ALEX MZIRAI K.

[alexmzirai@gmail.com](mailto:alexmzirai@gmail.com)

0720571166

**DOCUMENTATION FOR THE PALLADIUM HEALTH CENTRE RECORDS DEPARTMENT SYSTEM.**

The assignment involved implementing a web-based system that would allow the Palladium Health Centre Records department to automate her operations. The successful system should allow updating, searching, editing, deleting and viewing of patients in the database. All of this should be accomplished from a central registry.

The data set I chose to work with includes a database containing two tables/models named Patient and Kin, inside an app called ‘**data**’ which is pluggable within Django making it easier to re-use apps.

The Patients table holds information about the patients currently enrolled at the centre receiving treatment. The fields in this table include:

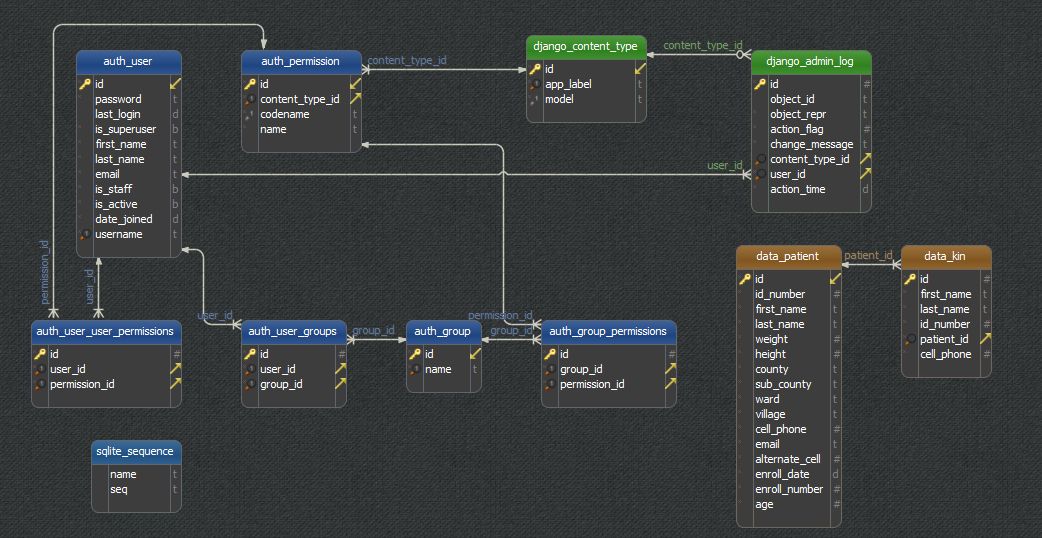
* Id number (National ID)
* First and last names
* Weight in Kilograms
* Height in centimetres
* Age
* County
* Sub-county
* Ward
* Village
* Cell phone number
* Email address
* Alternate cell phone number
* Enrol date (automatically added as current date on record creation in the database)
* Enrol number

The Kin table holds the information about the next of kin to the above mentioned Patients. The fields in kin include:

* First and last names
* Id number (National ID)
* Cell phone
* Patient (a foreign key linking to the Patients table).

Each patient can have one or more kin.

The database schema is as shown below:



How to run the application:

Since the application is written in Python and Django web framework, it can be started through the command “**python manage.py runserver**” while inside the main project directory that contains the **manage.py** file.

Django should then run at localhost (127.0.0.1) in port 8000 by default.

So the url will look something like **127.0.0.1:8000/**

The django application holding the logic for the application is called “**data**”. Therefore, to access the list of patients in the database, one would simply have to hit url **127.0.0.1:8000/data/patients**

The list of all next of kin will also be at **127.0.0.1:8000/kin.**

There is also an API that exposes the models in the database for access from mobile clients and devices and 3rd party applications that may need to consume the data.

The API logic is contained in a separate app called “**api**”. The API endpoints will therefore be located at 127.0.0.1:8000/api.

To view patient data via API, type in **127.0.0.1:8000/api/patients**.

The next of kin data can be viewed at **127.0.0.1:8000/api/kin.**

**Minimum Requirements**

These include:

* A windows or Linux machine with a web browser installed
* Ability to access localhost for web applications
* A database server (in this system I used SQLite but this can be changed to the more stable and robust PostgreSQL in production.
* Knowledge of Python and Django is preferred but not a requirement

**Dependencies**

These include:

1. Python programming language should be installed on the machine
2. Django web framework version 1.11
3. Gunicorn and WSGI on the production server
4. Bootstrap framework 3 or 4
5. JQuery library
6. Django REST framework