**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 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 for the program to combined patients lists quickly so that Nurses can be assigned to these 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.
* 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.
* 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.

**3. Scope**

**3.1 Features to be Tested**

**[ x ] Flag duplicate patients**

**[ x ] Export filtered lists**

**[ x ] Track Nurses by dates assigned to patients.**

**[ x ] Add columns for data to be combined into.**

**[ x ] Validate Exel files**

**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)
2. **Integration Testing**
   * Ensure correct data flow between modules (e.g., from lists to → filter engine → UI).
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** |
| --- | --- | --- | --- | --- |
| TC006 | OrganizeData | Name,DOB,MotherName,  Mother id, zipcode, city, address, phone number | Information filtered and organized into the input categories |  |
| TC007 | Find Patients Nurses | Zip code, Address, phone number | Show patient profile along with the name of the assigned Nurse |  |
| TC008 | Flag duplicates | Duplicates from combined data | Duplicate patients and families should have the first date it was found and some way to flag duplicate. |  |
| TC009 | Print all combined lists | Combined list of patients | Print all lists including duplicates |  |
| TC0010 | Combine list quickly | Import excel list over 5000 names | Combined lists in less than 3 seconds |  |
| TC0011 | Edit 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 | User name and password  setup | Provides access to combine lists from Medicaid and Database lists and maintain HIPAA |  |
| TC0013 | Program to keep unmatched data in a separate file | Combined data | A file with unmatched data |  |
| TC0014 | Access to Nurse Track record | Combined data | # of families attended by Nurses |  |

**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. |  |
|  |  |  |  |

**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 |  |

**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 search for  Nurse patients | Perform unit testing |
| Incorrect file when trying  To upload old data from  Past lists. | Give client choice of access to select from old data list. |
| Any excel file can be loaded | Make sure is an excel file and check for column quantities before loading |
|  |  |
| If System crashes under 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 log  history data | 7 days | 15 days Done |
| UI update | 10 days | 15 days Done |
| Load an Existing  Combined File | 10 days | 16 days Done |
| Test Execution | 7 days | 14 days |
| Bug Fixing & Retest  -Nurse Nan error  -Encryption Error  -screen clear  button | 14 days | 22days Done |
| UAT | 12 days | 17 days |

**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).
  + 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**).
  + Validate security aspects like role-based access control (**security model** that **restricts system access** based on a user's **role** within an organization).

**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.

**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.
  + 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 1000)
* 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 1000 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