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

1. **Introduction**
   1. Purpose

The purpose of this document is to describe in detail the functional and performance requirements of automatic test checking service so that the development team can understand and implement these requirements.

* 1. Range

This program can be used by teachers to manage automated grading of checklist-style tests and by students to verify the correctness of their answers. So the software will provide the basic features like upload and submission, automated grading, detailed feedback, etc.

1. **General description**
   1. Product Perspective

The software will serve as a mobile application for teachers and students to use on their smartphones to submit and grading the homework.

* 1. Product Functions
* Automated Grading
* Detailed Feedback
* Subject-Specific Tools
* Reminder and Notification
* Peer Review
* User Analytics
* Progress Tracking

1. **Specific requirements**
   1. Functional Requirements

3.1.1 Teacher Functionality

* The app should provide automated grading functionality to quickly assess the correctness of answers based on predefined criteria.
* It should offer detailed feedback on the corrections made, explanations for errors, and suggestions for improvement.
* Providing insights into performance trends, strengths, weaknesses, and areas needing more focus.

3.1.2 Student Functionality

* Customized tools for different subjects like math (equation editors), languages (grammar checks), or programming (code compilation).
* Reminders for submission deadlines, notifications for completed corrections, and upcoming tasks.
* Option for peer review where students can review each other’s work and provide feedback.
  1. Non-Functional Requirements

3.2.1 Performance

A good homework grading software should demonstrate the following characteristics to ensure an effective and efficient grading process:

* Accuracy: The software should accurately grade assignments based on predefined criteria, ensuring consistency and fairness in grading.
* Customization: Ability to customize grading criteria and rubrics to align with specific course requirements and learning objectives.
* Feedback Quality: Provide detailed and constructive feedback to students, helping them understand their mistakes and how to improve.

3.2.2 Usability

* The software should provide quick turnaround times for grading, allowing students to receive feedback promptly.
* Capable of handling assignments from a large number of students efficiently, without compromising on accuracy or speed.
* Support for grading different types of assignments, including written responses, code snippets, mathematical equations, and more.

3.2.3 Compatibility

* Seamless integration with learning management systems (LMS) or other educational platforms to streamline the grading process.
* Robust security measures to protect student data, ensure privacy, and comply with data protection regulations.
* The software should run on major operating systems (Windows).

1. **System Interfaces**
   1. User Interfaces

The user interface should include options for submission, correction, score viewing, detailed problem solutions, etc.

* 1. Hardware Interfaces

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

* 1. Software Interfaces
* Grading Interface
* Integration Interfaces
* Feedback Interface
* Communication Interface