**Pest Control CRM - Customer Detail Page Implementation**

**Overview**

Build a comprehensive customer detail page that displays when clicking on a customer from the customer list. This page should serve as a central hub for all customer-related information, prioritizing customer experience, scheduling efficiency, and technician management.

**Technical Stack**

* **Frontend**: React with TypeScript
* **Styling**: Tailwind CSS
* **Database**: Supabase
* **Animations**: Framer Motion (for micro-interactions)
* **State Management**: React hooks (useState, useEffect, useContext)
* **API**: RESTful endpoints with Supabase client

**Database Schema Requirements**

Ensure the following tables exist in Supabase:

**Primary Table: accounts**

-- Customer Profile Fields

customer\_id (PRIMARY KEY, UUID)

name (VARCHAR)

email (VARCHAR)

phone (VARCHAR)

address (TEXT)

city (VARCHAR)

state (VARCHAR)

zip\_code (VARCHAR)

billing\_address (TEXT)

payment\_method (VARCHAR)

billing\_cycle (VARCHAR)

created\_at (TIMESTAMP)

updated\_at (TIMESTAMP)

status (VARCHAR) -- active, inactive, suspended

property\_type (VARCHAR) -- residential, commercial

property\_size (VARCHAR)

access\_instructions (TEXT)

emergency\_contact (VARCHAR)

preferred\_contact\_method (VARCHAR)

**Supporting Tables**

Create these tables with proper foreign key relationships:

* service\_history - All past services with photos and notes
* contracts\_subscriptions - Active and historical contracts
* communication\_logs - All customer communications
* customer\_notes - Internal and technician notes
* pest\_activity\_tracking - Pest pressure and treatment effectiveness
* compliance\_tracking - Regulatory requirements
* technician\_certifications - Staff certification management
* chemical\_usage\_log - EPA compliance tracking
* customer\_photos - Comprehensive photo management
* business\_analytics - Predictive metrics
* customer\_documents - Contract and report storage

**Component Architecture**

**Main Container Component**

// CustomerDetailPage.tsx

interface CustomerDetailPageProps {

customerId: string;

}

const CustomerDetailPage: React.FC<CustomerDetailPageProps> = ({ customerId }) => {

// Component implementation

}

**Required Child Components**

1. **CustomerProfileDashboard** - Hero section with key customer info
2. **ServiceHistoryTimeline** - Interactive service history with analytics
3. **SmartContractManager** - Contract management with auto-renewal
4. **CommunicationHub** - Multi-channel communication center
5. **DualNotesSystem** - Internal vs technician notes
6. **PhotoGalleryManager** - Comprehensive photo organization
7. **BusinessIntelligencePanel** - Predictive analytics dashboard
8. **ComplianceTracker** - Regulatory and quality assurance

**Feature Implementation Requirements**

**1. Customer Profile Dashboard (Priority: Customer Experience)**

**Design Requirements:**

* Clean card-based layout with subtle shadows
* Customer property photo carousel with upload capability
* Status indicator (active/inactive/at-risk) with color coding
* Quick action buttons: Call, Email, SMS, Schedule Service
* Key metrics cards: Total Services, Active Contracts, Last Visit, Next Scheduled

**Functional Requirements:**

* Real-time data updates using Supabase subscriptions
* Auto-save on field changes with visual feedback
* Address validation with Google Maps integration
* Customer tenure recognition with loyalty badges
* AI-powered satisfaction scores with visual ratings

**Technical Implementation:**

// Key features to implement

- Photo upload with drag-and-drop

- Address autocomplete with Google Places API

- One-click communication with Twilio integration

- Real-time status updates

- Customer churn risk indicators

**2. Service History with Predictive Analytics**

**Visual Requirements:**

* Interactive timeline view with hover details
* Color-coded service types (prevention=green, treatment=orange, emergency=red)
* Before/after photo carousels with zoom functionality
* Weather data integration showing service conditions
* Pest activity heatmaps with seasonal trends

**Advanced Features:**

* Treatment effectiveness tracking with success rate charts
* Pest activity predictions based on historical data and weather
* Chemical usage compliance with EPA alerts
* Service pattern analysis for optimal intervals
* AI-recommended scheduling with technician skill matching

**3. Smart Communication Hub**

**Design Pattern:**

* WhatsApp/iMessage style conversation bubbles
* Rich media support (photos, documents, voice notes)
* Real-time typing indicators
* Video consultation capability

**Automation Features:**

* SMS service updates with technician ETA
* AI-suggested responses based on context
* Automated satisfaction surveys post-service
* Seasonal pest alerts for affected areas
* Referral program management

**4. Advanced Notes & Photo System**

**Dual Note Architecture:**

* Internal Notes: Office staff communications (gray theme)
* Technician Field Notes: Service visit notes with GPS (blue theme)
* Customer-Facing Notes: Shareable with customers (green theme)
* Safety Alerts: High-priority warnings (red theme)

**Photo Integration:**

* Before/during/after service photo management
* GPS coordinate capture for field photos
* Voice memo support with transcription
* Photo annotations with drawing tools
* Automatic customer account linking from mobile uploads

**Mobile Technician Features:**

* Voice-to-text note recording
* Offline note creation with sync
* @mention system for staff notifications
* Quick photo capture with batch upload

**5. Business Intelligence Dashboard**

**Predictive Analytics:**

* Customer churn risk scoring with retention strategies
* Revenue forecasting based on seasonal patterns
* Pest outbreak predictions using weather/historical data
* Treatment effectiveness by pest type and method

