
SCORE Project Description

Computer Training for Visually Impaired Automation Tool

1. Introduction

The aim of this project is to design and implement a tool that automate the evaluation of visually impaired students undergoing basic computer training at an NGO called Enable India (<http://www.enable-india.org>). At Enable India, each visually impaired trainee undergoes a series of teaching sessions on packages such as Word, Excel etc. This is followed by exercises that test the expertise of trainees on each module. Each test consists of a set of questions followed by a series of tasks that the trainee must perform on the computer. As of today, trainers spend a considerable amount of time and energy correcting these exercises manually. The accuracy of the corrections can also be affected due to the level of manual intervention involved. The goal of the project is make the process more efficient both with respect to time and the accuracy. The automation tool should allow the creation of the question bank of exercises and let the admin choose the questions from the pool for candidates and design specific question papers. The application must monitor the actions of the trainees during the test and at the end of the test, should come up with an evaluation of how well the trainee has performed.

1.1 Purpose of the document

The purpose of this document is to specify the requirements and preview some elements of the CTVI (Computer Training for Visually Impaired) Automation Tool (Here on referred to as CTVIAT or AT or the Automation Tool).

2. Product Description and Features

2.1 Current Process

The CTVI program consists of training visually impaired trainees with computer skills such as word processing, spreadsheets, document management etc. The trainees interact with the computer through Braille-based tools that act as replacements for the screen, the mouse and the keyboard.

During the program, each trainee undergoes a series of exercises that test his/her expertise on each module. Each test consists of a set of questions followed by a series of tasks that the trainee must perform on the computer. For example, a task might involve opening a Word Document, creating a title for the document, aligning the title to the center of the document, and then saving the document to a specified directory.

As of today, the trainers spend a considerable amount of time and energy correcting these exercises manually. The trainers correct each exercise by physically sitting at each computer, go through the list of tasks and decide whether the trainee has performed the task correctly. The accuracy of the corrections can also be affected due to the level of manual intervention involved.

2.2 Intent of the tool

The goal of the tool is to automate the process of correction of the CTVI exercises and make the process more efficient with respect to both time and the accuracy. The automation tool should allow the creation of the question bank of exercises and let the admin choose the questions from the pool for candidates and design specific question papers. The application must monitor the actions of the trainees during the test and at the end of the test, and it should come up with an evaluation of how well the trainee has performed.

3. Design

3.1 Data Model & Relationships

- Question Bank - A question bank is the set of all questions under each topic / module. The idea of creating a question bank is to build a repository of questions so that question papers can be created by choosing relevant questions. There is no limit for the number of questions that can be in a question bank.

The questions are broadly classified into 3 types

- Objective type
- Descriptive
- Action-based.

Objective type questions are multiple choice questions where the candidate is given 4-5 answer options from which he can choose the right answers. The answers for the objective type questions will be checked by the program after the completion of the test.

Descriptive questions have descriptive answers which cannot be compared / checked by the program for its accuracy. The answers provided by the user are compared with the standard answer (from the answer file) by the student herself after completion of the test.

Action-based questions require the candidate to perform a specified task on his computer while the test is being administered. The tasks could be anything ranging from creation of a folder, finding the contents of a folder, word-navigation tasks or editing of word documents.

- Question Paper - The list of questions picked up from the Question Bank under each topic (usually each question paper consists of around 10-15 questions picked by the admin / test creator). This will contain the questions as well as the correct option choice for these questions.
- Hints database - This will contain the hints required for action based questions (Action-based questions are tasks to be done by the candidate during the tests, e.g., creation of a folder in a specified location, renaming the folder etc).
- Analysis / Correct Answer list. This is the repository of the correct answers to the Question Bank. The input from this is also used to create the question paper.

3.2 Usage Scenario and User Profiles

The tool is designed to be used by two types of actors :

- Admin
- Candidate

The access privileges of the two groups are strictly demarcated and defined.

Admin : The admin has the privilege to create the questions for the question-bank, select the questions for the create and edit the questions that were previously set .

Candidate : The user who is administered with the test is termed as the candidate. The candidate logs into the program with his individual logins and takes the test.

