# Documento Architetturale Progetto Stage Exprivia

# GSS Employees Management System

# GSS (Gamma Security Services) is a worldwide military private security company (PMSC). For each on or off duty of its employees, registered in the management system, it is possible to view a personal card that presents all their data, such as:

# As person

# generic information

# photography

# Contacts

# As worker

# Military rank

# any current assignment (e.g. mission commander in xxx)

# current status (in service/resigned/fired/retired)

# chronology of service statuses, associated with a date

# Only some users of the system have the possibility to create, modify or delete employee data, while all others only have the possibility to consult them.

# 

# The system is usable via common web browsers, therefore end users haven't to install any software on their workstations. To ensure its maintainability, the system has been developed using cutting-edge technologies and the architecture has an approach open to new features.

# 

# In this GitHub repository, in addition to the source code, it is possible to find the following UML formatted documentation:

# 

# analysis of functional requirements

# use case analysis

# component diagram

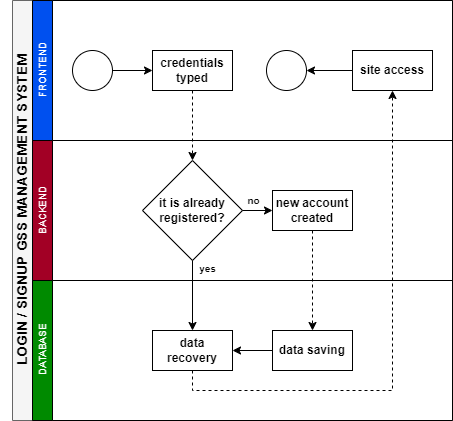
# ER diagram for defining the data model

# flowcharts for main features

# deployment diagram

# functional test document

Flowchart diagram



Software Requirements and Analysis

Frontend

FE - 01

The user interface will allow users to navigate employee data,

as person:

personal information (name, surname, date of birth)

photography

contacts (mobile number, email)

as worker:

qualification (the military rank, years of service)

any current assignment (e.g. mission commander in xxx)

current status (in service/resigned/fired/retired)

chronology of service statuses, associated with a date

FE - 02

The user interface will allow to filter the user research by department, years of service, etc...

FE - 03

The user interface will allow specific users to modify these employee data.

Backend

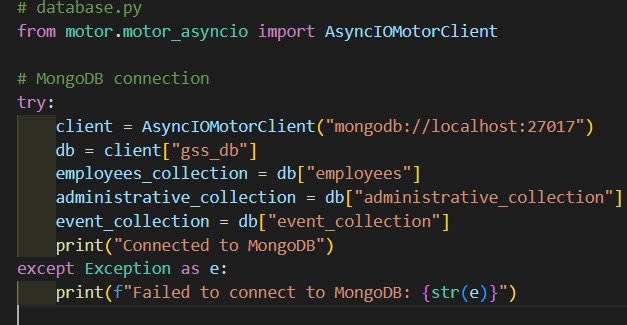
BE - 01

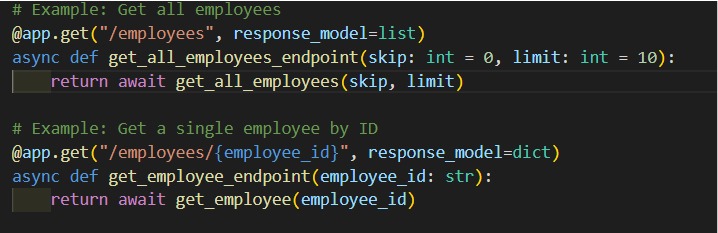
The system backend will be based on REST-API, ensuring access to employee data.

BE - 02

The system will expose an authentication mechanism based on JWT tokens to protect user information.

BE - 03

In order to secure communication between the frontend and backend, communication between frontend and backend will be sured by an SSL channel.



