**COLLEGE OF BUSINESS EDUCATION**



**DODOMA CAMPUS**

**UNVEILING HIDDEN TALENTS IN TANZANIA**

**NAME OF STUDENT: YOHANA ADREHEM LIKILIWIKE**

**REGISTRATION NO: 03.5347.01.02.2023**

**COURSE: BIT**

**SUBJECT NAME : PROGRAMMING IN JAVA**

**LECTURE NAME: MADAM ATUPELE CAIRO MWAITETE**

**NATURE OF WORK: INDIVIDUAL ASSIGNMENT**

**SUBMISSION DATE: 28TH JANUARY 2025**

**QUESTION**

1. A short report (not more than 10 pages) describing the features implemented, screenshots of the project interface and the challenges faced during the development.

**INTRODUCTION**

The Talent Showcase App is a Java Swing-based desktop application that allows users to share their talents by submitting their information. The primary purpose of this application is to create a platform for individuals to showcase their hidden skills and creativity while exploring and searching for other talent submissions.

This report highlights the features implemented, provides an overview of the user interface through screenshots, and discusses the challenges encountered during the development process.

Features Implemented

1. User Input and Form Validation

Users can input their name, talent, talent story, bio, and an optional video link.

Talent categories and levels can be selected from predefined dropdown menus.

Validation ensures that required fields (name, talent, story, and bio) are not left empty.

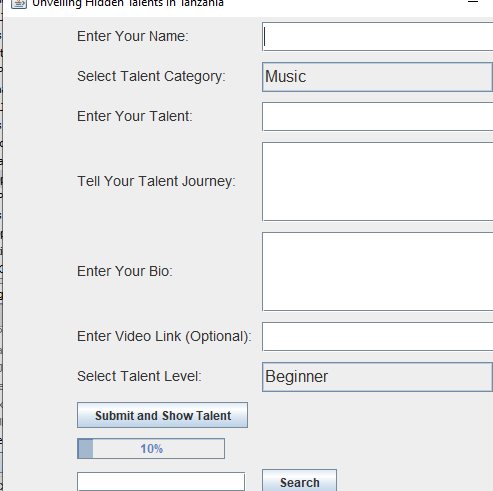


Fig: User Input Validation.

2. Talent Submission

Talent entries are submitted using the "Submit and Show Talent" button.

Submissions are saved locally to a file (talents.txt) to ensure persistence even after the application is closed.

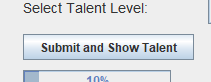


Fig:Talent Submit Button

3. Search Functionality

A search bar allows users to find specific talent entries by name or talent.

Results are displayed in the main display area with a message if no matches are found.



Fig: Search button

4. Talent Display

A scrollable text area is used to display all submitted talent entries.

The entries include name, talent, category, level, story, bio, and optional video link.

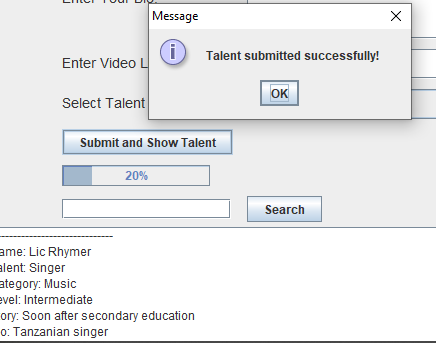


Fig: Talent show display.

5. Random Inspirational Quote

A motivational quote is displayed at the bottom of the interface, updating randomly when the application starts.



Fig: Random inspirational quote.

6. Progress Bar

A progress bar reflects the number of talents submitted, encouraging users to contribute more entries.

7. Data Persistence

Talent entries are saved to a file using ObjectOutputStream and loaded on application start with ObjectInputStream.

USER INTERFACE

1. Main Interface

The main interface consists of the following components:

A form for entering talent details.

A scrollable area for displaying submitted talent entries.

A search bar to find specific entries.

A progress bar and a motivational quote section.

