

2008

ACMESOFT

SmartPlan

Software Design Specification

Prepared by :

Kadir Kaan Evren

Mustafa Çağlar Yavuz

Candan Bölükbaş

Sami Akkuş



1	INTRODUCTION	5
1.1	The Scope of the Document	5
1.2	Project Information	5
1.3	A Brief Summary of the Work Done So Far	6
1.4	Design Objectives	6
1.5	Future Work	7
2	DESIGN CONSTRAINTS.....	7
2.1	Time Constraints.....	7
2.2	Hardware Constraints.....	7
2.3	System Constraints	7
2.4	Software Constraints	7
2.5	User Interface Constraints.....	8
2.6	Performance Constraints.....	8
3	SYSTEM MODULES.....	8
3.1	Login Module:.....	8
3.2	Registration & Role Management Module:	8
3.3	Application Initializer Module:	8
3.4	Database Module:	9
3.5	Course Management Module:	9
3.6	Warning Module:.....	9
3.7	Ranking Module:	9
3.8	Printing Module:.....	9
3.9	Interface Module:.....	9
3.10	Administration Module:	10
3.11	Versioning Module:	10
3.12	Poll Module:	10
4	MODELLING	10

4.1	Class Diagrams	10
4.2	Sequence Diagrams	15
5	DATA DESIGN.....	17
5.1	ER DIAGRAM.....	18
5.2	TABLE SCHEMAS	18
5.2.1	Table: tblServers.....	18
5.2.2	Table: tblOrganizations	19
5.2.3	Overview.....	20
5.2.4	Table: tblApplications.....	21
5.2.5	Table: tblRoles	22
5.2.6	Table: tblUsers.....	24
5.2.7	Table: tblUserRoles.....	25
5.2.8	Table: tblPrivileges	26
5.2.9	Table: tblAbstractGoals	27
5.2.10	Overview.....	27
5.2.11	Table: tblCourseObjectives.....	28
5.2.12	Overview.....	29
5.2.13	Table: tblCourses	30
5.2.14	Overview.....	30
5.2.15	Table: tblCourseTopics	31
5.2.16	Overview.....	31
5.2.17	Table: tblCourseTopicMaterials	33
5.2.18	Overview.....	33
5.2.19	Table: tblRanking	34
5.2.20	Overview.....	34
5.2.21	Table: tblCoursePoll.....	35
5.2.22	Overview.....	36

5.2.23	Table: tblCoursePoll.....	37
5.2.24	Overview.....	37
5.2.25	Table: tblCourseQuestions	38
5.2.26	Overview.....	38
5.2.27	Table: tblCourseQuestionChoices	39
5.2.28	Overview.....	39
5.2.29	Table: tblPollUserChoices.....	40
5.2.30	Overview.....	40
6	ARCHITECTURAL DESIGN	41
7	GUI DESIGN.....	51
7.1	General view.....	51
7.2	Adding new item	52
7.3	Updating an item.....	53
7.4	Listing items.....	54
8	DEVELOPMENT SCHEDULE	55
9	SOFTWARE AND HARDWARE REQUIREMENTS	55
9.1	Software Requirements.....	55
9.2	Hardware Requirements	56
10	TESTING STRATEGIES.....	56
10.1	Unit Testing	56
10.2	Integration Testing	56
10.3	Pilot Testing.....	57
10.4	Security Testing	57
10.5	Validation Testing.....	57

1 INTRODUCTION

1.1 *The Scope of the Document*

This report is the initial design report of ACMESOFT. In this report, firstly we mentioned the project description, what we have done so far, and objectives of our design. In the second part we explained the design constraints. In the third part we divided the pro

ject into modules, including classes and functions. We explained each module in detail. Then, in order to show the modules clearly we used class diagrams and sequence diagrams. In the data design part, in order to show the structure of the database tables we used ER diagrams. Before it we described the data objects. Also we prepared a data dictionary to list our data in the database. We divided the design into two parts, architectural and GUI design which is the web interface. We added screenshots showing our GUI design to the report.

1.2 *Project Information*

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.

In the mechanism of educational institutions, education programs are constructed based on abstract goals, application of these are based on course materials which can be seen as concrete goals, and course materials and topics together make up the course plan. Today, in educational institutions curriculums do not change often, while the course plans change very often. 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.

Moreover, our system does not delete any information entered; it archives all the entered information. The educators can take advantage of the archived course content, and the plans and materials of previous years. The course materials, topics and plans are given points by educators, so the users form regularity among all of them. 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.

1.3 A Brief Summary of the Work Done So Far

After the project topic was assigned to our group we gathered in regular and irregular meetings. We searched the internet to understand and realize the project better and we investigated the similar applications. We tried to understand the mapping concept.

During and before the preparation of the requirement analysis report, we worked on how to prepare use case, class, sequence and ER diagrams. We began to learn about the technology and software we will use in the project. We clearly described the use cases of the project. The preparation phase of the requirement analysis report improved very much our knowledge.

1.4 Design Objectives

Usability

Usability is very important in all web-based software projects. Since we have many features and processes for the users, to display these in the user interface is very important for the users' ease. Usability increases the admiration of the users.

Security

First of all, our system as a web-based one, keeps the user information in the database, which is not accessible to any user other than administrators. There is a password-login system, and administrators have the right to manage the system database, shortly they have full control over the whole system.

The other security issue is the security of server. The server machine should be in a secure environment, always open, and in the control of administrators.

1.5 Future Work

From now, we will complete our project prototype we have already started and we will show it at the end of the semester. During and after the preparation of the prototype we will get together and decide on what to do during the semester holiday and in the new semester.

At the beginning of the next semester we will start the implementation of our project. All our members are already able to use the specified software development tools for our project. Our periodical meetings as group and the spontaneous gatherings will continue in the next semester.

We plan to implement firstly our database and then our modules and the user interface of the system. When we complete the implementation of all modules, we will use unit testing to test each modules. After the integration we will apply integration testing, and when the product is almost ready we will apply pilot, security and validation testing strategies. At the end of this process our system will be ready to use and we will make its demonstration at the end of the spring semester.

2 DESIGN CONSTRAINTS

Below are the main design constraints of our project :

2.1 Time Constraints

Each phase of our project is scheduled according to strict deadlines. We have a month to finish the detailed design and also the prototype. The time factor is very determinant in our work, that is we determine the workload according to time.

2.2 Hardware Constraints

The main hardware constraint in our project is that we will need a server machine with Windows 2000/2003 Server, .NET Framework, EES, SQL Server, AJAX Control Toolkit installed in it. We will need the server machine at the second semester.

2.3 System Constraints

Our project is a web based system that means the security issues are very important for us. Because, there will be database connections, the users' information will be editable and kept in the database. To overcome the problems resulting from security system should be maintainable.

2.4 Software Constraints

We will use ASP .NET for both the architectural design and GUI design in our project. We chose it, because it provides us the functionality with its many useful libraries.

2.5 User Interface Constraints

Our web-based system has a GUI part. The user interface must be easy to use, has a high visual quality, and organized according to the users' ease.

2.6 Performance Constraints

Since our web-based system has a huge database, we will probably face the performance constraints which the similar software projects, highly depending on database operations face also. A database which hundreds of users access at the same time should be constructed professionally. We will concentrate on the problem during our implementation.

3 SYSTEM MODULES

3.1 Login Module:

This module is the one controlling the entrance of users to the system, if the user is authorized to enter to the system this is the one setting the sessions belonging to that user and other necessary sessions. Users must have a password given by the administrator in order to enter the system. The user name is unique and also in the case that user forgets his/her password the password reminding mechanism will be made active. The uniqueness of username and reminding password is being handled via e-mail confirmation system.

3.2 Registration & Role Management Module:

This is the module in which the admin or a user authorized by admin registers new users and in which the authorization of the users is arranged. Because there is no static user type exists in the desired system we had to design all system dynamically which resulted in fully dynamic roles with relevant privileges and authorizations. This module can move the user to the different levels of the hierarchy by assigning him special authorization as well as it can organize the hierarchy in an envisioned tree structure. This module takes our system one step forward from the classical static hierarchy structure. In addition to these, roles specific to user can take place as well as the roles designed for group.

3.3 Application Initializer Module:

This module is the module serving to users the rights provided by role management module and applications related to user after they log in to the system. It decides whether to initialize an application when user wants to access it by using the `getPrivilege` method which controls whether the user has right to access the application she selected.

3.4 Database Module:

This is the module managing all the processes related to database. This is the module handling the opening of a connection, execution of a query taken from another query, execution of stored procedures and all other similar processes.

3.5 Course Management Module:

This is the module organizing the hierarchy between an abstract objective and a course. In this module an authorized user manages the process till the preparation of course plan and course material after he/she enters a discrete and abstract objective to the system. In this process whether the course plan and course materials are proper to the standards those stated in the objectives is checked, and the warning module is informed if necessary.

3.6 Warning Module:

This takes a modification in any entry from the lowest to the highest level in the system from the course management module, and gives warning about the nodes where the modification occurred and the ones related to that node.

3.7 Ranking Module:

This is the module evaluating the application success of course curriculum and course materials, providing feedbacks and statistics. In this module teachers or students give points to the course materials and the form of lecturing. By rating the nodes in the lower level of hierarchy, the average rank of the higher level nodes are automatically obtained.

3.8 Printing Module:

This module is the one preparing the outputs in pdf format which the user obtains from the applications within her authorization. For example, after a course material is entered to system this material or any desired course material may be wanted to printout. This module gives printout to user in certain templates.

3.9 Interface Module:

Since there will be more than one organization and all organizations will have a specific logo and stil design, the system needs this module. When the web application of an organization is called this module finds the related interface and then loads.

3.10 Administration Module:

This is the module setting the system first time, performing the necessary adjustments, owning all the methods able to access all kind of data on the server, and creating and arranging user files and folders.

3.11 Versioning Module:

This is the module providing the filing of the nodes of any level which are affected by the changes occurring any time in the system and that these nodes are kept accessible when desired to use. This module, by different filtering ways provides access to versioned nodes for users and it suggests similar information about the topic.

3.12 Poll Module:

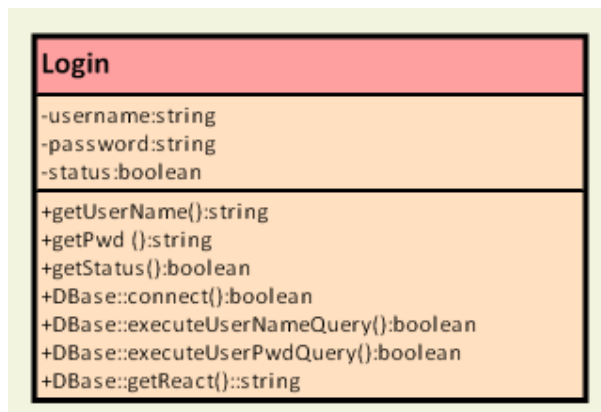
This module owns all necessary methods to design a poll and also edit it. Users can take polls via this module which directly interacts with versioning module that generates archives automatically. Because of the usual audience of polls are students the style of polls generally will be similar to nowadays pink colored end semester evaluation forms.

