```
package com.company;

public class System {
    private SecurityState _state;
    private UserCredentials _user;
    private UserFunctionsMacine userMachine;
    public System(UserCredentials user){
        _user = user;
        userMachine = new UserFunctionMachine(user);
        _state = logout.instance(user);
    }
    public void changeState(SecurityState newState){
        _state = newState;
    }
    public UserCredentials getUser(){return _user;}
    public void loginpassword(EncryptedString password){
        if(_state.loginpassword(this,password)) userMachine.loginpassword(password);
    }
    public void logout(){
        if(_state.logout(this)) userMachine.logout();
    }
    public void superuserpassword(EncryptedString superpassword){
        if(_state.superuserpassword(this,superpassword))
userMachine.superuserpassword(superpassword);
    }
    public void connect(){
        if(_state.connect(this)) userMachine.connect();
    }
    public void disconnect(){
        if(_state.disconnect(this)) userMachine.disconnect();
    }
    public void op1(){
        if(_state.op1(this)) userMachine.op1();
    }
    public void op2(){
        if(_state.op2(this)) userMachine.op2();
    }
    public void op3(){
        if(_state.op3(this)) userMachine.connect();
    }
    public void op4(){
        if(_state.op4(this)) userMachine.connect();
    }
}

abstract class SecurityState{
    boolean loginpassword(System controller, EncryptedString password){
        return false;
    }
    boolean logout(System controller){
        return false;
    }
    boolean superuserpassword(System controller, EncryptedString superpassword){
        return false;
    }
    boolean op1(System controller){
        return false;
    }
    boolean op2(System controller){
        return false;
    }
    boolean op3(System controller){
        return false;
    }
    boolean op4(System controller){
        return false;
    }
    void changeState(System controller, SecurityState newState){
        controller.changeState(newState);
    }

}

//logout class
class logout extends SecurityState{
    private static HashMap<UserCredentials, logout> instances = new
HashMap<Usercredentials, logout>();
    public static logout instance(userCredentials user){
        if(!instances.containsKey(user))
            instances.put(user, new Logout(user));
         return instances.get(user);
    }
    private UserCredentials _user;
    private int numFailedLogins;
    public boolean loginpassword(System controller, EncrypttedString password){
        if (controller.getUser().validate(password)) {
            numFailedLogins = 0;
            if (controller.getUser().isUser()) {
                changeState(controller, LoggedInUser.instance(_user));
            } else if (controller.getUser().isSuperUser()) {
                changeState(controller, LoggedInSupperUser.instance(_user));

            } else {
                changeState(controller, LoggedInAdmin.instance(_user));



            }
            return true;
        }
        }else{
        numFailedLogins++;
        if(numFailedLogins >= 5){
            numFailedLogins = 0;
            changeState((controller, Disconnect.instance(_user)));
        }
        return false;
    }
}
//loggedInSuperUser same for login Admin and login User
class LoggedInSupperUser extends SecurityState {
    private static HashMap<UserCredentials, loggedInSupperUser> instances = new
HashMap<Usercredentials, loggedInSupperUser>();
    public static LoggedInSupperUser instance(userCredentials user){
        if(!instances.containsKey(user))
            instances.put(user, new  LoggedInSupperUser(user));
        return instances.get(user);
        private UserCredentials _user;
        private int op3Count;
        public LoggedInSupperUser(UserCredentials user){
            _user = user;
            op3count = 0;
        }
        public boolean logout(System controller){
            changeState(controller,logout.instance(_user));
            return true;
        }
        public boolean op1(system controller){return ture; }
        public boolean op2(system controller){return ture; }
        public boolean op3(system controller){
            op3Count++;
            if(op3Count >50){
                op3Count =0;
                controller.Disconnect();
                return false;
            }
            return true;
        }
    }
}
```