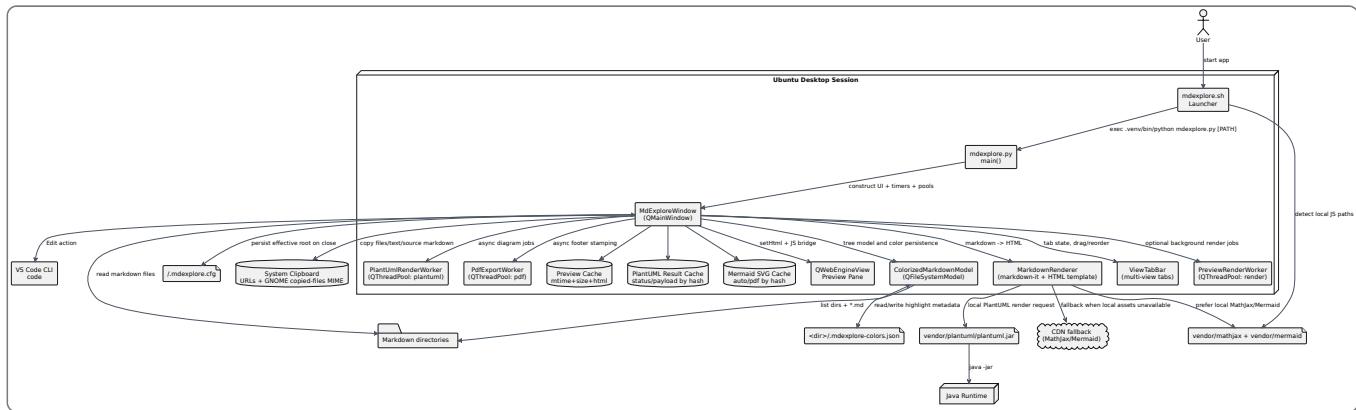


mdexplore UML

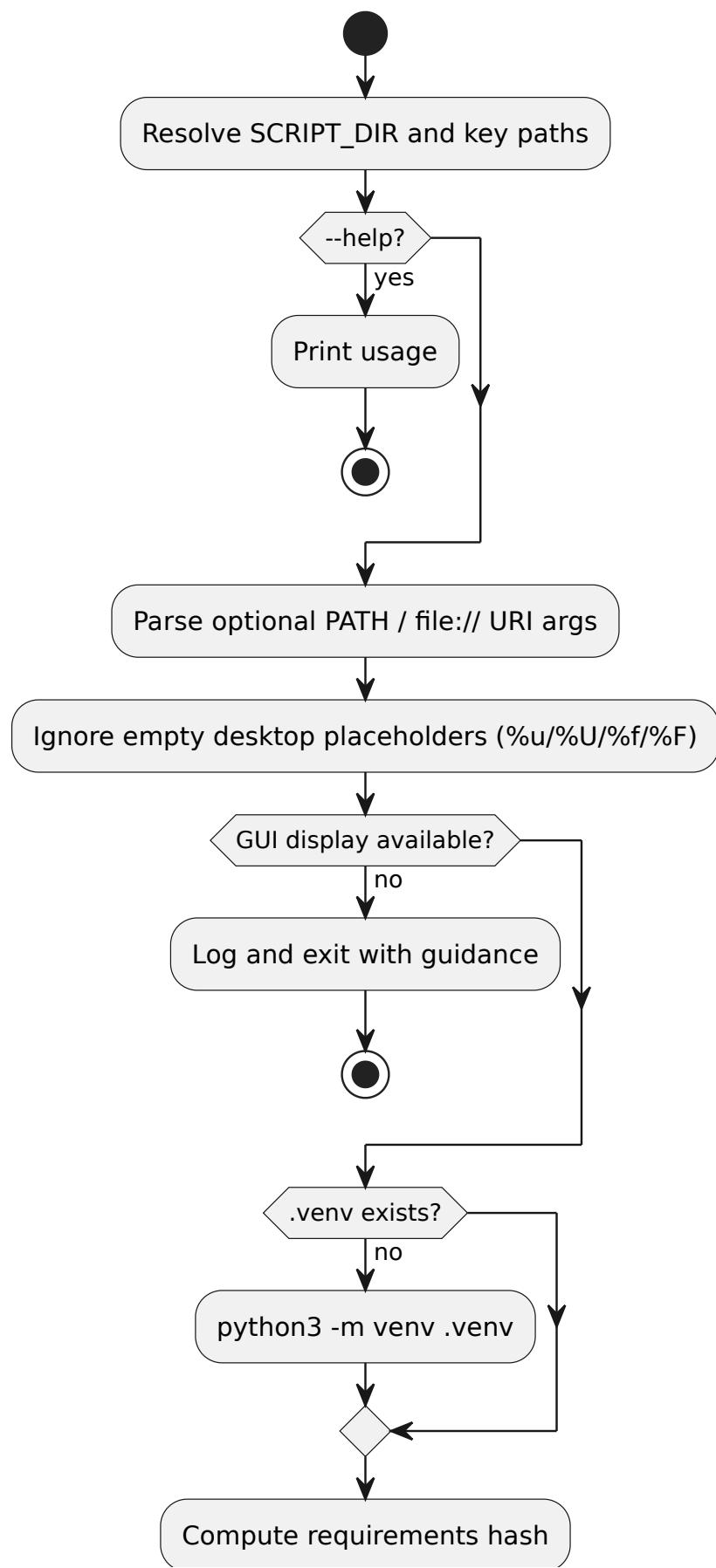
This document provides a comprehensive PlantUML view of the current `mdexplore` implementation. All diagrams are embedded so they can be rendered directly by `mdexplore` (or any Markdown viewer with PlantUML support).

