**Test Plan for Nursing Patient Filtering Application**

**1. Introduction**

This test plan outlines the testing strategy for the **Nursing Patient Filter Application**, to help ensure that the functions are fixed or updated to work as expected by our client Utah County Health department.

**2. Test Objectives**

✅As a Nurse Manager, or supervisor, I want to program to collect and organize patient data by categories(name,DOB, mother name, mother id, zipcode, city, address, phone number) so that these patients can be filtered accurately.

✅As a Nurse Director, Manager, or supervisor, I want for the program to find patient assigned nurses by name so that patients visiting Nurse’s can be easily added and assigned to different patients

✅As a Nurse Director, Manager or supervisor I want to edit saved patient lists. So that patient history information can be edited and be updated and fix previous records.

✅As a Nurse Director, Manager or supervisor, I want the program to flag(based on name and DOB) with a date of when the duplicate patients are found So that I can maintain paper trail of services provided to patients. And can avoid duplicate assignments

✅As a Nurse Director, Manager or supervisor, I want the program to keep unmatched data in a separate file. So that I can assigned these new patients to Nurses.

✅As a Nurse Director, Manager, or supervisor, I want the program to let me print any of the combined, unmatched, and duplicated list, so that a copy of patients can be given to the visiting nurse.

✅As a Nurse Director, Manager or supervisor, I want to only give access to those in Management positions, so that HIPAA compliance can be maintained. This was changed to only one User the Director.

✅As a Nurse Director, Manager, or supervisor, I want the program to give timed data on when Nurses are assigned to patients, so that a visiting record maybe kept for all patients.

NEW Test Objectives for M4

* As a Nurse Director, Manager, or supervisor, I want to have access to Nurses visiting records on their patients. So that Nurse’s performance can be measured.
* As a Nurse Director, Manager, or supervisor, I want to be able to write notes on the profile records of patients. So that the record can have a more thorough explanation of its progress status etc.
* As a Nurse Director, Manager or supervisor, I want the program to assign patient records by zip code. So that Nurse assignments can be done more quickly.
* As a Nurse Director, Manager or supervisor, I want to the program to sort more than 5000 family names in less than 5 seconds. So that quantity fluctuations of patients can be still handled when the lists grow to be larger.
* As a Nurse Director, Manager, or supervisor, I want the program to search by zip codes. So that patients can be quickly searched within the given zip-code.
* As a Nurse Director, Manager, or supervisor, I want the program to provide access to the old combined list. So that old data can be used as comparison with new lists, in case old duplicates creep in.
* As a Nurse Director, Manager, or supervisor, I want to search by all the categories provided by the program. So that one can have the options to search by any category.
* As a Nurse Director, Manager, or supervisor, I want assigned Nurses to be automatically assigned to the patient. So that there is less confusion on which buttons do what in the application.
* As a Nurse Director, Manager, or supervisor, I want the program to have a history of nurse assigned to the current patients. So that I can have traceability of Nurse assignments to this patients.

**3. Scope**

**3.1 Features to be Tested**

**[ x ] Process lists with 5,000 patients**

**[ x ] Note log to add notes to patient profiles**

**[ x ] Track Nurses by dates assigned dates not times assigned**

**[ IN PROGRESS] Add more columns for data to be combined**

**[ x ] Search for patients by zip code**

**[x] Batch Assign patients to a Nurse.**

**[x] Search by each ca**

**3.2 Features NOT to be Tested**

Will not have an external database integration (assumed functional)

**4. Test Strategy**

**4.1 Testing Levels**

1. **Unit Testing**
   * Verify individual functions, such as filtering logic(By, Zipcode, city, etc)
   * Verify search by individual categories(name, city, phone number)
   * Verify batch Assigning of Patients to a Nurse
2. **Integration Testing**
   * Ensure correct data flow between modules (e.g., from lists to → filter engine → UI).
   * Patient list imports → Filtering Engine → UI display
   * Combined file logic → Match engine → Duplicate/unmatched file output
   * Nurse assignment → Patient record update
   * Integration of note-logging with patient profiles
   * Access control for restricted features
   * Auto-assignment of nurses by ZIP code
3. **System Testing**
   * Validate the complete system's filtering accuracy and user workflows.
4. **User Acceptance Testing (UAT)**
   * Present to Nursing director to help validate the filtering results and usability.

**5. Test Cases**

**5.1 Functional Test Cases**

| **ID** | **Test Case** | **Input** | **Expected Output** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| TC007 | Find Patients Nurses | Zip code, Address, phone number | Show patients by Name, DOB, lastname, etc |  |
| TC008 | Add notes to Patient profiles | Combined list and Patient profile function | Add notes to patient profiles. |  |
| TC009 | Assign Nurse automatically | Combined list of patients profile | Name of the Nurse is assigned simply by typing her name. |  |
| TC0010 | Combine list quickly | Import excel list over 5000 names | Combined lists in less than 3 seconds |  |
| TC0011 | Edit all old patient lists | Import previous patient lists | Access old list along with and be able to update the list. |  |
| TC0012 | App only Accessible to Directors, Manager, supervisor, and specialist | A single user name and password will be used  setup | Provides access to combine lists from Medicaid and Database lists and maintain HIPAA |  |
| TC0012A | Nurse log | Patient profile | Patient assigned date to current Nurse |  |
| TC0013 | Filter 5000 patients | Combined data | Process the lists in less than 5 seconds |  |
|  |  |  |  |  |
| TC0014 | Program to keep unmatched data in a separate file | Combined data | A file with unmatched data |  |