4 MODELLING

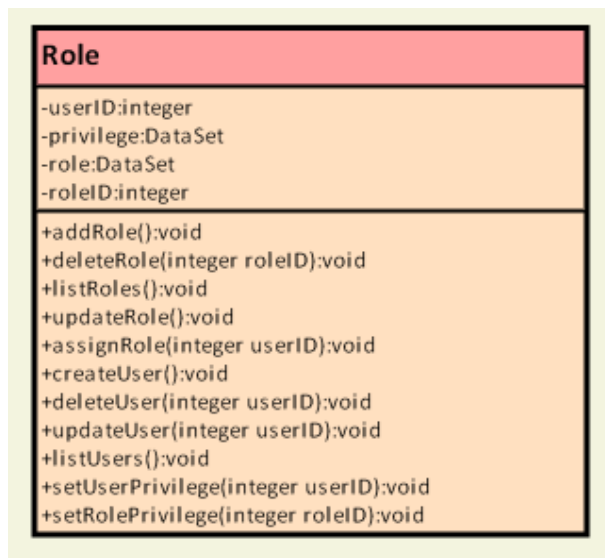
This part involves relevant class and sequence diagrams. The class diagrams that we have designed via divide and conquer technique will facilitate implementation phase. Meanwhile the sequence diagrams are designed for just some basic activities of our system should handle.

