

**Mauritius Registry Software**  
**Technical Spec**

**Team:**

**Andre Ortega**

**Joel Collins**

# Contents

<b>OVERVIEW .....</b>	<b>3</b>
<b>CONTEXT .....</b>	<b>3</b>
<b>PROPOSAL .....</b>	<b>3</b>
<b>GOALS .....</b>	<b>3</b>
<b>EXISTING SOLUTION .....</b>	<b>3</b>
<b>PROPOSED SOLUTION .....</b>	<b>4</b>
<b>1. Users .....</b>	<b>4</b>
<b>2. Use cases .....</b>	<b>4</b>
<b>3. Model.....</b>	<b>12</b>
<b>4. Code structure.....</b>	<b>14</b>
<b>OPERATIONS.....</b>	<b>14</b>
<b>DELIVERABLES .....</b>	<b>14</b>

## **OVERVIEW**

The UK Kidney Association (UKKA) will develop a Web application to capture data for renal patients in Mauritius. The data collection will incorporate the 4 data completion forms provided in the requirements.

## **CONTEXT**

Nephrologists from Mauritius who trained in the UK contacted the UKKA asking for advice on how to set up and develop a renal registry in Mauritius. Following discussions between Mauritius nephrologists and UKKA, the UKKA expressed the desire to support the development of an electronic system to capture data for renal patients in Mauritius. The system will collect relevant renal data on patients based on a set of defined paper forms supplied by Davy Ip (NRR A0.2 Registration, NRR B0.1 Assessment dialysis, NRR C0.1 Change of KRT Modality, NRR X0.1 Stopping Dialysis) to allow initial patient registration, assessment, edition of KRT modality and dialysis termination. Latest versions of the paper forms were also provided by Davy (NRR A0.5 Registrationcccc.pdf, NRR C0.4 Change of KRT Modality.pdf, NRR B0.3 Assessment dialysis.pdf, NRR X0.2 Patient Stopping Dialysis.pdf).

## **PROPOSAL**

The UKRR will develop a front-end for data entry and capture that will be based on the Nephwork style application which will be developed and customised for the Mauritius data collection to incorporate the 4 data completion forms. Before commencement of the development, the Mauritius nephrology team needs to finalise the data capture forms in terms of data specification, code lists and formats.

## **GOALS**

1. Allow registered users to record renal data organised into a database, which then can be viewed, listed and updated electronically.
2. Allow superusers/admin users to manage apps and models organised into a database, which can then be edited, viewed, and deleted electronically. With models, we refer to any element (for example Health Institution, Patient, Comorbidity, Disability) that is included in the web app.

## **EXISTING SOLUTION**

Nephrologists from Mauritius currently collect data based on the completion of 4 paper forms that are subsequently manually entered into an Excel spreadsheet.

## PROPOSED SOLUTION

The following describes the users, use cases, model and code structure included in the renal data collection system developed by the UKKA.

### 1. Users

There are two type of users that have access to the Mauritius web application: admin/superuser and staff/registered user.

- Admin/Superuser: Registered user that can access the administration site of the application for the management of any app and data model. As described above, models refer to any element (for example Health Institution, Patient, Comorbidity, Disability) that is included in the web app. This is a trusted site user to create and edit content related to the existing models. This type of user is also able to manage (create, view, edit, delete) the users of the application.
- Staff user/registered user: Registered user that can create, view and update patient information, assessments, modalities and related information.

The list of users (superuser and registered users) will be stored in the database by the Mauritius team in charge to register the users in the system.

### 2. Use cases

The following describes the list of functionalities (use cases) included in the Mauritius application. The description covers the user that is authorised to execute the use case, the preconditions, outcomes, basic and alternative flow of each functionality.

#### Use case: Manage users

- User: Superuser
- Precondition: Superuser is logged in the system.
- Successful outcome: The successful outcome will depend on the action executed by the superuser. Possible actions: create, edit, view, delete user.
- Minimal outcome: Feedback after action attempt.

