

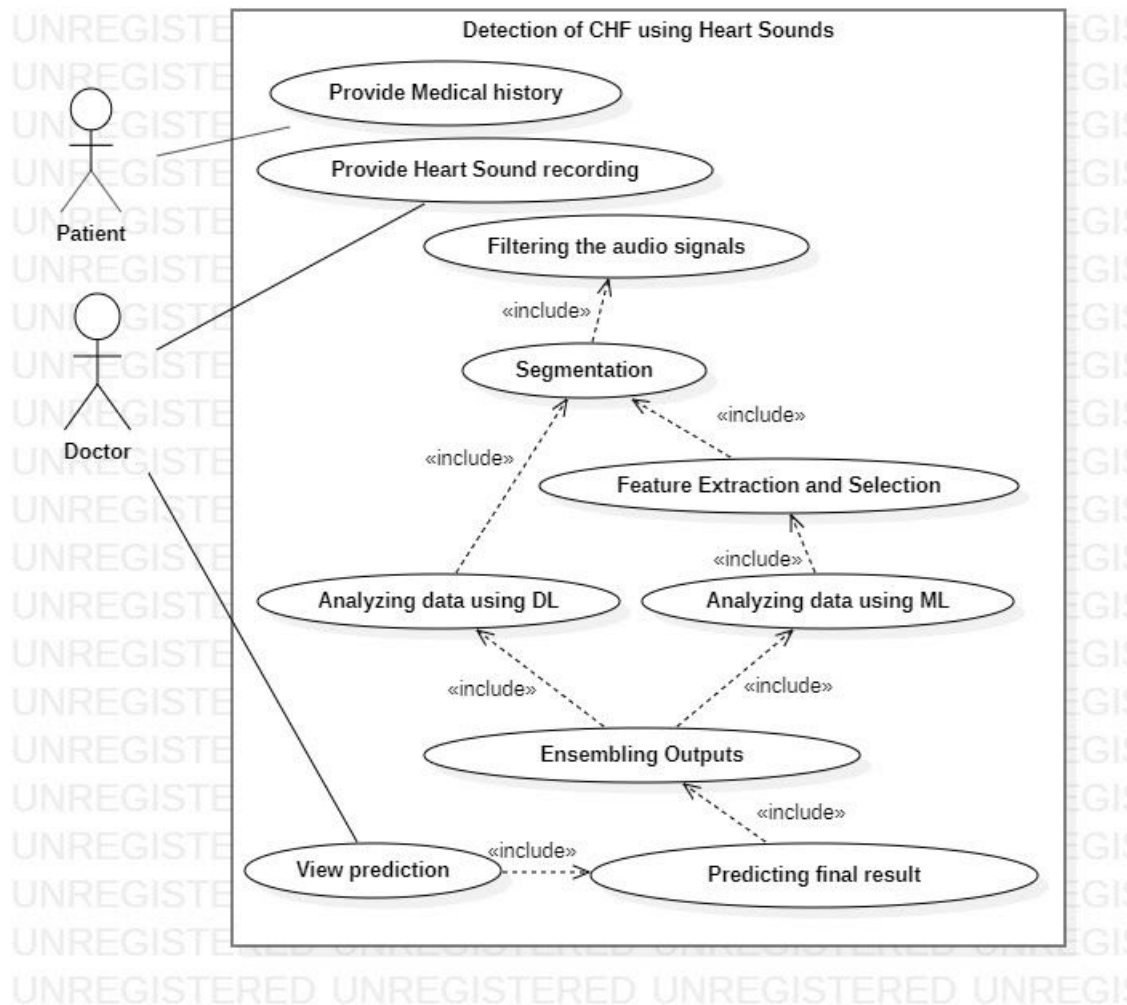
# FINAL SUBMISSION

## GROUP No - 9

Shreya Kajbaje 7062  
Shreya Mahajan 7060  
Riya Parekh 7056  
Rashmi Mokashi 7021