4.1 Class Diagrams

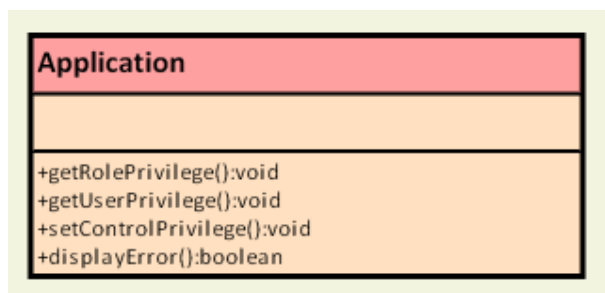
- Login Module



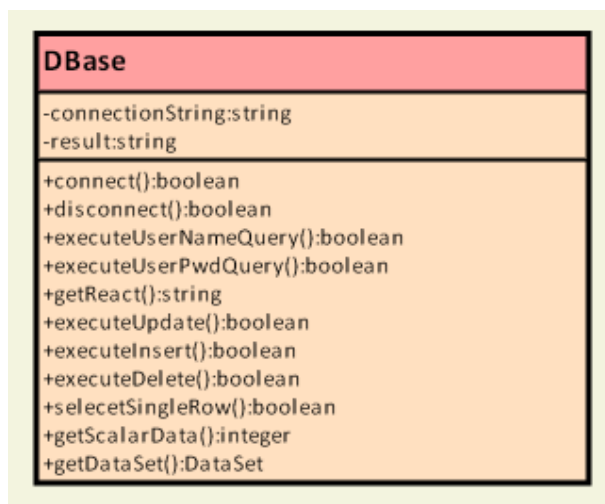
- Registration & Role Management Module



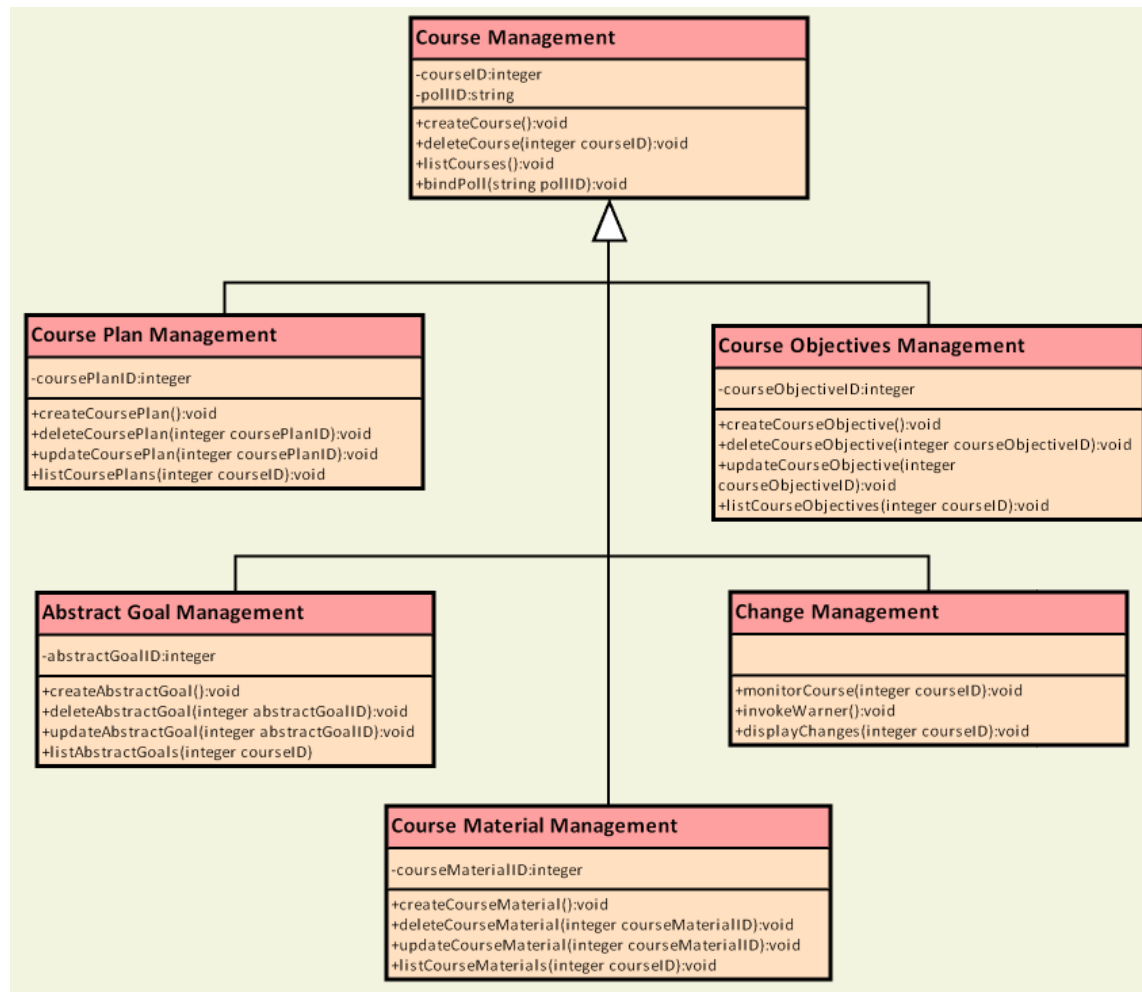
- Application_INITIALIZER Module



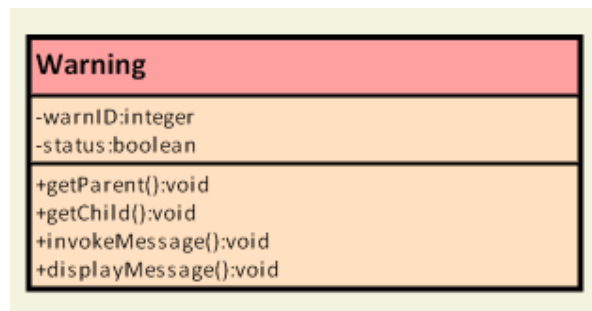
- Database Module



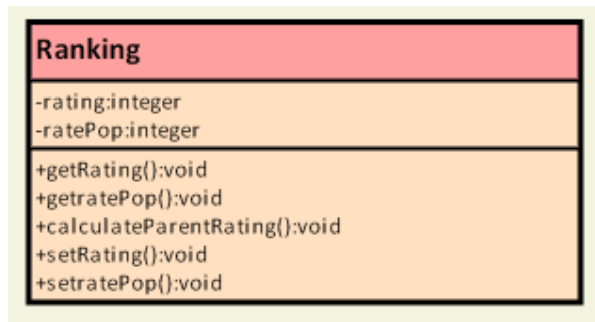
- Course Management Module



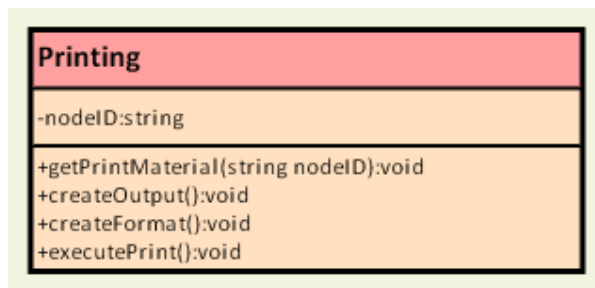
- Warning Module



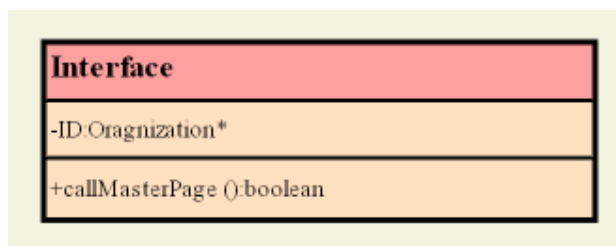
- Ranking Module



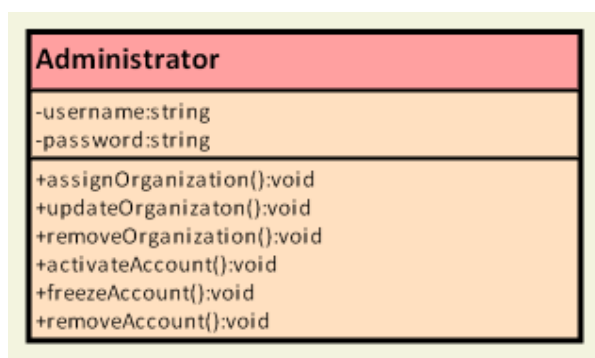
- Printing Module



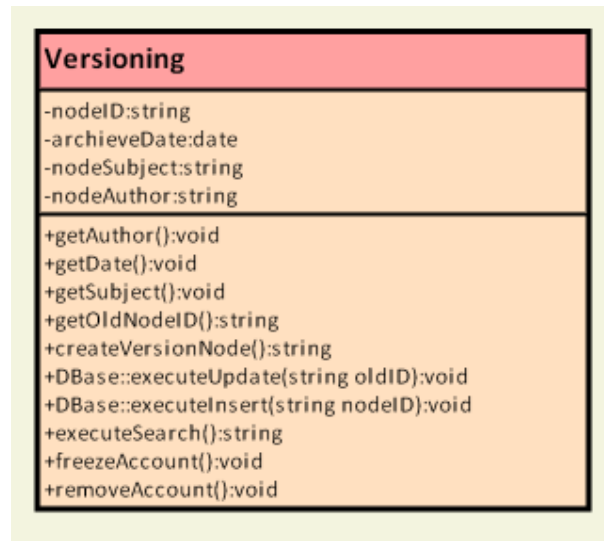
- Interface Module



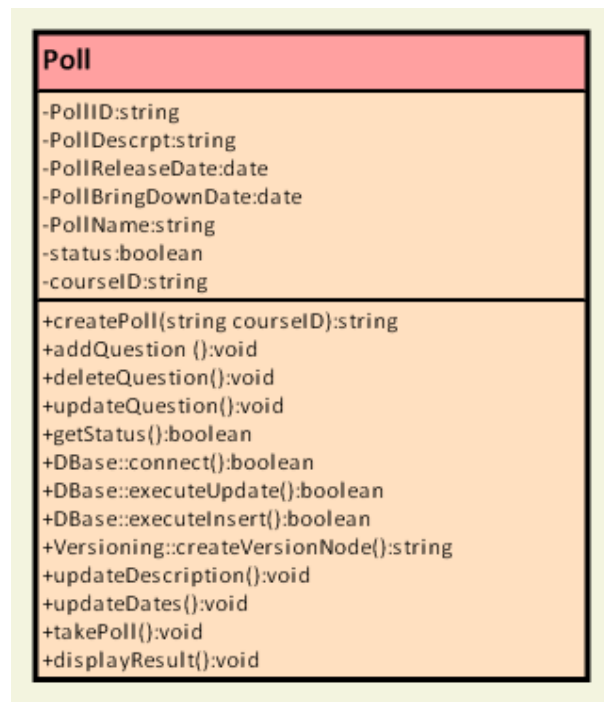
- Administration Module



- Versioning Module



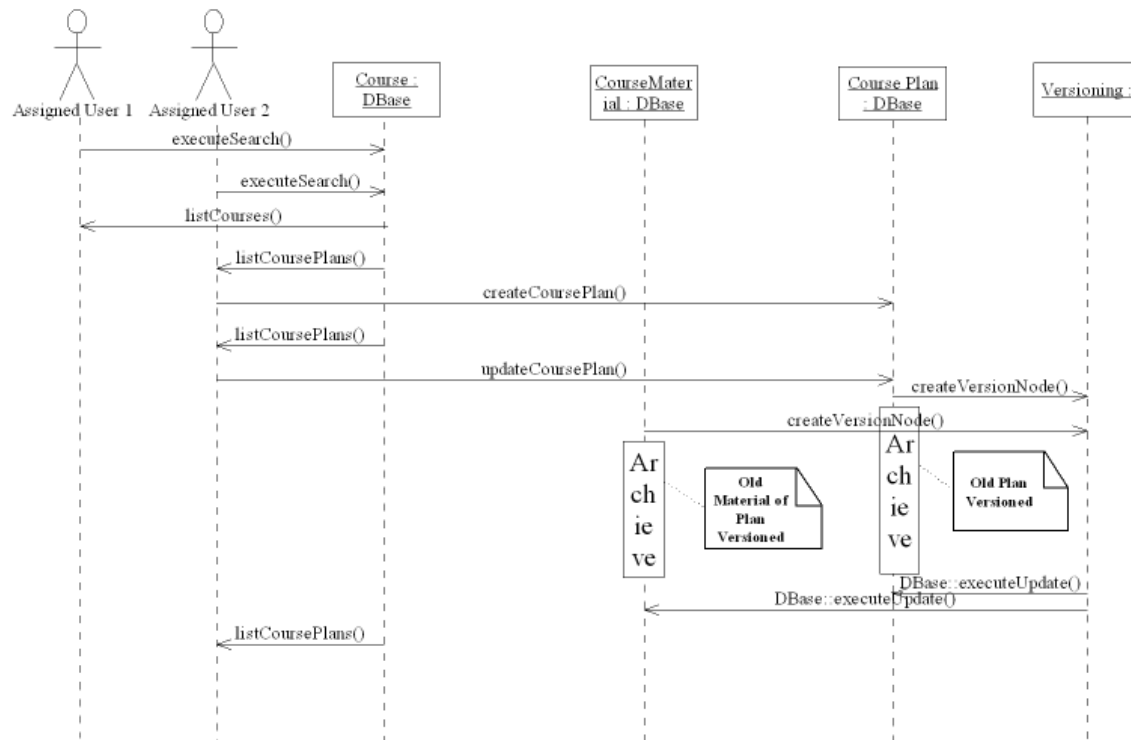
- Poll Module



4.2 Sequence Diagrams

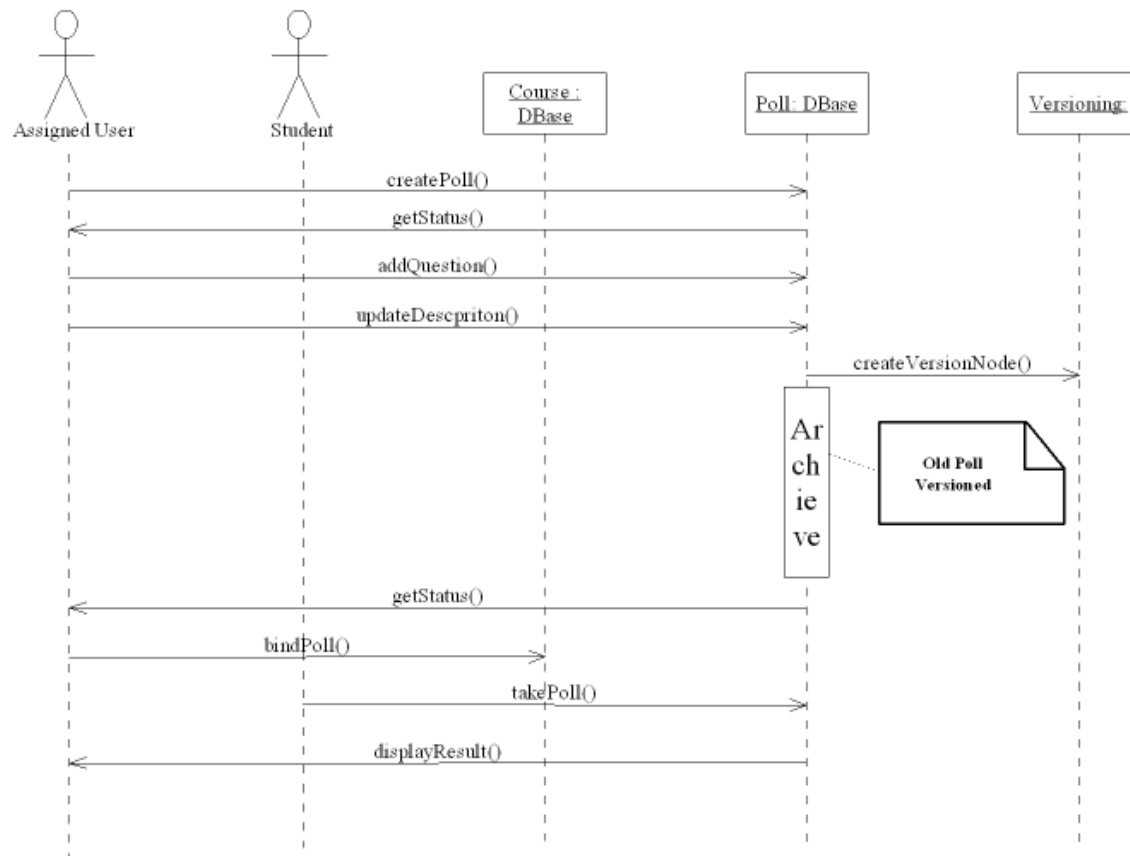
Addition of Course and Course Plan Activity

Course Addition and Course Plan Addition and Update with Relevant Course Material



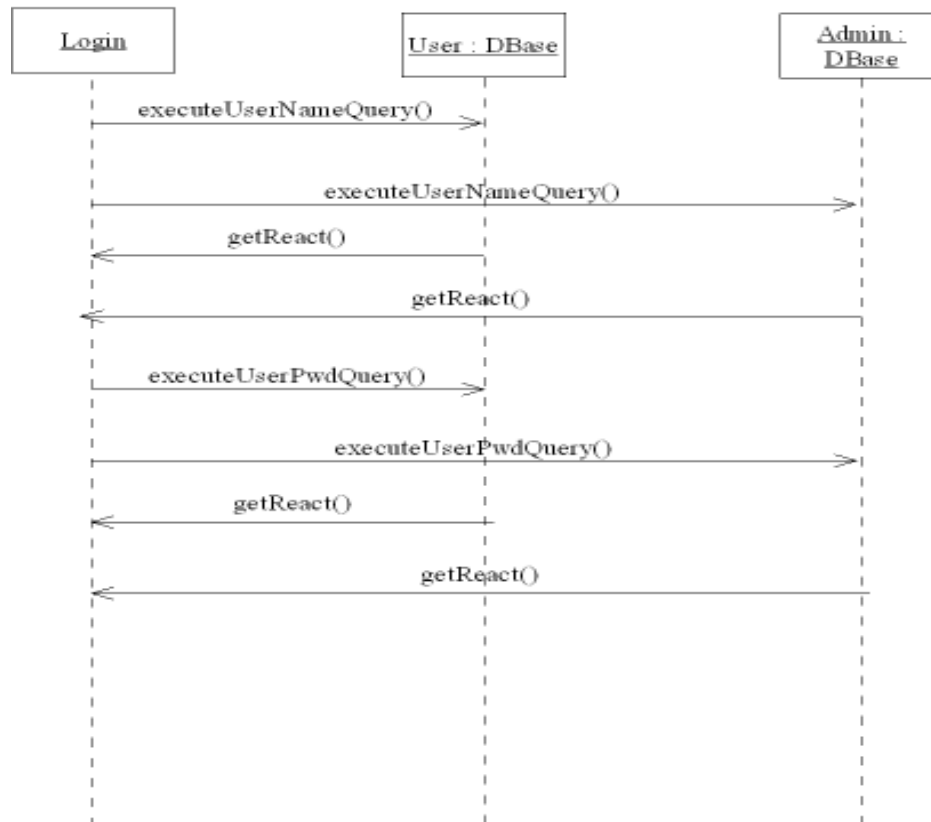
The Case of Creating and Taking Poll

Creating Poll and Taking Poll



The Login Activity

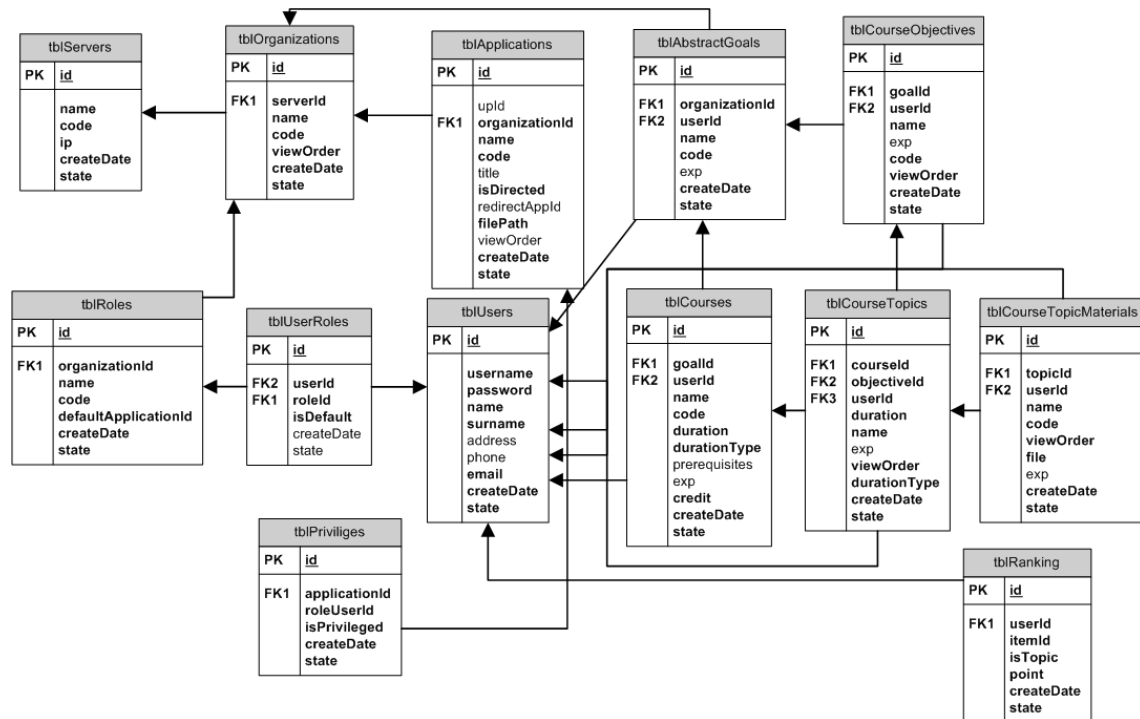
System Login

**5 DATA DESIGN**

We have 13 database objects manipulated by our software.

