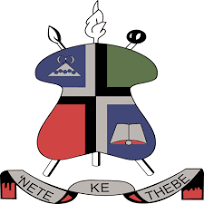
**Ntsoekhe Distributed Database Management System**



**By: DistributedIT**

***Members:***

## 202000715 – Mapeshoane.M

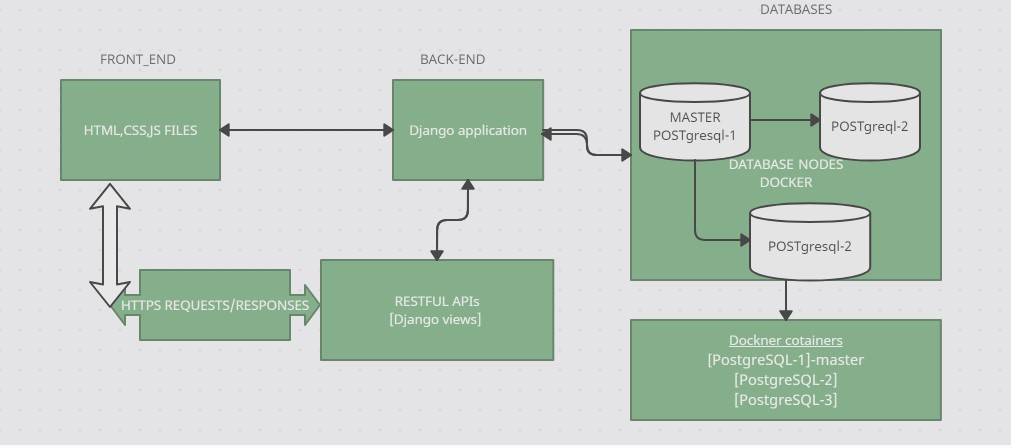
## 202000106 – Ntilane.R. N

## 201702785 – Libetso. T

## 202002429 – Hlaha. P.P

## 202002856 – Moriee. L

SYSTEM ARCHITECTURE DESIGN- NTSOEKHE HEALTHCARE SYSTEM



DESCRIPTION

1. Frontend:
   * Initiates HTTP requests to the backend when users interact with the user interface.
   * These requests could include actions such as submitting a form, fetching data, or updating information.
   * Receives HTTP responses from the backend containing the requested data or status updates.
   * Renders the received data or updates the UI accordingly to provide feedback to the user.
2. Backend (Django Application):
   * Listens for incoming HTTP requests from the frontend.
   * Handles each request by invoking the appropriate handler function based on the URL route and HTTP method (GET, POST, PUT, DELETE, etc.).
   * Interprets the request parameters, performs necessary data processing or business logic, and generates an appropriate response.
   * Utilizes the ORM (Object-Relational Mapping) provided by our chosen framework (Django) to interact with the SQLite database for data retrieval, storage, update, or deletion.
   * Implements security measures such as SSL/TLS for secure data transmission and authentication and authorization mechanisms to control access to sensitive data and functionalities.
   * Interacts with the database nodes encapsulated within Docker containers to perform data operations.
3. Database Nodes (POSTgresql):

* Each database management system (PostgreSQL. etc.) is encapsulated within its own Docker container.
  + Managed independently of the Django application.

