": {
    "@playwright/test": "^1.40.0",
    "@cucumber/cucumber": "^10.0.0",
    "@cucumber/playwright": "^1.0.0",
    "typescript": "^5.0.0",
    "axios": "^1.6.0",
    "joi": "^17.11.0",
    "faker": "^6.6.6",
    "dotenv": "^16.3.1",
    "allure-playwright": "^2.9.0"
  }
}
```

playwright.config.ts

```typescript
export default defineConfig({
  testDir: './tests',
  timeout: 30000,
  expect: { timeout: 5000 },
  use: {
    baseURL: 'https://pandashop.md',
    headless: true,
    screenshot: 'only-on-failure',
    video: 'retain-on-failure',
    locale: 'ru-RU'
  },
  projects: [
    { name: 'chromium', use: { ...devices['Desktop Chrome'] } },
    { name: 'firefox', use: { ...devices['Desktop Firefox'] } },
    { name: 'webkit', use: { ...devices['Desktop Safari'] } },
    { name: 'mobile', use: { ...devices['iPhone 13'] } }
  ]
});
```

🌐 Website Structure and Selectors (Pandashop.md)

Main Interface Elements

```typescript
```

```
// Selectors for auto-generating Page Objects
export const SELECTORS = {
  // Header Navigation
  LOGO: 'a[href="/"]',
  LANGUAGE_SWITCHER_RU: 'a:has-text("Ру")',
  LANGUAGE_SWITCHER_RO: 'a:has-text("Ro")',
  SEARCH_INPUT: 'input[placeholder*="Поиск"], input[placeholder*="Căutare"]',
  SEARCH_BUTTON: 'button[type="submit"]',
  CART_ICON: '[data-testid="cart-icon"], .cart-counter, .header-cart',
  CART_COUNTER: '.cart-counter, .cart-badge',
  USER_MENU: '.user-menu, .header-user',
  LOGIN_LINK: 'a:has-text("Вход"), a:has-text("Intrare")',
  REGISTER_LINK: 'a:has-text("Регистрация"), a:has-text("Înregistrare")',
  PHONE_NUMBERS: 'a[href^="tel:022"], a[href^="tel:079"], a[href^="tel:060"]',

  // Navigation Menu
  CATALOG_MENU: 'text="Каталог товаров", text="Catalogul produselor"',
  CATEGORIES: {
    ELECTRONICS: 'text="Электроника", text="Electronică"',
    APPLIANCES: 'text="Бытовая техника", text="Electrocasnice"',
    HOME_GARDEN: 'text="Дом и сад", text="Casă și grădină"',
    CONSTRUCTION: 'text="Строительство и ремонт", text="Construcții și reparații"',
    PETS: 'text="Товары для животных", text="Produse pentru animale"',
    SPORTS: 'text="Спорт", text="Sport"',
    TOURISM: 'text="Активный отдых и туризм", text="Recreere activă și turism"',
    HOBBY: 'text="Хобби и творчество", text="Hobby și creativitate"',
    KIDS: 'text="Детские товары", text="Produse pentru copii"',
    AUTO: 'text="Автотовары", text="Produse auto"',
    BEAUTY: 'text="Красота и здоровье", text="Frumusețe și sănătate"',
    FASHION: 'text="Fashion"',
    BUSINESS: 'text="Товары для бизнеса", text="Produse pentru afaceri"'
  },

  // Product Catalog
  PRODUCT_CARDS: '.product-card, .product-item',
  PRODUCT_TITLE: '.product-title, .product-name',
  PRODUCT_PRICE: '.product-price, .price',
  PRODUCT_OLD_PRICE: '.product-old-price, .old-price',
  PRODUCT_DISCOUNT: '.discount, .sale-badge',
  CASHBACK_AMOUNT: '.cashback-amount, .cashback',
  PRODUCT_IMAGE: '.product-image img, .product-photo img',
  PRODUCT_RATING: '.product-rating, .rating',
  PRODUCT_REVIEWS_COUNT: '.reviews-count, .product-reviews',
ADD_TO_CART_BTN: 'button:has-text("В корзину"), button:has-text("În coș")',
ADD_TO_FAVORITES: '.add-to-favorites, .wishlist-btn',
ADD_TO_COMPARE: '.add-to-compare, .compare-btn',