- tblServers
- tblOrganizations
- tblRoles
- tblUsers
- tblRoleUsers
- tblPrivileges
- tblApplications
- tblAbstractGoals
- tblCourseObjectives
- tblCourses
- tblCourseTopics
- tblCourseTopicMaterials
- tblRanking

5.1 ER DIAGRAM



5.2 TABLE SCHEMAS

5.2.1 Table: tblServers

tblServers	
PK	<u>id</u>
	name
	code
	ip
	createDate
	state

5.2.1.1 Overview

We use this table to store information related to the our server configuration. Each organization will be bound by a server.

5.2.1.2 Data Type Definition

Field Name	Data Type
id	GUID
name	VARCHAR(50)
code	VARCHAR(50)
ip	CHAR(15)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.1.3 Description

Field Name	Description
id	Unique identifier
name	Server name
code	Unique code
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.2 Table: tblOrganizations

tblOrganizations	
PK	<u>id</u>
FK1	serverId name code viewOrder createDate state

5.2.3 Overview

This table is necessary for storing the information related to organizations. Organization can be any educational institution e.g. METU, Hacettepe etc.

5.2.3.1 Data Type Definition

Field Name	Data Type
id	GUID
serverId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
viewOrder	INTEGER
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.3.2 Description

Field Name	Description
id	Unique identifier
serverId	Stores the related serverId
name	Server name
code	Unique code
viewOrder	The organizations will be ordered according to this number in ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.4 Table: tblApplications

tblApplications	
PK	<u>id</u>
FK1	upld organizationId name code title isDirected redirectAppld filePath viewOrder createDate state

5.2.4.1 Overview

In order to make our software flexible we make our table structure as dynamic as it can. We use this table to load and store our applications dynamically.

5.2.4.2 Data Type Definition

Field Name	Data Type
id	GUID
upld	VARCHAR(40)
organizationId	VARCHAR(40)
Name	VARCHAR(50)
Code	VARCHAR(50)
Title	VARCHAR(200)
isRedirected	TINYINT
redirectAppld	VARCHAR(40)
filePath	VARCHAR(500)
viewOrder	INTEGER
createDate	DATETIME: DEFAULT GETDATE()

State	TINYINT: DEFAULT 1
-------	--------------------

5.2.4.3 Description

Field Name	Description
id	Unique Identifier
upId	Parent application id
organizationId	Organization, the application related to it
name	Organization name
code	Unique code related to the organization
title	Header of the application
isRedirected	An application can be redirected to another application if it is set to 1
redirectAppld	Redirected application Id
filePath	Our system will load application dynamically. It is the ascx file path which our application initializer will load it
viewOrder	The applications will be ordered according to this number in the ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.5 Table: tblRoles

tblRoles	
PK	<u>id</u>
FK1	organizationId name code defaultApplicationId createDate state

5.2.5.1 Overview

Administrator of the system or a user assigned by administrator can create roles dynamically and assign privileges to new role. We store the role information in this table.

5.2.5.2 Data Type Definition

Field Name	Data Type
id	GUID
organizationId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
defaultApplicationId	VARCHAR(40)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.5.3 Description

Field Name	Description
id	Unique Identifier
organizationId	Organization Id
name	Role name
code	Unique role code
defaultApplicationId	Each role can have privileges to many applications. When the user log in the system, the system must know the default application.
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.6 Table: tblUsers

tblUsers	
PK	<u>id</u>
	username
	password
	name
	surname
	address
	phone
	email
	createDate
	state

5.2.6.1 Overview

This table stores information related to the user. We designed the system thinking that each user can work for different organization.

5.2.6.2 Data Type Definition

Field Name	Data Type
id	GUID
username	VARCHAR(50)
password	CHAR(8)
name	VARCHAR(50)
surname	VARCHAR(50)
address	VARCHAR(200)
phone	CHAR(10)
email	VARCHAR(100)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.6.3 Description

Field Name	Description
id	Unique Identifier
username	Username of the user to log in the system
password	8 characters password required to log in the system
name	User real name
surname	User's surname
address	Address of user
phone	Phone of user
email	Email of user
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.7 Table: tblUserRoles

tblUserRoles	
PK	<u>id</u>
FK2 FK1	userId roleId isDefault createDate state

5.2.7.1 Overview

We use this table to associate each user with some roles.

5.2.7.2 Data Type Definition

Field Name	Data Type
id	GUID
userId	VARCHAR(40)
roleId	VARCHAR(40)

isDefault	TINYINT
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.7.3 Description

Field Name	Description
id	Unique Identifier
userId	User Id
roleId	Role Id
isDefault	When the user enters the system, the system must know its default role.
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.8 Table: tblPrivileges

tblPrivileges	
PK	<u>id</u>
FK1	applicationId roleUserId isPrivileged createDate state

5.2.8.1 Overview

This relation assigns application privileges to each user or role.

5.2.8.2 Data Type Definition

Field Name	Data Type
id	GUID
applicationId	VARCHAR(40)
roleUserId	VARCHAR(40)

isPrivileged	TINYINT
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.8.3 Description

Field Name	Description
id	Unique Identifier
applicationId	Application Id
roleUserId	Role or user Id
isPrivileged	If this value is one, the user has the privilege to access the application
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.9 Table: tblAbstractGoals

tblAbstractGoals	
PK	<u>id</u>
FK1	organizationId
FK2	userId
	name
	code
	exp
	createDate
	state

5.2.10 Overview

This object contains the assigned users' or organization's abstract goal and related information.

5.2.10.1 Data Type Definition

Field Name	Data Type
id	GUID
organizationId	VARCHAR(40)

userId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
exp	VARCHAR(1000)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.10.2 Description

Field Name	Description
id	Unique Identifier
organizationId	Organization Id
userId	The id of user which inputs the data
name	Goal name
code	Goal code
exp	Explanation
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.11 Table: tblCourseObjectives

tblCourseObjectives	
PK	<u>id</u>
FK1	goalId
FK2	userId
	name
	exp
	code
	viewOrder
	createDate
	state

5.2.12 Overview

This object has a relationship with tblAbstractGoals.

5.2.12.1 Data Type Definition

Field Name	Data Type
id	GUID
goalId	VARCHAR(40)
userId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
exp	VARCHAR(1000)
viewOrder	INTEGER
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.12.2 Description

Field Name	Description
id	Unique Identifier
goalId	Organization Id
userId	The id of user which inputs the data
name	Goal name
code	Goal code
exp	Explanation
viewOrder	Course objectives will be ordered according to this number in the ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.13 Table: tblCourses

tblCourses	
PK	<u>id</u>
FK1 FK2	goalId userId name code duration durationType prerequisites exp credit createDate state

5.2.14 Overview

This object has a relationship with tblAbstractGoals.

5.2.14.1 Data Type Definition

Field Name	Data Type
id	GUID
goalId	VARCHAR(40)
userId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
prerequisites	VARCHAR(8000)
exp	VARCHAR(1000)
duration	VARCHAR(3)
durationType	VARCHAR(40)
viewOrder	INTEGER
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.14.2 Description

Field Name	Description
id	Unique Identifier
goalId	Organization Id
userId	The id of user which inputs the data
name	Course name
code	Course code
prerequisites	Course prerequisite ID list
exp	Explanation
duration	The course duration
durationType	Month, Week, Day, Hour
viewOrder	Course will be ordered according to this number in the ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.15 Table: tblCourseTopics

tblCourseTopics	
PK	<u>id</u>
FK1	courseId
FK2	objectiveId
FK3	userId
	duration
	name
	exp
	viewOrder
	durationType
	createDate
	state

5.2.16 Overview

Each Course consists of some topics. We use this table to store the topics related to course topics.

5.2.16.1 Data Type Definition

Field Name	Data Type
id	GUID
courseId	VARCHAR(40)
userId	VARCHAR(40)
objectiveId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
exp	VARCHAR(1000)
duration	VARCHAR(3)
durationType	VARCHAR(40)
viewOrder	INTEGER
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.16.2 Description

Field Name	Description
id	Unique Identifier
courseId	Course Id
userId	The id of user which inputs the data
objectiveId	Objective Id
name	Topic name
code	Topic code
exp	Explanation
duration	The topic duration
durationType	Month, Week, Day, Hour

viewOrder	Topics will be ordered according to this number in the ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.17 Table: tblCourseTopicMaterials

tblCourseTopicMaterials	
PK	<u>id</u>
FK1	topicId
FK2	userId
	name
	code
	viewOrder
	file
	exp
	createDate
	state

5.2.18 Overview

This table stores the course materials related to topic.

5.2.18.1 Data Type Definition

Field Name	Data Type
id	GUID
topicId	VARCHAR(40)
userId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
file	VARCHAR(500)
exp	VARCHAR(1000)
viewOrder	INTEGER

createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.18.2 Description

Field Name	Description
id	Unique Identifier
topicId	Topic Id
userId	The id of user which inputs the data
name	Material name
code	Material code
file	File Path
exp	Explanation
viewOrder	Materials will be ordered according to this number in the ascending order
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.19 Table: tblRanking

tblRanking	
PK	<u>id</u>
FK1	userId itemId isTopic point createDate state

5.2.20 Overview

This table stores the information related to the topic ranking. Assigned users can rank a course topic. Total ranking will show us the course achievement.

5.2.20.1 Data Type Definition

Field Name	Data Type
id	GUID
userId	VARCHAR(40)
itemId	VARCHAR(40)
isTopic	TINYINT
point	INTEGER
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.20.2 Description

Field Name	Description
id	Unique Identifier
userId	User Id
itemId	Item Id
isTopic	isTopic or Material
point	The point given to each item
createDate	Created when the row is created and it stores the date and time.
state	It stores the current status of the related row.

5.2.21 Table: tblCoursePoll

tblCoursePoll	
PK	<u>id</u>
FK1	courseId name code startDate endDate createDate state

5.2.22 Overview

We use this table to prepare questionnaires to the students or privileged users.

5.2.22.1 Data Type Definition

Field Name	Data Type
id	GUID
courseId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
startDate	DATETIME
endDate	DATETIME
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.22.2 Data Type Definition

Field Name	Description
id	Unique Identifier
courseId	Course Id
name	Poll name
code	Poll code
startDate	Poll's showing duration's start date
endDate	Poll's showing duration's end date
createDate	Created when the row is created and it stores the date and time.
state	Created when the row is created and it stores the date and time.

5.2.23 Table: tblCoursePoll

tblCoursePoll	
PK	<u>id</u>
FK1	courseId name code startDate endDate createDate state

5.2.24 Overview

We use this table to prepare questionnaire to the students or privileged users.

5.2.24.1 Data Type Definition

Field Name	Data Type
id	GUID
courseId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
startDate	DATETIME
endDate	DATETIME
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.24.2 Data Type Definition

Field Name	Description
id	Unique Identifier
courseId	Course Id
name	Poll name
code	Poll code
startDate	Poll's showing duration's start date

endDate	Poll's showing duration's end date
createDate	Created when the row is created and it stores the date and time.
state	Created when the row is created and it stores the date and time.

