ScamGuard: AI Scam Protection for Seniors

Design Document v1.0

1. Executive Summary

Project Name: ScamGuard

Purpose: A macOS desktop application designed to protect senior citizens from AI-powered scams by monitoring emails and analyzing screenshots for deepfakes, voice cloning attempts, and other AI-generated fraudulent content.

Target Users: Senior citizens (65+) and their trusted contacts (family members, caregivers)

Core Value Proposition: Proactive, user-friendly protection against increasingly sophisticated AI scams with educational explanations rather than just warnings.

2. Product Overview

2.1 Problem Statement

Senior citizens are increasingly targeted by AI-powered scams including:

- Deepfake videos impersonating family members or authority figures
- AI-generated voice cloning in phone scams
- Sophisticated phishing emails with AI-generated content
- Fake customer service interactions

Traditional security solutions are often too technical and reactive. Seniors need proactive, understandable protection.

2.2 Solution

A macOS desktop app that:

1. Monitors Gmail accounts for suspicious emails (automated)
2. Analyzes screenshots on-demand for AI-generated content (manual trigger)
3. Provides clear, non-technical explanations of detected threats
4. Alerts trusted contacts when significant threats are detected
5. Empowers seniors with understanding rather than fear

3. Core Features

3.1 Email Monitoring (Automated)

Functionality:

- Polls Gmail account every 25-30 minutes via Gmail API
- Analyzes email content, attachments, and metadata
- Detects AI-generated text, suspicious links, and phishing attempts
- Flags emails with deepfake images or AI-generated content

User Experience:

- Passive background monitoring after initial OAuth setup
- Non-intrusive banner notifications when threats detected
- Email remains in inbox (not moved or deleted)

3.2 Screenshot Analysis (Manual Trigger)

Functionality:

- Menu bar icon with "Check for AI" option
- User clicks when suspicious of content on screen (video call, website, image)
- Captures screenshot and sends to Gemini Vision API
- Analyzes for AI watermarks, deepfake indicators, synthetic media

User Experience:

- Always visible menu bar icon for quick access
- Single click to trigger analysis
- Results shown within 3-5 seconds
- Clear explanation of findings

3.3 Fraud Detection System

Technology Stack:

- **Primary Model:** Google Gemini Vision API (multi-modal)
- **Rationale:** Built-in watermark detection for Gemini-generated images
- **Detection Targets:**
  - Deepfake videos and images
  - AI-generated text patterns
  - Voice cloning indicators (for future call analysis)
  - Phishing email patterns
  - Suspicious URLs and attachments

Detection Strategy:

- Prioritize false positives over false negatives (better safe than sorry)
- Multi-factor analysis (content + metadata + context)
- Confidence scoring for transparency

3.4 Alert System

For Seniors (Primary User):

- **Format:** Banner notification (non-blocking)

- **Content:**
  - Clear headline: "This might be a scam"
  - Simple explanation: "This email asks for personal information and may not be from your real bank"
  - Recommended action: "Don't click any links. Call your bank directly."
  - No technical jargon or model confidence scores

For Trusted Contacts:

- **Trigger Options:**
  1. Every time senior receives alert (high vigilance)
  2. Only when senior requests help (senior-controlled)
- **Delivery:** SMS or Email (user preference during setup)
- **Content Format:** "Alert: [Senior's name] has been warned of a scam. Here is a summary of the situation: [Brief description]. Consider checking in with them."

3.5 Trusted Contact Management

Setup:

- Add up to 5 trusted contacts
- Each contact provides: Name, Phone, Email, Relationship
- Can be added during onboarding or later
- Each contact configures their notification preferences

Privacy:

- Seniors control what information is shared
- Contacts cannot access the app or monitor without alerts
- Seniors can remove contacts at any time

---

4. Technical Architecture

4.1 System Overview

```
┌─────────────────────────────────────────────────────┐
│                 macOS Desktop App                     │
│  ┌───────────────────────────────────────────────┐  │
│  │        User Interface Layer (SwiftUI)         │  │
│  └───────────────────────────────────────────────┘  │
│                                                     │
│  ┌───────────────────────────────────────────────┐  │
│  │          Application Logic Layer              │  │
│  │ • Email Poller (25-30 min intervals)          │  │
│  │ • Screenshot Capture Service                  │  │
│  │ • Alert Manager                               │  │
│  │ • Contact Notification Service                │  │
│  └───────────────────────────────────────────────┘  │
│                                                     │
│  ┌───────────────────────────────────────────────┐  │
│  │           Integration Layer                   │  │
│  │  • Gmail API Client (OAuth)                   │  │
│  │  • Gemini Vision API Client                   │  │
│  │  • SMS/Email Gateway (Twilio/SendGrid)        │  │
│  └───────────────────────────────────────────────┘  │
│                                                     │
│  ┌───────────────────────────────────────────────┐  │
│  │           Data Storage Layer                  │  │
│  │  • Local SQLite DB (alerts, settings)         │  │
│  │  • Secure Keychain (API keys, tokens)         │  │
│  └───────────────────────────────────────────────┘  │
│                                                     │
└─────────────────────────────────────────────────────┘
```