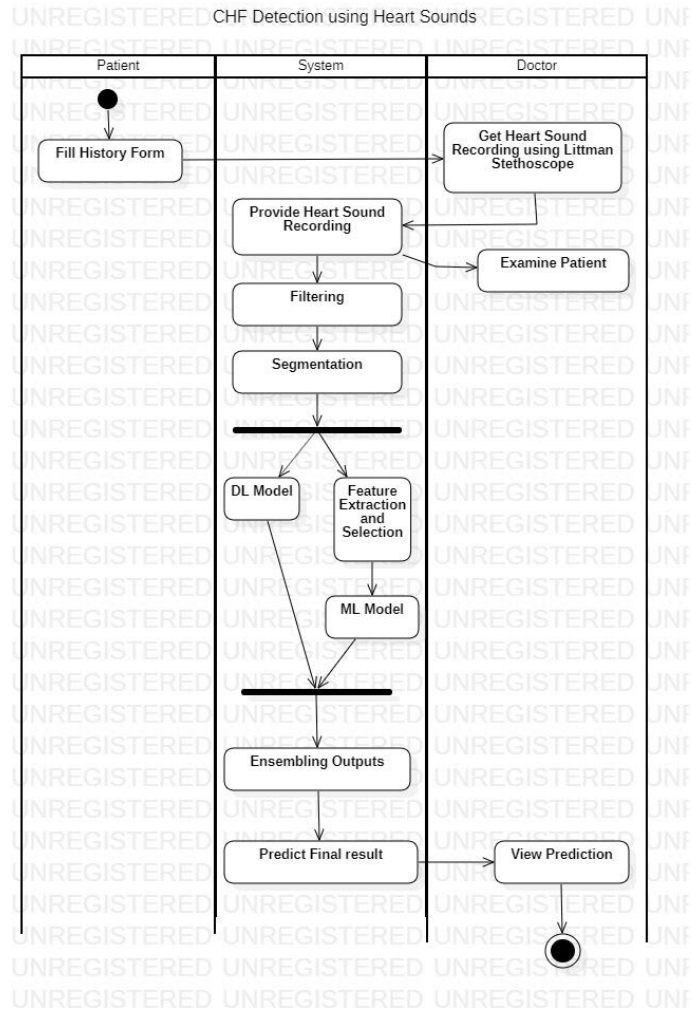
## Assignment 2

Use Case Model -



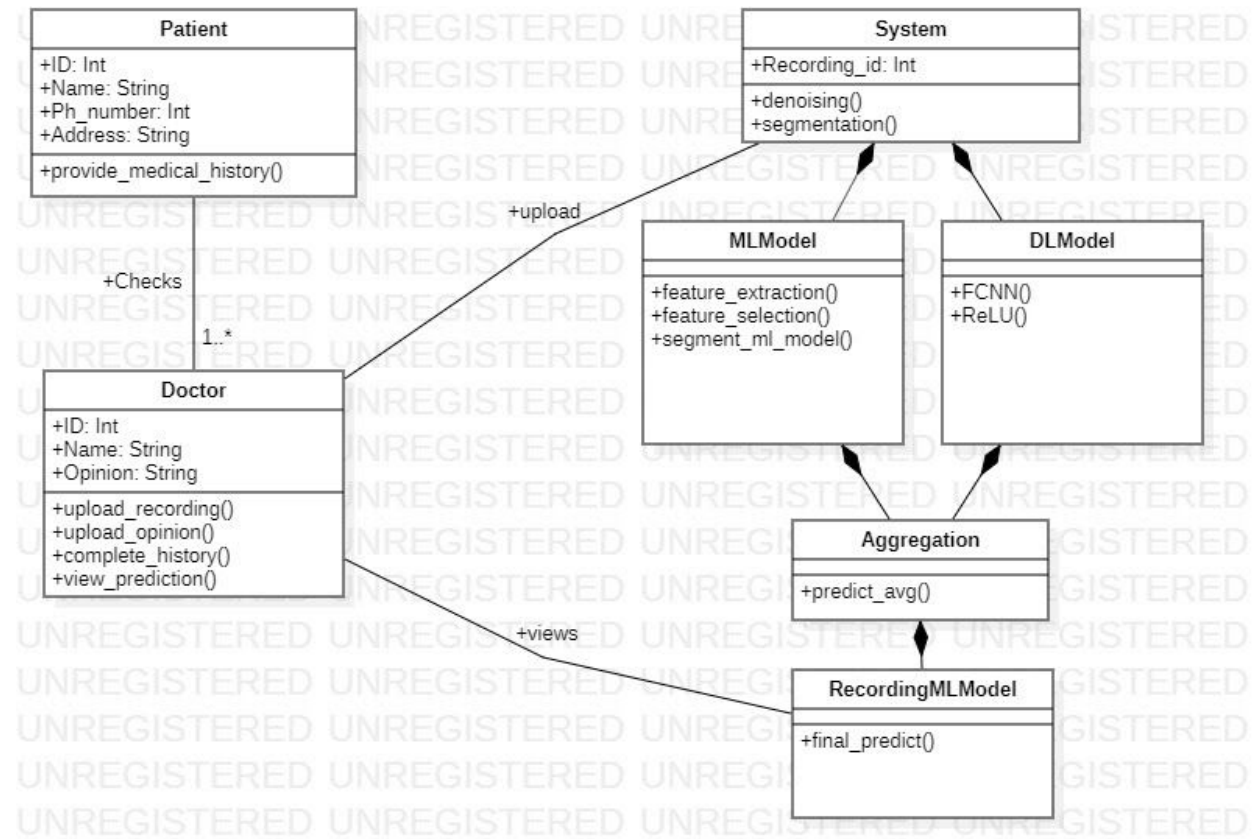
# Assignment 3

## Activity Model -



## Assignment 4 & 5

Class Model -



Code -

```
#include<iostream>
using namespace std;
```

```
class Patient {
public:
    int ID;
    string Name;
    int ph_number;
    string Address;
    Patient(){
        ID = 404;
        Name = "abc";
```

