Cursor Session Checklist - Interactive Paper Processor

Date: October 12, 2025

Goal: Implement interactive menu system (8 tasks, ~2 hours)

Reference: `CURSOR_TASKS.md` (detailed implementation guide)

Before You Start (5 minutes)

✅ Pre-Session Setup

☐ **Stop the daemon** if running

```bash
python scripts/stop_paper_processor.py
```

☐ **Check git status** - commit or stash current changes

```bash
git status
git add -A
git commit -m "checkpoint: before interactive menu implementation"
```

☐ **Open required files** in Cursor
☐ `scripts/paper_processor_daemon.py` (main work)
☐ `shared_tools/zotero/paper_processor.py` (Task 8)
☐ `shared_tools/zotero/local_search.py` (Task 8)
☐ `CURSOR_TASKS.md` (reference)
☐ **Terminal ready**

```bash
cd /mnt/f/prog/research-tools
conda activate research-tools
```

☐ **Test files ready** (for testing each task)
☐ At least 2-3 sample PDFs in a test directory
☐ Know which papers exist in your Zotero (for Task 7)

Task 1: Menu Display Functions (15 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

- [ ] Open `scripts/paper_processor_daemon.py`
- [ ] Find line 79 (after `setup_logging` method)
- [ ] Copy-paste `display_metadata` method from CURSOR_TASKS.md
- [ ] Copy-paste `display_interactive_menu` method
- [ ] Save file

Testing

- [ ] Add test code at end of file (from Task 1 in CURSOR_TASKS.md)
- [ ] Run test:

```bash
python scripts/paper_processor_daemon.py
```

- [ ] Verify menu displays correctly
- [ ] Verify input validation works
- [ ] Remove test code

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: add interactive menu display functions"
```

Time: _____ minutes | **Issues:** _____

Task 2: Menu Integration (15 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

- [ ] Replace `process_paper` method (lines ~140-180)
- [ ] Add `move_to_skipped` method
- [ ] Save file

Testing

☐ Start daemon:

```bash
python scripts/paper_processor_daemon.py
```

☐ Copy test PDF to scanner directory:

```bash
cp /path/to/test.pdf /mnt/i/FraScanner/papers/EN_test.pdf
```

☐ Verify:
☐ Metadata displays
☐ Menu appears
☐ Option '4' (skip) works → moves to skipped/
☐ Option '5' (manual) works → stays in place
☐ Option 'q' (quit) stops daemon
☐ Stop daemon: Ctrl+C

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: integrate interactive menu into processing loop"
```

Time: _____ minutes | **Issues:** _____

Task 3: Local Zotero Search (20 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

☐ Add `search_and_display_local_zotero` method
☐ Update `process_paper` - replace else block for choice '3'
☐ Save file

Testing

☐ Start daemon
☐ Copy test PDF (preferably one that exists in Zotero)

- [ ] Choose option '3'
- [ ] Verify:
- [ ] Searches local Zotero DB
- [ ] Displays matches with all fields
- [ ] Shows sub-menu
- [ ] 'b' (back) works
- [ ] Stop daemon

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: add local Zotero search to interactive menu"
```

Time: _____ minutes | **Issues:** _____

Task 4: Failed Extraction Workflow (20 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

- [ ] Add `handle_failed_extraction` method
- [ ] Add `manual_metadata_entry` method
- [ ] Add `convert_zotero_item_to_metadata` method
- [ ] Update `process_paper` - fix metadata handling (around line 155)
- [ ] Save file

Testing

- [ ] Start daemon
- [ ] Use poor quality scan OR image file (to trigger failure)
- [ ] Verify guided workflow:
- [ ] Document type selection
- [ ] DOI/ISBN prompt
- [ ] Author search in Zotero
- [ ] Match selection (1-10 or 0 for none)
- [ ] Manual entry if no match
- [ ] Stop daemon

Commit

```bash
bash
```

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: add failed extraction guided workflow"
```

Time: _____ minutes | **Issues:** _____

Task 5: Use Metadata As-Is (20 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

- [ ] Add `use_metadata_as_is` method
- [ ] Add `get_file_info` method
- [ ] Update `process_paper` - replace placeholder for choice '1'
- [ ] Save file

Testing

- [ ] Start daemon
- [ ] Copy test PDF with good metadata
- [ ] Choose option '1'
- [ ] Verify:
- [ ] Filename proposed correctly
- [ ] Can confirm or customize filename
- [ ] Duplicate check works (if file exists)
- [ ] Copies to G:/publications/
- [ ] Prompts for Zotero
- [ ] Moves original to done/
- [ ] Check: File in publications/, original in done/
- [ ] Stop daemon

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: implement use metadata as-is workflow"
```

Time: _____ minutes | **Issues:** _____

Task 6: Metadata Editing (15 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

☐ Add `edit_metadata_interactively` method
☐ Update `process_paper` - replace placeholder for choice '2'
☐ Save file

Testing

☐ Start daemon
☐ Copy test PDF
☐ Choose option '2'
☐ Edit various fields (title, authors, year, etc.)
☐ Verify:
☐ All fields editable
☐ Enter keeps current value
☐ Changes show in summary
☐ Can proceed or cancel
☐ Proceeds to standard workflow
☐ Stop daemon

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: implement metadata editing workflow"
```

Time: _____ minutes | **Issues:** _____

Task 7: Zotero Match Selection (20 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

☐ Add `attach_to_existing_zotero_item` method
☐ Update `process_paper` - complete Zotero search handling (choice '3')
☐ Save file