4.2 Technology Stack

Frontend:

- **Framework:** SwiftUI (native macOS)
- **Menu Bar Integration:** NSStatusItem
- **Notifications:** UserNotifications framework

Backend Services:

- **Email:** Gmail API (REST) with OAuth 2.0
- **AI Detection:** Google Gemini Vision API
- **Notifications:** Twilio (SMS), SendGrid (Email)
- **Storage:** SQLite for local data, macOS Keychain for credentials

Development:

- **Language:** Swift 5.9+
- **Min macOS Version:** macOS 12.0 (Monterey)
- **Package Manager:** Swift Package Manager

4.3 Data Flow

Email Monitoring Flow:

```
1. Background timer triggers every 25-30 minutes
2. Gmail API fetches new emails since last check
3. Each email passed to Gemini API for analysis
4. Results stored in local database
5. If threat detected:
   a. Show banner notification to senior
   b. Log incident with timestamp and details
   c. If trusted contacts opted in → send notifications
```

Screenshot Analysis Flow:

```
1. User clicks menu bar icon → "Check for AI"
2. App captures current screen screenshot
```

```
3. Screenshot sent to Gemini Vision API
4. API analyzes for AI-generated content
5. Results parsed and translated to simple language
6. Display banner with findings and explanation
7. If threat detected → same alert flow as email
```

4.4 API Integration Details

Gmail API:

- **Scope:** `gmail.readonly` (read-only access)
- **Authentication:** OAuth 2.0 with refresh tokens
- **Rate Limits:** 250 quota units per user per second (sufficient for 30-min polling)
- **Data Accessed:** Email body, headers, attachments (metadata only)

Gemini Vision API:

- **Model:** `gemini-1.5-pro-vision` or latest
- **Input:** Base64-encoded images, email text
- **Prompt Strategy:** "Analyze this content for signs of AI generation, deepfakes, or scam indicators. Provide confidence score and reasoning."
- **Rate Limits:** Monitor usage, implement exponential backoff

Notification Services:

- **Twilio SMS:** For real-time urgent alerts
- **SendGrid Email:** For detailed summaries
- **Fallback:** Local notifications if services unavailable

---

5. User Experience Design

5.1 Onboarding Flow

Step 1: Welcome & Purpose

- Large, clear text explaining what the app does
- Emphasis on protection and peace of mind
- "Continue" button (large, high contrast)

Step 2: Gmail Connection

- "Connect your email to start protection"
- Large "Connect Gmail" button
- OAuth flow in default browser
- Success confirmation: "✓ Email connected"

Step 3: Alert Preferences

- "How would you like to be notified?"
- Radio buttons (large touch targets):

  - "Show me a message on my computer" (default)
  - "Also send me an email"
- Can be changed later in settings

Step 4: Trusted Contacts (Optional)

- "Add people who can help if you need it"
- "Skip for now" option prominent
- If adding: simple form (Name, Phone/Email, Relationship)
- Can add up to 5 contacts

Step 5: Tutorial

- Quick visual guide showing:
  1. Menu bar icon location
  2. How to use "Check for AI" feature
  3. What alerts look like
- "Start Protection" button to complete

5.2 Main Interface

Menu Bar Icon:

- Shield icon (always visible)
- Green when active, yellow during scan, red if threat detected
- Click reveals dropdown menu:
  - "Check for AI" (with keyboard shortcut)
  - "View Recent Alerts"
  - "Settings"
  - "Help & Support"
  - "Quit"

Alert Banner Design:

- Large, colorful banner (warning orange/red)
- Big, clear icon (⚠)
- Headline: 18pt bold font
- Explanation: 16pt regular font, 2-3 sentences max
- Action buttons: "Learn More" | "Dismiss" | "Get Help"
- Stays on screen until dismissed (but can be minimized)

Settings Panel:

- Simple, organized sections:
  - Email Monitoring (on/off toggle, last check time)
  - Trusted Contacts (add/edit/remove)
  - Alert Preferences
  - About & Privacy Policy

5.3 Accessibility Features

For Seniors:

- **Large Text:** Minimum 16pt for body text, 18pt+ for headings
- **High Contrast:** WCAG AAA compliant color ratios
- **Clear Language:** 6th grade reading level, no jargon
- **Simplified Navigation:** Maximum 3 clicks to any feature
- **Keyboard Shortcuts:** All features accessible via keyboard
- **VoiceOver Support:** Full screen reader compatibility

---

6. Security & Privacy

6.1 Data Handling

What We Store Locally:

- Alert history (last 90 days, then auto-deleted)
- User preferences and settings
- Trusted contact information (encrypted)
- OAuth tokens (in macOS Keychain)

What We DON'T Store:

- Full email content (only metadata and analysis results)
- Screenshots (deleted immediately after analysis)
- Financial information or passwords
- Browsing history or other personal data

6.2 Privacy Controls

User Consent:

- Explicit opt-in for each monitoring feature
- Can disable email monitoring anytime
- Screenshots only taken on manual trigger
- Clear privacy policy in plain language

Data Transmission:

