BRYCE PAGEL

CS 499 Milestone Three

Dr. Sherin Aly

8/3/2025

Milestone Three

1. **Briefly describe the artifact. What is it? When was it created?**

The artifact I enhanced is batchRename (now renamed as Barron Pagel File Utilities), a Python‐based utility originally developed around March 2025 to streamline the bulk renaming of discovery response files in my family-law paralegal firm.

In this update, I implemented the ability to create and manage client records and introduced a scanning tool that recursively parses a selected directory tree to identify PDF filenames that match the naming convention used by the bulk rename tool. These files – following formats such as x[ACCT] - [YYYY][MM][DD].pdf or x[ACCT] - [YYYY][MM].pdf – are automatically parsed for account number and statement date, validated against existing records, and added to the database if not previously logged. The user can also filter, view, and open any stored file directly from the application via a graphical interface.

1. **Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

This artifact demonstrates a full-stack approach to building maintainable, user-centric tools with persistent data. The database enhancement showcases my ability to design and normalize a relational schema, connect UI components to underlying database queries, validate input and prevent duplicate entries, and build systems that scale beyond single-session use.

It also illustrates thoughtful engineering choices around user configurability. By default, the SQLite database is stored in a hidden .bpfu directory under the user’s home folder – isolated from working directories for consistency and privacy – but the path is editable via a persistent config.json. The same applies to debug logs and backup locations, reflecting an intentional, user-aware design philosophy. Although the current file browsing interface is still in progress and lacks advanced filtering or export options, the enhancement meets the requirements by logging statement data, enabling basic record inspection, and enabling users to open source files directly within the app.

1. **Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

Yes, this enhancement meets the outcomes tied to databases, security, and tool design. I demonstrated the use of well-founded techniques by integrating SQLite with modular query logic, building validation layers for duplicate detection, and linking form inputs to structured database operations. I began to develop a stronger security mindset by ensuring that data entry is validated, stored safely, and never arbitrarily executed. The decision to use a local SQLite instance over a networked database inherently limits many attack vectors and keeps sensitive data within the user’s control.

This enhancement also expands on earlier design choices – such as the .bpfu structure and settings menu – by aligning persistence, organization, and user preferences with the evolving goals of the tool.

1. **Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

The hardest part of this enhancement was building a system that needed to interpret existing user data (in the form of file names) and maintain relational consistency while still remaining user-friendly. Extracting structured data from filename patterns required resilient parsing logic, especially when considering edge cases or historical inconsistencies in naming conventions.

Additionally, while SQLite itself is straightforward to use, connecting it to GUI controls in a maintainable way required careful architecture. I had to design flexible, reusable functions that could interface with the database without coupling logic to specific widgets. Balancing database access, recursive folder scanning, duplicate detection, and file path linking into one cohesive user flow forced me to think like both a backend and frontend developer.

I am still iterating on the visual display of statement records – currently, the viewing and filtering UI is functional but not polished. I plan to enhance this further by implementing export-to-Excel functionality, search refinements, and a more intuitive layout. Even so, the current state meets the milestone expectations, and I consider it a solid foundation for future improvement.