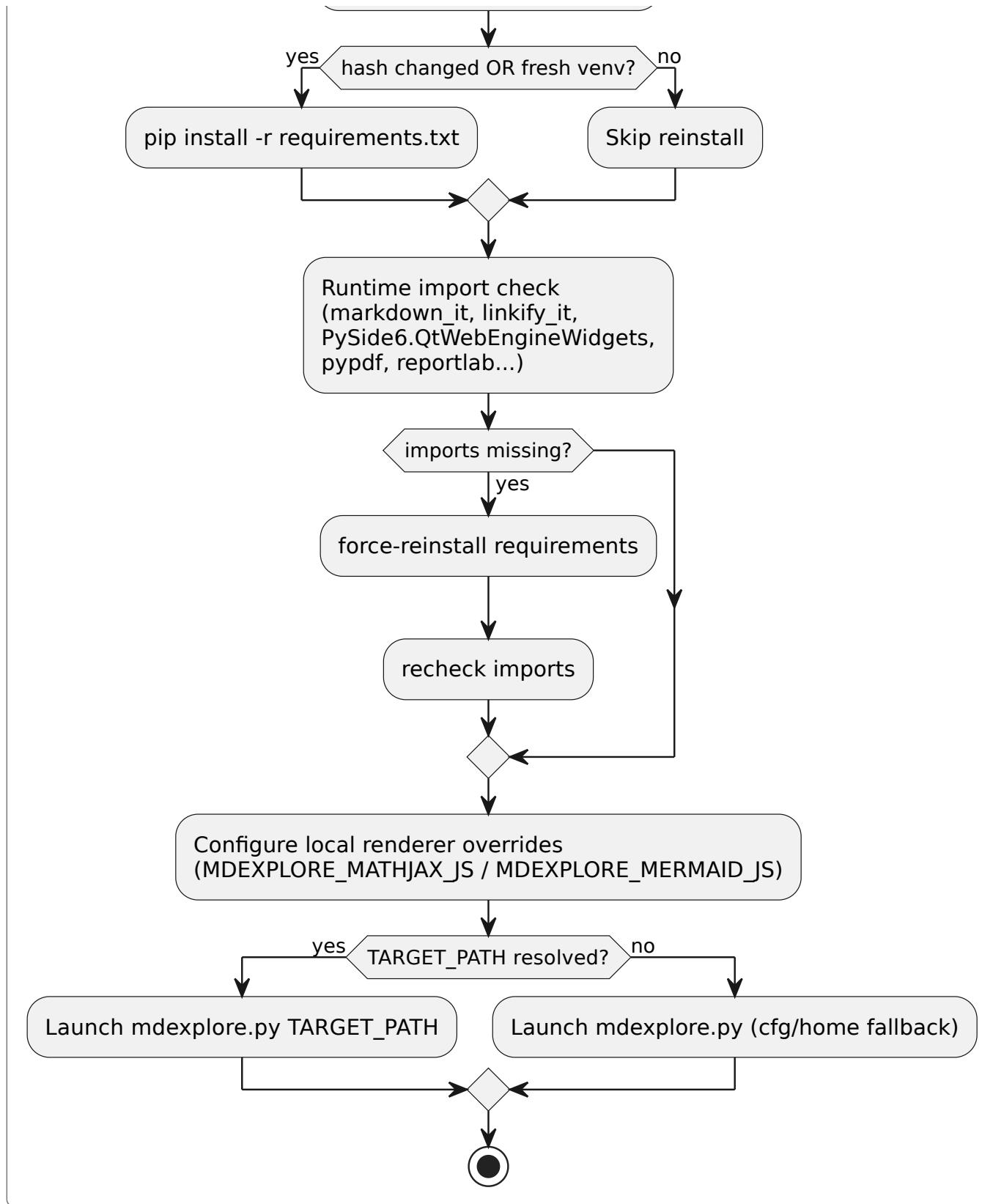
Detailed render/caching forks (GUI vs PDF, JS Mermaid vs Rust Mermaid, cache mode ownership, and restore behavior) are intentionally documented in `RENDER-PATHS.md`. This UML file keeps those areas abstracted at system/class boundaries to avoid duplicating low-level render logic across two docs.