[illegible]

```

    void denoising();
    void segmentation();
    System(){
        cout<<endl<<"Recording ID: 101"<<endl;
    }

};

class MLModel: public System{
    void feature_extraction();
    void feature_selection();
    void segment_ml_model();
};

class DLModel: public System{
    void FCNN();
    void ReLU();
};

class Aggregation: public System{
    void predict_avg(MLModel ml, DLModel dl);
};

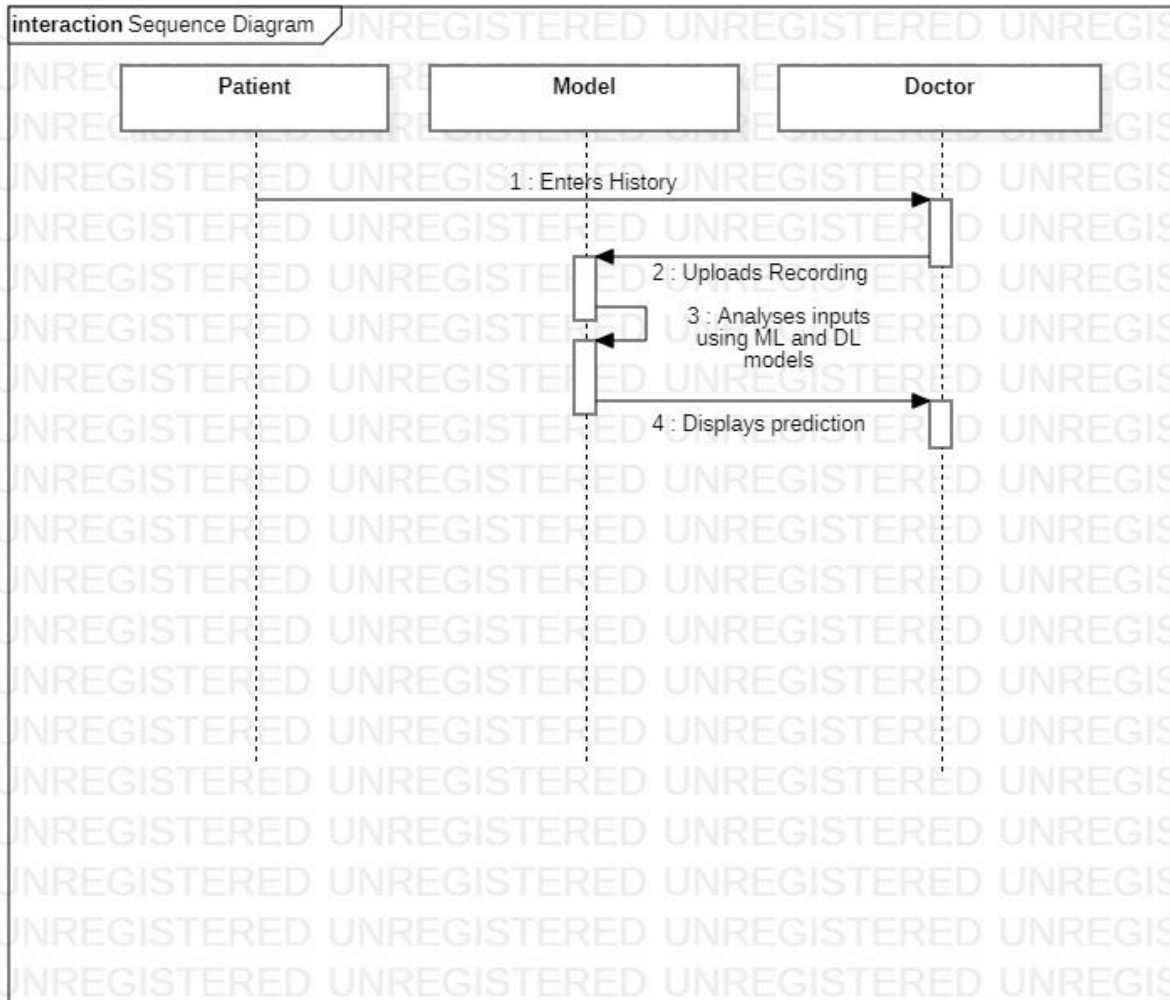
class RecordingMLModel: public System{
    Aggregation aggr;
    void final_predict(Aggregation aggr);
};

int main(){
    cout<<"Welcome!!"<<endl<<endl;
    Doctor d1;
    System s1;
}

