A PROJECT ON

"STUDENT PERFORMANCE MONITORING SYSTEM"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
'Anudha Chambers', 203 Shaniwar Peth,
Near Gujar Hospital,
KARAD – 415 110.
MH-INDIA

SUBMITTED BY:
Kamlesh Kumar, Kumar Rahul, Manish Gupta, Mohnish,
Niteen Khedkar, Nitin Kurade, Pankaj Kumar Verma,
Pawan Kumar, Priyank Agrawal, Pushpendra Singh Dhakad,
Saurav Gupta, Suchitra R. Biswas and Yogesh Surve

UNDER THE GUIDENCE OF:

MR. RAHUL SANSUDDI
Senior Faculty Member
Sunbeam Institute of Information Technology, KARAD

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Sarang Patil (Director, SIIT Karad), Mr. Prashant Lad (Course Coordinator, SIIT, Karad) and Mr. Rahul Sansuddi (Our Project Guide and Senior Faculty Member, SIIT Karad).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty, especially Mr. Nitin Jadhav, and the staff members of Sunbeam Institute of Information Technology, Karad for their support.

TEAM SUNBEAM
(DAC-15 to DAC-27)
DAC August 2005 Batch,
SIIT Karad

TABLE OF CONTENTS

1. Introduction of Project

2. Requirements

2.1 Functional Requirements

2.1.1 On-line Examination Entry
2.1.2 Examination Conduct
2.1.3 Practice Sessions
2.1.4 Progressive Track Record
2.1.5 Periodic Performance Statements
2.1.6 Periodic Performance Statements
2.1.7 C-DAC Evaluation Record Generation

2.2 Non-Functional Requirements

2.2.1 Interface
Go to Appendix B for user interfaces
2.2.2 Performance
2.2.3 Constraint
2.2.4 Other Requirements

3. Design

4. Coding Standards Implemented

5. Test Report

6. Project Management Related Statistics

1. INTRODUCTION TO PROJECT

This project is all about developing a software subsystem, which facilitates Scheduling, Configuration, Conduct, Review of Online examinations and tracking of student's performance along with facility for Practice sessions.

The first part of the project is for setting the examination pattern for the online examination. This has been implemented by using web application (ASP.NET). The actors for this part are Student, Faculty, Course coordinator and Staff. The Course coordinator and the faculty are responsible and authorized to set the examination pattern. The student and staff can enter the questions through the system. These questions, after being approved by respective faculty, are stored in the final database.

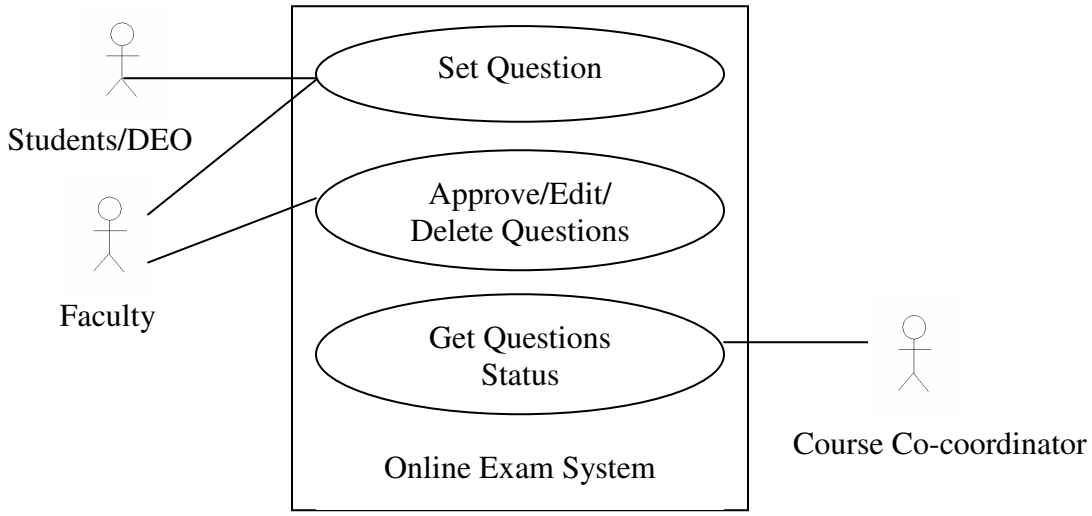
The second part of the project is for conducting Online Examination, and Practice Examination as per the set pattern. Practice Session consists of practice examinations in which a student can select the subject and corresponding topics for the examination. The actors in this part are the students who can take the practice and online examination.