#### Use case: Manage application objects

- User: Superuser
- Precondition: Superuser is logged in the system.
- Successful outcome: The successful outcome will depend on the action executed by the superuser. Possible actions: create, edit, view, delete object.
- Minimal outcome: Feedback after action attempt.

#### Use case: Reset password

- User: Registered user, Superuser
- Precondition: User exists in the system.
- Successful outcome: User's new password.
- Minimal outcome: Feedback after reset password attempt.

- Basic flow 1:
  1. The user clicks on the "Change Password" option from the Options menu.
  2. The user enters his old password and new password is entered twice correctly.
  3. The password is reset by the system, and it displays a successful message.
- Basic flow 2:
  1. The user clicks on the "Forgot your password?" option at the login page.
  2. The user enters his/her email to receive the instructions to reset the password.
  3. The system confirms the instructions were sent to the user.
  4. The user is informed about successful password reset attempt and follows instructions to reset the password.
- Alternative path for Basic flow 1:
  - 2a. The user enters an invalid old password and/or the new password is re-entered incorrectly.
  - 3a. The system displays an error message.

Mauritius server requirement for this functionality:

Mauritius needs to provide the settings for an SMTP email server to the developers so they can include them in the configuration file of the web app.

### **Use case: Register patient**

- User: Registered user, Superuser
- Precondition: User is logged in. Patient (N.I.C No or passport No) does not exist in the selected Health institution and unit.
- Successful outcome: Patient is created.
- Minimal outcome: Feedback after registration attempt.
- Basic flow:
  1. The user is logged in.
  2. The user selects "Register" from the Patients menu option.
  3. The system displays a blank patient registration form.
  4. The user enters the patient identification (N.I.C No or passport No) and all the required fields (See requirements document: Notes to forms A,B,C,X.docx)/enters required fields and additional information/completes the form. The user submits the information.
  5. The system verifies that the patient does not exist in the health institution and unit using the patient identification, which is unique for every patient.
  6. The patient registration is correct.
- Alternative paths:
  - 5a. The patient already exists, and the user fails to add a new patient.
  - 6a. The system informs the user that the patient already exists.
  - 4b. User enters invalid information.
  - 5b. The system validates the information.
  - 6b. The user fails to create a new patient and the system informs the user of the form fields to correct.

**Use case: View patient registration information**

- User: Registered user, Superuser
- Precondition: User is logged in.
- Successful outcome: Patient information.
- Minimal outcome: Feedback after searching attempt.
  
- Basic flow 1:
  1. The user is logged in.
  2. The user selects "List" from the Patients menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit number.
  4. The user selects the "View" option of a record from the displayed list.
  5. The system displays the patient information.
  
- Basic flow 2:
  1. The user is logged in.
  2. The user selects "List" from the Patient menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
  4. The user enters the search term and presses the search button.
  5. The system verifies that the patient exists.
  6. The system displays the matches.
  7. The user selects the "View" option of the desired record.
  8. The system displays the patient's information.
  
- Alternative path for Basic flow 1,2:
  - 3a. The system informs the user that there are no patients.
  
- Alternative path for Basic flow 2:
  - 5b. The patient does not exist.
  - 6b. The system informs the user that there are not matching records.

**Use case: Edit patient registration**

- User: Registered user, Superuser
- Precondition: User is logged in.
- Successful outcome: Updated patient information.
- Minimal outcome: Feedback after searching or updating attempt.
  
- Basic flow 1:
  1. The user is logged in.
  2. The user selects "List" from the Patient menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
  4. The user selects the "Edit" option of a record from the displayed list.
  5. The system displays the patient registration form with current information.
  6. The user enters the patient new information and submits the form.

