Testable Functional Requirements

1. Read Excel Files

- **Description:**
  The application shall enable the user to select and load exactly two Excel files using a file dialog interface.

2. Validate Excel File Formats

- **Description:**
  The application shall ensure each selected file has a `.xls` or `.xlsx` extension and that each filename does not exceed 100 characters before processing.

3. Combine Data

- **Description:**
  The application shall merge data from the two loaded Excel files by matching records where `Mother_First_Name`, `Mother_Last_Name`, and `Child_Date_of_Birth` are identical after normalization.

4. Display Combined Data

- **Description:**
  The application shall display the merged data in a list view, showing `Mother_IDs`, `Child_Names`, and `Date_of_Birth` for each matched record.

5. Check Minimum Files Requirement

- **Description:**
  The application shall disable the data combination feature until both Excel files are successfully loaded.

6. Normalize Name Fields

- **Description:**
  The application shall normalize `Mother_First_Name` and `Mother_Last_Name` by

removing all spaces, converting all letters to lowercase, and eliminating all non-alphanumeric characters before merging.

7. Error Handling

- **Description:**
  The application shall display specific error messages in message boxes for various error scenarios, including but not limited to:
  ### Unreadable Excel Files
  - **Condition:** When an Excel file fails to load.
  - **Message:**
    `"Failed to load combined data file: [error details]"`
- ### Invalid File Formats
  - **Condition:** When a user attempts to open a combined data file before combining.
  - **Message:**
    `"No combined data file found. Please combine data first."`
  - **Condition:** When a user tries to combine data without selecting two files.
  - **Message:**
    `"Please read two Excel files first."`
- ### Encryption Key Issues
  - **Condition:** When a user attempts to generate a new encryption key while one already exists.
  - **Message:**
    `"To generate a new key, delete the previous key."`
  - **Condition:** When a user tries to delete an encryption key that does not exist.
  - **Message:**
    `"No key exists."`
- ### Encryption/Decryption Issues
  - **Condition:** When encryption fails.
  - **Message:**
    `"Encryption Unsuccessful."`
  - **Condition:** When decryption fails due to an incorrect key.
  - **Message:**
    `"Error Decrypting File.\nEncryption Key Wrong."`
  - **Condition:** When attempting to decrypt without an existing key.
  - **Message:**
    `"Key does not exist."`
- ### File Operations
  - **Condition:** When attempting to open the combined data file and it does not exist.

    - **Message:**
      `"The combined data file does not exist."`
    - **Condition:** When attempting to open unmatched data in Excel and the file does not exist.
    - **Message:**
      `"The unmatched data file does not exist."`
    - **Condition:** When opening the combined data file in Excel fails.
    - **Message:**
      `"Error opening Excel file: [error details]"`
- **Profile Viewing Issues**
    - **Condition:** When no data is available to display in the combined data view.
    - **Message:**
      `"No data available to display."`
    - **Condition:** When no data is found for a selected child profile.
    - **Message:**
      `"No data found for [Child_First_Name] [Child_Last_Name]."`
    - **Condition:** When loading a child profile fails due to an unexpected error.
    - **Message:**
      `"Error loading profile: [error details]"`
- **General Warnings**
    - **Condition:** When the user attempts to perform actions that require certain prerequisites (e.g., combining data without loading files).
    - **Message:**
      Appropriate warning messages shall be displayed to guide the user.

8. Child Profile Display

- **Description:**
  The application shall open a new window displaying a detailed child profile, including all associated data from the combined records, when a user double-clicks on a child's name in the list view.