// Product Details Page
PRODUCT_DETAILS: '.product-details, .product-info',
PRODUCT_GALLERY: '.product-gallery, .product-images',
PRODUCT_CHARACTERISTICS: '.product-characteristics, .specifications',
QUANTITY_INPUT: 'input[type="number"], .quantity-input',
QUANTITY_PLUS: '.quantity-plus, .qty-plus',
QUANTITY_MINUS: '.quantity-minus, .qty-minus',
IN_STOCK_STATUS: '.in-stock, .availability',
OUT_OF_STOCK_STATUS: '.out-of-stock, .unavailable',

// Search and Filters
SEARCH_RESULTS: '.search-results, .products-list',
SEARCH_RESULTS_COUNT: '.results-count, .found-products',
PRICE_FILTER: '.price-filter, .filter-price',
PRICE_MIN: 'input[name="price_min"], #price-min',
PRICE_MAX: 'input[name="price_max"], #price-max',
BRAND_FILTER: '.brand-filter, .filter-brand',
CATEGORY_FILTER: '.category-filter, .filter-category',
APPLY_FILTERS: 'button:has-text("Применить"), button:has-text("Aplică")',
RESET_FILTERS: 'button:has-text("Сбросить"), button:has-text("Resetare")',
SORT_SELECT: '.sort-select, .sorting',

// Shopping Cart
CART_PAGE: '.cart-page, .shopping-cart',
CART_ITEMS: '.cart-items, .cart-products',
CART_ITEM: '.cart-item, .cart-product',
CART_ITEM_IMAGE: '.cart-item-image img',
CART_ITEM_NAME: '.cart-item-name, .product-name',
CART_ITEM_PRICE: '.cart-item-price, .item-price',
CART_ITEM_QUANTITY: '.cart-item-quantity input, .quantity-input',
CART_ITEM_TOTAL: '.cart-item-total, .item-total',
REMOVE_ITEM: '.remove-item, .delete-item',
CART_SUBTOTAL: '.cart-subtotal, .subtotal',
CART_DISCOUNT: '.cart-discount, .discount-amount',
CART_CASHBACK: '.cart-cashback, .cashback-amount',
CART_TOTAL: '.cart-total, .total-amount',
CONTINUE_SHOPPING: 'button:has-text("Продолжить покупки"), button:has-text("Continuă cumpărăturile")',
PROCEED_TO_CHECKOUT: 'button:has-text("Оформить заказ"), button:has-text("Plasați comanda")',
CLEAR_CART: 'button:has-text("Очистить корзину"), button:has-text("Golește coșul")',
// User Authentication (EMAIL ONLY - NO SMS)
LOGIN_FORM: '.login-form, .auth-form',
LOGIN_EMAIL: 'input[type="email"], input[name="email"]',
LOGIN_PASSWORD: 'input[type="password"], input[name="password"]',
LOGIN_SUBMIT: 'button[type="submit"]:has-text("Войти"), button[type="submit"]:has-text("Intrare")',
REMEMBER_ME: 'input[type="checkbox"]',
FORGOT_PASSWORD: 'a:has-text("Забыли пароль"), a:has-text("Ați uitat parola")',

// Registration Form (EMAIL ONLY)
REGISTER_FORM: '.register-form, .registration-form',
REGISTER_EMAIL: 'input[name="email"], input[type="email"]',
REGISTER_PASSWORD: 'input[name="password"], input[type="password"]',
REGISTER_CONFIRM_PASSWORD: 'input[name="password_confirmation"], input[name="confirm_password"]',
REGISTER_FIRST_NAME: 'input[name="first_name"], input[name="firstname"]',
REGISTER_LAST_NAME: 'input[name="last_name"], input[name="lastname"]',
REGISTER_PHONE: 'input[name="phone"], input[type="tel"]',
REGISTER_AGREE_TERMS: 'input[type="checkbox"]',
REGISTER_SUBMIT: 'button[type="submit"]:has-text("Зарегистрироваться"), button[type="submit"]:has-text("Înregistr

