**Automatic Test Checking Service (ATCS) - Software Requirements Specification (SRS)**

**1. Introduction**

This document outlines the Software Requirements Specification (SRS) for an Automatic Test Checking Service (ATCS) written in Python. This program will be used by teachers to manage automated grading of checklist-style tests and by students to get results.

**2. Overall Description**

The ATCS is a desktop application designed to streamline the grading of checklist-based tests. It leverages Optical Character Recognition (OCR) technology to automatically recognize student answers on scanned test sheets.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Teacher Functionality**

* **Grading:**
  + Upload scanned answer sheets (image files) for a specific test.
  + The system automatically recognizes answers on the test sheets using OCR.
  + The system automatically grades each test based on predefined checklist items and answer choices (marked or unmarked).
  + The system generates a report with:
    - Individual student scores and detailed answer breakdown.

**3.1.2 Student Functionality**

* **Test Upload:**
  + Upload scanned test sheets (image files) for a specific test.
  + The system automatically recognizes student answers on the test sheets using OCR.
  + The system automatically grades each test based on predefined checklist items and answer choices (marked or unmarked).
  + The system generates a report with:
    - Individual student scores and detailed answer breakdown.

**3.2 Non-Functional Requirements**

* **Performance:**
  + The application should be responsive and provide quick feedback during test creation, grading, and report generation.
  + OCR processing time should be optimized for efficient test grading.
* **Usability:**
  + The user interface should be intuitive and easy to navigate for both teachers and students.
  + Provide clear instructions and feedback throughout the workflow.
  + Offer clear visual cues for checklist items and answer marking on test sheet templates (if applicable).
* **Compatibility:**
  + The application should run on major operating systems (Windows).
  + Support various image formats for scanned test sheets (e.g., JPG, PNG, PDF).

**4. System Interfaces**

**4.1 User Interfaces**

* A GUI for,uploading scanned tests, viewing reports.

**4.2 Hardware Interfaces**

The application should function on standard personal computers with a webcam (optional for electronic test submission).

**4.3 Software Interfaces**

* The system will utilize an OCR library for answer recognition.

**5. Design Constraints**

* The program should be developed in Python for portability and ease of use.
* The OCR library should be open-source and well-maintained.
* Consider limitations of OCR technology (handwriting recognition accuracy).

**6. Quality Attributes**

* Accuracy: The OCR engine should provide a high degree of accuracy in recognizing student answers.
* Reliability: The application should function consistently and without errors during test creation, grading, and reporting.
* Maintainability: The code should be well-documented and modular for ease of future modifications.

**7. Revision History**

* Include a table to track the version number, date, and description of changes made to the SRS document.

This SRS focuses on the core functionalities of uploading tests, uploading answers, and getting results. It expands on functionalities compared to previous versions.