Database

DB - 01

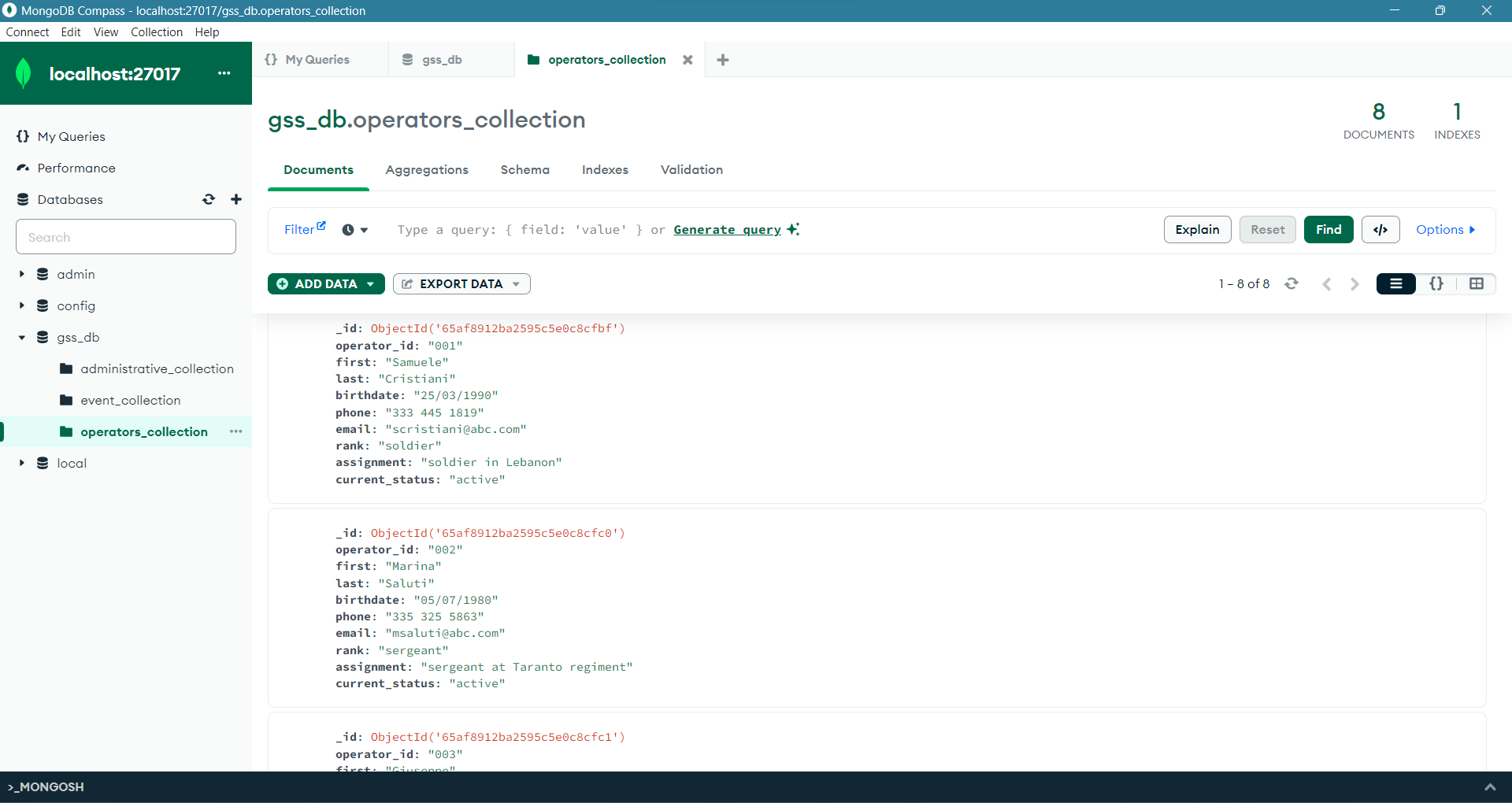
In order to store employees' data, a non-relational database will be used.