// Checkout Process (SAFE MODE - NO FINAL SUBMISSION)
CHECKOUT_STEPS: '.checkout-steps, .order-steps',
DELIVERY_FORM: '.delivery-form, .address-form',
DELIVERY_FIRST_NAME: 'input[name="delivery_first_name"]',
DELIVERY_LAST_NAME: 'input[name="delivery_last_name"]',
DELIVERY_PHONE: 'input[name="delivery_phone"]',
DELIVERY_EMAIL: 'input[name="delivery_email"]',
DELIVERY_CITY: 'input[name="city"], select[name="city"]',
DELIVERY_STREET: 'input[name="street"], input[name="address"]',
DELIVERY_BUILDING: 'input[name="building"], input[name="house"]',
DELIVERY_APARTMENT: 'input[name="apartment"], input[name="apt"]',
DELIVERY_POSTAL_CODE: 'input[name="postal_code"], input[name="zip"]',
DELIVERY_METHOD: '.delivery-method, .shipping-method',
DELIVERY_COURIER: 'input[value="courier"], label:has-text("Курьерская доставка")',
DELIVERY_PICKUP: 'input[value="pickup"], label:has-text("Самовывоз")',
DELIVERY_NOTES: 'textarea[name="notes"], textarea[name="comment"]',

// Payment Method Selection (SAFE MODE)
PAYMENT_METHODS: '.payment-methods, .payment-options',
PAYMENT_CASH: 'input[value="cash"], label:has-text("Наличными")',
PAYMENT_CARD: 'input[value="card"], label:has-text("Картой")',
PAYMENT_TRANSFER: 'input[value="transfer"], label:has-text("Переводом")',

// Order Review (STOPPING POINT)
ORDER_REVIEW: '.order-review, .order-summary',
  ORDER_ITEMS_REVIEW: '.order-items, .order-products',
  ORDER_DELIVERY_INFO: '.delivery-info, .shipping-info',
  ORDER_PAYMENT_INFO: '.payment-info',
  ORDER_TOTALS_REVIEW: '.order-totals, .order-summary-totals',
  // ⚠️ DO NOT INCLUDE FINAL SUBMIT BUTTONS
  // PLACE_ORDER_BTN: 'button:has-text("Подтвердить заказ")', // EXCLUDED FOR SAFETY

  // Promotional Elements
  PROMO_BANNERS: '.promo-banner, .promotion',
  PROMO_CODE_INPUT: 'input[name="promo_code"], input[name="coupon"]',
  APPLY_PROMO_CODE: 'button:has-text("Применить"), button:has-text("Aplică")',

  // Popular Categories (Footer)
  POPULAR_CATEGORIES: '.popular-categories',

  // Newsletter Subscription
  NEWSLETTER_FORM: '.newsletter, .subscription',
  NEWSLETTER_EMAIL: 'input[name="newsletter_email"]',
  NEWSLETTER_SUBMIT: 'button:has-text("Подписаться"), button:has-text("Abonare")',

  // Mobile Specific
  MOBILE_MENU_TOGGLE: '.mobile-menu-toggle, .hamburger',
  MOBILE_MENU: '.mobile-menu, .mobile-nav',
  MOBILE_SEARCH_TOGGLE: '.mobile-search-toggle',

  // Language Context
  RUSSIAN_CONTENT: '[lang="ru"], .lang-ru',
  ROMANIAN_CONTENT: '[lang="ro"], .lang-ro'
};
```

URL Structure and Routing

```
typescript
```

```
export const ROUTES = {
  HOME: '/',
  CATALOG: '/catalog',
  PRODUCT: '/product/:id',
  SEARCH: '/search',
  CART: '/cart',
  CHECKOUT: '/checkout', // SAFE MODE - will not complete
  LOGIN: '/login',
  REGISTER: '/register',
  PROFILE: '/profile',
  CATEGORY: (category: string) => `/catalog/${category}`
};

export const CATEGORIES = [
  'electronics', 'appliances', 'home-garden', 'construction',
  'pets', 'sports', 'tourism', 'hobby', 'kids', 'auto',
  'beauty', 'fashion', 'business'
];

export const LANGUAGES = ['ru', 'ro'];
```

📝 Gherkin Scenarios (BDD) - SAFE MODE

Critical Path Features (No Real Orders)

Feature: Product Discovery and Cart Management

