File - main_db.py

```
document = {
   "ship_imo": 9876543,
   "date": Date("2016-05-18T16:00:00Z"),
   "ship_name": "RMTCourier",
   "historical":False,
   "daily_data": [
   {
         "identifier":"rpm",
         "name": "RPM",
         "reported":70,
         "processed": 70,
         "is_outlier": False,
         "preprocessor_results":"Passed",
         "z_score": -2.1,
         "unit":"rpm",
         "statement":"RPM is Low",
         "predictions":{
               "m3":[71,72,73],
               "m6": [71,72,73],
               "m12": [71,72,73],
               "ly": [71,72,73],
               "dd": [71,72,73]
         }
   },
  {
         "identifier":"speed",
         "name": "Speed",
         "reported":70,
         "processed": 70,
         "is_outlier": False,
         "preprocessor_results":"Passed",
         "z_score": -2.1,
         "unit":"rpm",
         "statement":"RPM is Low",
         "predictions":{
               "m3":[71,72,73],
               "m6": [71,72,73],
               "m12": [71,72,73],
               "ly": [71,72,73],
               "dd": [71,72,73]
         }
   }
   ],
   "weather_api": [
       {
           "identifier":"tempC",
           "value":25
       },
       {
           "identifier":"swell",
           "value":6
       }
   ],
    "position_api": [
       {
           "identifier":"lat",
           "value":18.520430
       },
```

File - main_db.py

```
        {
            "identifier":"long",
            "value":73.856743
        }
    ],
     "indexes": [
        {
            "identifier":"index1",
            "value":5
        },
        {
            "identifier":"index2",
            "value":8
        }
    ],


[
}
```

File - daily_data.py

```
"""
Collection to store recieved daily data as it is.

Process in short:
User uploads DD(Daily Data) for particular ship on particular data.
Before cleaning or analysing, first we have to store data as it is as we might
need to afterwards for compliance.
So this collection is used to store received data as it is.

Whether to use fields names provided by user in DD file or to use fields names
decided by us is not yet concluded.
As different ships might have different naming conventions. Example say for
Wind Force it could be 'w_force','wind','wind_force'
in their file, but for it to be consistent we can store with our identifiers.
"""

# Method 1  - Data in Array of Objects
document = {
    "ship_imo": 9876543,
    "ship_name": "RMTCourier",
    "date": Date("2016-05-18T16:00:00Z"),
    "historical":False,
    "nav_data_details":{
                "upload_datetime": Date("2016-05-18T16:00:00Z"),
                "file_name":"daily_data19June20.xlsx",
                "file_url":"aws.s3.xyz.com",
                "uploader_details":{"userid":"xyz","company":"sdf"},
    }
    "engine_data_details":{
                "upload_datetime": Date("2016-05-18T16:00:00Z"),
                "file_name":"daily_data19June20engine.xlsx",
                "file_url":"aws.s3.xyz.com",
                "uploader_details":{"userid":"xyz","company":"sdf"},
    }
    "data_available_nav": ['rpm','speed','w_force'],
    "data_available_engine": ['er_temp','er_hum','jwc1_fwin_temp'],

    "data_nav": [
    {       "identifier":"rpm",
            "reported": 70,

    },
    {       "identifier":"speed",
            "reported": 10,
    },
    ],
    "data_engine": [
    {       "identifier":"er_temp",
            "reported": 70,
    },
    {       "identifier":"er_hum",
            "reported": 10,
    },
    ],
}
```

File - report_data.py

```
# Generic Report. Each report will have it's own collection.


document = {
    "ship_imo": 9876543,
    "ship_name": "RMTCourier",
    "date": Date("2016-05-18T16:00:00Z"),
    "historical":False,
    "upload_datetime": Date("2016-05-18T16:00:00Z"),
    "file_name":"lub_oil_data_xyz.xlsx",
    "file_url":"aws.s3.xyz.com",
    "uploader_details":{"userid":"xyz","company":"sdf"},
    "data_available": ['viscocity','temp','basicity'],
    "data": [
    {       "identifier":"viscocity",
            "reported": 70

    },
    {       "identifier":"temp",
            "reported": 46
    }
    ]
}
```

File - ship_configs.py

```
"""
Few terms:

Daily Data: User upload data for their ship daily on Aranti. Data is right now
in the form on Excel file.

"""



"""
Stores all static data about each ship. There are around 50 types of static
fields for
each of the ship, like length, year, port etc. Those are stored here.

Also, each ship might have different machinary, say about 10 systems with total
 200 fields which we
will recieve daily in form of excel file. So field 'data_available' will have
list of all those fields
which are expected to recieve daily.

Suppose user wants to add new machinery to their ship i.e. he wants to upload
new kind of daily data.
Say new cooler is fitted, then this 'data_available' should be modified.

Each time when preprocecssor code will be running, it will check this '
data_available' field to see which of the
fields to extract from dailydata.

A new ship document will be added to this collection once new ship is oboarded
on Aranti.

"""


document = {
        "ship_imo":987654, #MongoDB Index FIeld. Single Index
        "ship_name":"RMT Courier",
        "added_on": Date("2016-05-18T16:00:00Z"),#Date when ship was added
        "grt":1234, #static field #1
        "length":1234, #static field #2
        "50th_field":1234 #Upto 50 static fields of ship data like above two.
        "data_available_nav": ['rpm','speed','w_force'],
        "data_available_engine": ['er_temp','er_hum','jwc1_fwin_temp'],
        "limits_nav":[
        {
        "identifier":"w_force",
        "max":14,
        "min":0,
        },
        {
        "identifier":"speed",
        "max":50,
        "min":0,
        },
        ],
```

File - ship_configs.py

```
        "limits_engine":[
        {
        "identifier":"rpm",
        "max":120,
        "min":50,
        },
        {
        "identifier":"er_temp",
        "max":20,
        "min":60,
        },
        ],
}
```

```
"""
Sample for Multiparametric model configurations. Similar for time- trending.

"""

document = {"name":"speedfoc",
           "training_features":["speed",'w_force',"slip","others"],
           "target_feature":"foc",
           "model_filename":"speedfoc.pickle",
           "durations":["3m","6m","9m","12m","dd"],
           "model_type":"abc",
           "confidence_alpha":0.95,
           "standardize":True,
           "polynomials":2,
           "polynomials_interaction":True,
           }
```

File - training_logs.py

```
"""
Sample model log - Stores information whenever model is trained.
"""

document = {
    "name":"speedfoc",
    "ship_imo":9876543,
    "creation_date": Date("2016-05-18T16:00:00Z"),
    "duration":"3m",
    "model_filename":"speedfoc.pickle",
    "model_url":"aws.s3.xys",
    "training_data":100,
    "training_score":0.87,
}
```