**CONTROLLER**

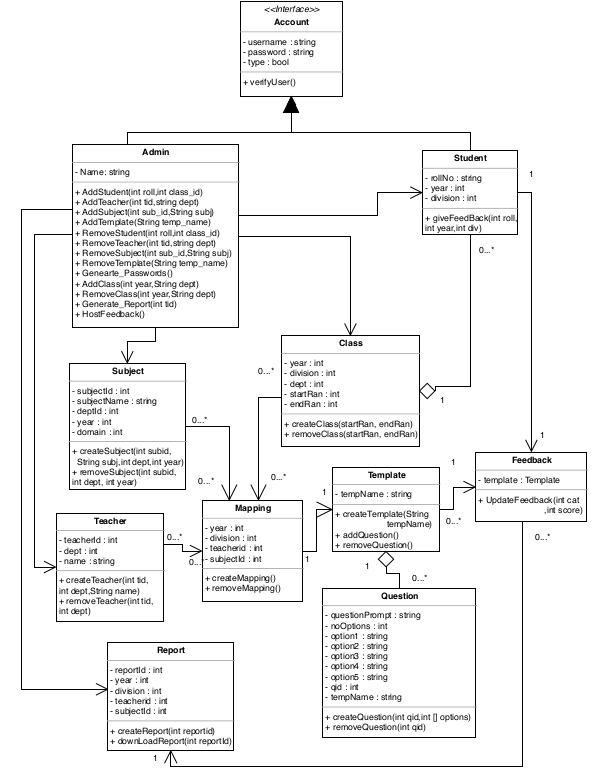
**Problem:**

What first object beyond the UI layer receives and coordinates “controls” a system operation?

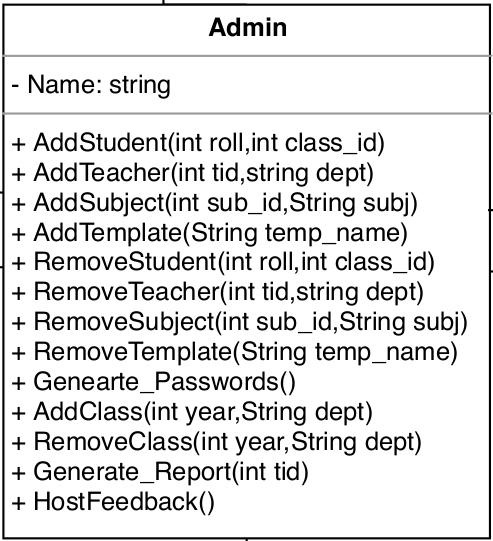
**Solution:**

Assign the responsibility to an object representing one of these choices:  
– Represents the overall “system”, “root object”, device that the software is running within, or a major subsystem (these are all variations of a facade controller)  
– Represents a use case scenario within which the system operation occurs (a use case or session controller)

This principle implementation depends on high level design of our system but general we need always define object which orchestrate our business transaction processing. At first glance, it would seem that the [MVC](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller" \t "/home/devashish/Documents\\x/_blank)Controller in Web applications/API’s is a great example here (even the name is the same) but for me it is not true. Of course it receives input but **it shouldn’t coordinate a system operation** – it should **delegate it** to separate service or Command Handler:



**In this project:**



* Admin class is the controller object.
* It represents the overall “system” or “root object” that the software is running within or a major subsystem
* Admin class is used to handle and connect most other objects.
* It controls objects like Student, Teacher, Subject, Template, Class and Report.
* Admin object is responsible to interact with user and accordingly add, modify or delete the data of the system.
* It is also responsible to carry out some of the crucial operations of the system such as Generate\_Passwords and Generate\_Reports

Admin Object controls the User actions right from the beginning of the system. After Login, user need to fill the data as well as host the feedback. After that Students will give the respective feedback for faculty and then reports are available to view. All of these major actions are controlled by Admin object

**Admin controls Class:**

The admin object is responsible for creation of the class. But after creation, there are some other major activities that the class object performs. These activities are

* Getting mapped with the teacher and subject
* Create students according to the assigned roll number
* Modify any of the contents with respect to class.

**Admin controls Subject:**

The admin object is responsible for creation of the Subject. But after creation, there are some other major activities that the subject object performs. These activities are

* Getting mapped with the teacher and class
* Modify any of the contents with respect to subject.

**Admin controls Teacher:**

The admin object is responsible for creation of the Teacher. But after creation, there are some other major activities that the teacher object performs. These activities are

* Getting mapped with the subject and class
* Modify any of the contents with respect to teacher.

**Admin controls Question:**

The admin object is responsible for creation of the Questions and also Templates. But after creation, there are some other major activities that the Question object performs. These activities are

* Getting mapped with the respective template
* Getting mapped with the teacher class subject mapping
* Modify any of the contents with respect to question and its template.

**Admin controls Other Major operations:**

The admin object is responsible for creation of the Report and Passwords. But after creation, there are some other major activities that the Report object performs. These activities are

* Generating reports
* Creating passwords
* Hosting Feedback
* Modify any of the contents with respect to reports.

**Class Structure in this project:**

// Super class Account (Base class of Student and Admin)

class Account {

// Name, Email and Password

private String username, password;

// Initialize name and email to NULL

Account() {

name = null;

password = null;

}

// Methods to get Name, Email and Password

public String getName() {

/\* Get Name \*/

return null;

}

public String getPassword() {

/\* Get Password \*/

return null;

}

// Protected methods to set Name, Email and password provided

protected void setName(String username) {

/\* Set Name \*/

}

protected void setPassword(String password) {

/\* Set password \*/

}

private boolean isValid(String username, String password) {

if(quantity == NULL || password == NULL)

return false;

return true;

}

public Boolean verifyUser(String username, String password) {

if (isValid(username, passsword))

{

this.username = username;

this.password = password;

return true;

}

else

{

// Error

return false;

}

}

class Admin extends Account {

// Attribute of Admin

private String name;

// Get name of admin

public String getName() {

return null;

}

public void addStudent(int roll,int class\_id) {

/\* add details of student \*/

}

public void addTeacher(int tid,String dept) {

/\* add details of teacher \*/

}

public void addSubject(int sub\_id,String subj) {

/\* add subjects \*/

}

public void addTemplate(String temp\_name) {

/\* add template of questions \*/

}

public void removeStudent(int roll,int class\_id) {

/\* remove student \*/

}

public void removeTeacher(int tid,String dept) {

/\* remove teacher \*/

}

public void removeSubject(int sub\_id,String subj) {

/\* remove subject \*/

}

public void removeTemplate(String temp\_name) {

/\* remove template \*/

}

public void generatePasswords() {

/\* generate passwords \*/

}

protected void addClass(int year,String dept) {

/\* add class \*/

}

public void removeClass(int year,String dept) {

/\* remove class \*/

}

public void generateReport() {

/\* generate passwords \*/

}

}