7. The system validates the information.
  8. The system informs the user about the successful update.
- Basic flow 2:
    1. The user is logged in.
    2. The user selects "List" from the Patient menu option.
    3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
    4. The user enters the search term and presses the search button.
    5. The system verifies that the patient exists.
    6. The system displays the matches.
    7. The user selects the "Edit" option of the desired record.
    8. The system displays the patient form with current information.
    9. The user enters the patient new information and submits the form.
    10. The system validates the information.
    11. The system informs the user about the successful update.
  - Alternative path for Basic flow 1,2:
    - 3a. The system informs the user that there are no patients.
  - Alternative path for Basic flow 1:
    - 6b. User enters invalid information.
    - 7b. The system validates the information.
    - 8b. The user fails to update the patient and the system informs the user of the form fields to correct.
  - Alternative paths for Basic flow 2:
    - 5b. The patient does not exist.
    - 6b. The system informs the user that there are not matching records.
    - 9c. User enters invalid information.
    - 10c. The system validates the information.
    - 11c. The user fails to update the patient and the system informs the user of the form fields to correct.

### **Use case: Add assessment for dialysis patient**

- User: Registered user, Superuser
- Precondition: User is logged in. Associated patient exists and is in dialysis mode.
- Successful outcome: Patient assessment is created.
- Minimal outcome: Feedback after creation attempt.
- Basic flow:
  1. The user is logged in.
  2. The user selects "List" patients' option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
  4. The user selects the "Add dialysis assessment" option of a record from the displayed list.

5. The system displays a list of assessments and an option to add a new dialysis assessment.
6. The user selects "New dialysis assessment".
7. The system displays a blank patient assessment form.
8. The user enters all the required fields /enters required fields and additional information/completes the form and submits the information.
9. The system validates the information.
10. The system informs the user about the successful assessment creation.

- Alternative path:

8a. User enters invalid information.

9a. The system validates the information.

10a. The user fails to create a new assessment for the selected patient and the system informs the user of the form fields to correct.

- Alternative path:

5b. The system displays the following information:

There are no records.

Note: The patient should be in dialysis to register a new dialysis assessment.

#### **Use case: Edit assessment for dialysis patient**

- User: Registered user, Superuser
- Precondition: User is logged in. Assessment exists.
- Successful outcome: Updated patient assessment.
- Minimal outcome: Feedback after updating attempt.

- Basic flow:

1. The user is logged in.
2. The user selects "List" from the Patients menu option.
3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
4. The user selects the "Add dialysis assessment" option of a record from the displayed list.
5. The system displays a list of assessments and an option to add new assessment.
6. The user selects the "Edit" option of an assessment chosen from the displayed list.
7. The system displays the assessment form with current information.
8. The user enters the assessment new information and submits the form.
9. The system validates the information.
10. The system informs the user about the successful update.

- Alternative path:

5a. The system informs the user that there are no patient assessments.

- Alternative path:

8b. User enters invalid information.

9b. The system validates the information.

10b. The user fails to update the assessment and the system informs the user of the form fields to correct.



### **Use case: View patient assessment information**

- User: Registered user, Superuser
- Precondition: User is logged in.
- Successful outcome: Patient's assessment information.
  
- Basic flow 1:
  1. The user is logged in.
  2. The user selects "List" from the Patients menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit number.
  4. The user selects the "Add dialysis assessment" option of a record from the displayed list.
  5. The system displays a list of assessments and an option to add new assessment.
  6. The user selects the "View" option of an assessment chosen from the displayed list.
  7. The system displays the assessment details.
  
- Basic flow 2:
  1. The user is logged in.
  2. The user selects "List" from the Patient menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit number.
  4. The user selects the "Add dialysis assessment" option of a record from the displayed list.
  5. The system informs the user that there are no assessments.

### **Use case: Start/Change of KRT modality for a patient**

Note: Take into consideration that in the patient registration form, the user can provide a chronology of previous and present KRT modalities for a patient (as described in the paper form NRR A0.2 Registration). The modality filled as current/present in the registration form will be affected by this functionality.

- User: Registered user, Superuser
- Precondition: User is logged in. Associated patient exists.
- Successful outcome: The patient is in a new KRT modality.
- Minimal outcome: Feedback after creation attempt.
  
