

Analyze User Request Constraints:

```
# Analyze User Request Constraints

# Overview
Ensure the accuracy of constraint analysis in user requests reaches **≥95%**

# Execution Plan

## **1. Constraint Identification Checklist**
[ ] **Technical Constraints**: List hardware/software limitations (e.g., "Must work on iOS 15+").
[ ] **Budget Constraints**: Specify financial limits (e.g., "Max $500/month").
[ ] **Timeline Constraints**: Define deadlines (e.g., "Launch by Q3 2024").
[ ] **Regulatory Constraints**: Note compliance requirements (e.g., "GDPR-compliant").
[ ] **User Constraints**: Document user preferences (e.g., "No dark mode").
[ ] **Performance Constraints**: Specify speed/latency requirements (e.g., "Load time <1s").
[ ] **Scalability Constraints**: Define growth expectations (e.g., "Support 10,000 users").

## **2. Validation Process**
  **Test Case Template**
  - **Constraint**: [e.g., "Must support 1000+ concurrent users"]
  - **Validation Method**: [e.g., "Load test with 1200 users"]
  - **Expected Outcome**: [e.g., "No crashes or latency >2s"]
  - **Actual Outcome**: [Fill] after testing
  - **Pass/Fail**: ✓ ✗

## **3. Quality Assurance**
  **Accuracy Metric**: ≥95% of constraints must be correctly identified and validated.
  **Review Frequency**: Weekly cross-checks with stakeholders.
  **Documentation**: Store all test results in a shared folder.

# Distribution & Maintenance
  **Publish**: Deploy the validated test on the venture's free website.
  **Promote**: Share via newsletter and social media.
  **Maintain**: Monthly updates based on user feedback.
  **Feedback Loop**: Collect user feedback via a dedicated form.

# Rules
  **Kill Rule**: Stop if 3 iterations fail to generate revenue.
  **Boost Rule**: Prioritize if revenue exceeds $1M.
  **Clone Rule**: Duplicate if revenue hits $50K.
  **Pause Rule**: Temporarily halt if no feedback for 2 months.
```