**Compliance Management:**

* EPA compliance tracking with automated alerts
* Technician certification monitoring
* Chemical usage logging for regulatory reporting
* Service quality checklist enforcement

**6. Advanced Integration Features**

**Financial Integration:**

* Payment processing with automated billing
* QuickBooks/Xero synchronization
* Revenue reporting with profitability analysis

**Marketing Automation:**

* Mailchimp/HubSpot integration for campaigns
* Review management with automated responses
* Lead scoring and nurturing workflows

**Mobile-First Requirements**

**Responsive Design Priorities**

1. **Mobile Technician Interface**
   * Large touch targets for gloved hands
   * GPS-enabled check-ins at locations
   * Comprehensive photo capture with quality validation
   * Voice-to-text capabilities
   * Offline functionality with auto-sync
2. **Field Photo Management**
   * Before/during/after service photos
   * GPS coordinates embedded in all photos
   * Photo annotation tools for issue marking
   * Batch photo upload for multiple locations
   * Automatic customer linking via work orders

**Performance Requirements**

**Loading Benchmarks**

* Initial page load: < 2 seconds
* Data filtering/searching: < 500ms
* Photo uploads: Progress indicators for files > 1MB
* Mobile responsiveness: Smooth on 3G connections

**Scalability Targets**

* Support 100+ concurrent users
* Database queries: < 100ms response times
* Handle 10,000+ documents efficiently
* Process 1000+ API requests per minute

**Implementation Phases**

**Phase 1: Core Functionality (Week 1-2)**

1. Customer Profile Dashboard with basic info display
2. Service History timeline with photo viewing
3. Basic notes system (internal + technician)
4. Photo upload and basic management

**Phase 2: Intelligence Features (Week 3-4)**

1. Predictive analytics integration
2. Advanced scheduling with AI recommendations
3. Communication automation setup
4. Compliance tracking system

**Phase 3: Advanced Features (Week 5-6)**

1. Business intelligence dashboard
2. Third-party API integrations
3. Mobile optimization for technicians
4. Customer portal features

**Phase 4: Optimization (Week 7-8)**

1. Performance optimizations
2. Advanced analytics and reporting
3. Security hardening
4. Testing and quality assurance

**Security & Compliance Requirements**

**Data Protection**

* Role-based access control for different user types
* Audit logging for all customer data changes
* Data encryption at rest and in transit
* GDPR/CCPA compliance with retention policies

**Quality Assurance**

* WCAG 2.1 AA accessibility compliance
* Cross-browser testing (Chrome, Firefox, Safari, Edge)
* Mobile device testing (iOS/Android)
* Performance monitoring and optimization

**API Endpoints Required**

Create the following Supabase RPC functions or REST endpoints:

// Customer data retrieval

GET /api/customers/{customerId}/profile

GET /api/customers/{customerId}/service-history

GET /api/customers/{customerId}/contracts

GET /api/customers/{customerId}/communications

GET /api/customers/{customerId}/notes

GET /api/customers/{customerId}/photos

GET /api/customers/{customerId}/analytics

// Real-time updates

POST /api/customers/{customerId}/notes

PUT /api/customers/{customerId}/profile

POST /api/customers/{customerId}/photos/upload

POST /api/customers/{customerId}/communications

// Predictive features

GET /api/customers/{customerId}/churn-risk

GET /api/customers/{customerId}/service-recommendations

GET /api/customers/{customerId}/pest-predictions

**Success Metrics**

**Customer Experience Targets**

* Customer satisfaction scores > 4.5/5
* Response time to inquiries < 2 hours
* Self-service portal adoption > 60%
* Customer retention rate > 90%

**Operational Efficiency Goals**

* Route optimization saves 15% travel time
* Service completion rate > 95%
* Photo documentation compliance > 90%
* Automated communication reduces manual work by 40%

**Business Growth Objectives**

* Customer lifetime value increase by 25%
* New customer acquisition through referrals up 30%
* Revenue per technician increase by 20%
* Market expansion capabilities

**Development Guidelines**

**Code Quality Standards**

* TypeScript strict mode enabled
* ESLint and Prettier configuration
* Comprehensive error handling
* Unit tests for critical functions
* Component documentation with Storybook

**State Management Pattern**

* Use React Context for global customer data
* Local state for component-specific UI states
* Supabase real-time subscriptions for live updates
* Optimistic updates for better UX

**File Structure**

src/

├── components/

│ ├── customer/

│ │ ├── CustomerDetailPage.tsx

│ │ ├── CustomerProfileDashboard.tsx

│ │ ├── ServiceHistoryTimeline.tsx

│ │ ├── SmartContractManager.tsx

│ │ ├── CommunicationHub.tsx

│ │ ├── DualNotesSystem.tsx

│ │ ├── PhotoGalleryManager.tsx

│ │ ├── BusinessIntelligencePanel.tsx

│ │ └── ComplianceTracker.tsx

├── hooks/

│ ├── useCustomerData.ts

│ ├── useRealtimeUpdates.ts

│ └── usePredictiveAnalytics.ts

├── types/

│ └── customer.types.ts

└── utils/

├── supabase.ts

├── analytics.ts

└── integrations.ts

This implementation should create a comprehensive, modern customer management interface that significantly enhances the user experience while providing powerful business intelligence and operational efficiency tools.