Product Requirements Document (PRD): UI-TARS Desktop v3.0

1. Project Overview

The goal of version 3.0 is to transform UI-TARS Desktop into a robust, production-ready local AI agent. The primary focus is on stabilizing local VLM communication and optimizing performance for high-frequency task execution.

2. Core Objectives

- **Fix Action Parsing**: Implement a resilient heuristic parser in `actionParser.ts` to correctly extract coordinates `[x, y]` from non-standard local VLM outputs (Ollama/LM Studio).
- **Native Ollama Integration**: Replace the generic OpenAI SDK with a dedicated `OllamaModel` class to communicate directly with Ollama' s native API (`/api/chat`).
- **Performance Optimization**: Enable multi-threaded processing to support "thousands of requests per hour" without UI freezing.
- **Enhanced Debugging**: Add a real-time log panel in the React frontend to monitor coordinate generation and mouse actions.

3. Technical Requirements

- **Backend**: Node.js / TypeScript (Electron environment).
- **VLM Provider**: Local Ollama instance running on port 11434.
- **Model**: UI-TARS-1.5-7B (GGUF Quantized).
- **Permissions**: Application must bypass Windows UAC to maintain persistent Administrator privileges for mouse/keyboard control.

4. Success Criteria

- The mouse must move to the correct coordinates within 2 seconds of model response.
- The system should handle fallback parsing when JSON formatting fails.
- Zero-latency communication between the Electron frontend and the local VLM server.