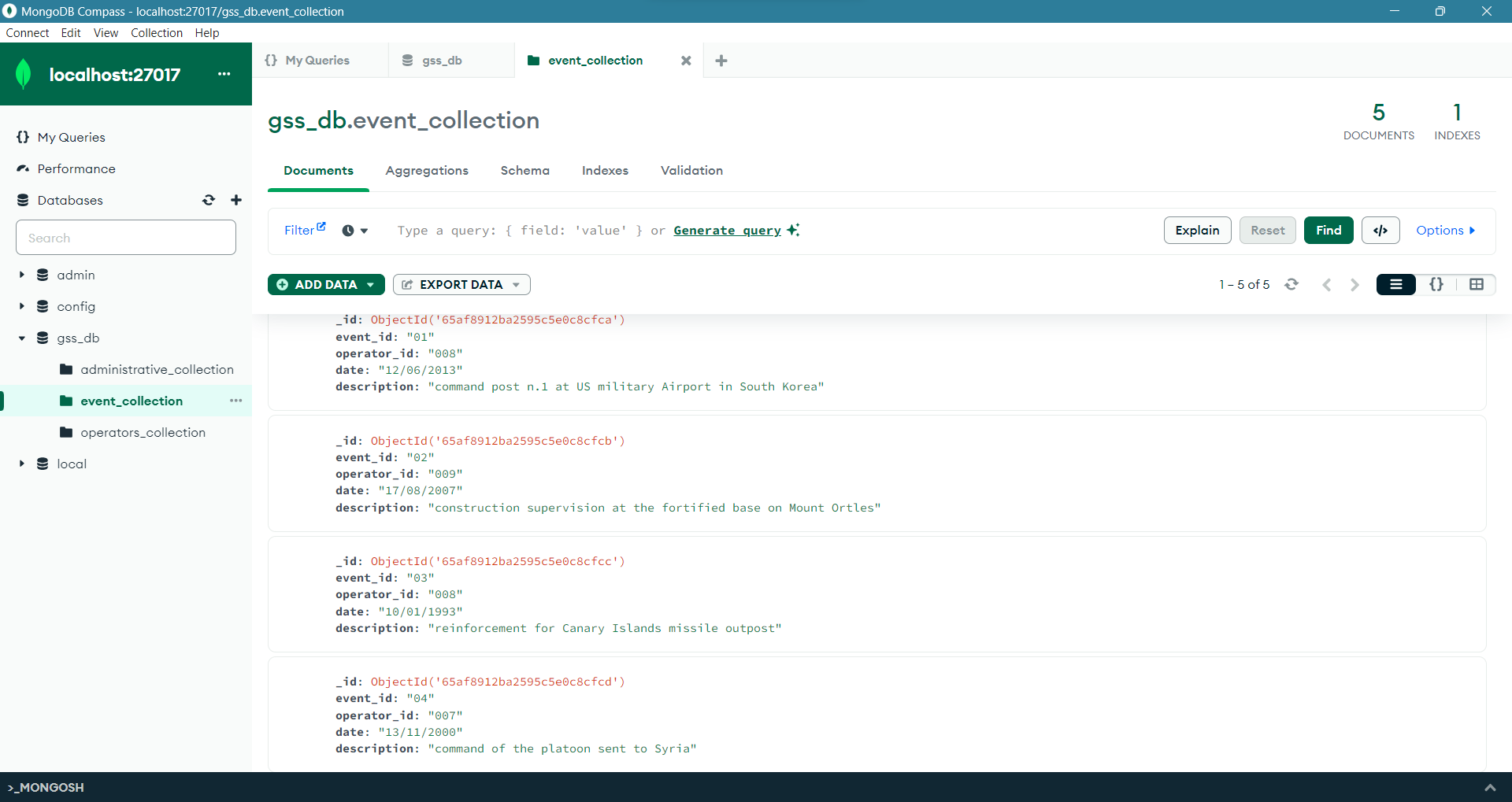
DB - 02

