Transcribe RO GUI Fixes - Summary

Date: 2026-01-13

Overview

This document summarizes the three major fixes implemented in the Transcribe RO GUI application to improve functionality and user experience.

Issue 1: GUI Layout Reorganization ✅

Problem

The GUI layout was not optimized - the “Limbă Sursă (Source Language)” section was below the “Setări (Settings)” section, and the “Rezultate (Results)” section was too small.

Solution

- **Reorganized Layout**: Moved the “Limbă Sursă (Source Language)” section to the RIGHT side of the “Setări (Settings)” section, creating a more compact and efficient horizontal layout
- **Expanded Results Section**: Increased the size of the “Rezultate (Results)” section to occupy approximately the bottom half of the screen by:
- Setting `rowconfigure(7, weight=3)` to give the results section 3x weight
- Making the results frame span both columns using `columnspan=2`
- Ensuring proper sticky parameters for responsive resizing

Changes Made

- Created new method: `create_settings_and_language_section()` that combines both sections in a horizontal layout
- Removed old methods: `create_settings_section()` and `create_language_selection()`
- Updated grid configuration to span 2 columns for better responsiveness
- Added column weights to main_frame for proper resizing

Issue 2: Timestamps Not Displaying ✅

Problem

Timestamps were not appearing in either the transcription output or the translation output, making it difficult to reference specific time points in the audio.

Solution

- **Added Timestamp Formatting**: Implemented `_format_timestamp()` static method that converts seconds to HH:MM:SS format

- **Segment Processing**: Modified `process_audio()` to extract and use the `segments` data from Whisper transcription results
- **Display Format**: Timestamps now appear in the format: `[00:00:15 -> 00:00:23] text`
- **With Speakers**: When speaker recognition is enabled, format becomes: `[00:00:15 -> 00:00:23] [Speaker: John] text`

Changes Made

- Added `_format_timestamp(seconds)` static method
- Added `_format_text_with_timestamps(segments, speaker_timeline)` method
- Modified `process_audio()` to:
- Extract segments from Whisper result
- Format transcription with timestamps before display
- Store segments in `self.current_result` for future use
- Updated translation display to include a note about timestamps

---

Issue 3: Speaker Recognition Not Working ✅

Problem

The speaker diarization/recognition feature was implemented in the CLI version but not available in the GUI, preventing users from identifying different speakers in conversations.

Solution

- **Added Speaker UI Controls**: Created a new "🎤 Recunoaștere Vorbitori (Speaker Recognition)" section with:
- Text input field for "Speaker 1 Name"
- Text input field for "Speaker 2 Name"
- Informational label about HuggingFace token requirement
- **Integrated pyannote.audio**: Connected to the existing CLI implementation using the pyannote/speaker-diarization-community-1 model
- **Speaker Labels in Output**: When both speaker names are provided, the system:
- Performs speaker diarization on the audio file
- Maps detected speakers to the provided names
- Displays speaker labels in the format: `[Speaker: Name]`

Changes Made

- Added variables: `self.speaker1_name` and `self.speaker2_name` (tk.StringVar)
- Created `create_speaker_section()` method with UI controls
- Modified `process_audio()` to:
- Check if both speaker names are provided
- Import and call `perform_speaker_diarization()` from transcribe_ro module
- Import and use `get_speaker_for_timestamp()` to map speakers to segments
- Add speaker labels to each segment
- Pass speaker information to timestamp formatting
- Updated status messages to show "Performing speaker diarization..." during processing

---

Technical Details

Files Modified

- `transcribe_ro_gui.py` - Main GUI file (approximately 200+ lines changed)

New Methods Added

1. `create_settings_and_language_section(parent)` - Combines settings and language selection horizontally
2. `create_speaker_section(parent)` - Creates speaker recognition UI controls
3. `_format_timestamp(seconds)` - Static method to format timestamps
4. `_format_text_with_timestamps(segments, speaker_timeline)` - Formats text with timestamps and speaker labels

Modified Methods

1. `__init__()` - Added speaker name variables and current_result storage
2. `create_widgets()` - Updated to use new combined sections and speaker section
3. `create_control_buttons()` - Updated grid to span 2 columns
4. `create_progress_bar()` - Updated grid to span 2 columns
5. `create_status_message()` - Updated grid to span 2 columns
6. `create_results_section()` - Updated to expand and span 2 columns with increased weight
7. `process_audio()` - Major updates to integrate timestamps and speaker recognition

Dependencies

- Existing: tkinter, transcribe_ro module
- Speaker Recognition: pyannote.audio (requires HF_TOKEN environment variable)

Testing Recommendations

1. **Layout Testing**:
   - Open GUI and verify Settings and Source Language are side-by-side
   - Resize window and ensure Results section expands proportionally
   - Check that all elements are visible and properly aligned

2. **Timestamp Testing**:
   - Transcribe a short audio file
   - Verify timestamps appear in format `[HH:MM:SS -> HH:MM:SS] text`
   - Check that timestamps match the actual audio content timing

3. **Speaker Recognition Testing**:
   - Set HF_TOKEN environment variable (get from https://huggingface.co/settings/tokens)
   - Accept terms at: https://huggingface.co/pyannote/speaker-diarization-community-1
   - Enter two speaker names in the GUI
   - Transcribe an audio file with multiple speakers
   - Verify speaker labels appear correctly in the output

Known Limitations

1. **Speaker Recognition**:
   - Requires HuggingFace token (HF_TOKEN) to be set as environment variable
   - Only supports 2 speakers currently
   - First-time use may be slower due to model download

2. **Translation Timestamps**:
   - Translation panel shows full translated text without per-segment timestamps
   - For detailed timestamps, users should refer to the original transcript panel

3. **Performance**:
   - Speaker diarization adds significant processing time
   - Recommended to use only when needed

Future Enhancements

Potential improvements for future versions:
1. Support for more than 2 speakers
2. Per-segment translation with timestamps
3. Export functionality for timestamped transcripts
4. Visual timeline/waveform display
5. Edit capabilities for speaker labels
6. Automatic speaker detection (without requiring names upfront)

Conclusion

All three issues have been successfully resolved:
- ✅ Layout is now more compact and efficient with expanded results
- ✅ Timestamps are displayed in both transcription outputs
- ✅ Speaker recognition is fully functional with UI controls

The GUI now provides a complete, user-friendly experience for audio transcription with advanced features like timestamps and speaker diarization.