5.2.25 Table: tblCourseQuestions

tblCoursePollQuestions	
PK	<u>id</u>
FK1	coursePollId name code viewOrder exp createDate state

5.2.26 Overview

This table stores the questions for each poll.

5.2.26.1 Data Type Definition

Field Name	Data Type
id	GUID
coursePollId	VARCHAR(40)
name	VARCHAR(50)
code	VARCHAR(50)
viewOrder	INTEGER
exp	VARCHAR(1000)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.26.2 Data Type Definition

Field Name	Description
id	Unique Identifier
coursePollId	Related Poll Id
name	Question name
code	Question
viewOrder	Questions will be ordered according to this number in the ascending order
exp	Explanation
endDate	Poll's showing duration's end date
createDate	Created when the row is created and it stores the date and time.
state	Created when the row is created and it stores the date and time.

5.2.27 Table: tblCourseQuestionChoices

tblPollQuestionChoices	
PK	<u>id</u>
FK1	questionId name code viewOrder exp createDate state

5.2.28 Overview

This table stores the choices for each question.

5.2.28.1 Data Type Definition

Field Name	Data Type
id	GUID
questionId	VARCHAR(40)

name	VARCHAR(50)
code	VARCHAR(50)
viewOrder	INTEGER
exp	VARCHAR(1000)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

5.2.28.2 Data Type Definition

Field Name	Description
id	Unique Identifier
coursePollId	Related Poll Id
name	Question name
code	Question
viewOrder	Choices will be ordered according to this number in the ascending order
exp	Explanation
createDate	Created when the row is created and it stores the date and time.
state	Created when the row is created and it stores the date and time.

5.2.29 Table: tblPollUserChoices

tblPollUserChoices	
PK	<u>id</u>
FK1 FK2	choiceId userId createDate state

5.2.30 Overview

This table stores the user choices related to polls

5.2.30.1 Data Type Definition

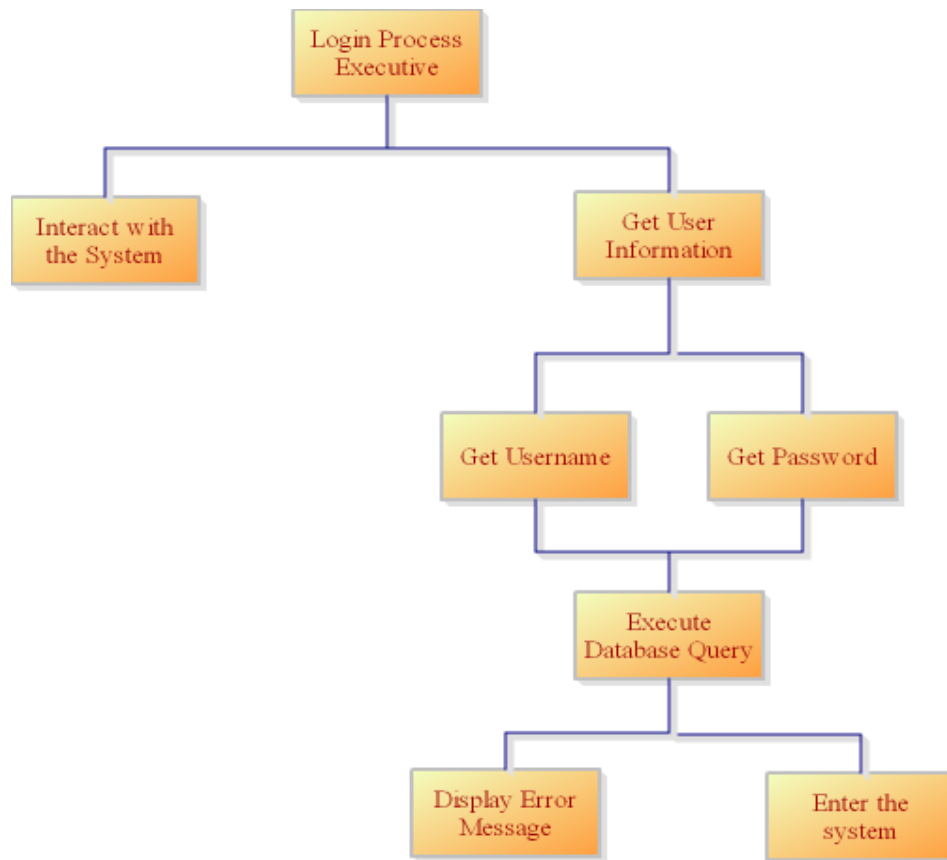
Field Name	Data Type
id	GUID
choiceId	VARCHAR(40)
userId	VARCHAR(40)
code	VARCHAR(50)
createDate	DATETIME: DEFAULT GETDATE()
state	TINYINT: DEFAULT 1

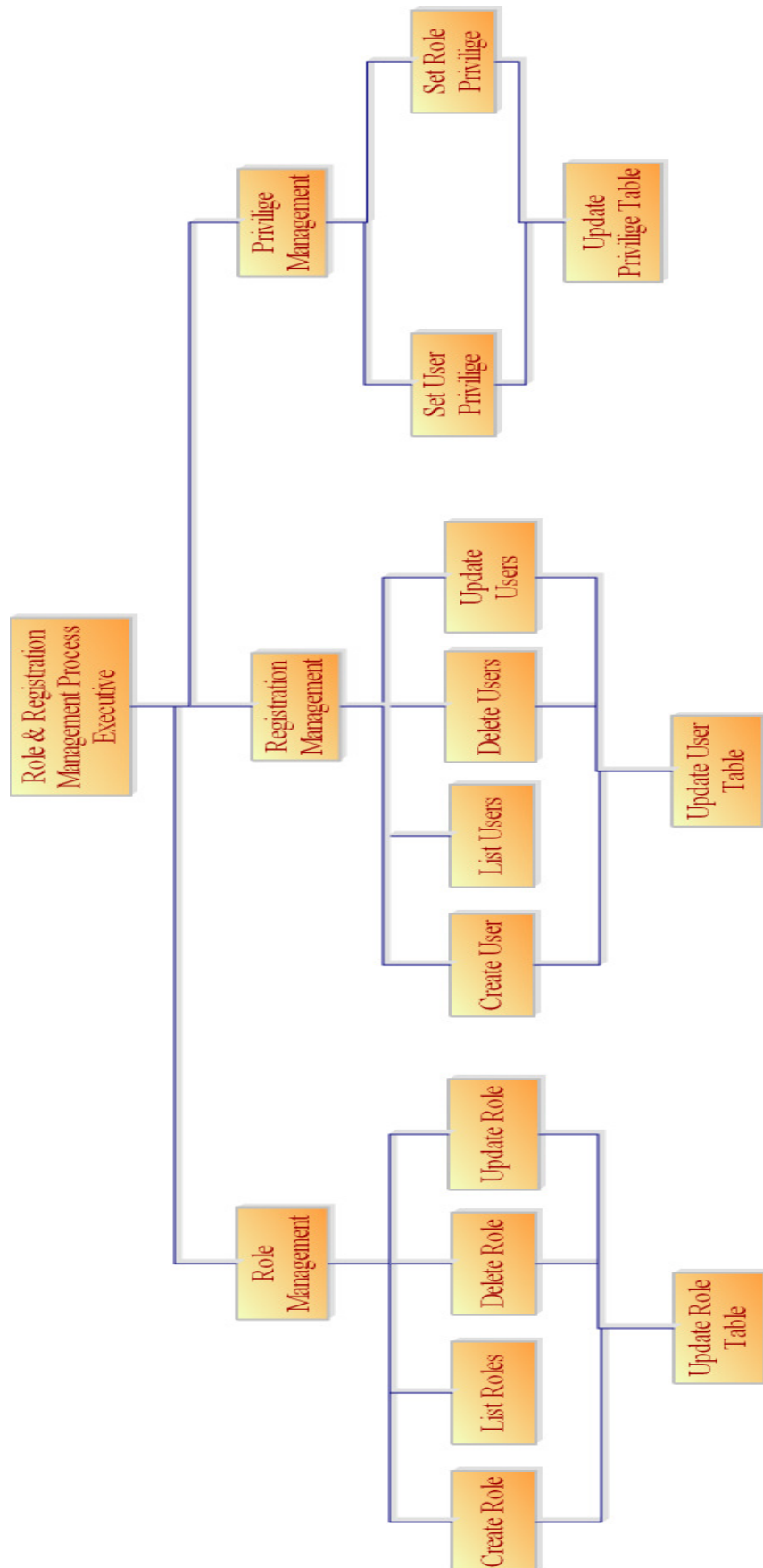
5.2.30.2 Data Type Definition

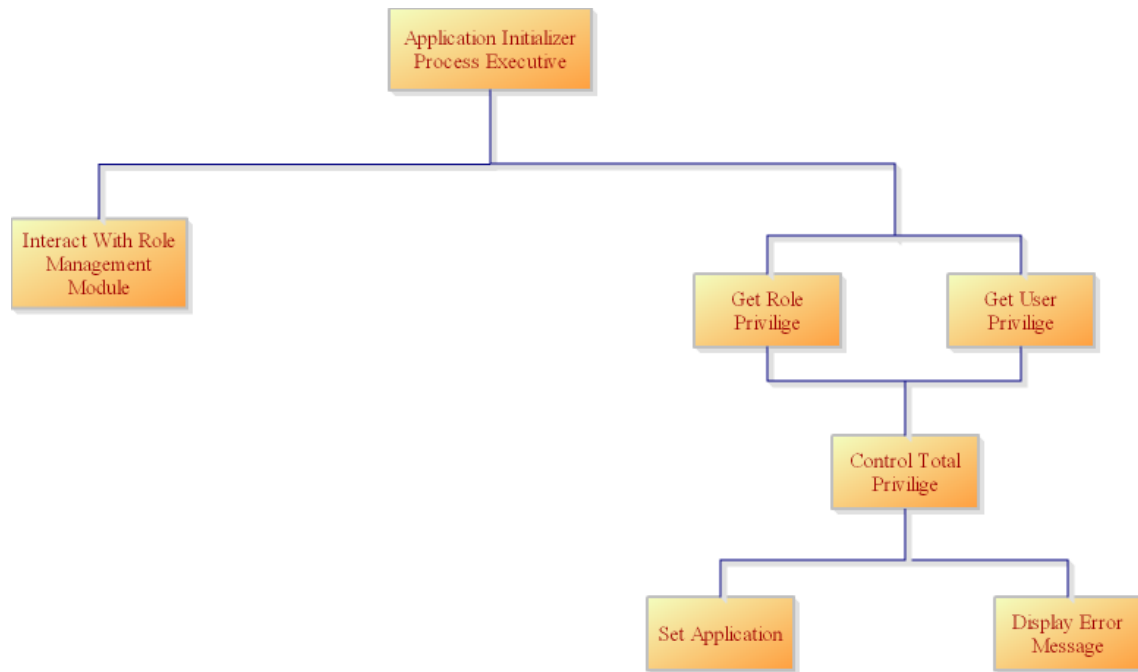
Field Name	Description
id	Unique Identifier
choiceId	Choice Id
userId	User Id
createDate	Created when the row is created and it stores the date and time.
state	Created when the row is created and it stores the date and time.