**5.2 Security Test Cases**

| **ID** | **Test Case** | **Expected Outcome** | **Pass/Fail** |
| --- | --- | --- | --- |
| ST001 | Unauthorized access attempt | Access denied |  |
| ST002 | Encryption | Lists should not be readable. | Pass |
|  |  |  |  |

**5.3 Performance Test Cases**

| **ID** | **Test Case** | **Test Method** | **Expected Outcome** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| PT001 | Load test (5000 patients) | Simulate large dataset | System filters within 3 seconds | Pass |

**6. Tools and Resources**

* **Testing Framework:** Selenium, PyTest (for automation)
* **Security Scanner:** OWASP ZAP , Postman Security tests, SoapUi, REST- Assured for vulnerability scanning
* **Performance Tools:** JMeter for load testing, Locust, Gatling

**7. Risk Analysis**

| **Risk** | **Mitigation** |
| --- | --- |
| Incorrect Amount of columns could make the program crash | Instruct the Client on best way to use the software and possibilities for a later refactor to make number columns possibly dynamic |
| Assigned Nurse to patient information is not encrypted | Maybe better for client to have a windows folder with old patient lists where client can put in place a password to access it . |
|  |  |
| If System crashes under more than a 5,000 patient load | Conduct performance and stress testing |
| Security Risk based on the inability to create a new user name and password | Implement a way for only user to have access to resetting their password. |
|  |  |

**8. Test Schedule**

| **Phase** | **Start Date** | **End Date** |
| --- | --- | --- |
| Nurse profile Notes | 14 days | 18 days Done |
| Load all Existing  Combined File | 14days | 21days Done |
| Test Execution | 18days | 22 days IN PROGRESS |
| UAT | 14days | 17 days IN PROGRESS |
| Auto Assign Nurse to patients | 14 days | 21 days Done |
| Filter all categories | 14 days | 22 days Done |
| Combine more than 5,000 patients | 9 days | 22 days Done |
| Nurse assigned Log | 14 days | 22 days Done |

**4.1 Testing Levels**

**Unit Testing (White-Box)**

* **Approach:** White-box testing
* **What should be tested?**
  + Verify individual functions, such as filtering logic and input validation.
  + Check how the filtering algorithm processes inputs
  + Ensure correct handling of edge cases (empty records, invalid inputs).
  + Ensure correct handling of edge cases with quantity of patients
  + Ensure correct handling of edge cases with zip\_ codes ( more than one zip code or less than 1)
  + Use **code coverage tools** to check execution paths.
  + Verify Nurse log keeps date and time Nurse was assigned

**Integration Testing (Gray-Box)**

* **Approach:** Combination of black-box and white-box testing
* **What should be tested?**
  + Ensure correct data flow between modules (patient List**→ filter app → UI**).
  + Patient list imports/exports
  + Assignment logic
  + Filtering logic and output of all lists combined, duplicates, unmatched.
  + Note window logic and output .

Validate:

* The results of filtering are saved to the correct files and format.
* The Assignment is presented correctly to the indicated patient.
* Imports and exports are correctly read by the program and presented
* Notes windows and notes added to it are saved properly and in the correct profile.

**System Testing (Black-Box)**

* **Approach:** Black-box testing
* **What should be tested?**
  + Validate the **complete system's** filtering accuracy and efficiency
  + Test patient search and filtering functionality without looking at the internal implementation.
  + Ensure that different roles (nurse, admin) receive only the appropriate patient data.
  + Validate date function filters by multiple parameters or categories

**User Acceptance Testing (UAT) (Black-Box)**

* **Approach:** Black-box testing
* **What should be tested?**
  + Have **nurses and administrators** validate the filtering results for usability and correctness.
  + Verify that the system its easy to use and understand how to operate the program.
  + Conduct real-world scenario testing to ensure the system is robust and meets client requirements.
  + Get feedback on ease of use and UI design.

How to Generate Sheets for testing

This project uses the fake data generator to create Excel files to test the application

* Database\_data.xlsx
* Mediaid\_data.xlsx

1. How to create these files?

* Run generate\_test\_data.py , from the terminal run python generate\_test\_data.py
* Follow the prompts How many name to generate(500 to 5000)
* Do you want to add duplicates and how many
* To include unmatched data and which file to add the data to.
* You should get these two files database\_data.xlsx and Medicaid\_data.xslx
* These will be saved in the working directory
* Load the database\_data file first and then Medicaid\_data second.

1. Once loaded combined the files with the combine button.

Example:

1. Test for generate 5000 names, add 20 duplicates and add 10 unmatched to the database. Load Database first and then Medicaid
2. Press button Combine data
3. View , export duplicates and unmatched lists.
4. Pip install pandas faker openpyxl