### **Software Requirements Specification (SRS)**

### **Project: VA Computer Guy Website Redevelopment**

Version: 1.1

Date: August 16, 2025

### **1. Introduction**

#### **1.1 Purpose**

This document provides a detailed specification of the requirements for the new vacomputerguy.com website. Its purpose is to serve as the guiding document for the design, development, and testing phases of the project. It aims to provide a clear, unambiguous understanding of the system to be built, ensuring that the final product aligns with the strategic vision and business goals of VA Computer Guy.

#### **1.2 Scope**

The project scope includes the complete redesign and development of the public-facing website. This encompasses a new user interface (UI), an improved user experience (UX), integration with third-party services (Square API), development of new customer-facing features (Instant Quote Generator, Repair Status Tracker), and migration of existing SEO value. The project will deliver a fully responsive, secure, and performant website accessible on modern web browsers across desktop and mobile devices.

This SRS does not cover internal administrative tools beyond the specified Headless CMS and the technician-facing interface for updating repair statuses.

#### **1.3 Definitions, Acronyms, and Abbreviations**

* **API:** Application Programming Interface
* **CMS:** Content Management System
* **CTA:** Call to Action
* **SEO:** Search Engine Optimization
* **SRS:** Software Requirements Specification
* **UI:** User Interface
* **UX:** User Experience
* **WCAG:** Web Content Accessibility Guidelines

### **2. Overall Description**

#### **2.1 Product Perspective**

The new website will replace the existing vacomputerguy.com and will be the primary digital entry point for all potential and existing customers. It must integrate seamlessly with the existing business operations, specifically the **Square platform**, which is used for appointment booking and payment processing. The system architecture will be a decoupled frontend application communicating with a set of serverless functions that interface with external APIs and a dedicated database for custom features.

#### **2.2 Product Functions**

* **Inform:** Clearly present services, pricing, and company information.
* **Engage:** Provide tools like an Instant Quote Generator to capture user interest and pre-qualify leads.
* **Convert:** Facilitate seamless, integrated appointment booking and protection plan sign-ups, increasing conversion rates.
* **Support:** Offer post-service support through a Repair Status Tracker and accessible contact methods, improving customer satisfaction.
* **Build Authority:** Establish VA Computer Guy as a local expert through a content-rich blog and detailed service pages, driving organic traffic.

#### **2.3 User Characteristics**

1. **Potential & Existing Customers:** Residents and small business owners in the Virginia Beach / Hampton Roads area. They may be under stress due to technology issues and require a simple, fast, and reassuring user experience. Technical proficiency will vary widely.
2. **Site Administrators / Staff:** VA Computer Guy staff who will manage blog content via the CMS and update repair statuses. They are assumed to be technically proficient in their respective domains but not necessarily in web development.

#### **2.4 Constraints**

* **C-1:** The system **shall** use the Square API for all appointment scheduling and payment processing to maintain continuity with existing business systems.
* **C-2:** The design **shall** adhere to the branding guidelines (logos, color palette, fonts) of VA Computer Guy.
* **C-3:** The system **shall** be hosted on a modern, scalable platform (Vercel or Netlify) that supports serverless functions and continuous deployment.
* **C-4:** The chosen Headless CMS **must** have a user-friendly interface for non-technical staff.

#### **2.5 Assumptions and Dependencies**

* **A-1:** VA Computer Guy will provide high-quality professional photography of the team, workspace, and service vehicles in required digital formats.
* **A-2:** The Square API will be available during development and will provide the necessary endpoints and documentation for seamless integration.
* **A-3:** All valuable content and a comprehensive list of high-performing URLs from the current site will be provided for the SEO migration strategy.

### **3. Specific Requirements**

#### **3.1 Functional Requirements**

##### **FR-1: Instant Quote Generator**

* **FR-1.1:** The system shall present an interactive, multi-step form on the homepage.
* **FR-1.2 (Data Collection):** The form shall collect the following data points:
  + **FR-1.2.1:** User Type (Home / Business).
  + **FR-1.2.2:** Device Type (e.g., Laptop, Desktop, Mac, PC, Phone).
  + **FR-1.2.3:** Issue Category (e.g., Slow Performance, Virus, Cracked Screen, Won't Turn On).
  + **FR-1.2.4:** Service Location (In-Store / On-Site).
* **FR-1.3 (Logic):** The system shall use a predefined pricing matrix to calculate and display an estimated price range based on the user's selections.
* **FR-1.4 (Output):** The final step shall display the estimated price range and a prominent CTA to "Book This Repair," which pre-fills the booking form with the selected service.

##### **FR-2: Integrated Appointment Booking**

* **FR-2.1:** The system shall embed the entire appointment booking process within a single, seamless interface on the website.
* **FR-2.2 (API):** The system shall make real-time calls to the Square Appointments API to fetch available services, staff, and time slots.
* **FR-2.3 (User Flow):**
  + **FR-2.3.1:** User selects a service category and specific service.
  + **FR-2.3.2:** User views a calendar of available dates and times.
  + **FR-2.3.3:** User selects a time slot and enters their contact information (Name, Email, Phone).
  + **FR-2.3.4:** User confirms the appointment. No payment is required at this stage.
* **FR-2.4 (Confirmation):** Upon successful booking, the system shall display a confirmation page with all appointment details and a unique booking ID. An automated confirmation email shall be triggered.
* **FR-2.5 (Upsell):** The confirmation page shall feature a module promoting the relevant Protection Plan with a special offer for new repair customers.