9. Save Combined Data

- **Description:**
  The application shall provide an option to save the merged data to a new Excel file named `combined_matched_data.xlsx` after the merge process is completed. The saved Excel file shall include the following columns:
    - **Mother_ID**
    - **Mother_First_Name**
    - **Mother_Last_Name**
    - **Child_First_Name**

    - **Child_Last_Name**
    - **Child_Date_of_Birth**
    - **Street**
    - **City**
    - **State**
    - **ZIP**
    - **Phone_#**
    - **Mobile_#**
    - **Assigned Nurse**

10. Display Unmatched Entries

- **Description:**
  The application shall save unmatched records to a separate Excel file named `unmatched_data.xlsx` after the data combination process is completed. The saved Excel file shall include the following columns:
    - **Source**
    - **Child_ID**
    - **Mother_First_Name**
    - **Mother_Last_Name**
    - *(Include any additional relevant columns present in the unmatched data)*
- Additionally, the application shall display the list of unmatched records in a designated area of the user interface, allowing users to view and interact with the unmatched data. Users shall have the ability to:
    - **View Unmatched Data:** Open the `unmatched_data.xlsx` file in Excel.
    - **Expand Details:** Double-click on a record to view additional information not displayed in the primary columns.

11. Encrypt Data Files

- **Description:**
  The application shall encrypt data files using the Advanced Encryption Standard (AES) with the following specifications:
    - **Encryption Scope:**
        - The application shall encrypt the Medicaid and Database data files upon saving.
    - **Encryption Key Management:**
        - The encryption key shall be stored in a file named `key.txt`.
        - The application shall provide functionalities to generate, delete, and manage the encryption key through dedicated buttons:
            - **Generate Encryption Key:**
              Allows users to create a new encryption key. If a key already

exists, the application shall prompt the user to delete the existing key before generating a new one.
        - **Delete Encryption Key:**
          Allows users to delete the existing encryption key after confirming the action. Deleting the key will render all previously encrypted files unusable.
    - **Encryption Process:**
        - Upon invoking the encryption feature, the application shall encrypt the selected data files using AES and save the encrypted versions.
    - **User Feedback:**
        - **Successful Encryption:**
          Upon successful encryption, the application shall display a message box with the message:
          `"Encryption Successful."`
        - **Failed Encryption:**
          If encryption fails for any reason, the application shall display a message box with the message:
          `"Encryption Unsuccessful."`
    - **Error Handling:**
        - If a user attempts to encrypt files without an existing encryption key, the application shall display an error message:
          `"Key does not exist."`
        - Any exceptions or errors during the encryption process shall be logged, and appropriate error messages shall be displayed to the user.

12. Decrypt Data Files

- **Description:**
  The application shall decrypt encrypted data files using the encryption key with the following specifications:
    - **Decryption Scope:**
        - The application shall decrypt the encrypted Medicaid and Database data files to make the decrypted data available for display within the application.
    - **Decryption Key Management:**
        - The application shall utilize the encryption key stored in `key.txt` to perform decryption.
    - **Decryption Process:**
        - Users can select encrypted files to decrypt through a file dialog interface.
        - The application shall verify the existence of `key.txt` before attempting decryption.
    - **User Feedback:**

        - **Successful Decryption:**
          Upon successful decryption, the application shall make the decrypted data available for display and notify the user with a message box: `"Decryption Successful."`
        - **Failed Decryption:**
          If decryption fails due to an incorrect key or corrupted file, the application shall display a message box with the message: `"Error Decrypting File.\nEncryption Key Wrong."`
        - **Missing Key:**
          If `key.txt` does not exist, the application shall display a warning message box: `"Key does not exist."`
    - **Error Handling:**
        - Any exceptions or errors during the decryption process shall be logged, and appropriate error messages shall be displayed to the user.
        - If a user attempts to decrypt a file without selecting a valid encrypted file, the application shall prompt the user to select a valid file.

13. Assign Nurses

- **Description:**
  The application shall provide functionalities to assign nurses to children through the user interface, including:
    - **Assign Nurse Functionality:**
      Users shall be able to assign a nurse to a child by entering the nurse's name in the child's profile window. This action shall update the `Assigned Nurse` field in the combined data.
    - **Update Data and Save:**
      Upon assigning a nurse, the application shall update the corresponding entry in the combined data DataFrame and save the changes to the `combined_matched_data.xlsx` file to ensure data persistence.
    - **View Nurse Statistics:**
      The application shall offer an option to view nurse assignment statistics, displaying information such as the most and least assigned nurses. This feature shall assist in managing nurse workloads and identifying staffing needs.
    - **User Feedback:**
        - **Successful Assignment:**
          After successfully assigning a nurse, the application shall display a message box with the message: `"Nurse '[Nurse_Name]' assigned successfully."`
        - **Failed Assignment:**
          If assigning a nurse fails due to any reason (e.g., data inconsistency), the

application shall display a message box with the message:
`"Failed to assign nurse."`