1. System Architecture

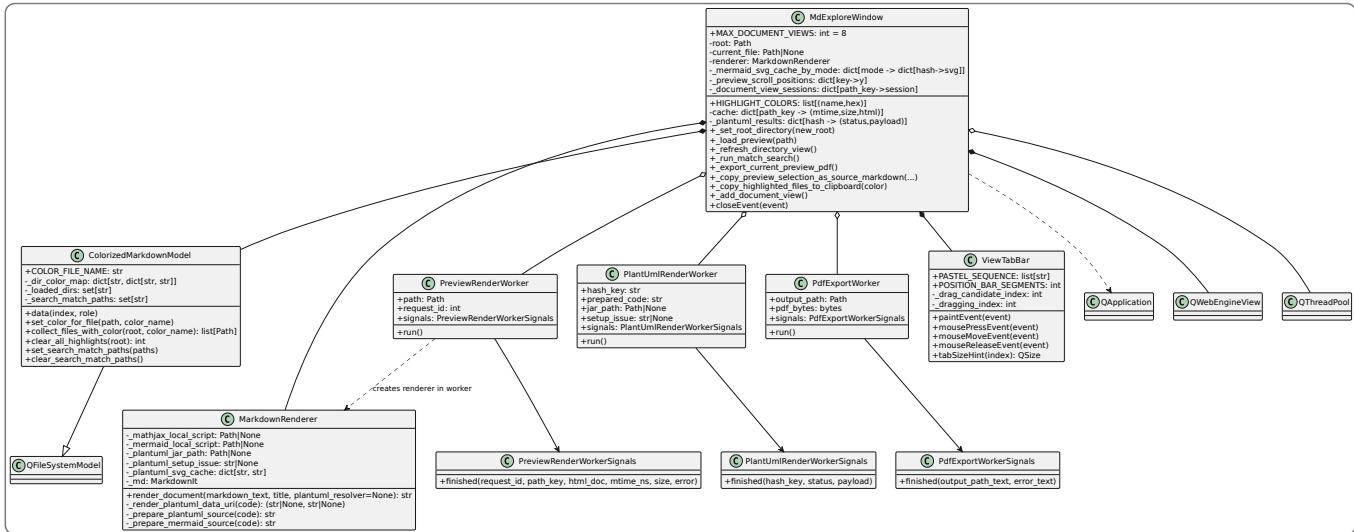


2. Launcher Runtime Activity (`mdexplore.sh`)



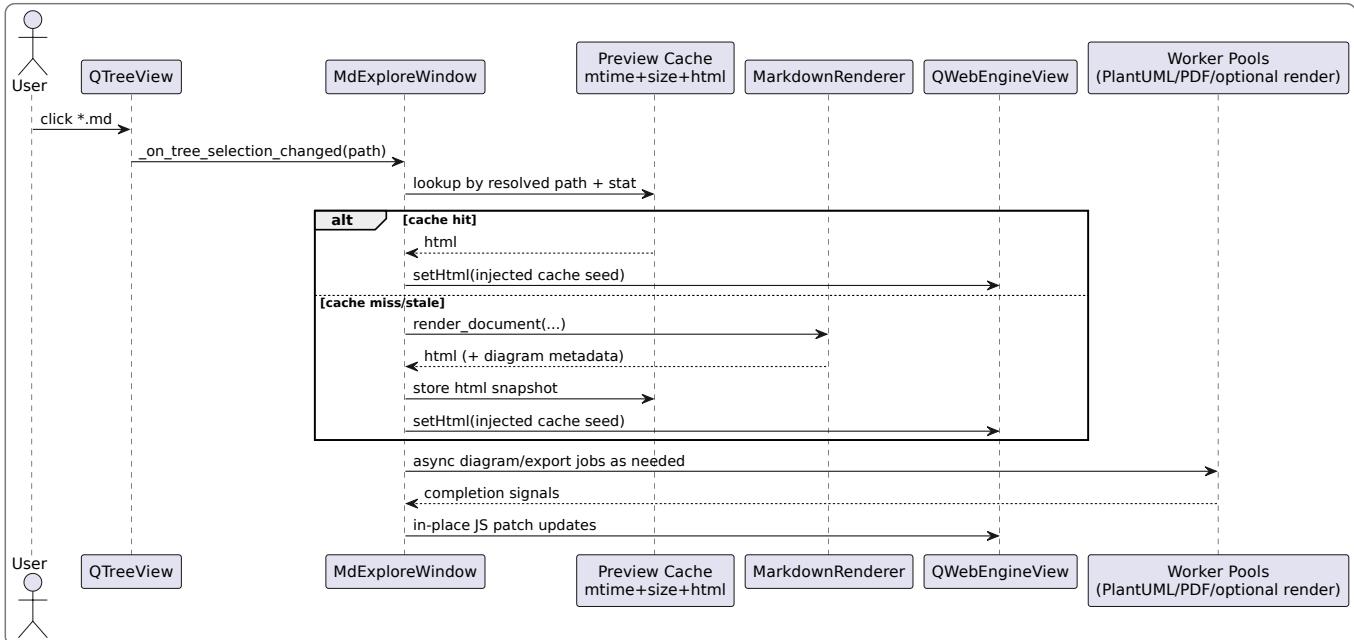


