**Storage Structure**

**Group-1 Team-2**

Vatsal Soni - 2018201005

Darshan Kansagara - 2018201033

Dhawal Jain - 2018201065

**Subsystem and APIS**

### **Authentication Service**

* 1. API:- /user\_authenticate
     1. Input:-
        1. User’s credential (UserName, Password)
     2. Output:-
        1. Success / Error message

### **Deployment Service**

* 1. API:- /deployService
     1. Input:-
        1. Modelname
        2. FileName
        3. ModelPath
        4. DeployIp
        5. DeployPort
        6. DeployPassword
        7. DeployUserName
     2. Output:-
        1. Deploy successfull / Error

### **Scheduling Service**

* 1. API:- /ScheduleService
     1. Input:-
        1. Modelname
        2. StartTime
        3. EndTime
        4. Repeat
        5. Count
        6. Interval
        7. Repeat\_Period
     2. Output:-
        1. Successfully scheduled / unable to invoke

### **Invocation Service**

* 1. API:- /invocationService/start
     1. Input:-
        1. Modelname
        2. InputStream
        3. DeployIp
        4. DeployPort
        5. DeployPassword
        6. DeployUserName
     2. Output:-
        1. Model Up successfully
  2. API:- /invocationService/end
     1. Input:-
        1. Modelname
        2. DeployIp
        3. DeployPort
        4. DeployPassword
        5. DeployUserName
     2. Output:-
        1. Model Down successfully

### **Notification & Action Service**

* 1. API:- /notification
  2. Input:-
     1. Predicted output of model
  3. Output:-
     1. Run the action code / Send message

### **Logging Service**

* 1. API:- /loggingService
     1. Input:-
        1. Log message
     2. Output:-
        1. Write into log file

### **Monitoring Service**

* 1. API:- /monitoring/model\_status
     1. Input:-
        1. User Id
     2. Output:-
        1. Model Name
        2. Gateway / Server Instance (ip and port)
        3. Status ( UP / Down )
  2. API:- /monitoring/gateway\_status
     1. Input:-
        1. None
     2. Output:-
        1. Gateway Id
        2. Sensor list
        3. Sensor status ( UP / Down )
        4. Gateway Status ( UP / Down )

### **Inferencing Service**

* 1. API:- /inferenceSerivce
     1. Input:-
        1. Usecase (Test data)
        2. Action file
     2. Output:-
        1. Inferencing rate

### **Authorization Service**

* 1. API:- /authorizationService
     1. Input:-
        1. User credentials
     2. Output:-
        1. Valid / Invalid
        2. List of services that user can access

## **Storage structures**

All these **Data Stored in DB** and We used in-memory data structure used for our services

namely:

**List, Dictionary, JSON, Queue,**

**Model Information :**

1. Model Name
2. Scheduling info
3. Gateway information
4. User ID
5. status

**Scheduling information :**

1. Model Name
2. Deployment socket
3. Deployment username
4. Deployment Machine Uname & Password
5. Starttime
6. Endtime
7. Repeat interval
8. count

**Sensor information**

1. Sensor ID
2. Gateway ID
3. Sensor type
4. Sensor status

**Gateway Information**

1. Gateway ID
2. List of Sensor
3. Gateway status

**User credential**

1. Username
2. Password

Model data and config file corresponds to a model will be stored in **Network File System(NFS)** that hat lets a computer user view and optionally store and update files on a remote computer as though they were on the user's own computer ad thus allow a very fast access to the shared data on remote machine.

In case of our platform the main 3 files that are widely used by almost every microservice

Are namely:

1 Model files

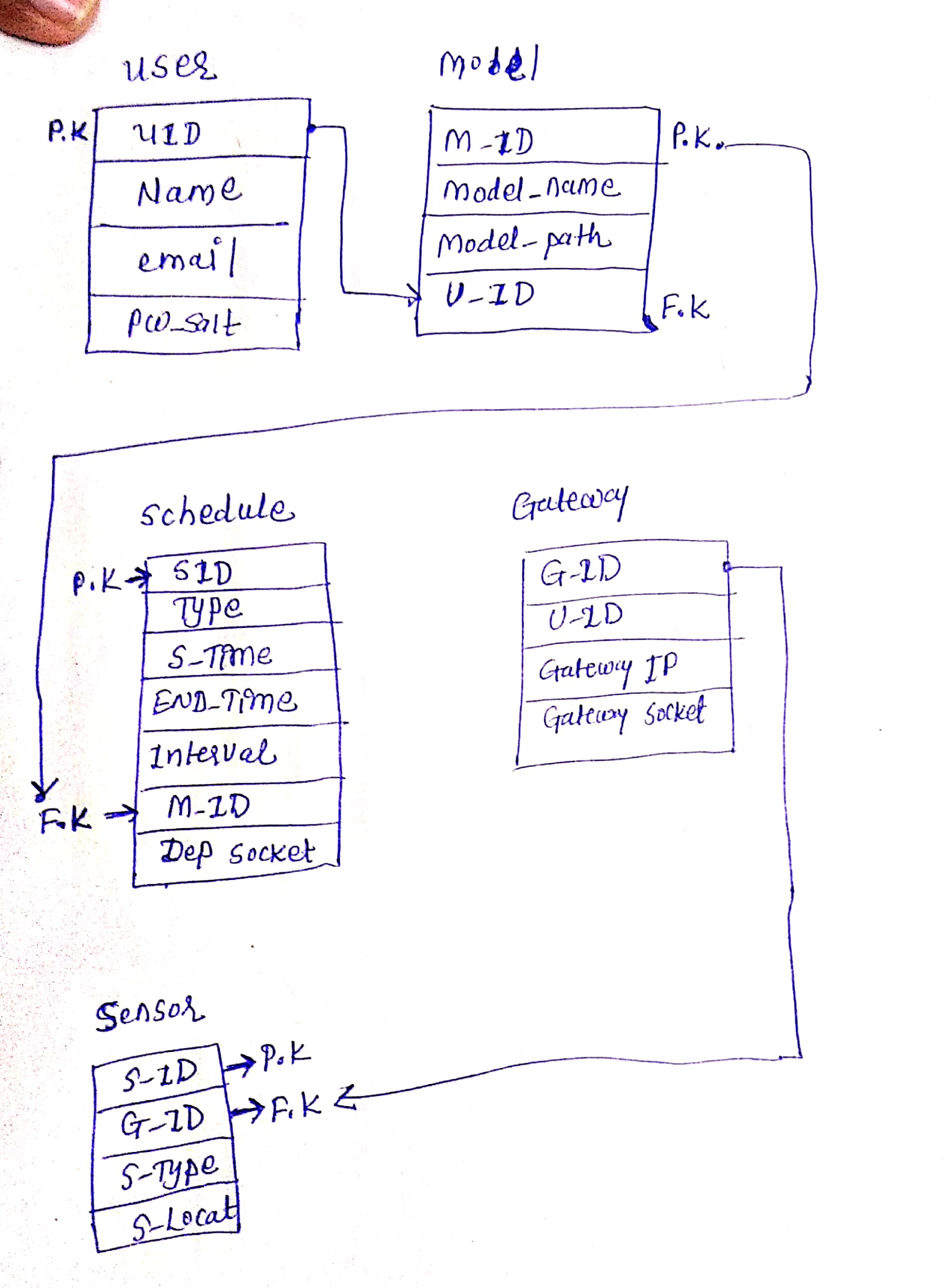
2 Config files

3 Action files

**Format of NFS**

In NFS, there is separate folder for each user and each user’s model data, config file will be stored in this folder.

**DataBase Diagram**

****