- **Error Handling:**
  - If a user attempts to assign a nurse without selecting a valid child profile, the application shall display an error message:
    `"No child profile selected for nurse assignment."`
  - Any exceptions or errors during the nurse assignment process shall be logged, and appropriate error messages shall be displayed to the user.
- **Interface Elements:**
  - **Assign Nurse Button:**
    Within the child profile window, there shall be an "Assign Nurse" button that, when clicked, opens a dialog for entering the nurse's name.
  - **Nurse Statistics Button:**
    The main combined data window shall include a "Nurse Statistics" button that opens a window displaying assignment statistics.
- **Data Integrity:**
  The application shall ensure that nurse assignments do not result in data duplication or inconsistency by validating inputs and maintaining unique associations between nurses and children.

Testable Non-Functional Requirements

1. Usability

- **Description:**
  The application shall provide a user interface that allows users to perform primary operations (reading Excel files, combining data, encrypting/decrypting files, assigning nurses) intuitively,which means all buttons will be labeled with their function they do. Ex: Read Excel File will read an excel file.

2. Performance

- **Description:**
  The application shall complete the data reading and merging processes within **5 seconds** when handling two Excel files, each with up to **1,000 rows** of data. Encryption and decryption operations shall complete within **10 seconds** for files of previously stated size.

3. Error Tolerance

- **Description:**
  The application shall remain operational without crashing when encountering missing

files, invalid file formats, encryption/decryption errors, or data mismatches. It shall notify the user of specific issues through descriptive error messages and logs that include the error name, ensuring that the application can recover from unexpected conditions without fully exiting the app.

4. Security

- **Description:**
  The application shall ensure the confidentiality and integrity of sensitive data by implementing the following security measures:
  - **Data Encryption:**
    All sensitive data files, including Medicaid and Database files, shall be encrypted using the Advanced Encryption Standard (AES) before saving.
  - **Key Management:**
    Encryption keys shall be securely stored in a file named `key.txt`. The application shall restrict access to this file to authorized users only.
  - **Access Control:**
    The application shall prevent unauthorized access to encrypted files by requiring the correct encryption key for decryption.
  - **Data Integrity:**
    The application shall verify the integrity of data during encryption and decryption processes, ensuring that data is not corrupted or tampered with.
  - **Logging:**
    All encryption and decryption activities, along with any security-related events, shall be logged for audit purposes.

5. Portability

- **Description:**
  The application shall be compatible with the following operating systems without requiring additional software installations beyond standard system updates:
  - **Windows:** Versions 10 and later.
  - **macOS:** Versions Catalina (10.15) and later.

6. Maintainability

- **Description:**
  The application's source code shall adhere to best practices for maintainability, including:
  - **Design Patterns:**
    Utilizing the Command Design Pattern, with each button action implemented as an extensible command class to facilitate the addition of new features without modifying existing code structures.

    - **Modular Architecture:**
      Organizing code into clear, modular components that include pre and post conditional comments
    - **Documentation:**
      Providing comprehensive inline documentation and external documentation to assist developers in understanding and modifying the codebase.
    - **Logging:**
      Implementing robust logging mechanisms to aid in debugging and monitoring application behavior.

7. Reliability

- **Description:**
  The application shall consistently perform its intended functions without failure. It shall handle exceptions gracefully, ensuring that users are informed of issues without disrupting the overall user experience.

8. Scalability

- **Description:**
  While the current requirements specify handling up to 1,000 rows of data, the application architecture shall be designed to accommodate future increases in data volume with less than 5% performance degradation.

9. Compliance

- **Description:**
  The application shall comply with HIPAA regulations and standards, ensuring that sensitive information is handled in accordance with legal and organizational policies.