```
gherkin
```

```
Feature: Product Discovery and Cart Management
  As a customer
  I want to browse and add products to my cart
  So that I can prepare my order for review

  Background:
    Given I am on the Pandashop homepage
    And the website is fully loaded

  @critical @smoke
  Scenario: Complete shopping flow to order review (SAFE MODE)
    When I search for "iPhone"
    And I select the first product from search results
    And I verify product details are displayed correctly
    And I add the product to cart
    And I navigate to cart
    And I verify product appears in cart with correct details
    And I proceed to checkout
    And I fill delivery information with valid data
    And I select "Курьерская доставка" delivery method
    And I select "Наличными курьеру" payment method
    And I review my order details
    Then I should see complete order summary with:
      | Product details | ✓ |
      | Delivery info   | ✓ |
      | Payment method  | ✓ |
      | Total amount    | ✓ |
    # ⚠️ STOP HERE - DO NOT PLACE ACTUAL ORDER

  @multilingual @critical
  Scenario: Language switching preserves cart state
    Given I have added 2 products to cart in Russian
    When I switch language to Romanian
    Then the interface should display in Romanian
    And my cart should preserve all 2 products
    And product names should be translated appropriately
    And cart totals should remain identical
    When I switch back to Russian
    Then everything should be preserved in Russian

  @search @smoke
  Scenario: Product search and filtering
    When I search for "телефон"
```

```gherkin
    Then I should see search results for phones
    And results count should be displayed
    When I apply price filter from 5000 to 15000 lei
    And I apply brand filter for "Apple"
    Then filtered results should match criteria
    When I reset all filters
    Then I should see all search results again
```

Feature: Cart Management (Safe Mode)

```gherkin
Feature: Cart Management
  As a customer
  I want to manage items in my shopping cart
  So that I can control my selection before checkout

  @cart @critical
  Scenario: Add and modify cart items
    Given I am on the homepage
    When I navigate to "Электроника" category
    And I add first product to cart
    And I continue shopping
    And I add second product to cart
    And I navigate to cart page
    Then I should see 2 products in cart
    When I increase quantity of first item to 3
    Then cart total should be recalculated correctly
    When I remove second item from cart
    Then cart should contain only first item with quantity 3

  @cart
  Scenario: Cart persistence across language switching
    Given I have 3 different products in cart
    When I switch language from Russian to Romanian
    Then cart should maintain all 3 products
    And quantities should remain unchanged
    And prices should remain the same
    And product names should be in Romanian
```

Feature: User Registration (Email Only)

```gherkin
```

```
Feature: User Registration
  As a new customer
  I want to create an account using email
  So that I can save my preferences and track orders

  @registration @critical
  Scenario: Register with email only (no SMS)
    Given I am on the registration page
    When I fill registration form with valid email data:
      | Email            | test.user@example.com |
      | Password         | SecurePass123!        |
      | Confirm Password | SecurePass123!        |
      | First Name       | Тест                  |
      | Last Name        | Пользователь          |
      | Phone            | +37360123456          |
    And I agree to terms and conditions
    And I submit registration form
    Then I should receive email confirmation
    And I should be able to login with these credentials
    # ⚠️ NO SMS VERIFICATION

  @login @critical
  Scenario: Login with valid credentials
    Given I have a registered account
    When I navigate to login page
    And I enter valid email and password
    And I click login button
    Then I should be logged in successfully
    And I should see my name in user menu
```

Feature: Checkout Process (Safe Mode)

gherkin

```
Feature: Checkout Process
  As a customer
  I want to fill checkout forms completely
  So that I can review my order before placement