Testing

- [ ] Start daemon
- [ ] Copy PDF that exists in Zotero
- [ ] Choose option '3'
- [ ] Select a match (1-5)
- [ ] Verify:
- [ ] Detects if item has PDF
- [ ] Offers keep/replace/cancel options
- [ ] Generates filename correctly
- [ ] Copies to publications/
- [ ] Attaches to Zotero item
- [ ] Moves original to done/
- [ ] Check Zotero: PDF attached correctly
- [ ] Stop daemon

Commit

```bash
git add scripts/paper_processor_daemon.py
git commit -m "feature: implement Zotero match selection and PDF attachment"
```

Time: _____ minutes | **Issues:** _____

Task 8: Missing API Methods (10 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Implementation

- [ ] Open `shared_tools/zotero/paper_processor.py`
- [ ] Add `attach_pdf_to_existing` method (after line 230)
- [ ] Save file
- [ ] Open `shared_tools/zotero/local_search.py`
- [ ] Add `search_by_author` method
- [ ] Save file

Testing

- [ ] Test attachment method:

```python
```

```python
from shared_tools.zotero.paper_processor import ZoteroPaperProcessor
from pathlib import Path

processor = ZoteroPaperProcessor()
# Use a real item key from your Zotero
test_key = "ABCD1234"
test_pdf = Path("/mnt/g/My Drive/publications/test.pdf")
result = processor.attach_pdf_to_existing(test_key, test_pdf)
print(f"Result: {result}")
```

☐ Test author search:

```python
from shared_tools.zotero.local_search import ZoteroLocalSearch

searcher = ZoteroLocalSearch()
results = searcher.search_by_author("Smith")
print(f"Found {len(results)} papers")
```

Commit

```bash
git add shared_tools/zotero/paper_processor.py
git add shared_tools/zotero/local_search.py
git commit -m "feature: add attach_pdf_to_existing and search_by_author methods"
```

Time: _____ minutes | **Issues:** _____

Final Integration Testing (20 min)

Status: 🟪 Not Started | ⏳ In Progress | ✅ Done

Test Scenarios

Test 1: Successful Extraction

☐ Scan paper with DOI
☐ Metadata extracts correctly
☐ Choose "use as-is"
☐ File copied to publications/
☐ Added to Zotero successfully
☐ Original in done/

Test 2: Failed Extraction

- [ ] Use poor quality scan
- [ ] Guided workflow starts
- [ ] Document type selection works
- [ ] Can complete manual entry
- [ ] File processed correctly

Test 3: Existing Zotero Item

- [ ] Scan paper already in Zotero
- [ ] Option 3 finds match
- [ ] Can attach PDF
- [ ] Handles existing PDF correctly

Test 4: Edit Metadata

- [ ] Choose option 2
- [ ] Edit fields successfully
- [ ] Processing completes

Test 5: Skip/Cancel

- [ ] Skip document (option 4) → moves to skipped/
- [ ] Manual processing (option 5) → stays in place
- [ ] Quit (q) → daemon stops cleanly

Issues Found

_____

_____

_____

_____

Final Commits

```bash

```

```bash
# Update implementation plan
git add implementation-plan.md
git commit -m "docs: mark Phase 4.5 interactive menu as complete"

# Create session summary
git add SESSION_SUMMARY_$(date +%Y-%m-%d).md
git commit -m "docs: add session summary for interactive menu implementation"

# Tag this milestone
git tag -a v0.2.0-interactive-menu -m "Interactive menu system complete"
```

Post-Implementation

Update Documentation

☐ Update `implementation-plan.md` - mark Phase 4.5 complete
☐ Update `SCANNER_SETUP.md` - add interactive workflow description
☐ Create session summary (date, time spent, issues, solutions)

Cleanup

☐ Delete `paper-processor-spec.md` (redundant)
☐ Archive old planning documents if needed

Next Steps

☐ Configure Epson scanner buttons
☐ Process backlog of scanned papers
☐ Refine prompts based on usage
☐ Consider adding batch mode for backlog

Emergency Stop Procedure

If you need to stop mid-task:

1. **Commit current work** (even if incomplete):

```bash
git add -A
git commit -m "wip: task X in progress - stopping point"
```

2. **Note where you stopped:**

- Task number: _____

- What's working: _____

- What's not working: _____

- Next step: _____

3. **Safe resume:**
   - Review this checklist

   - Continue from noted task

   - Re-test previous completed tasks if needed

Time Tracking

Task	Estimated	Actual	Notes
Task 1	15 min	_____	_____
Task 2	15 min	_____	_____
Task 3	20 min	_____	_____
Task 4	20 min	_____	_____
Task 5	20 min	_____	_____
Task 6	15 min	_____	_____
Task 7	20 min	_____	_____
Task 8	10 min	_____	_____
Testing	20 min	_____	_____
Total	**~2.5 hrs**	_____	

Common Issues & Solutions

Issue: Import errors

Solution: Make sure conda environment activated:

```bash
conda activate research-tools
```

Issue: Daemon won't start

Solution: Check PID file and kill stale process:

```bash
cat /mnt/i/FraScanner/papers/.daemon.pid
ps aux | grep paper_processor
kill <PID>
rm /mnt/i/FraScanner/papers/.daemon.pid
```

Issue: Local Zotero search fails

Solution: Check database path in config:

```bash
cat config.personal.conf | grep zotero_db_path
ls -la <path_from_config>
```

Issue: File operations fail

Solution: Check permissions and paths:

```bash
ls -la /mnt/i/FraScanner/papers/
ls -la /mnt/g/My\ Drive/publications/
```

Ready to implement! 🚀

Pro Tips:

- ✅ Test after EVERY task
- ✅ Commit after EVERY task
- ✅ Don't rush - quality over speed
- ✅ Take breaks between tasks
- ✅ Stop if stuck - ask for help

Good luck! You've got detailed instructions for everything.