- Basic flow:
  1. The user is logged in.
  2. The user selects "List" from the Patients menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
  4. The user selects the option "Start/Change modality" of a record from the displayed list.
  5. The system displays a list of modalities (if there is any) and an option to start a new modality.
  6. The user selects "Start new modality".
  7. The system displays a blank modality form. The form depends on the existing of previous modalities and the modality to be set (this is according to the paper form NRR C0.1 Change of KRT Modality).
  8. The user enters all the required fields and submits the information.
  9. The system validates the information.

10. The system informs the user about the successful modality set. This would be the current/present KRT modality for the patient.

- Alternative path:

8a. User enters invalid information.

9a. The system validates the information.

10a. The user fails to set a new modality for the selected patient and the system informs the user of the form fields to correct.

#### **Use case: Edit patient's KRT modality information**

- User: Registered user, Superuser

- Precondition: User is logged in, the modality to edit exists.

- Successful outcome: Updated patient's KRT modality.

- Minimal outcome: Feedback after updating attempt.

- Basic flow:

1. The user is logged in.

2. The user selects "List" from the Patients menu option.

3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.

4. The user selects the option "Start/Change modality" of a record from the displayed list.

5. The system displays a list of modalities (if there is any) and an option to start a new modality.

6. The user selects the "Edit" option of a patient's modality chosen from the displayed list.

7. The system displays the modality form with current information.

8. The user enters the modality new information and submits the form.

9. The system validates the information.

10. The system informs the user about the successful update.

- Alternative path:

5a. The system informs the user that there are no patient modalities.

- Alternative path:

8b. User enters invalid information.

9b. The system validates the information.

10b. The user fails to update the modality information and the system informs the user of the form fields to correct.

#### **Use case: View patient KRT modality information**

- User: Registered user, Superuser

- Precondition: User is logged in.

- Successful outcome: Patient's modality information.

- Basic flow 1:

1. The user is logged in.

2. The user selects "List" from the Patients menu option.

3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit number.
4. The user selects the “Start/Change modality” option of a record from the displayed list.
5. The system displays a list of modalities and an option to add new modality.
6. The user selects the “View” option of a modality chosen from the displayed list.
7. The system displays the modality details.

- Basic flow 2:

1. The user is logged in.
2. The user selects "List" from the Patient menu option.
3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit number.
4. The user selects the “Start/Change modality” option of a record from the displayed list.
5. The system informs the user that there are no records.

**Use case: Stop dialysis**

- User: Registered user, Superuser
- Precondition: User is logged in. This option only appears in the system if a patient has a current KRT modality.
- Successful outcome: Updated patient’s KRT modality (HD or PD). Once this form is submitted, the user won’t be able to start/change a patient KRT modality.
- Minimal outcome: Feedback after updating attempt.

- Basic flow:

1. The user is logged in.
2. The user selects “List” from the Patients menu option.
3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
4. The user clicks on “Stop dialysis” of a patient chosen from the displayed list.
5. The user completes the form (as described in the paper form NRR X0.1 Stopping Dialysis) and submits the information.
6. The system validates the information.
7. The system informs the user about the successful update.

- Alternative path:

- 3a. The system lists the patients.
- 4a. The option Edit dialysis stop appears if the form has been previously saved.

- Alternative path:

- 5b. User enters invalid information.
- 6b. The system validates the information.
- 7b. The user fails to stop dialysis and the system informs the user of the form fields to correct.

**Use case: Add new health institution**

- User: Superuser
- Precondition: User is logged in. Health institution does not exist.

- Successful outcome: Health institution is created.
- Minimal outcome: Feedback after creation attempt.
  
- Basic flow:
  1. The user is logged in.
  2. The user selects the correspondent "Add" option.
  3. The system displays a blank health institution form.
  4. The user enters the required fields/ completes the form and submits the information.
  5. The system verifies that the health institution does not exist.
  6. The creation is correct.
  