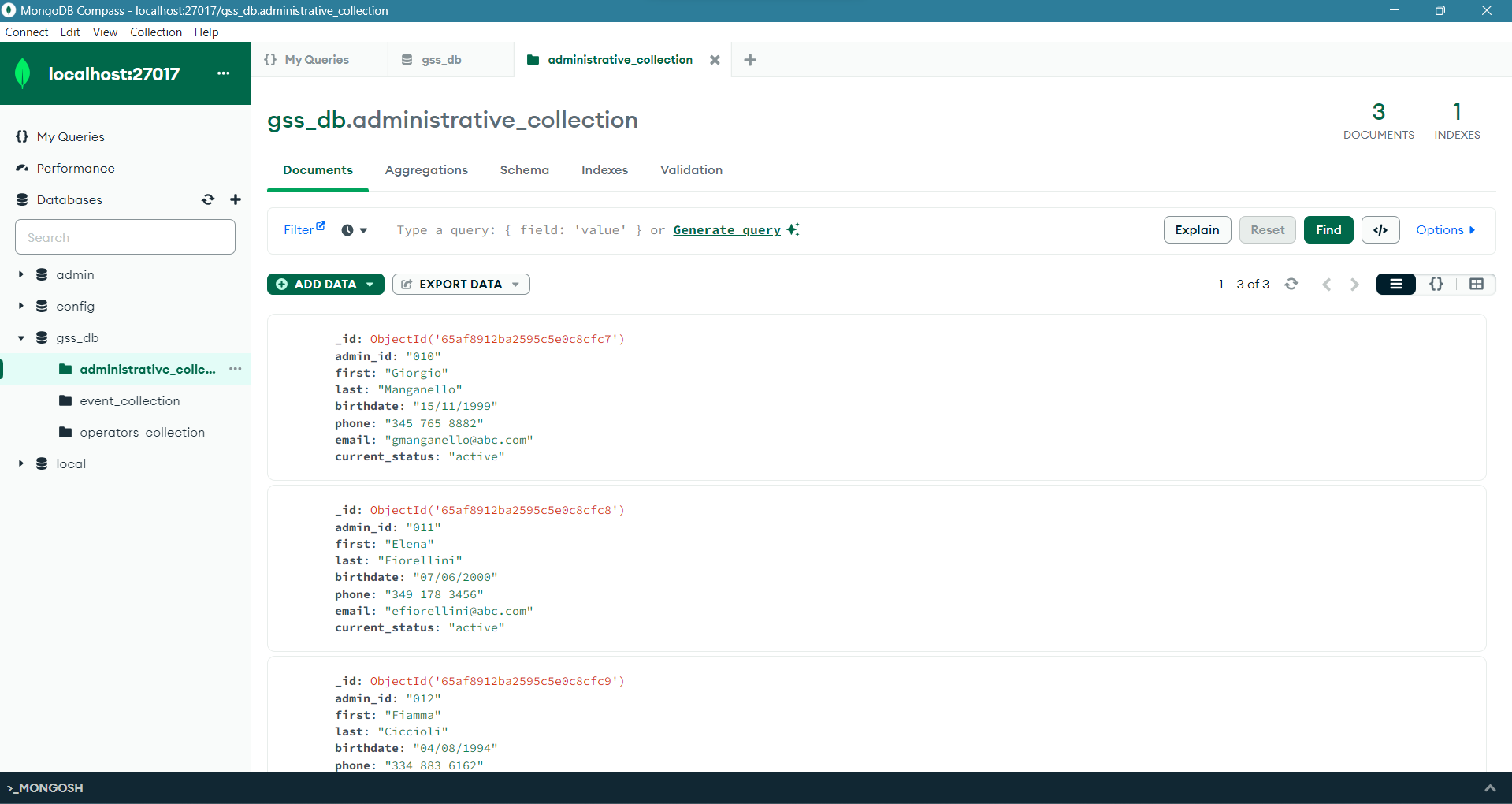
Each document contains employee's data.

