



# GUI Testing Guide

---

This guide helps you test the Transcribe RO GUI application to ensure it works correctly.

## Pre-Testing Checklist

---

Before testing, ensure:

- [ ] Python 3.8+ is installed: `python3 --version`
- [ ] Virtual environment is activated (if using one)
- [ ] All dependencies are installed: `pip install -r requirements.txt`
- [ ] FFmpeg is installed: `ffmpeg -version`
- [ ] tkinter is available: `python3 -m tkinter` (should open a test window)
- [ ] You have at least one audio file for testing (MP3, WAV, M4A, etc.)

## Test Scenarios

---

### Test 1: Launch the GUI

**Purpose:** Verify the GUI launches without errors

**Steps:**

1. Run the launcher script:
  - Linux/macOS: `./run_gui.sh`
  - Windows: `run_gui.bat`
  - Or: `python3 transcribe_ro_gui.py`

1. Wait for the GUI window to appear

**Expected Result:**

- Window opens with title “Transcribe RO - Audio Transcription & Translation”
- Window size is approximately 1200x800 pixels
- All UI elements are visible:
  - Header with title and subtitle
  - File selection section
  - Settings section (Model Size, Device)
  - Language selection section with radio buttons
  - Control buttons (Start Transcription, Stop)
  - Progress bar (not animating)
  - Status message (gray text: “Ready. Please select an audio file to begin.”)
  - Two empty text panels (Original Transcript, Romanian Translation)
  - Copy and Save buttons for each panel

**Status:**  Pass /  Fail

---

### Test 2: File Selection

**Purpose:** Verify file picker works correctly

**Steps:**

1. Click the “Browse...” button
2. File picker dialog should open
3. Navigate to an audio file
4. Select the file and click “Open”

**Expected Result:**

- File picker opens with audio format filters
- Selected file path appears in the text field
- Status message updates: “Selected: [filename]” in blue text
- Clear button is functional

**Status:**  Pass /  Fail

---

## Test 3: Settings Configuration

**Purpose:** Verify settings controls work

**Steps:**

1. Click on the Model Size dropdown
2. Select different options (tiny, base, small, medium, large)
3. Click on the Device dropdown
4. Select different options (auto, cpu, mps, cuda)

**Expected Result:**

- Both dropdowns are functional
- Selected values are displayed
- No errors occur

**Status:**  Pass /  Fail

---

## Test 4: Language Selection

**Purpose:** Verify language radio buttons work

**Steps:**

1. Scroll through the language selection area
2. Click on different language radio buttons
3. Verify Romanian is selected by default

**Expected Result:**

- Radio buttons are selectable
- Only one language can be selected at a time
- Scrolling works if there are many languages
- Romanian (Română) is pre-selected

**Status:**  Pass /  Fail

---

## Test 5: Transcription Process (Short Audio)

**Purpose:** Test basic transcription functionality

**Test Audio:** Use a short (30-60 second) audio file in English

**Steps:**

1. Select an audio file using Browse button
2. Set Model Size to “base”
3. Set Device to “auto”
4. Keep source language as default (Romanian) - will auto-detect
5. Click “ Start Transcription”

**Expected Result:**

- Process button becomes disabled
- Stop button becomes enabled
- Progress bar starts animating
- Status updates:
  - “Loading Whisper model...” (orange)
  - “Transcribing audio...” (orange)
  - “Detected language: en. Translating to Romanian...” (orange)
  - “✓ Transcription and translation complete! Detected language: en” (green)
- Original transcript appears in left panel (English)
- Romanian translation appears in right panel
- Success dialog appears
- Buttons return to normal state

**Status:** Pass / Fail

---

## Test 6: Romanian Audio (No Translation)

**Purpose:** Test special case where audio is already in Romanian

**Test Audio:** Use a Romanian audio file (or select Romanian as source language)

**Steps:**

1. Select a Romanian audio file
2. Set source language to “Romanian (Română)”
3. Click “ Start Transcription”

**Expected Result:**

- Transcription completes
- Left panel shows Romanian transcript
- Right panel shows message: “✓ Source audio is already in Romanian. No translation needed...”
- Status: “✓ Transcription complete! Detected language: Romanian (no translation needed)”

**Status:** Pass / Fail

---

## Test 7: Copy Functionality

**Purpose:** Test copy to clipboard feature

**Steps:**

1. After a successful transcription (with results in both panels)
2. Click “ Copy” button under the Original Transcript panel
3. Paste in a text editor (Ctrl+V / Cmd+V)
4. Click “ Copy” button under the Romanian Translation panel
5. Paste in a text editor

**Expected Result:**

- First copy pastes the original transcript
- Success message: “Text copied to clipboard!”
- Second copy pastes the Romanian translation
- No errors occur

**Status:**  Pass /  Fail

---

## Test 8: Save Functionality

**Purpose:** Test save to file feature

**Steps:**

1. After a successful transcription
2. Click “ Save” button under Original Transcript
3. Choose a location and filename
4. Click Save
5. Click “ Save” button under Romanian Translation
6. Choose a location and filename
7. Click Save

