

J-SCANNER

AN OMR READER AND QUIZ MANAGEMENT SYSTEM

A PROJECT REPORT
BY

TEAM NO. 01

ABHIRAJ GHOSE (E23CSE0014)
PALLAV SHARMA (E23CSE0022)



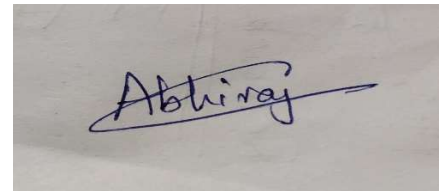
SUBMITTED TO
SCHOOL OF COMPUTER SCIENCE ENGINEERING AND
TECHNOLOGY, BENNETT UNIVERSITY
GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

18 April 2024

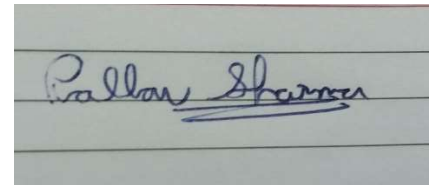
DECLARATION

We hereby declare that the work which is being presented in the report entitled “J-Scanner”, is an authentic record of our own work carried out under the guidance of Mr. Chandra Shekhar Ram during the period from January 2024 to April 2024 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by us for the award of any other degree elsewhere.

A photograph of a handwritten signature in blue ink on a light-colored surface. The signature is 'Abhiraj' with a long horizontal stroke extending to the right.

Abhiraj Ghose
(Enroll. No. E22CSE0014)

A photograph of a handwritten signature in blue ink on lined paper. The signature is 'Pallav Sharma' with the last name underlined.

Pallav Sharma
(Enroll. No. E22CSE0022)

TABLE OF CONTENTS

- 1. INTRODUCTION
 - 1.1 PROBLEM DESCRIPTION
 - 1.2 BACKGROUND
 - 1.3 OBJECTIVES
- 2. PROJECT OUTLINE
 - 2.1 TOOLS AND TECHNOLOGIES USED
 - 2.2 PROGRAMMING LANGUAGE USED
 - 2.3 LIBRARIES AND IMPORTS USED
- 3. PROJECT DESIGN
 - 3.1 DATA FLOW
 - 3.2 ALGORITHMS USED
 - 3.3 DESIGN FLOWCHART
- 4. ONLINE RESOURCES AND POSTS

1. INTRODUCTION

In the realm of modern education, the integration of technology has become indispensable, revolutionizing traditional teaching and learning methodologies. The advent of digital tools has facilitated the automation of various educational processes, enhancing efficiency and accessibility. In this context, the development of J-Scanner emerges as a significant contribution to the field, offering a comprehensive solution for document digitization and text processing in educational settings.

J-Scanner is a Java-based Optical Character Recognition (OCR) project. It aims to streamline document digitization and text processing tasks using Tesseract OCR technology.

This project caters to both teachers and students, providing essential functionalities for assessment management. For teachers, features include test creation, conclusion, and result viewing, simplifying evaluation processes. Students can log in, attempt tests, and review results conveniently, enhancing their learning experience.

Utilizing Java programming language, IntelliJ IDEA IDE, and Tesseract OCR library, J-Scanner ensures accurate and efficient text extraction from scanned documents and images. With its intuitive interface and robust functionality, J-Scanner serves as a valuable tool for educational institutions and individual users alike, facilitating streamlined assessment processes and fostering academic excellence.

1.1. Problem Description

The traditional methods of document digitization and assessment management are often cumbersome and inefficient, leading to delays in evaluation and potential inaccuracies in grading. Moreover, the lack of standardized tools for OCR and test management exacerbates these challenges, hindering the adoption of digital solutions in educational institutions. Addressing these

issues, J-Scanner seeks to provide a reliable and user-friendly system for digitizing documents and managing assessments, thereby enhancing the overall efficiency and effectiveness of educational processes.