- All API calls over HTTPS/TLS
- Email content sent to Gemini only for analysis (not stored)
- No data sold or shared with third parties
- Gemini API set to no-training mode

6.3 Compliance

- GDPR considerations for data minimization
- California Consumer Privacy Act (CCPA) compliance
- Clear data retention and deletion policies

---

7. Implementation Phases

Phase 1: MVP (Proof of Concept) - 8 weeks

Week 1-2: Foundation

- macOS app skeleton with SwiftUI
- Basic menu bar integration
- Settings storage (SQLite)

Week 3-4: Email Integration

- Gmail API OAuth implementation
- Email polling service (30-min intervals)
- Basic email content extraction

Week 4-5: AI Detection

- Gemini Vision API integration
- Email analysis pipeline
- Threat classification logic

Week 6-7: Alert System

- Banner notification UI
- Trusted contact management
- SMS/Email notification service

Week 8: Polish & Testing

- Onboarding flow
- Bug fixes and refinements
- Internal testing with seniors

Phase 2: Screenshot Analysis - 2 weeks

- Screenshot capture implementation
- Menu bar "Check for AI" feature
- Visual content analysis pipeline

Phase 3: Enhancement - 4 weeks

- Alert history dashboard
- Improved explanations with examples
- Performance optimization
- Beta testing with target users

Future Phases (Post-POC)

- iOS mobile app version
- Android call monitoring
- Voice analysis for recorded calls

- Browser extension for web protection
- Multi-language support

8. Success Metrics

Primary Metrics

- **Detection Accuracy:** % of actual scams caught (target: 95%+)
- **False Positive Rate:** % of legitimate content flagged (target: <10%)
- **User Satisfaction:** Survey score (target: 4.5/5)
- **Active Users:** % of installed users checking weekly (target: 70%+)

Secondary Metrics

- Time to first detection
- Number of trusted contacts added per user
- Feature usage (email vs screenshot analysis)
- User retention after 30/60/90 days

Qualitative Metrics

- User testimonials and feedback
- Reduction in reported scam losses
- Trusted contact satisfaction
- Ease of use ratings from seniors

9. Risks & Mitigations

Risk	Impact	Likelihood	Mitigation
Seniors find app too complex	High	Medium	Extensive user testing, simplified UI, in-person training option
High false positive rate causes alert fatigue	High	Medium	Tune detection thresholds, allow feedback to improve model
Gmail API access revoked/limited	High	Low	Implement fallback methods, maintain OAuth compliance
Gemini API costs exceed budget	Medium	Medium	Rate limiting, caching, local pre-filtering
Privacy concerns from seniors/families	High	Low	Transparent privacy policy, local-first processing where possible
Senior forgets to install/check app	Medium	High	Onboarding by family member, periodic reminder emails

10. Open Questions & Decisions Needed

1. **Pricing Model:** Free for POC, but long-term sustainability?
   - Options: Freemium, family subscription, one-time purchase
2. **Data Retention:** How long should alert history be kept?
   - Current: 90 days, but should users be able to export?
3. **Escalation Paths:** What if senior dismisses alerts repeatedly?
   - Should trusted contacts be auto-notified after X dismissals?
4. **Multi-Account Support:** Should one app monitor multiple Gmail accounts?
   - Use case: Seniors with multiple email addresses
5. **Offline Mode:** What happens if internet connection drops?
   - Should there be cached detection capabilities?

---

11. Appendix

A. User Personas

Primary Persona: Margaret (Senior User)

- Age: 72
- Tech comfort: Low-Medium (uses email, FaceTime)
- Main concern: Protecting savings from scams
- Motivation: Wants to stay independent, not burden family
- Pain points: Confusing security warnings, fear of technology

Secondary Persona: Sarah (Trusted Contact)

- Age: 45, Margaret's daughter
- Tech comfort: High
- Main concern: Mother's safety while respecting independence
- Motivation: Peace of mind without constant check-ins
- Pain points: Too many false alarms, unclear action items

B. Competitive Analysis

Existing Solutions:

- **Cluely:** Desktop monitoring app (inspiration for architecture)
- **RoboKiller:** Call blocking (doesn't handle AI scams specifically)
- **Norton/McAfee:** General security (too complex for seniors)
- **Gmail's built-in filters:** Basic phishing detection (not AI-focused)

ScamGuard Differentiators:

1. AI-specific scam detection (deepfakes, voice cloning)

2. Senior-first UX with educational approach
3. Trusted contact integration
4. Manual screenshot analysis for on-demand checking
5. Explanatory alerts instead of technical warnings

C. Technical Glossary for Stakeholders

- **OAuth:** Secure method for connecting to Gmail without sharing passwords
- **API:** Way for our app to communicate with external services (Gmail, Gemini)
- **Gemini Vision:** Google's AI that can analyze both text and images
- **Deepfake:** AI-generated fake video or image of a real person
- **Menu Bar:** Top bar on Mac screen with clock, WiFi icon, etc.
- **SQLite:** Lightweight database for storing information on user's computer

---

Document Control

Version: 1.0
Last Updated: January 2026
Owner: [Project Team]
Status: Ready for Review
Next Review: After POC completion