```

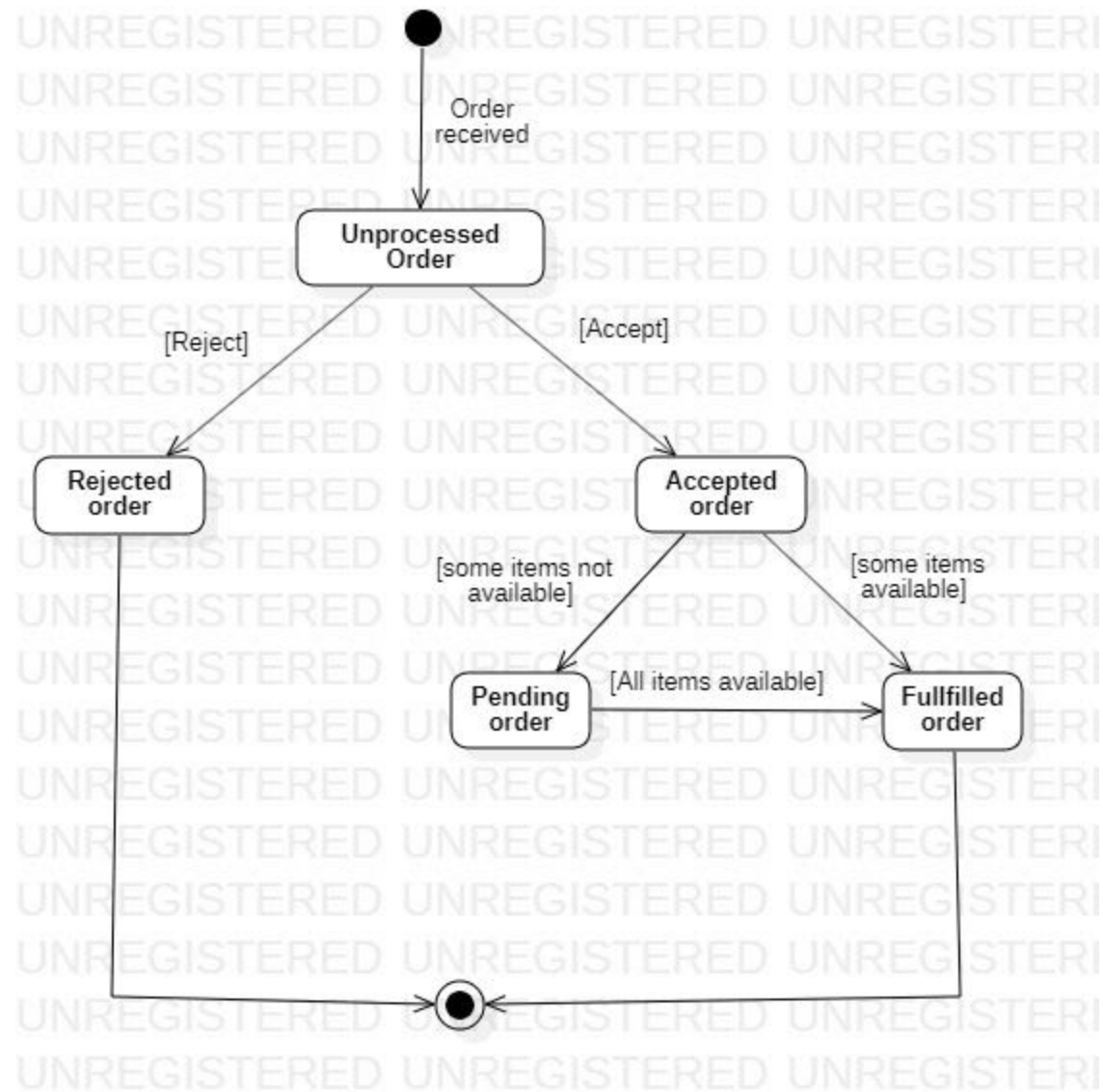
# Assignment 6

Sequence Model -



# Assignment 7

State Model -



Code -

```
#include<iostream>
#include <stdlib.h>
#include<time.h>
#include<math.h>
#define MAX 2
```

```
using namespace std;
```

```
class item{
public:
    int quantity;
    int order_no;
    int available;

    item(){
        quantity = 3;
        srand(time(NULL));
        available = rand() % 2;
    }
    order(int quan){
        if(quantity>0){
            cout<<"Order received"<<endl;
            cout<<"Unprocessed Order"<<endl;

        }
        if(quan<quantity){
            cout<<"Order Accepted!"<<endl;
            if(available==0){
                cout<<"Order Pending!"<<endl;

            }
            while(available==0){
                srand(time(NULL));
                available = rand() % 2;
            }
            if(available==1){
                cout<<"Order Fullfilled!"<<endl;
            }
        }
        else{
            cout<<"Order Rejected!"<<endl;
        }

        cout<<"END STATE!!!"<<endl;

    }
};
```



```

int main(){
    cout<<"START STATE"<<endl;
    item i1;
    i1.order(2);

    return 0;

}

```

/\*

OUTPUT:

PS C:\Users\Manisha\Desktop> ./a.exe

START STATE

Order received

Unprocessed Order

Order Fullfilled!

END STATE!!!

PS C:\Users\Manisha\Desktop> ./a.exe

START STATE

Order received

Unprocessed Order

Order Accepted!