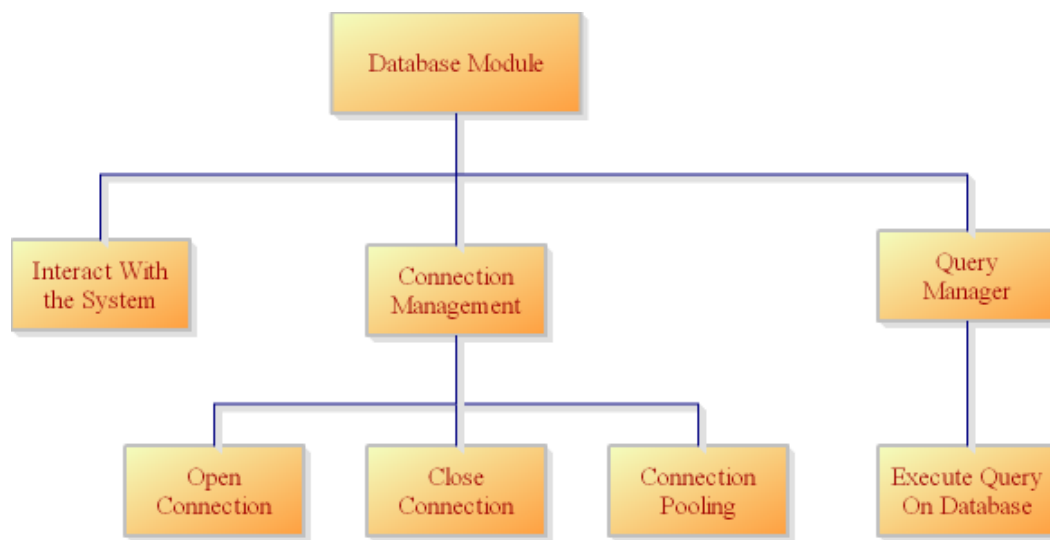
6 ARCHITECTURAL DESIGN

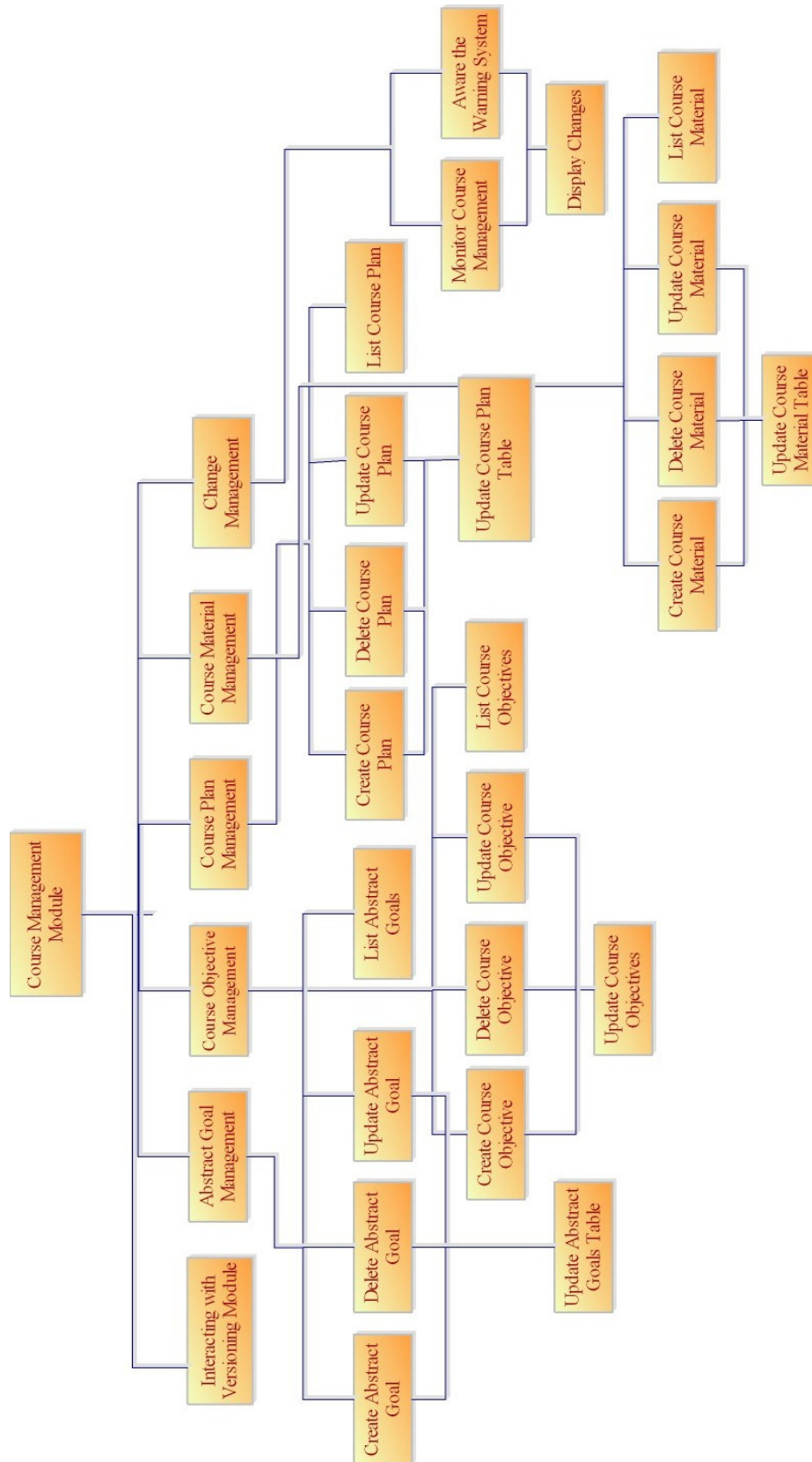
In this section the modules which were described and clarified previously, takes place with structure charts. The final status for each module is given in the following figures.

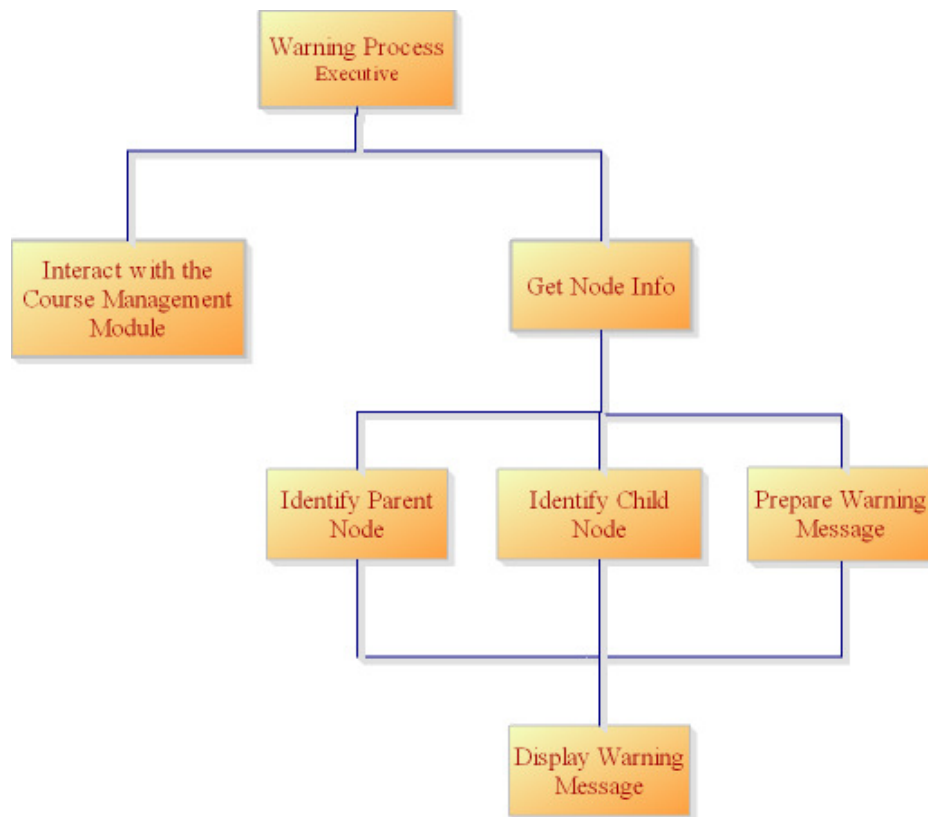
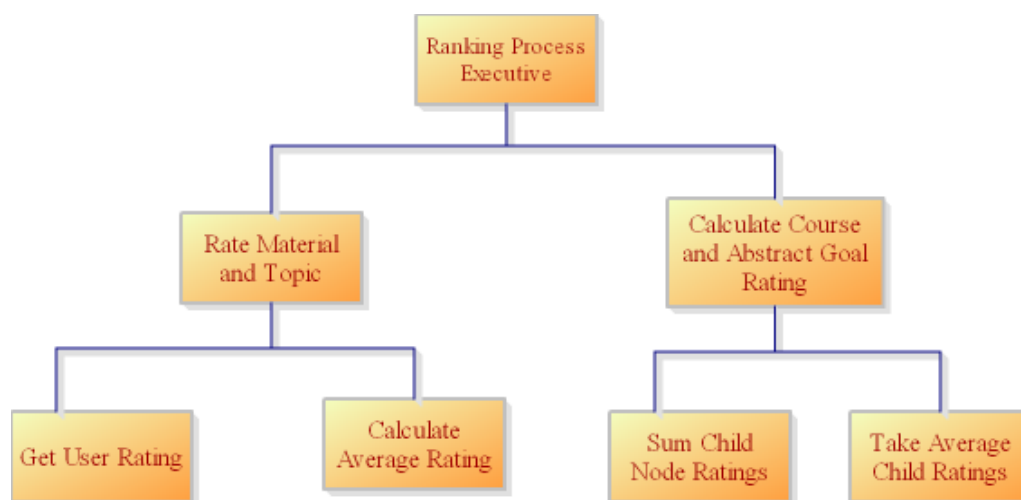
Structure Chart of Login Process BranchingStructure Chart of Registration & Role Management Branching