- Alternative paths:
  - 5a. The health institution already exists, and the user fails to add a new one.
  - 6a. The system informs the user that the health institution already exists.
  
  - 4b. User enters invalid information.
  - 5b. The system validates the information.
  - 6b. The user fails to create a new health institution and the system informs the user of the form fields to correct.

This same action of creating new record is currently activated for the models : Comorbidity, Disability and Hd unit. The superuser can create, edit, or remove any object related to these models.

#### **Use case: View patient historical data**

This displays a historical of every health institution the patient has been transferred/moved in.

- User: Registered user, Superuser
- Precondition: User is logged in.
- Successful outcome: Patient history.
- Minimal outcome: Feedback after reading attempt.
  
- Basic flow:
  1. The user is logged in.
  2. The user selects "List" from the Patients menu option.
  3. The system displays a paginated list of patients and a search bar to search patients by N.I.C or passport number, name, surname, health institution or unit.
  4. The user selects the option "View hospital history" of a record from the displayed list.
  5. The system presents the list of health institutions with the date of change for the selected patient.
  
- Alternative paths:
  - 5a. The system informs the user that there is no history.

### **3. Model**

Patient

Comorbidity  
HealthInstitution  
HDUnit  
Disability

Relationships :

PatientRegistration

Define the main relationship patient 1 -> 1 registration \* -> 1 health institution

i.e. Every patient is registered in one main health institution and one health institution can be linked to one or many registrations records.

PatientRenalDiagnosis

Define the relationship patient 1 -> \* renal diagnosis

Suggestion: Table for renal diagnosis codes. It was originally designed that way but Mauritius requested the change to enter the renal diagnosis code in the registration form.

PatientKRTModality

Define the main relationship patient 1 -> \* KRT modalities \* -> 1 HD unit

i.e. Every patient can have multiple KRT modalities and one HD unit can be linked to one or many KRT modalities (dialysis modality).

PatientAKImeasurement

Define the main relationship patient 1 -> 1 AKI set of measurements (creatinine, egrf, hb).

PatientAssessment:

Define the main relationship patient 1 -> \* assessments \* -> \* comorbidities, disabilities

i.e. A patient can have multiple assessments and one to many comorbidities and disabilities can appear in one or many assessments.

PatientDialysisAssessment

Define the relationship assessment 1 -> 1 dialysis assessment

i.e. An assessment type is a dialysis assessment.

PatientLPAssessment

Define the relationship assessment 1 -> 1 assessment for the laboratory parameters

i.e. An assessment evaluates laboratory parameters.

PatientMedicationAssessment

Define the relationship assessment 1 -> 1 set of medications

An assessment includes a list of medications.

PatientStop

Define the main relationship patient 1 -> 1 stop dialysis record

The documentation will be automatically updated for any future change in the models or views in the admin side of the application under the DOCUMENTATION option.

## 4. Code structure

As found in the code repository <https://github.com/renalreg/mauritius-renaldataregistry>, the **src** folder contains the files to arrange the applications of the Mauritius renal data registry project. The project folder is found in **src/mauritiusrenalregistry**.

The code to modify the Mauritius renal data registry application is found in **src/renaldataregistry**. This folder includes the files to modify any model (models.py), view (views.py), form (forms.py), route (urls.py) of the Mauritius renal data registry application. The folder src/templates/renaldataregistry contains all the templates of the renaldataregistry application.

The folder src/utils includes the file for custom form validation.

The code for the users application is found in **src/users**.

## OPERATIONS

### Deployment

The software will be provided as a docker image with appropriate docker compose scripts to allow easy deployment and updates over the internet. The README file of the code's repository includes documentation about the deployment.

## DELIVERABLES

### Source code

The application source code will be made available under the GNU Affero GPL v3.0 license in the UK Renal Registry repository during development. Upon completion, source code may be transferred to the client upon request for further development outside of the UK Renal Registry.

### Technical Spec

Documentation on project objectives, list of functionalities, model description and deployment.