##### **FR-3: Protection Plan Showcase & Sign-up**

* **FR-3.1:** The system shall have a dedicated, visually engaging page for Protection Plans.
* **FR-3.2:** The page shall feature an interactive comparison table for Residential and Business plans, detailing features like "Proactive Monitoring," "Data Backup," "Antivirus," etc.
* **FR-3.3:** The table shall include a toggle to switch between Monthly and Annual pricing, dynamically updating the displayed price and highlighting the annual savings.
* **FR-3.4 (Sign-up):** The "Sign Up Now" CTA shall lead to a multi-step form that collects user information and payment details.
* **FR-3.5 (Payment):** The payment form shall securely process the subscription payment via the Square Subscriptions API.

##### **FR-4: Live Repair Status Tracker**

* **FR-4.1:** The system shall provide a public-facing page at /repair-status for customers.
* **FR-4.2 (Authentication):** Users shall enter their unique Repair Ticket Number and the Last Name associated with the ticket to access status information.
* **FR-4.3 (Statuses):** The system shall display one of the following statuses:
  + Received: "We've received your device and it's in the queue for diagnosis."
  + In Diagnosis: "Our technician is currently diagnosing the issue."
  + Awaiting Parts: "We've ordered a necessary part. Estimated arrival: [Date]."
  + In Repair: "Your device is actively being repaired."
  + Quality Check: "The repair is complete and is undergoing final testing."
  + Ready for Pickup: "Your device is ready for pickup!"
* **FR-4.4 (Technician Interface):** A secure, simple backend interface (not part of the public site) shall allow technicians to log in, search for a ticket number, and update its status from a dropdown menu.

##### **FR-5: Content Management & SEO**

* **FR-5.1:** The system shall integrate with a Headless CMS (e.g., Sanity.io) for managing the "Tech Tips" blog, individual service pages, and the "About Us" page.
* **FR-5.2 (Metadata):** The CMS shall provide fields for editors to manage SEO metadata for each page, including Meta Title, Meta Description, and Open Graph tags.
* **FR-5.3 (Redirects):** A comprehensive 301 redirect map shall be implemented at the hosting level to map all high-value old URLs to their new counterparts.
* **FR-5.4 (Performance):** All content pages shall be rendered using Server-Side Rendering (SSR) or Static Site Generation (SSG) for optimal load times and crawlability.

#### **3.2 External Interface Requirements**

* **EIR-1: Square API:** The system will interface with the Square API for retrieving appointment slots, creating appointments, and processing subscription payments. All API communication will be handled server-side to protect credentials.
* **EIR-2: Live Chat Service:** The system will embed a JavaScript widget from a third-party provider (e.g., Tawk.to). The widget shall load asynchronously to avoid impacting page performance.

#### **3.3 User Interface Requirements**

* **UI-1:** The website shall be fully responsive and functional across the latest two versions of all major browsers and devices.
* **UI-2:** The design shall be clean, modern, and professional, instilling trust and competence.
* **UI-3:** Navigation shall be intuitive, with a persistent main menu and clear CTAs on every page.
* **UI-4:** All forms shall include client-side and server-side validation with clear, user-friendly error messages.

#### **3.4 Non-Functional Requirements**

* **NFR-1: Performance**
  + **NFR-1.1:** Largest Contentful Paint (LCP) shall be under 2.5 seconds.
  + **NFR-1.2:** The website shall achieve a Google Lighthouse performance score of 90+ for both mobile and desktop.
* **NFR-2: Security**
  + **NFR-2.1:** All traffic shall be served over HTTPS.
  + **NFR-2.2:** All user input shall be sanitized to prevent XSS and other injection attacks.
  + **NFR-2.3:** API keys shall be stored as secure environment variables on the server.
* **NFR-3: Usability & Accessibility**
  + **NFR-3.1:** The website shall meet WCAG 2.1 AA accessibility standards, including keyboard navigation and screen reader compatibility.
  + **NFR-3.2:** The core user flow of booking an appointment shall be completable in 5 steps or fewer from the homepage.

#### **3.5 Data Model (High-Level)**

* **RepairTicket:**
  + ticketId (String, Primary Key)
  + customerLastName (String)
  + status (Enum: Received, In Diagnosis, etc.)
  + notes (String, optional)
  + lastUpdated (Timestamp)

### **4. System Architecture**

The system will be built using a modern JAMstack architecture.

* **Frontend:** A Next.js (React) application will be responsible for the UI and all client-side interactions.
* **Backend Logic:** API routes within Next.js (serverless functions) will handle all business logic, including communication with the Square API and the repair status database. This ensures security and separation of concerns.
* **Database:** A simple, scalable database like Firestore or Supabase will be used exclusively for the Repair Status Tracker feature.
* **CMS:** A Headless CMS will serve content to the Next.js frontend via an API.

### **5. Proposed Technology Stack**

* **Frontend:** React (Next.js)
* **Language:** TypeScript
* **Styling:** Tailwind CSS
* **UI Components:** Shadcn/UI
* **Booking & Payments:** Square API
* **Content Management:** Headless CMS (e.g., Sanity.io)
* **Custom Backend/Database (for Repair Tracker):** Serverless Functions with Firestore/Supabase
* **Live Chat:** Tawk.to or Crisp
* **Deployment/Hosting:** Vercel or Netlify