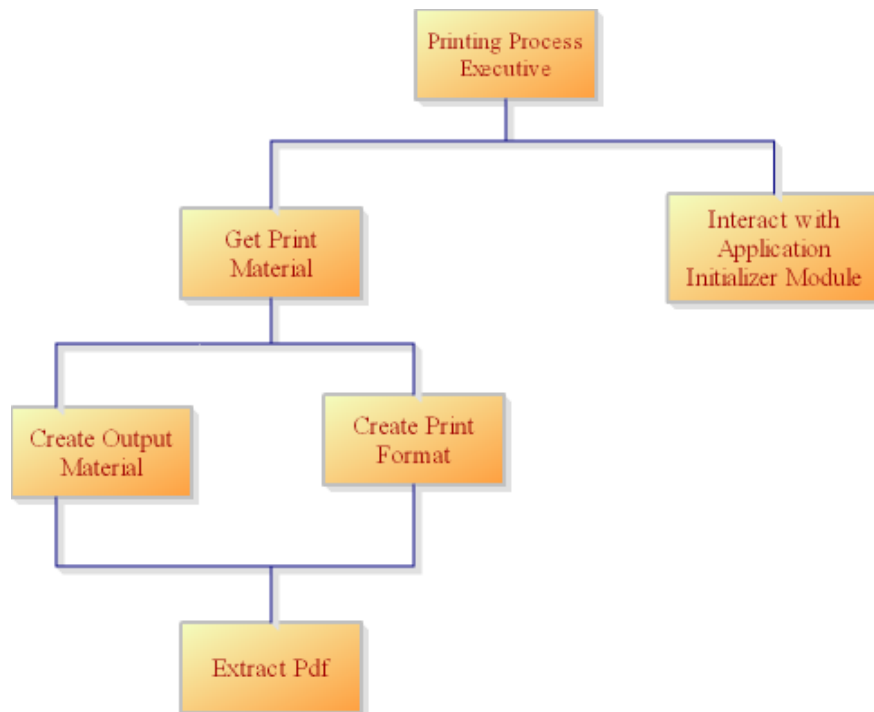
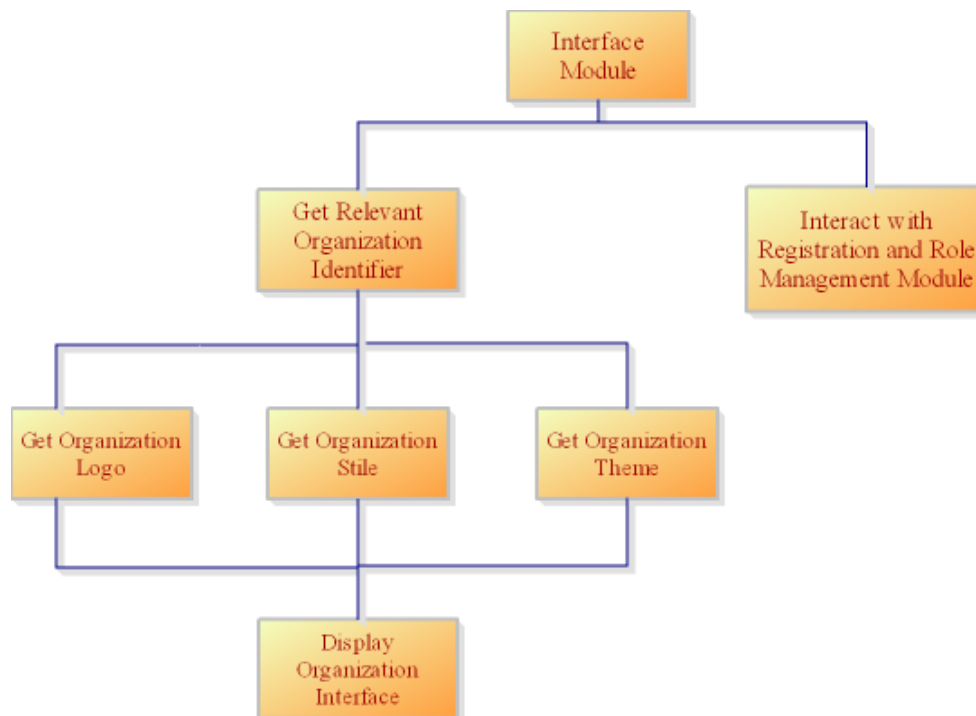


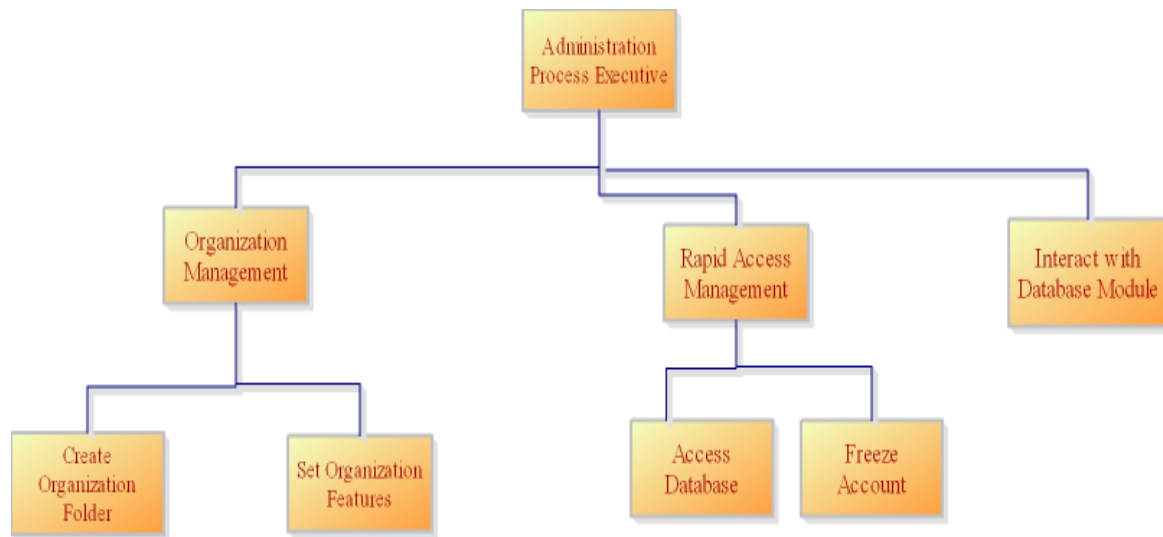
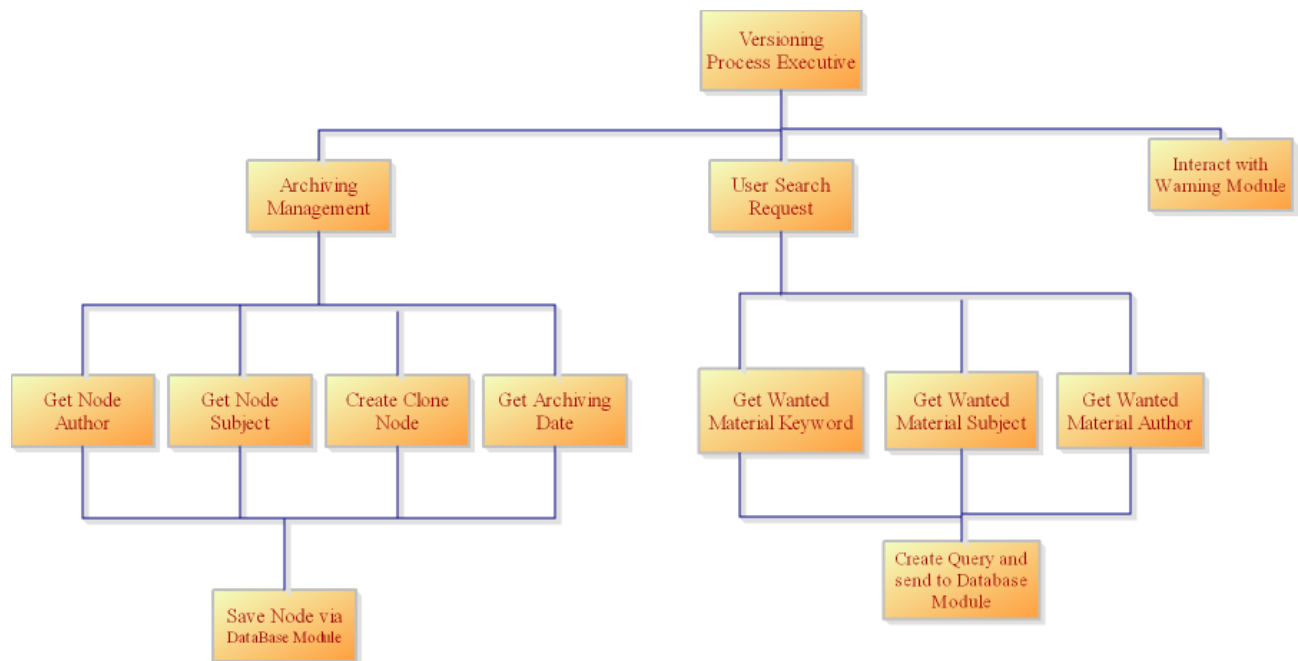
Structure Chart of Application Initializing Process Branching

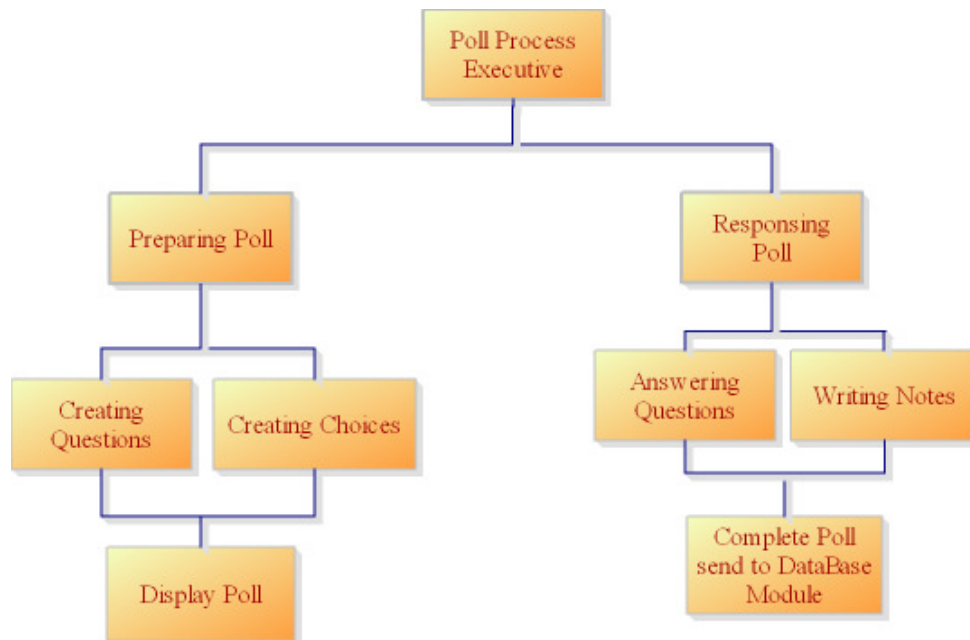
Structure Chart of Database BranchingStructure Chart of Course Management Process Branching



Structure Chart of Warning Process BranchingStructure Chart of Ranking Process Branching

Structure Chart of Print Process BranchingStructure Chart of Interface Branching

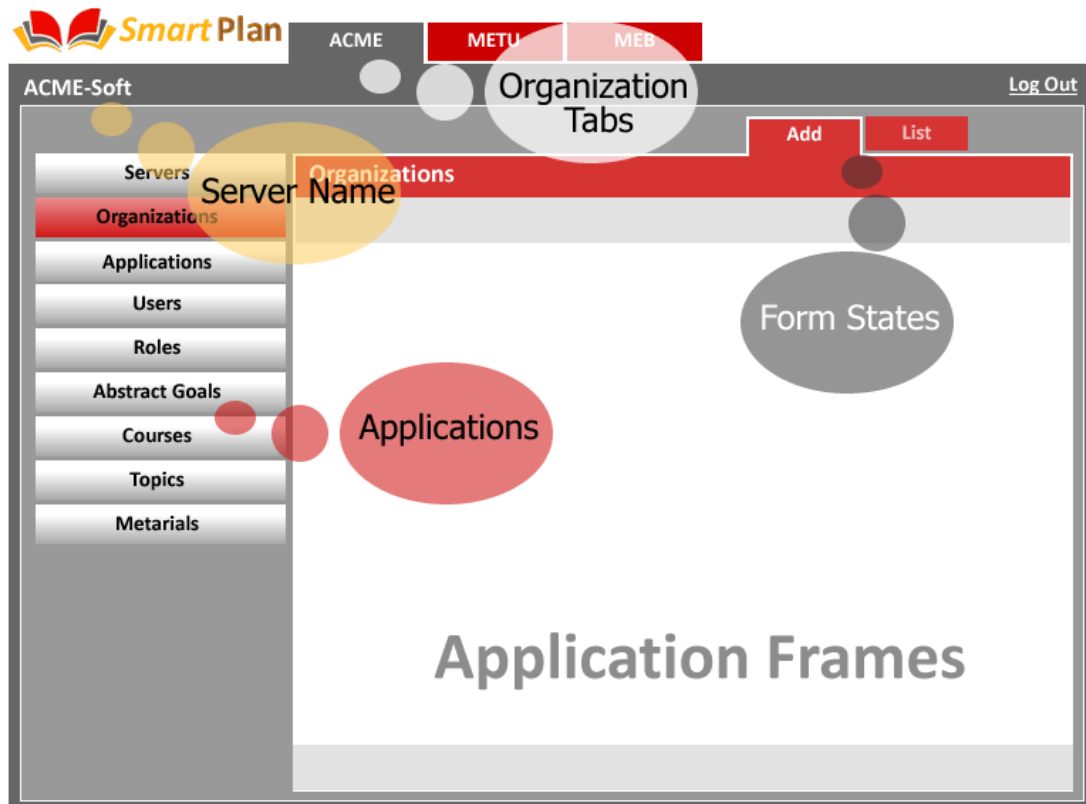
Structure Chart of Administration BranchingStructure Chart of Versioning Process Branching

Structure Chart of Poll Activity Branching

7 GUI DESIGN

We have three different kinds of user interfaces which are primarily used throughout in the software. Because our framework will load each form and grid dynamically, we show the general structure of our user interfaces and insert, delete, update screenshot.