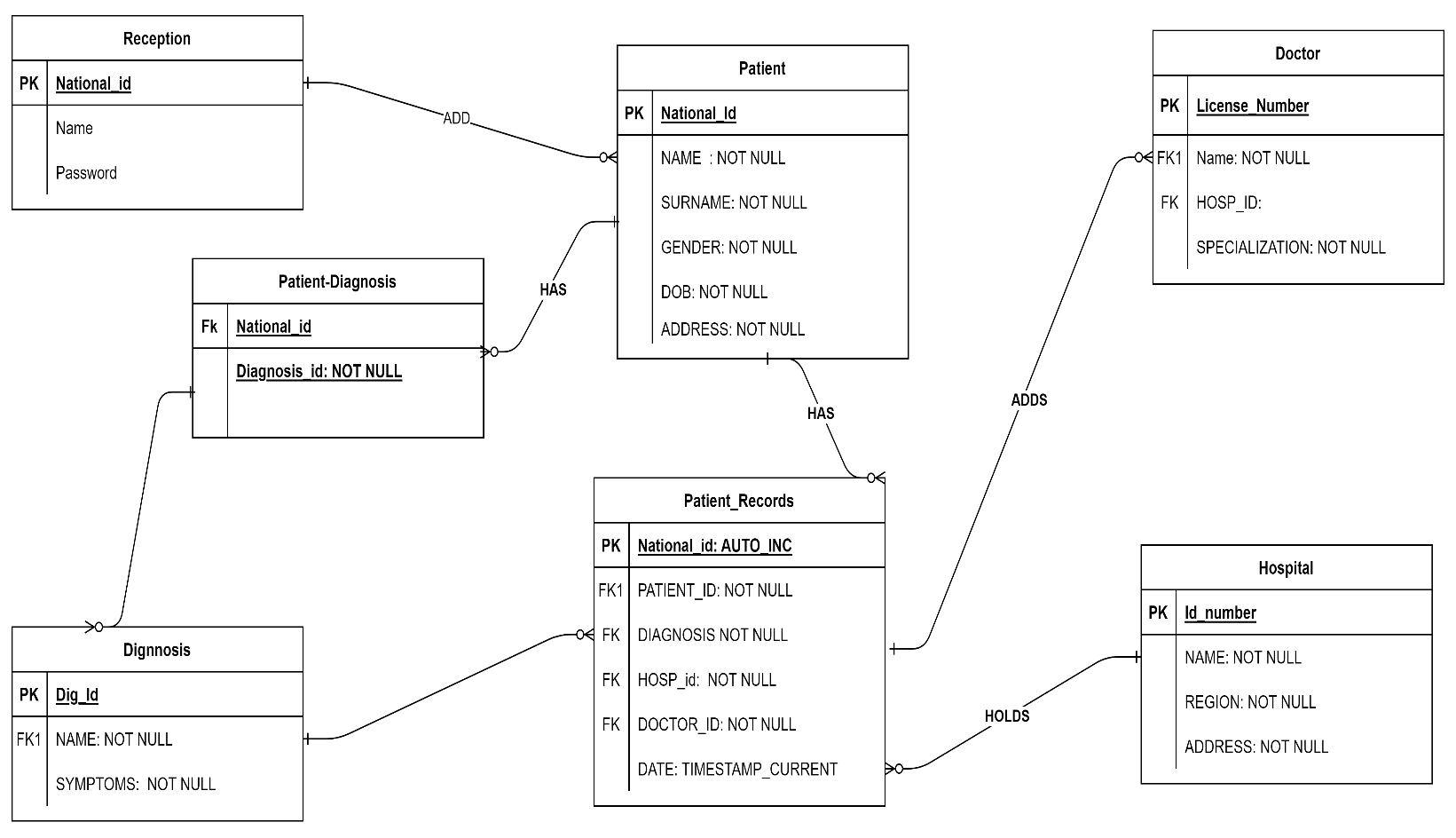
Receive requests from the Django application for database operations.

* + Store application data locally within the Docker containers.

1. RESTful APIs:
   * Implemented within the backend to define the interface between the frontend and backend components.
   * Receives HTTP requests from the frontend and routes them to the appropriate handler functions.
   * Translates incoming requests into actions to be performed by the backend, such as querying the database, processing data, or invoking business logic.
   * Serializes data returned by the backend into JSON or XML format before sending it back to the frontend as an HTTP response.
   * Enables efficient communication between different components of the system by providing a standardized way for them to interact.

In conclusion, the frontend interacts with the backend by sending HTTP requests, the backend processes these requests, interacts with the SQLite database nodes to perform necessary data operations, and generates HTTP responses that are sent back to the frontend. The RESTful APIs serve as the intermediary layer that facilitates communication between the frontend and backend components, ensuring seamless interaction and data exchange within the system. It is clear that RESTful APIs are part of the Django application and interact with the frontend and backend components, while the database nodes are encapsulated within Docker containers and are interacted with by the Django application.

**ENTITY RELATIONSHIP MODEL(ERD)**



Description:

Our ERD show that in our System, there are 7 Entities currently, Namely:

* Receptionist.
* Patient.
* Doctor.
* Patient\_Records.
* Hospital.
* Diagnosis.

A last Entity **(Patient\_Diagnosis)** is derived from the many-to-many relationship between the Patient and Diagnosis entities. The table Patient\_Records contain information for almost all the other entities but for data consistency and integrity, store the primary keys for those entities.

Moreover, Diagnosis and patient\_Records Entities are associative entities because Record or Diagnosis can not happen on its own.

**References:**

1. *Website: Docker Documentation*
2. *Django Documentation. (n.d.). Django Documentation.*
3. *Fielding, R. (n.d.). Representational State Transfer (REST). [Doctoral dissertation, University of California, Irvine]. Architectural Style REST.*