**Expected Result:**

- Save dialog opens with suggested filename
- Files are created successfully
- Success message: “Text saved to: [path]”
- Files contain the correct content (can be verified by opening in text editor)

**Status:**  Pass /  Fail

---

## Test 9: Window Resizing

**Purpose:** Test responsive layout

**Steps:**

1. Drag window edges to make it smaller
2. Drag window edges to make it larger
3. Observe how UI elements adjust

**Expected Result:**

- Window resizes smoothly
- Text panels adjust proportionally
- No UI elements are cut off or overlap
- Minimum size is enforced (window doesn't get too small)

**Status:**  Pass /  Fail

---

## Test 10: Stop Functionality

**Purpose:** Test process cancellation

**Steps:**

1. Start a transcription
2. While processing (during “Loading model” or “Transcribing”), click “ Stop”
3. Confirm when prompted

**Expected Result:**

- Confirmation dialog appears
- Processing stops
- Status message: “Processing stopped by user.” (red)
- UI returns to ready state

**Status:**  Pass /  Fail

---

## Test 11: Error Handling - No File Selected

**Purpose:** Test validation

**Steps:**

1. Without selecting a file, click “ Start Transcription”

**Expected Result:**

- Error dialog appears: “Please select an audio file first.”
- No processing starts

**Status:**  Pass /  Fail

---

## Test 12: Error Handling - Invalid File

**Purpose:** Test file validation

**Steps:**

1. Select a non-audio file (e.g., a text file or image)
2. Click “ Start Transcription”

**Expected Result:**

- Either file picker filters prevent selection, or
- Error message during processing
- GUI remains functional

**Status:**  Pass /  Fail

---

## Test 13: GPU Acceleration (If Available)

**Purpose:** Test GPU usage

**Prerequisites:** Apple Silicon Mac or NVIDIA GPU

**Steps:**

1. Select an audio file
2. Set Device to:
  - "mps" (Apple Silicon)
  - "cuda" (NVIDIA)
  - Or "auto" (should auto-detect)
3. Start transcription

**Expected Result:**

- Status shows GPU device configuration
- Transcription completes faster than CPU
- No errors related to GPU

**Status:**  Pass /  Fail /  N/A (No GPU available)

---

## Test 14: Different Model Sizes

**Purpose:** Test all model options

**Steps:**

1. Test with "tiny" model
2. Test with "base" model
3. Test with "small" model
4. (Optional) Test with "medium" or "large" models

**Expected Result:**

- All models load successfully
- Larger models take longer but may be more accurate
- No crashes or errors

**Status:**  Pass /  Fail

---

## Test 15: Long Audio File

**Purpose:** Test with longer audio (5+ minutes)

**Steps:**

1. Select a longer audio file
2. Use "base" or "small" model
3. Start transcription
4. Monitor progress messages

**Expected Result:**

- Processing takes longer
- Status messages update regularly
- Translation may show "Translating chunk X..." messages

- Completes successfully
- No timeout or crash

**Status:** Pass / Fail / N/A (No long audio available)

---

## Common Issues and Solutions

### Issue: GUI doesn't launch

**Solution:**

```
# Check Python version
python3 --version # Should be 3.8+

# Check tkinter
python3 -m tkinter # Should open a test window

# Install tkinter if missing (Linux)
sudo apt-get install python3-tk

# Reinstall dependencies
pip install -r requirements.txt
```

### Issue: “Module not found” errors

**Solution:**

```
# Make sure you're in the right directory
cd /path/to/transcribe_ro

# Activate virtual environment
source venv/bin/activate # Linux/macOS
venv\Scripts\activate # Windows

# Reinstall dependencies
pip install -r requirements.txt
```

### Issue: Translation not working

**Solution:**

- Check internet connection (requires Google Translate API)
- Verify deep-translator is installed: `pip list | grep deep-translator`
- Reinstall: `pip install deep-translator`

### Issue: GPU not detected

**Solution:**

```
# Check PyTorch installation
python3 -c "import torch; print(torch.__version__)"

# Check GPU availability
python3 -c "import torch; print(torch.backends.mps.is_available())" # Apple Silicon
python3 -c "import torch; print(torch.cuda.is_available())" # NVIDIA
```

## Test Results Summary

---

Fill in after completing all tests:

- **Total Tests:** 15
- **Passed:** \_\_\_\_
- **Failed:** \_\_\_\_
- **N/A:** \_\_\_\_
- **Pass Rate:** \_\_\_\_%

## Notes

---

Use this space to record any additional observations, bugs, or suggestions:

[Your notes here]

## Reporting Issues

---

If you encounter bugs:

1. Note the test scenario that failed
  2. Record the exact error message
  3. Check system information:
    - OS and version
    - Python version
    - Available RAM
    - GPU type (if applicable)
  4. Check the console output for detailed errors
  5. Open an issue on GitHub with all the above information
- 

Happy Testing! 