3.3 Requirements based on Functional Flow and Use Cases

UC01: Use Case 1 – My computer exercise (action-based)

- The program asks the user to create a folder in a specified location in his computer.
- The user should be able to 'alt-tab' and use the windows explorer.
- The user creates the folder in the specified location and clicks the 'DONE' button in the program screen.
- The program checks the presence of the folder in the specified location.
- If the folder was successfully created, the program announces this and goes to the next question.
- If the folder was not created successfully, the program alerts the user and gives step-wise hints to help him complete the task.

UC02: Use Case 2 – Word Navigation exercise (action-based + objective type)

- The program gives the instruction for the user to open a specified MS word file from a specified location.
- The candidate opens the file and clicks the "DONE" button on the program screen.
- The program instructs the user to perform a certain key stroke (e.g., INSERT +T) and gives an objective-type question based on it (e.g., what does JAWS speak on performing the specified key stroke).
- The candidate performs the action and answers the question. If the candidate's choice corresponds to the correct answer, the program moves to the next action. Otherwise, the program asks the candidate to try again. (The program does not move to the next question since this is an action-based question and requires the candidate to get it right to move on to the next action)

UC03: Use Case 3 – Word Formatting exercise (action-based + objective type)

- The program instructs the candidate to go to a particular location and correct the formatting errors in the given MS Word document. (The instructions could give detailed scenarios on what the document is and how many errors the document has)
- The candidate opens the document and corrects / makes the formatting changes. After completion of the tasks, the candidate saves the file and provides the program with the location of the corrected document.
- The program compares the candidate's corrected document with that of the 'correctly formatted document' and gives a report on what are the formatting tasks the user was not able to do.
- If the tasks were done correctly, the program gives alternate ways of doing the same formatting tasks. The program also captures and tells the time taken by the candidate to perform the task.

3.4 Requirements

3.4.1 General Requirements

REQ01: The exercise topics should have the ability to be separated into modules of instructions (e.g., editing, dialog, windows explorer, jaws help, word, excel etc)

REQ02: The tool should also support the ability to add mp3 files as instructions/answer options.

- REQ03:* The tool should have the ability to create specific question types (Objective / Descriptive and Action-based)
- REQ04:* The time taken by the candidate to complete the exercises should be captured by the tool.
- REQ05:* The marks scored should be captured in a format that can be sent / evaluated at a later stage. The program should interpret the responses and the results of the tests.
- REQ06:* In case the user desires to complete the exercise in installments, the tool should support the option for the user to continue from the point where he left.
- REQ07:* In case the user desires, he/she should be able to repeat an exercise. The tool should record and report all exercises that have been repeated.

3.4.2 Accessibility Specific Requirements

- REQ08:* Students should be able to re-read the instructions/questions word by word or line by line or letter by letter. If the user cannot understand the word/ sentence, he/she can use their arrow keys.
- REQ09:* The candidate should be able to choose multiple choice answers by the use of Numeric keys (instead of having to deal with radio buttons or combo boxes). The intended user may not have the required expertise to work with these advanced windows controls .

3.4.3 Other User Specific Requirements

- REQ10:* The instructions/questions and the answer choices should be editable by the ADMIN at any stage of the exercise/question bank preparation.
- REQ11:* The user (candidate) should have the ability to choose the desired topic/module that he wants to be quizzed on (e.g., editing, dialog, windows explorer, jaws help, word, excel etc)
- REQ12:* The admin should be able to generate individual reports based on each individual user's performance.

3.4.4 System Specific Requirements

- REQ13:* The instructions/questions should be stored in Unicode format such that it is compatible and supports other languages.

3.4.5 Security Requirements

- REQ14:* The candidates should have individual logins to the application so that they can review their results, check their answers vis-à-vis that of the correct ones.

3.5 Assumptions and dependencies

3.5.1 Data Storage

The input files for the questions are stored in text files. At this point of time, implementation of a database interface is not incorporated.

3.5.2 Operating Environment

The tool is designed to operate in standard Windows (XP, Vista or Windows 7) environment.

3.6 Mode of interaction between stakeholder and developing team

The expected time for the project is about 3-4 months. The developing team is encouraged to interact as often as required with the stakeholder, especially during the early requirements collection and design stage. The developing team is free to contact the stakeholder by email (kapilv@microsoft.com) throughout the year. The team and the stakeholders can also arrange regular (bi-weekly) updates over Skype.