1.2. Background

Traditional assessment methods often involve manual evaluation of written tests, which can be time-consuming and prone to errors. Additionally, the proliferation of digital content necessitates efficient means of converting physical documents into digital formats for archival and analysis purposes. Recognizing these challenges, the J-Scanner project leverages modern technology to streamline these processes, providing educators and students with a user-friendly platform for creating, administering, and assessing tests.

1.3. Objectives

The primary objective of the J-Scanner project is to develop a robust and intuitive platform for document digitization and assessment management. Specific objectives include:

- Implementing Optical Character Recognition (OCR) technology to accurately extract text from scanned documents and images.
- Providing teachers with tools for creating, administering, and concluding tests, simplifying assessment management processes.
- Offering students a user-friendly interface for accessing and attempting tests, as well as reviewing their results.
- Ensuring compatibility with standard file formats and integrating seamlessly with existing educational systems.
- Enhancing the overall efficiency and effectiveness of educational processes through automation and digitization.

By achieving these objectives, J-Scanner aims to revolutionize document digitization and assessment management in educational institutions, fostering a more streamlined and efficient learning environment.

2. PROJECT OUTLINE

The J-Scanner project was conceived as a solution to streamline the process of conducting tests and assessments in educational institutions, particularly in scenarios where physical presence is limited or impractical. Motivated by the increasing demand for remote learning and assessment tools, as well as the need for efficient test management systems, the project aimed to provide a user-friendly, automated platform for creating, administering, and evaluating tests.

The structure of the J-Scanner project reflects a comprehensive approach to test management, encompassing multiple layers of functionality and integration with external tools and libraries. At its core, the project features a layered architecture comprising presentation, business logic, data access, external tools, utility, and external integration layers. This modular design ensures scalability, flexibility, and maintainability, allowing for seamless integration of additional features and enhancements.

The primary purpose of the J-Scanner project is to address the challenges associated with traditional test administration methods, such as manual grading, scheduling conflicts, and resource constraints. By leveraging technology to automate various aspects of the testing process, the project aims to improve efficiency, accuracy, and accessibility for both educators and students. The vision for the project is to create a platform that empowers educators to design and deliver assessments effectively, while providing students with a convenient and reliable means of demonstrating their knowledge and skills.

In the context of educational institutions, the implementation of J-Scanner can lead to significant improvements in test management practices, allowing educators to focus more on instructional activities and student engagement. By reducing the administrative burden associated with test creation, administration, and grading, educators can devote more time and resources to

personalized instruction, feedback, and academic support. Additionally, the adoption of J-Scanner aligns with broader trends in digital learning and assessment, contributing to the organization's commitment to innovation and educational excellence.

2.1. Tools and Technologies Used

- **Java:** A versatile and widely-used programming language known for its platform independence and robustness.
- **IntelliJ IDEA:** An integrated development environment (IDE) specifically designed for Java development, offering advanced features and tools for code editing, debugging, and version control.
- **Tesseract OCR:** An open-source OCR engine capable of recognizing text within images and converting it into editable text.
- **Git:** A distributed version control system used for tracking changes in source code during software development.
- **GitHub:** A web-based platform for hosting and collaborating on Git repositories, facilitating team collaboration and version control management.
- **LucidChart:** A web-based platform for creating flowcharts, mind maps, and other visual diagrams to facilitate project planning and design.
- **Google Docs:** A cloud-based word processor that allows collaborative editing and sharing of documents in real-time, enhancing team productivity and communication.
- **Google Slides:** A web-based presentation tool for creating, editing, and sharing slide presentations, offering collaborative features and seamless integration with Google Drive.
- **YouTube:** A popular video-sharing platform where users can upload, view, and share videos, making it an ideal platform for showcasing project demonstrations and tutorials.
- **ClipChamp:** An online video editing tool that enables users to create, edit, and customize videos with ease, offering a range of features for enhancing and refining video content.

2.2. Programming Language Used

Java: Utilized as the primary programming language for developing the J-Scanner project due to its platform independence, object-oriented approach, and extensive libraries.