3. Core Class Diagram

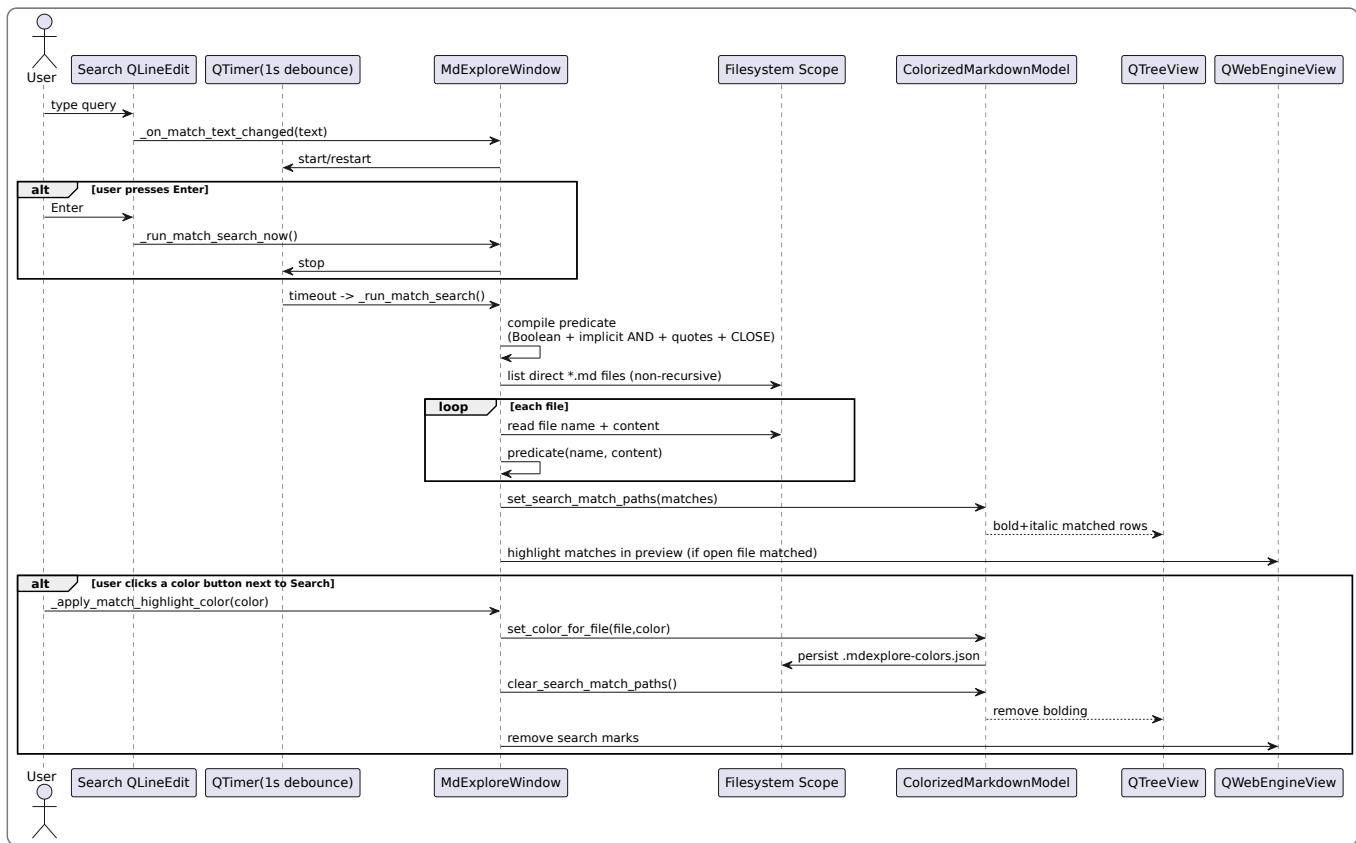


4. Preview Pipeline (Abstracted)

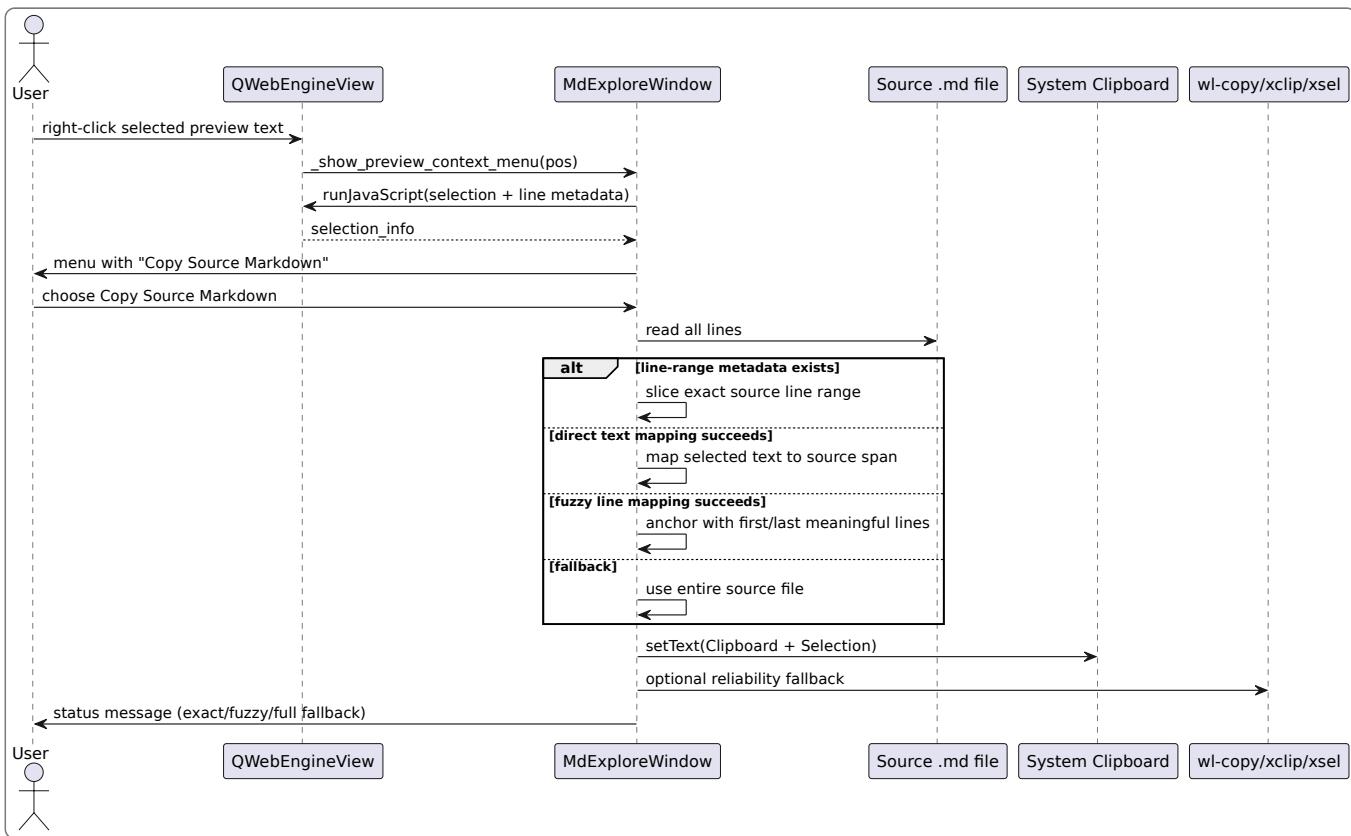
The detailed render/cache branches are covered in `RENDER-PATHS.md`. This section intentionally keeps an architectural boundary view only.