  @checkout @critical
  Scenario: Complete checkout form filling (SAFE MODE)
    Given I have products in my cart
    When I proceed to checkout
    And I fill delivery information:
      | First Name   | Иван                  |
      | Last Name    | Петров                |
      | Email        | ivan.petrov@example.com |
      | Phone        | +37360123456          |
      | City         | Кишинев               |
      | Street       | ул. Штефан чел Маре   |
      | Building     | 124                   |
      | Apartment    | 15                    |
      | Postal Code  | MD-2012               |
    And I select "Курьерская доставка" as delivery method
    And I select "Картой онлайн" as payment method
    And I add delivery notes "Звонить за час до доставки"
    Then I should see complete order review with:
      | Delivery address | Кишинев, ул. Штефан чел Маре, 124-15 |
      | Delivery method  | Курьерская доставка                  |
      | Payment method   | Картой онлайн                        |
      | Order items      | All cart products listed             |
      | Total amount     | Calculated correctly                 |
    # ⚠️ STOP AT REVIEW - DO NOT SUBMIT ORDER

  @checkout @validation
  Scenario: Checkout form validation
    Given I have products in my cart
    When I proceed to checkout
    And I try to submit form with missing required fields
    Then I should see validation errors for:
      | Field    | Error Message                   |
      | Email    | Email обязателен для заполнения |
      | Phone    | Телефон обязателен              |
      | City     | Город обязателен                |
      | Street   | Адрес обязателен                |
      | Building | Номер дома обязателен           |
    When I fill all required fields correctly
    Then validation errors should disappear
    And I should be able to proceed to review
```

Test Data Templates (Safe Mode)

```
typescript
```

```
export const TEST_DATA = {
  USERS: {
    VALID_USER: {
      email: () => `test.user.${Date.now()}@example.com`,
      password: 'SecurePassword123!',
      confirmPassword: 'SecurePassword123!',
      firstName: 'Тест',
      lastName: 'Пользователь',
      phone: '+37360123456'
    },

    EXISTING_USER: {
      email: 'existing.user@example.com',
      password: 'ExistingPass123!'
    }
  },

  SEARCH_TERMS: {
    RUSSIAN: ['iPhone', 'телефон', 'холодильник', 'телевизор'],
    ROMANIAN: ['telefon', 'frigider', 'televizor', 'laptop']
  },

  DELIVERY_ADDRESS: {
    CHISINAU: {
      firstName: 'Иван',
      lastName: 'Петров',
      email: 'ivan.petrov@example.com',
      phone: '+37360123456',
      city: 'Кишинев',
      street: 'ул. Штефан чел Маре',
      building: '124',
      apartment: '15',
      postalCode: 'MD-2012',
      notes: 'Звонить за час до доставки'
    },

    BALTI: {
      firstName: 'Мария',
      lastName: 'Иванова',
      email: 'maria.ivanova@example.com',
      phone: '+37368234567',
      city: 'Бельцы',
      street: 'ул. Независимости',
      building: '45',
      apartment: '8',
      postalCode: 'MD-3100',
      notes: 'Не звонить в домофон'
    }
  },

  DELIVERY_METHODS: [
    'Курьерская доставка',
    'Самовывоз',
    'Экспресс доставка'
  ],

  PAYMENT_METHODS: [
    'Наличными курьеру',
    'Картой онлайн',
    'Банковский перевод'
  ],

  FILTER_PRICES: {
    MIN: 100,
    MAX: 50000,
    RANGES: [
      { min: 1000, max: 5000 },
      { min: 5000, max: 15000 },
      { min: 15000, max: 30000 }
    ]
  }
};
```

🎯 Priority Test Cases for Safe Mode

Week 1: Core Functionality (150 test cases)

Day 1-2: Smoke Testing (40 cases)

```
typescript
```

```
const SMOKE_TESTS = [
  'Homepage loads in Russian',
  'Homepage loads in Romanian',
  'Language switching works',
  'Navigation menu functional',
  'Search functionality basic',
  'Product catalog displays',
  'Add to cart works',
  'Cart page displays items',
  'Cart counter updates',
  'Checkout form loads (no submission)'
];
```

Day 3-5: Extended Functional (110 cases)

```
typescript
```

```
const CORE_FUNCTIONALITY = {
  MULTILINGUAL: [
    'Russian interface display',
    'Romanian interface display',
    'Language switching preserves cart',
    'Search works in both languages',
    'Category navigation in both languages',
    'Form labels translation',
    'Error messages translation',
    'Cart preservation during language switch'
  ],