2.3. Libraries and Imports Used

- `java.io`: Provides classes for reading and writing files, streams, and other input/output operations in Java.
- `java.io.file`: Offers support for file and file system operations, including file reading, writing, and manipulation.
- `java.util.stream`: Introduces functional-style operations to Java collections, allowing for concise and expressive manipulation of data streams.
- `java.lang.Process`: Represents a subprocess, allowing for the execution of external processes and interaction with their input/output streams.
- `java.lang.Runtime`: Provides access to the Java runtime environment, enabling the execution of system commands and processes.
- `java.util.Scanner`: Enables parsing of primitive types and strings from various input streams, simplifying user input and file reading operations.
- `java.io.FileWriter`: Allows for writing character-based data to a file in Java, supporting the creation and appending of text files.
- `java.io.BufferedReader`: Provides efficient reading of characters from input streams, offering buffering and line-oriented reading capabilities.
- `java.io.InputStreamReader`: Converts bytes from an input stream into characters, facilitating the reading of text from byte-oriented streams.
- `java.io.FileReader`: Reads text from character files, providing convenient methods for reading character-based data.
- `java.util.Map`: Represents a collection of key-value pairs, offering efficient retrieval, insertion, and deletion operations.
- `java.util.HashMap`: Implements the Map interface using a hash table, providing fast access to elements based on their keys.
- `java.util.Arrays`: Provides utility methods for manipulating arrays, including sorting, searching, and filling operations.
- `java.util.List`: Represents an ordered collection of elements, supporting dynamic resizing and efficient element access.

- `java.util.ArrayList`: Implements the `List` interface using a dynamic array, offering fast element insertion and removal operations.
- `java.lang.ProcessBuilder`: Facilitates the creation and management of processes in Java, providing methods for building and executing system commands and scripts.

3. PROJECT DESIGN

3.1. Data Flow

Inputs:

- Users select login or signup.
- For login:
 - Users choose to log in as a teacher or a student.
- Provide username and password.
- For signup:
- Users select whether they are a teacher or a student.
- Enter personal details such as name, username, password, and email.
- Students provide their enrollment number and a teacher's 5-letter code.

Processes:

- Authentication:
 - Validate username and password against stored credentials for login.
 - Check if the provided teacher's code exists for student signup.
- User Selection:
 - Based on user input, direct users to appropriate functionalities.
- Test Creation:
 - Teachers create new tests, inputting test names and answer keys.
 - Answer keys are processed using Tesseract OCR to extract answer patterns.
- Test Attempt:
 - Students select ongoing tests to attempt.
 - Answer patterns are processed using Tesseract OCR to record student responses.
- Test Result Processing:
 - Conclude tests, updating status and calculating student scores.

- View test results, displaying overall scores and individual responses.