Order Fullfilled!

END STATE!!!

PS C:\Users\Manisha\Desktop> ./a.exe

START STATE

Order received

Unprocessed Order

Order Accepted!

Order Fullfilled!

END STATE!!!

PS C:\Users\Manisha\Desktop> ./a.exe

START STATE

Order received

Unprocessed Order

Order Accepted!

Order Pending!

Order Fullfilled!

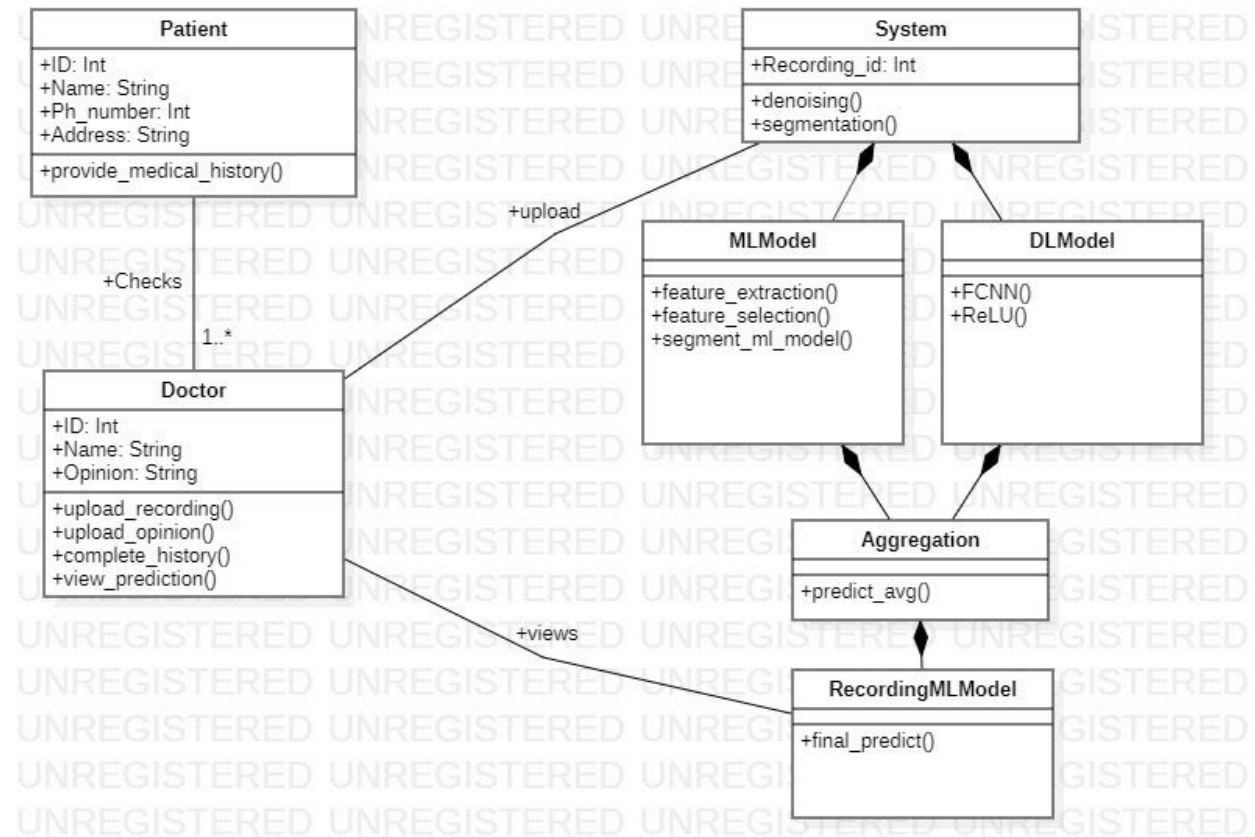
END STATE!!!

PS C:\Users\Manisha\Desktop>

\*/

# Assignment 8

GRASP Pattern -



# Assignment 9

GOF Pattern -