  PRODUCT_CATALOG: [
    'Product listing display',
    'Product detail page',
    'Product images and gallery',
    'Price and discount display',
    'Cashback calculation display',
    'Stock availability display',
    'Product characteristics',
    'Related products'
  ],

  CART_OPERATIONS: [
    'Add product to cart',
    'Update quantity in cart',
    'Remove product from cart',
    'Clear entire cart',
    'Cart total calculations',
    'Cashback calculations in cart',
    'Cart state persistence',
    'Cross-language cart maintenance'
  ]
};
```

Week 2: User Journey & Forms (120 cases)

User Authentication (40 cases)

```
typescript
```

```
const AUTH_TESTS = [
  // Email-only registration (NO SMS)
  'Registration with valid email',
  'Registration with invalid email',
  'Registration with weak password',
  'Registration with mismatched passwords',
  'Registration with missing fields',
  'Registration form validation',

  // Login scenarios
  'Login with valid credentials',
  'Login with invalid email',
  'Login with wrong password',
  'Login form validation',
  'Remember me functionality',
  'Logout functionality'
];
```

Checkout Forms (80 cases) - SAFE MODE

```
typescript
```

```
const CHECKOUT_TESTS = [
  // Form filling (NO FINAL SUBMISSION)
  'Delivery form with valid data',
  'Delivery form validation',
  'Phone number format validation',
  'Email format validation',
  'Address field requirements',
  'City selection (Moldova cities)',
  'Postal code validation',

  // Delivery method selection
  'Courier delivery selection',
  'Pickup delivery selection',
  'Express delivery selection',
  'Delivery cost calculation',

  // Payment method selection
  'Cash payment selection',
  'Card payment selection',
  'Transfer payment selection',

  // Order review (STOPPING POINT)
  'Order summary display',
  'Delivery info verification',
  'Payment method verification',
  'Order totals verification',
  'Product list verification'
  // ⚠️ NO ACTUAL ORDER PLACEMENT
];
```

Week 3: Advanced Features (80 cases)

Search and Filtering (40 cases)

```
typescript
```

```
const SEARCH_FILTER_TESTS = [
  'Text search in Russian',
  'Text search in Romanian',
  'Search results pagination',
  'Search suggestions',
  'Empty search results',
  'Price range filtering',
  'Brand filtering',
  'Category filtering',
  'Combined filters',
  'Filter reset functionality'
];
```

Mobile Experience (40 cases)

```
typescript

const MOBILE_TESTS = [
  'Mobile navigation menu',
  'Mobile search functionality',
  'Mobile product browsing',
  'Mobile cart management',
  'Mobile checkout forms',
  'Click-to-call phone numbers',
  'Mobile responsive layout',
  'Touch-friendly interactions'
];
```

Week 4: Cross-browser & Polish (50 cases)

Browser Compatibility (40 cases)

```
typescript

const BROWSER_TESTS = {
  CHROME: ['Core flow', 'Forms', 'Mobile view'],
  FIREFOX: ['Core flow', 'Forms', 'Mobile view'],
  SAFARI: ['Core flow', 'Forms', 'Mobile view'],
  EDGE: ['Core flow', 'Forms', 'Mobile view']
};
```

🔧 Test Implementation Guidelines (Safe Mode)

Page Object Model Structure

```
typescript
```

```
export abstract class BasePage {
  constructor(protected page: Page) {}

  async goto(url: string) {
    await this.page.goto(url);
  }

  async switchLanguage(lang: 'ru' | 'ro') {
    const switcher = lang === 'ru' ? SELECTORS.LANGUAGE_SWITCHER_RU : SELECTORS.LANGUAGE_SWITCHER_RO;
    await this.page.click(switcher);
    await this.page.waitForLoadState('networkidle');
  }