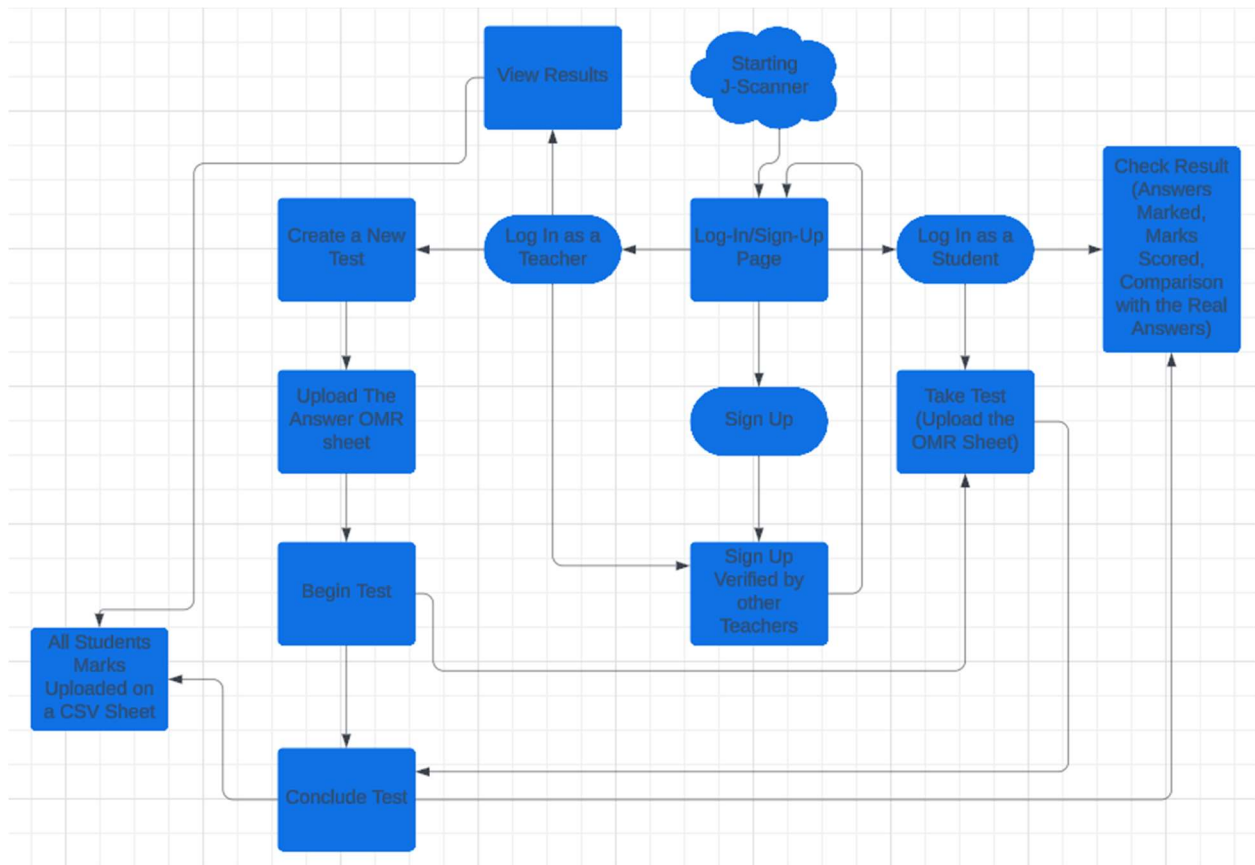
Outputs:

- Login/Signup Confirmation:
 - Confirmation message upon successful login/signup.
- Test Selection:
 - List of available tests for teachers and students to choose from.
- Test Results:
 - Display of overall scores and individual responses for concluded tests.
- Data Saving:
 - User Credentials: Teacher and student credentials stored in separate CSV files.
- Test Data:
 - Test details, answer keys, and student responses saved in CSV files.

3.2. Algorithms Used

- **Authentication Algorithm:**
 - Algorithm to authenticate users during login/signup by verifying their credentials against stored data.
- **Tesseract OCR Processing Algorithm:**
 - Algorithm to process answer keys and student responses using Tesseract OCR for optical character recognition.
- **Test Scoring Algorithm:**
 - Algorithm to calculate scores for student responses based on the comparison with correct answers in the answer key.
- **Test Conclusion Algorithm:**
 - Algorithm to conclude tests by updating their status and calculating final scores for students.
- **Data Parsing Algorithm:**
 - Algorithm to parse CSV files containing test data, user credentials, and other relevant information.
- **User Input Handling Algorithm:**
 - Algorithm to handle user inputs during test selection, login/signup, and other interactions with the system.
- **File I/O Algorithm:**
 - Algorithm to read from and write to CSV files for storing and retrieving test data, user credentials, and other data entities.
- **Error Handling Algorithm:**
 - Algorithm to handle errors and exceptions that may occur during the execution of various functionalities within the system.

3.1. Design Flowchart



4. ONLINE RESOURCES AND POSTS

- Github: <https://github.com/AetherSparks/J-Scanner>
- Youtube: <https://youtu.be/CAqFo5dy2EI?si=vu8mRlT21u5IfBaF>
- LinkedIn: https://www.linkedin.com/posts/abhiraj-ghose-b41b63282_j-scanner-java-project-by-abhiraj-ghose-activity-7185501095811510274-u03C?utm_source=share&utm_medium=member_desktop