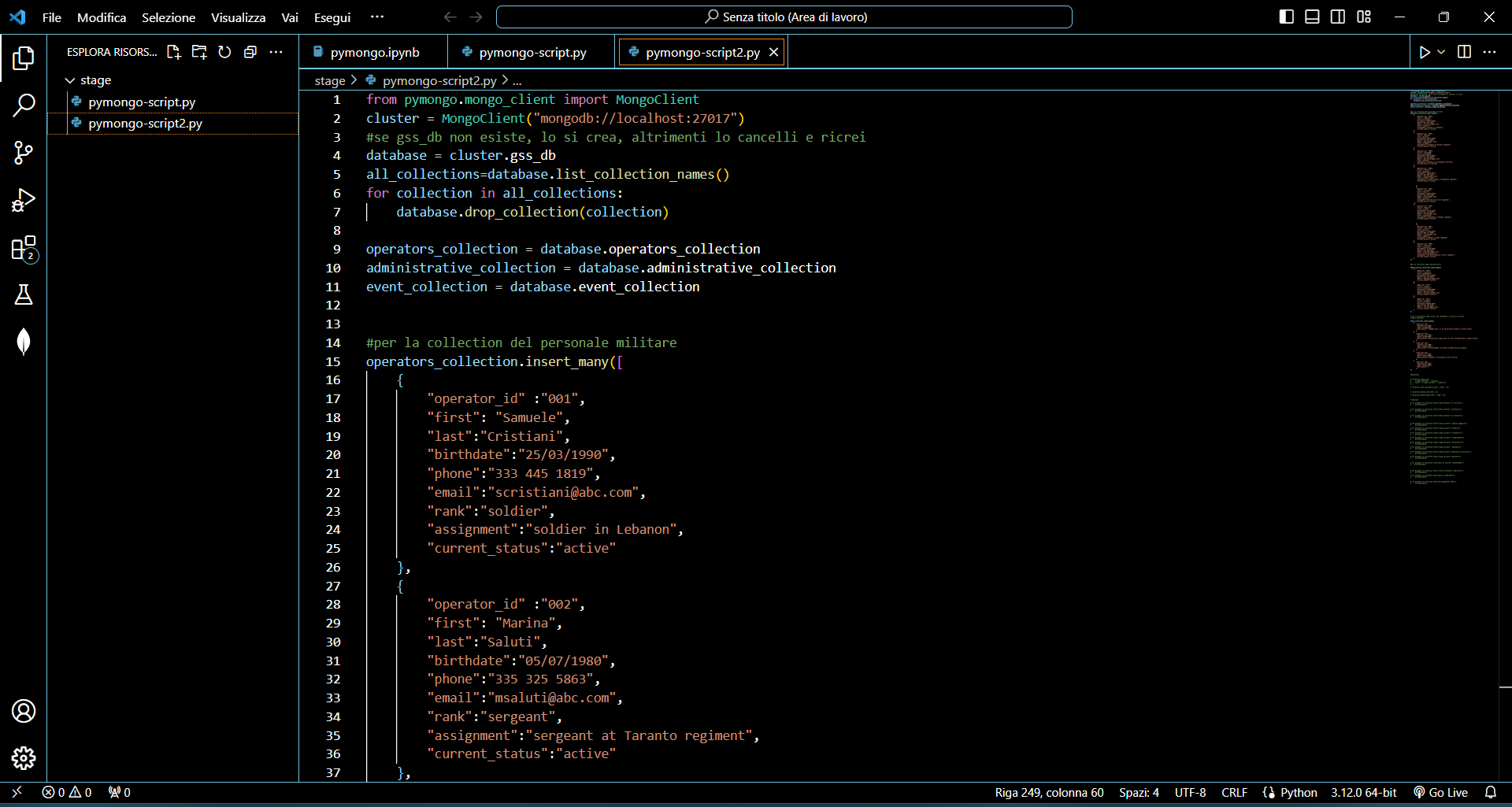
DB - 03

Each document has an unique id, in order to search or update each employee's data.

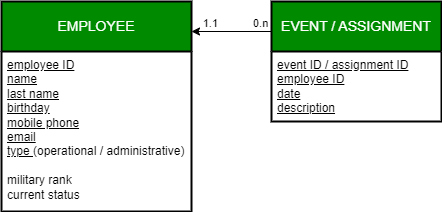








ER Model Diagram



Architectural Schema - Bozza

1. Start

2. Check User Access Level

- If Administrator

- Access to employee data creation, modification, and deletion

- If Other User

- Access only to view employee data

3. User Action

- Choose to View or Modify Employee Data

4. If View Data

- Display Employee Information

- Personal Information

- Event Information

5. If Modify Data

- Request User Input

- Update Personal Information

- Update Event Information

6. Frontend Request

- Send request to Backend API

7. Backend Process

- Handle API request

- Access Database

- Fetch or Update Employee Data or Event

8. Database Interaction

- Perform CRUD operations on Employee Data and Events

9. System Update

- Update Employee or Events Data in the Database

10. Backend Response

- Send response to Frontend

11. Frontend Update

- Update UI based on the response

12. End