

## Project Description

Many sectors are benefiting the development in technology. However, the sector of education, especially in our country, is not benefiting the technological developments at the desired level. The goal of our project is to implement a curriculum system which will help remove the troubles and tardiness resulting from too much bureaucratic procedure and from the seriousness of the whole process and which will bring significant efficiency gains. That is why we chose the name SmartPlan for our project.

The results those will occur in this case can be observed at least in one semester. One of the main gainings of our system is just becomes significant at this point and makes the educators able to foresee the system they constructed. A system that will be able to show what a changed course topic matches among a range of standards or not, and in addition to these, that will be able to make suggestions when a new course topic will be entered by benefiting the data entered to the system before will significantly reduce the workloads of educators. Another basic improvement of our system is that it will remove a serious loss of efficiency by removing the troubles of education system which very human based. Namely, in the case there is a change in the course plan, the change is confirmed by the convention of commissions for which averagely 15 days are passed and this leads to a serious time hazard. In applications waiting for approval, our system ties the approving and approved levels of the hierarchy, confirmation is achieved by a single mouse click removing this serious inefficiency.

Also, the students are subjected to a quiz at the end of the semester, and the ratings of them are archived in the system.

Briefly, the system frees the users from processes requiring bureaucratic confirmations, and physical effort, constructs a standard in education; it makes the results of the plan, material and topic modifications predictable, so it removes many handicaps of the current education system.

### Using the MultiView and View controls

A large percentage of the code in a typical application is devoted to validating the input received from the user. ASP.NET dedicates five server controls to making this step more manageable: RequiredFieldValidator, CompareValidator, RangeValidator, RegularExpressionValidator, and CustomValidator. These validation controls all inherit from the abstract BaseValidator class, which is derived from the Label class. An additional control can be used in conjunction with the validators, ValidationSummary, to present the set of errors in a cohesive and consistent manner.

Each of these validation controls has a ControlToValidate property used to set the ID of the target control. When the form is posted, the Validate method is called automatically; when validation fails, the ErrorMessage text is displayed. For some browsers, the validation controls are able to take advantage of client-side script to perform the validation check and display an error message

without completing the postback.

**Tip:** If necessary, you can disable client-side validation by using the Page directive's ClientTarget attribute. Setting this directive to downlevel causes ASP.NET to treat the browser as though it does not support JScript.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ITEXT SYSTEM "http://www.lowagie.com/iText/itext.dtd">
<itext creationdate="Fri Aug 17 13:15:42 CEST 2001" producer="iTextXML by
lowagie.com">
```

*The Display property is used to indicate whether space should be reserved in the rendering for the error message. It can be set to one of the two ValidatorDisplay enumeration values: Static or Dynamic. The Static setting, where space is allocated but not immediately filled, works only when the client supports client-side scripting and the Page directive's ClientTarget attribute has not been set to downlevel.*

From time to time, it's useful to gather controls together in several panes and give the user the opportunity to page through the panes. During the lifetime of ASP.NET 1.0, Microsoft released several rich dynamic (though officially unsupported) controls that emitted DHTML instead of regular HTML. A trio of these controls, the TabStrip, the MultiView (an older version), and the PageView, worked together to form essentially a set of tabbed panes.