SmartOS – Voice/Chat-Based AI Operating SystemAssistant

Version 1.0

Prepared for

myOnsite Healthcare, LLC.

Document Control

Rev. No.	Description of Change	Effective Date
1.0	Initial Release	22nd Aug 2025

Authored By

Name	Role	Signature	Date
myOnsite Healthcare			22nd Aug 2025

Reviewed and approved By

Name	Role	Signature	Date

SmartOS – Voice/Chat-Based AI Operating System Assistant

Project Brief

SmartOS is a lightweight AI OS assistant that transforms natural language (text or voice) into executable system tasks. Designed as a 'local ChatGPT meets Windows Spotlight meets AutoGPT', it empowers users to control and operate desktop systems using plain English commands. From writing essays to opening apps and toggling connectivity, SmartOS enables hands-free, intelligent automation on the user's OS.

Time Allocation: 5 hours

Focus Areas: - Voice Assistant Integration - Desktop Automation - AI Command Parsing - System Control Layer

System Overview

- **Processes voice or text commands to understand intent**
- **Executes system-level actions (e.g., open apps, connect WiFi, write files)**
- **Automates content creation and OS navigation**
- **Maintains fallback chat interface for manual operations**

Data & Requirements

- Input data: Natural language requirements, user stories, or commands
- Quantity: Unlimited
- Evaluation dataset: 50 test cases split across 3 tiers (easy: 40%, medium: 40%, hard: 20%)
- Characteristics: Ambiguity, varied domain language, real-world scenarios

Technical Requirements

- Primary: Natural Language Understanding (NLU), Task Execution Automation
- Secondary: Headless Browser Automation / Shell-level OS commands
- Integration: Git, System APIs, Voice Recognition, File Management

Advanced/Challenging Requirements

- Dynamic command interpretation and fallback handling
- Screenshot capture and error logging (for TestSmith)
- Background execution support
- Multi-modal interaction (text/voice)

Output Schema / API Specification

- JSON and Excel-based result logging
- Fields: Test Name, Description, Status, Error, Screenshot link
- CLI/Voice return messages and file output for SmartOS

Evaluation Framework

- Metric 1: Accuracy (Manual validation of execution correctness)
- Metric 2: Latency (Time to execute task)
- Metric 3: Robustness (Recovery from bad inputs)
- Performance benchmarking across all task categories

System Adaptation / Flexibility Requirements

- Must support JSON-based input and output
- Easily adaptable to different domains (e.g., healthcare, finance)
- Hot-swap between CLI and GUI (SmartOS) or languages (TestSmith AI)

Implementation Challenges

- Parsing and grounding ambiguous natural language
- Handling OS-level security and permission contexts
- Maintaining deterministic outputs for automated evaluation

Evaluation Complexity

- Tier 1: Simple tasks (e.g., open app)
- Tier 2: Multi-step tasks (e.g., edit and save a file)
- Tier 3: Integrative tasks (e.g., build & deploy project or run multi-case tests)

Advanced Challenge Specifications

- Zero-downtime requirement for CLI tools
- API hot-reload for plugin extensions
- Multi-tiered fallback strategies for assistant

Deliverables

1. System:
- Executable packages or Docker containers
2. Evaluation:
- Test cases, logs, metrics dashboard
3. Documentation:
- Installation guide, user manual, API specs

Success Criteria

- Pass rate >90% on evaluation dataset
- Response time <3 seconds for 80% of tasks
- Full autonomous task execution without manual correction