  async waitForLoad() {
    await this.page.waitForLoadState('networkidle');
  }
}

export class CheckoutPage extends BasePage {
  async fillDeliveryForm(data: DeliveryData) {
    await this.page.fill(SELECTORS.DELIVERY_FIRST_NAME, data.firstName);
    await this.page.fill(SELECTORS.DELIVERY_LAST_NAME, data.lastName);
    await this.page.fill(SELECTORS.DELIVERY_EMAIL, data.email);
    await this.page.fill(SELECTORS.DELIVERY_PHONE, data.phone);
    await this.page.fill(SELECTORS.DELIVERY_CITY, data.city);
    await this.page.fill(SELECTORS.DELIVERY_STREET, data.street);
    await this.page.fill(SELECTORS.DELIVERY_BUILDING, data.building);
    await this.page.fill(SELECTORS.DELIVERY_APARTMENT, data.apartment);
    await this.page.fill(SELECTORS.DELIVERY_POSTAL_CODE, data.postalCode);

    if (data.notes) {
      await this.page.fill(SELECTORS.DELIVERY_NOTES, data.notes);
    }
  }

  async selectDeliveryMethod(method: string) {
    await this.page.click(`text="${method}"`);
  }

  async selectPaymentMethod(method: string) {
    await this.page.click(`text="${method}"`);
  }
  async reviewOrder(): Promise<OrderReview> {
    // ⚠️ SAFE MODE: Only review, never submit
    const review = await this.page.locator(SELECTORS.ORDER_REVIEW);
    await expect(review).toBeVisible();

    return {
      deliveryInfo: await this.getDeliveryInfo(),
      paymentMethod: await this.getPaymentMethod(),
      orderItems: await this.getOrderItems(),
      totals: await this.getOrderTotals()
    };
  }

  // ⚠️ INTENTIONALLY EXCLUDED FOR SAFETY
  // async placeOrder() {
  //   // This method is excluded to prevent real orders
  // }
}
```

Safety Utilities

```
typescript
```

```
export class SafeTestUtils {
  // Prevent accidental order submission
  static async blockOrderSubmission(page: Page) {
    await page.addInitScript(() => {
      // Block any form submissions to order endpoints
      const originalSubmit = HTMLFormElement.prototype.submit;
      HTMLFormElement.prototype.submit = function() {
        const action = this.action || window.location.href;
        if (action.includes('/order') || action.includes('/checkout/submit')) {
          console.warn('⚠️ BLOCKED: Order submission prevented in test mode');
          return false;
        }
        return originalSubmit.call(this);
      };
    });
  }

  static generateSafeEmail(): string {
    return `test.safe.${Date.now()}@example.com`;
  }

  static generateSafePhone(): string {
    return `+37360${Math.random().toString().slice(2, 8)}`;
  }
}
```

📊 Test Execution & Safety

Safety Checklist

```
typescript
```

```
const SAFETY_CHECKLIST = {
  BEFORE_TESTS: [
    'Verify test data uses safe email addresses',
    'Confirm no SMS endpoints are called',
    'Check that checkout stops at review',
    'Validate no real orders can be placed'
  ],

  DURING_TESTS: [
    'Monitor for unexpected order submissions',
    'Verify cart operations don\'t affect real inventory',
    'Check language switching works correctly',
    'Ensure form validations work'
  ],

  AFTER_TESTS: [
    'Clean up any test user accounts',
    'Clear any test cart data',
    'Verify no real orders were created',
    'Review test execution logs'
  ]
};
```

CI/CD Configuration

```
yaml
```

```
# .github/workflows/safe-tests.yml
name: Pandashop Safe Tests
on: [push, pull_request]
jobs:
  safe-test:
    runs-on: ubuntu-latest
    env:
      SAFE_MODE: true
      NO_ORDER_SUBMISSION: true
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
      - run: npm ci
      - run: npx playwright install
      - name: Run Safe Tests
        run: npm run test:safe
      - name: Verify No Orders Created
        run: npm run verify:no-orders
```

🎯 Implementation Priority

1. **First Generate**: Safe Page Objects (no order submission methods)
2. **Then**: Form filling and validation helpers
3. **After**: Multilingual support utilities
4. **Next**: Cart management without checkout completion
5. **Finally**: Cross-browser compatibility tests

Safety Features:

- ✅ No SMS integration
- ✅ No final order submission
- ✅ Email-only registration
- ✅ Checkout stops at review
- ✅ Safe test data generation
- ✅ Order submission blocking

This specification provides complete foundation for safely testing Pandashop.md without creating real orders or SMS issues!