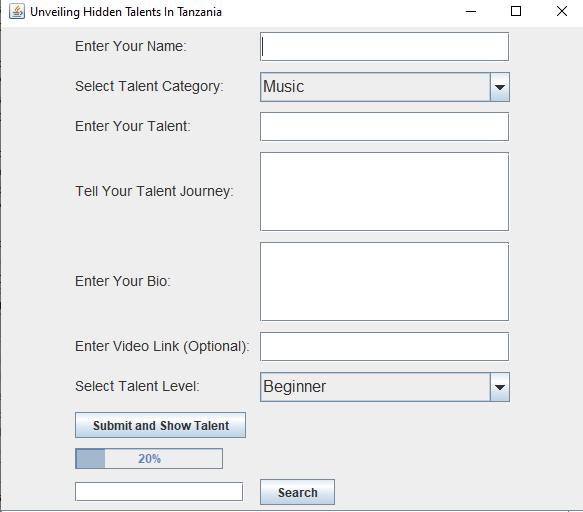


Fig: A full user Requirements sumbittion form.

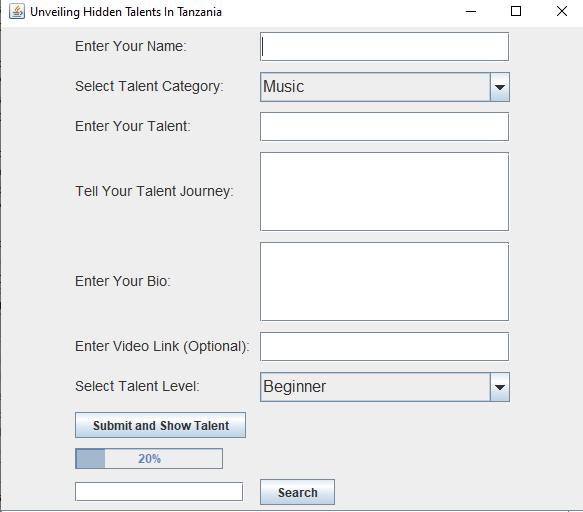
2. Form Fields

The form includes:

Text fields for Name, Talent, and Video Link.

Combo boxes for selecting Talent Category and Level.

Text areas for Talent Story and Bio.



3. Search Functionality

A search bar allows users to query submitted talents.

Results matching the search are displayed in the main text area.



Challenges Faced During Talent Show Case Development;

GUI Design Complexity:

Aligning components using GridBagLayout required careful management of constraints.

Designing a user-friendly interface while maintaining functionality was time-intensive.

Data Persistence:

Serialization and deserialization of complex objects (like TalentEntry) required proper exception handling.

Ensuring file operations didn’t lead to data corruption was a key challenge.

Search Efficiency:

Implementing an intuitive and responsive search mechanism involved optimizing string comparisons for user queries.

Error Handling:

Ensuring that the application didn’t crash when encountering empty fields or corrupted data files was crucial.

Progress Bar Updates:

Dynamically updating the progress bar based on the number of talents required calculations tied to the submission logic.

Conclusion

The Talent Showcase App successfully fulfills its objective of providing a platform for users to share and discover talents. Despite some challenges during development, the application is robust, user-friendly, and ensures data persistence.

Future improvements could include:

A richer multimedia integration for uploading images or videos directly.

Improved search capabilities using advanced filtering options.

A database backend for better scalability.

References

Java Swing Documentation

The official documentation for Java Swing provides an in-depth understanding of components like JFrame, JPanel, JTextField, JTextArea, JComboBox, JScrollPane, and JButton.

GridBagLayout Guide

The layout manager GridBagLayout is used in the code for flexible component placement. This guide explains its constraints and usage.

Object Serialization in Java

The code uses ObjectOutputStream and ObjectInputStream for saving and loading talent entries. This tutorial provides a comprehensive explanation of Java serialization.

Event Handling in Swing

The code makes extensive use of ActionListener for button actions. This tutorial explains the event-driven programming model in Swing.

JProgressBar Component

The JProgressBar component is used to show progress based on the number of talent submissions. This guide explains how to use it.

Java ComboBox Guide

The use of JComboBox for selecting talent categories and levels is a key feature of the application.

Exception Handling in Java

The application handles exceptions related to file I/O and serialization. This resource provides a detailed explanation of Java’s exception handling mechanisms.