7.1 General view



7.2 Adding new item

The screenshot displays the ACME-Soft Smart Plan application interface. At the top, there is a logo with a red book icon and the text 'Smart Plan'. Below the logo, there are three tabs: 'ACME', 'METU', and 'MEB'. The 'ACME' tab is selected. The main window is titled 'ACME-Soft'. On the left side, there is a vertical menu with the following items: 'Servers', 'Organizations' (highlighted in red), 'Applications', 'Users', 'Roles', 'Abstract Goals', 'Courses', 'Topics', and 'Metarials'. The main area of the window is titled 'Organizations' in a red header. Below the header, there are four input fields: 'Name:', 'Code:', 'Servers:', and 'Explanation:'. The 'Name:' field is a text input. The 'Code:' field is a text input. The 'Servers:' field is a dropdown menu. The 'Explanation:' field is a large text area. At the bottom right of the main area, there is a 'Save' button.

Smart Plan

ACME METU MEB

ACME-Soft

Servers

Organizations

Applications

Users

Roles

Abstract Goals

Courses

Topics

Metarials

Organizations

Name:

Code:

Servers:

Explanation:

Save

7.3 Updating an item



ACME METU MEB

ACME-Soft

Update List

Organizations

Servers
Organizations
Applications
Users
Roles
Abstract Goals
Courses
Topics
Metarials

Name:

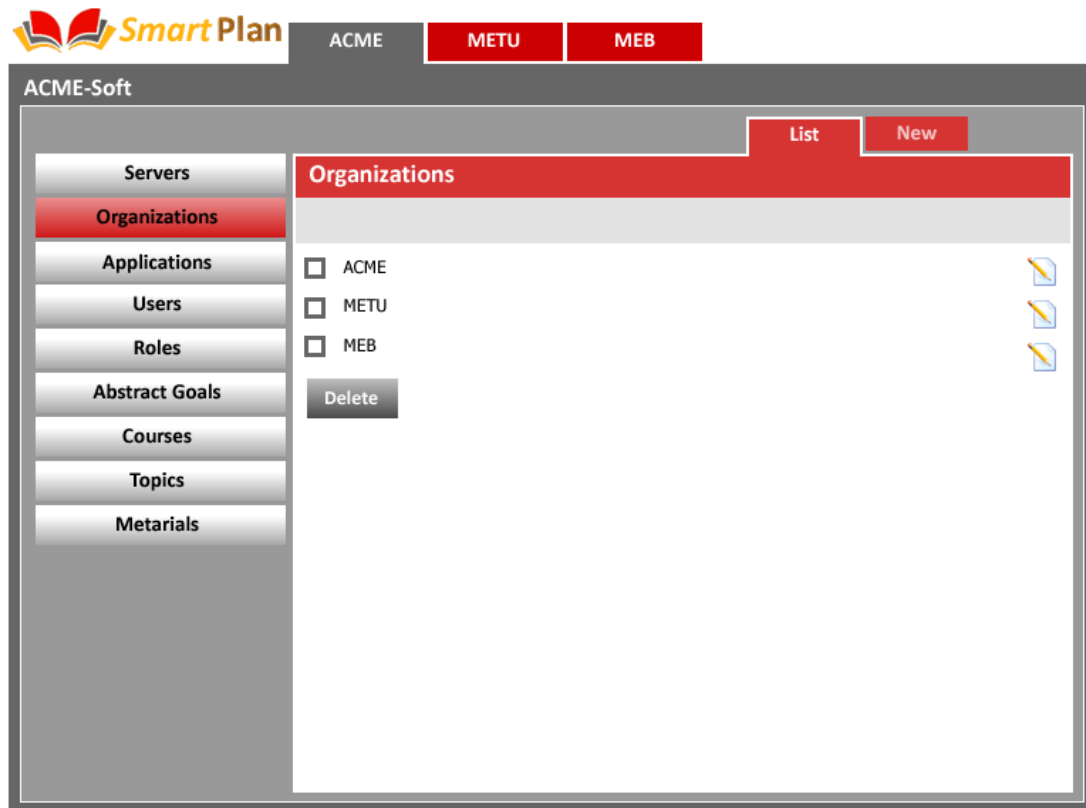
Code:

Servers:

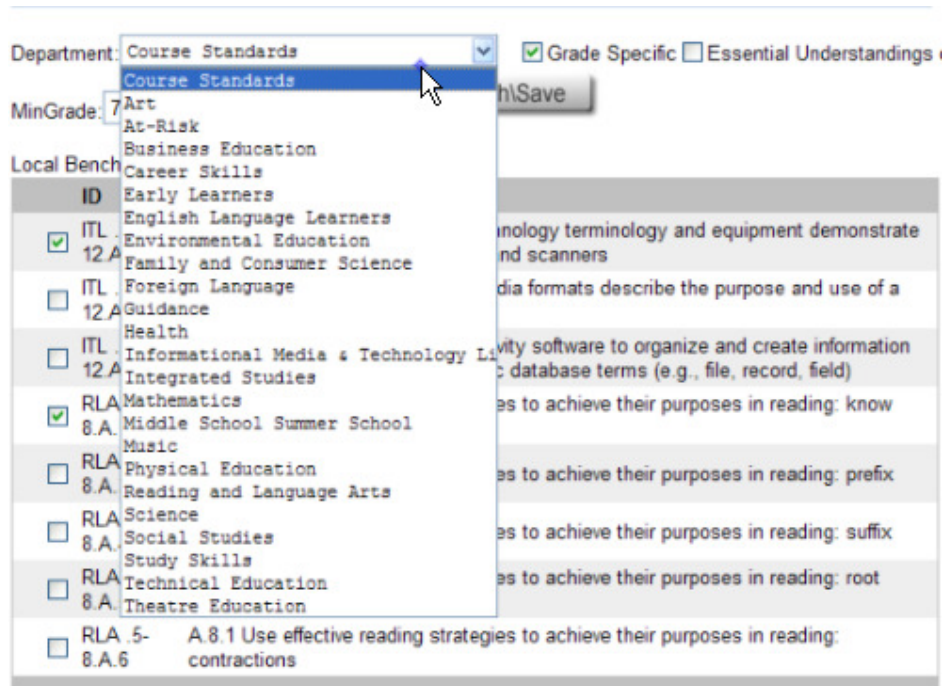
Explanation:

Update

7.4 Listing items



Below picture shows the grid design that we are planning to use at the stage of course design.



References: www.buildyourowncurriculum.com

After design stage of a course there will be a result like below picture.

Advanced English - 7		Reading and Language Arts Grade 7, 0 Credits
Course Overview		
Here is my course overview		
Scope and Sequence		Instructional Topics
Timeframe	Unit	
3 Week(s)	Poetry	1. The life of Poe
2 Week(s)	Short Stories	
Materials and Resources		
some sample materials		
Prerequisites		
recommendation from 6th grade teacher		
Course Details		
UNIT: Poetry -- 3 Week(s)		
Here is a district configurable section, often used for essential questions		
TOPIC: The life of Poe -- 3 Day(s)		
<u>Key Concepts</u>		
Students will be able to explain how Poe's life affected his poetry		
UNIT: Short Stories -- 2 Week(s)		

References: www.buildyourowncurriculum.com

8 DEVELOPMENT SCHEDULE

Development schedule is at the end of the report.

9 SOFTWARE AND HARDWARE REQUIREMENTS

9.1 Software Requirements

We have to use software development tools and programming languages in sophisticated level during the implementation process of our project. Predetermining which software tools to use for the project development is necessary, since it affects our design choices and to choose the most suitable software development tools is important in the success of the project.

- **.Net Framework 2.0**

The .Net Framework is an integral Windows component and it supports building and running the the next generation of applications and Web Services. Because .Net Frameworks provides lots of components and class libraries, it easy to develop windows/web applications. It also provides a managed execution environment, simplified development and deployment.

- **Microsoft Sql Server 2005**

MS SQL Server is a multithreaded, multiuser SQL database management system (DBMS). It is very powerful database management system and it is used lots of companies.

9.2 Hardware Requirements

Minimal hardware requirements for our project are:

For Development :

A PC with the following configuration will be needed:

- Intel Pentium IV or AMD Athlon 1.7 GHz CPU
- 2GB MB DDR2 RAM 80 GB Hard Disk Space
- Internet Connection

End User :

A PC with the following minimum configuration will be needed:

- Intel Pentium4 or AMD Athlon 1 GHz CPU
- 256 MB DDR2 RAM
- Internet Connection

10 TESTING STRATEGIES

We are planning to test our project in different aspects and dimensions after the implementation phase. The aim of this part is to detect errors and bugs of the application. We hope that our software will work properly after a fine testing. We will use some techniques in order to complete the final release. These techniques are as follows:

10.1 Unit Testing

The aim of this kind of testing is to verify whether the smallest testable pieces of the application are working properly or not. In this phase of testing we test each unit separately before integrating it to the complete application. Since finding the possible error in the integrated project is crucial, this stage is relatively important. Also this stage ensures that integration test may only have integration errors and hopefully has no unit dependent errors appear.

10.2 Integration Testing

This testing stage is a little extended form of unit testing. We will have done unit testing while integration phase therefore we will only deal with the integration and combining errors instead of

unit errors. The aim is to combine small units in an error-free manner. This stage results in properly integrated units and ready-to-use modules. We will test combined units, classes and diagrams i.e. related tables.

10.3 Pilot Testing









Considering this kind of testing we will release our application to the real world and wait for feedback from users. Since our project has various types of users, roles, rights and a lot of users it requires real-world testing. For this purpose we are going to deliver our alpha release project to Bora Güngören, and expect feedback from him.

10.4 Security Testing

The system has different types of users and roles which results in many user dependent errors. Moreover the number of users may cause overload and duplicate hazards. In this testing stage we will be dealing with some security issues such as ensuring users are as single-sign-on.

10.5 Validation Testing

This kind is the last of testing strategies those we are going to use. The aim is to produce the well working final release. Obviously, after this test the project is ready to distribute and demonstration. Although, the final release may still have some rare errors, we are confident to handle these errors by releasing service packages.

Project: firstSemester.mpp Date: Sun 1/18/09	Task		Milestone		External Tasks	
	Split		Summary		External Milestone	
	Progress		Project Summary		Deadline	