5. Search + Highlight + Apply Color Flow

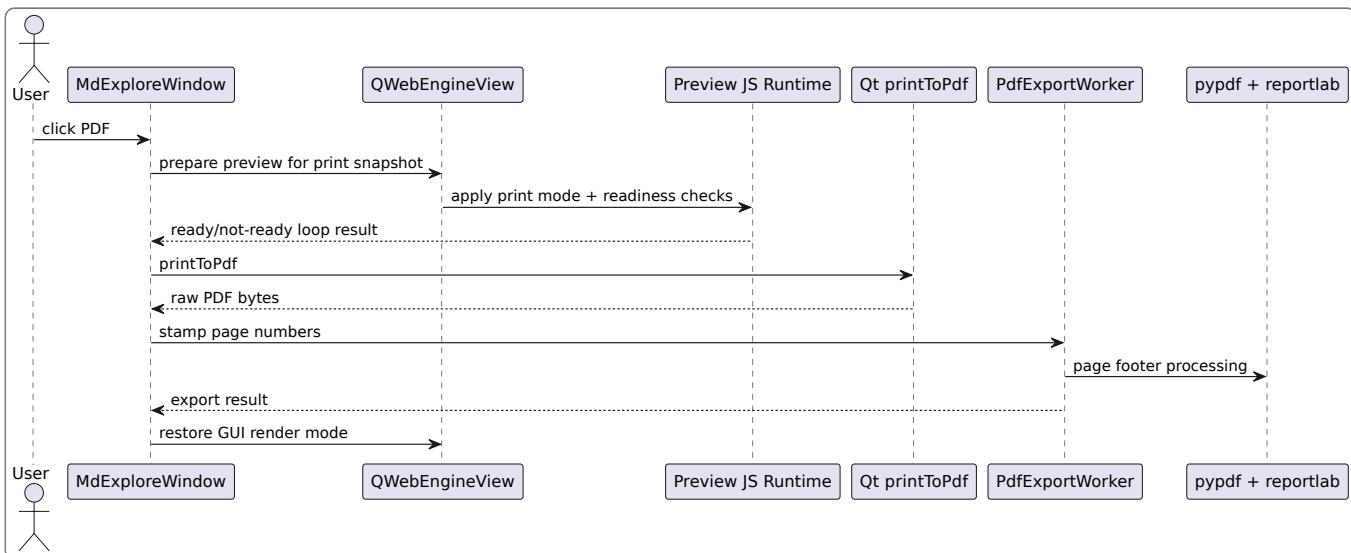


6. Preview Context Menu: Copy Source Markdown

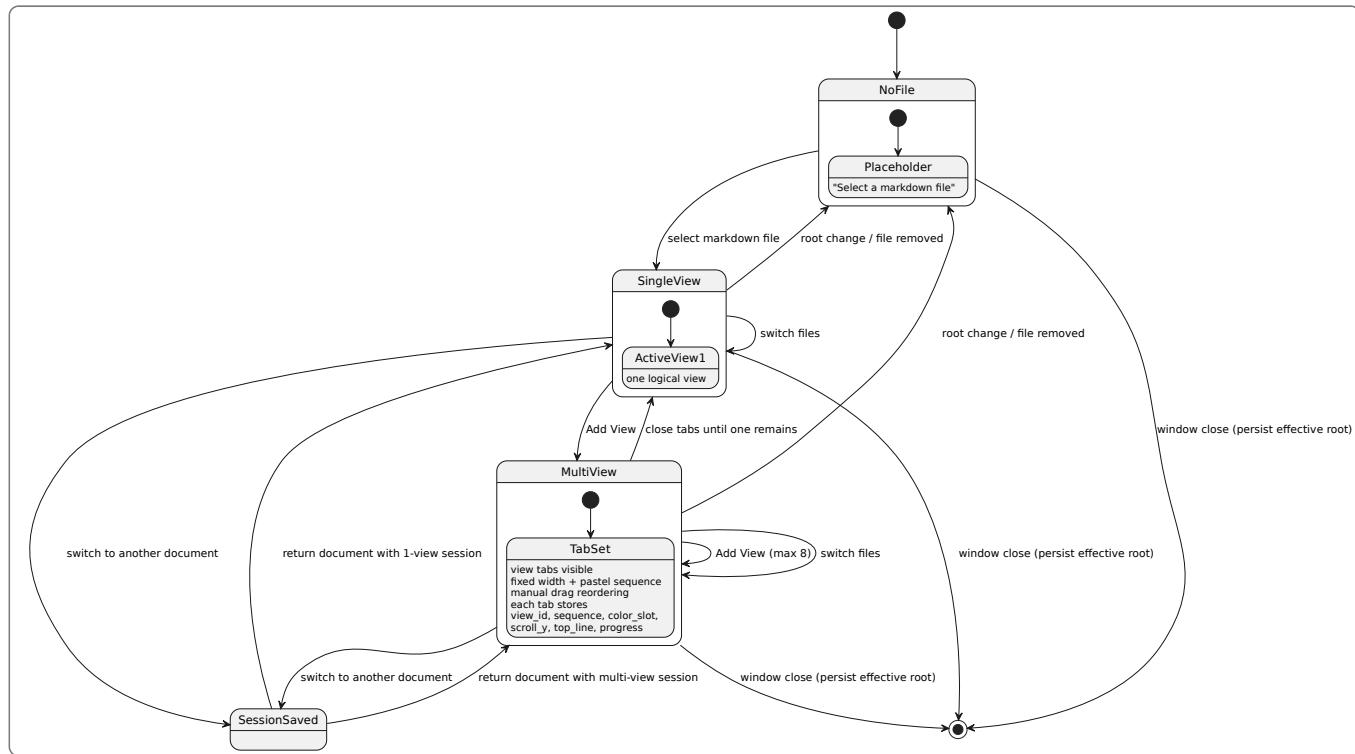


7. PDF Export Pipeline (Abstracted)

Detailed PDF mode branching (JS Mermaid grayscale path vs Rust Mermaid default-themed PDF cache path) is documented in [RENDER-PATHS.md](#).



8. Document View/Tab Lifecycle



Notes

- Diagrams are based on current code in `mdexplore.py` and `mdexplore.sh`.
- Render/caching branch internals are intentionally abstracted here and documented in `RENDER-PATHS.md` to keep a single authoritative deep map.
- Worker/threadpool usage is intentionally separated by concern:
 - render pool (preview HTML jobs),
 - PlantUML pool (diagram jobs),
 - PDF pool (post-processing/stamping).
- PlantUML rendering is non-blocking in UI flow: placeholders are rendered first, then patched in place as jobs finish.
- TODO (known issue): diagram zoom/pan restore (Mermaid and PlantUML) is not yet consistently reliable when leaving a document and returning in the same app run.