The final part of the project is for generating the report of student's performance. This includes the total evaluation of student's performance. The total evaluation record consists of the sum of all evaluations based on online examination evaluation, internal lab evaluation and evaluation based on other academic activities.

2. REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

2.1.1 On-line Examination Entry:



U1: Question Bank Entry

There is a question entry interface that is intended to facilitate the user (student, faculty or data entry operator) to enter question/s in the question bank stored in the database. User has to select various categories (subject, chapter, topic, level of difficulty etc)

Scenario 1: Mainline Sequence

1.Student/DEO: Select Question entry option

2.System: Display form to enter the question header, options and correct answers.

3. Student/DEO: Enter the necessary values.

4.System: Put the corresponding available questions in temporary table.

U2: Question Categorization, Approval and Updating

The questions entered by the user get stored in the temporary table. The specific faculty member will verify the questions as per their categories and answers. He will also be able to edit the question/header. After approval he

will save the data in the final database (accessed using web services). Faculty can update the question bank at any time.

`

Scenario 1: Mainline Sequence

1. Faculty: Select Question entry option
2. System: Display form to enter the question header, options and correct Answers.
3. Faculty: Enter the necessary values for editing purpose.
4. System: Put the corresponding available questions in Permanent Database.

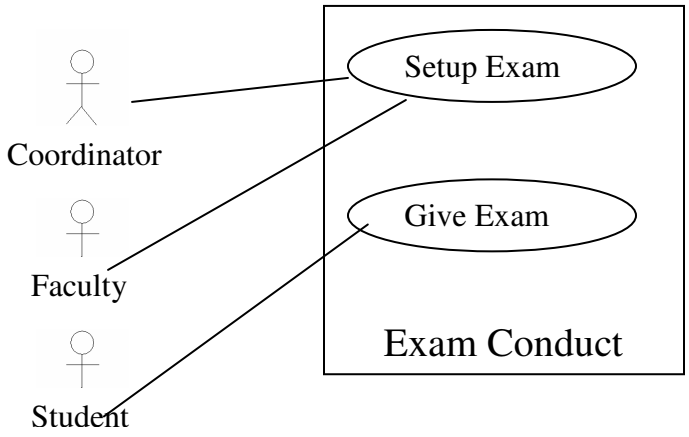
U3: Question Bank Status Reporting

There is a provision of statistical report generation to show the availability of questions in the database.

Scenario 1: Mainline Sequence

1. Faculty: Select Question entry option
2. System: Display form to enter the question header, options and correct Answers.
3. Faculty: Enter the necessary values for editing purpose.
4. System: Put the corresponding available questions in Permanent Database.

2.1.2 Examination Conduct



U1: Setting Up Examination

The course coordinator or the concerned faculty is authorized to set up the examination pattern. He can set the number of questions (either in % or in numbers), topic, level of difficulties and examination duration etc.
[Validations should be taken care of by the system]

Scenario 1: Mainline Sequence

1. User: Enter Login ID and Login Type (faculty or course- coordinator)
2. System: Authentication and authorization.
3. User: Select various options.
4. System: Validate and enter the pattern in database

Scenario 2: At step 4 of the mainline sequence

4. System: Displays the message "Incorrect entry please. Reenter the options".

5. User: Edit the wrong field.
6. System: Validate and enter the pattern in database.

U2: Examination Conduct

The student after being authenticated by his unique ID can give the online examination. The result of examination can be shown to the student immediately after finish of examination or afterwards. Each examination has a unique ID.

Scenario 1: Mainline Sequence

1. User: Enter login ID & password.
2. System: Authenticate user.

3. User: Take examination.
4. System: Provide questions and Evaluate questions.

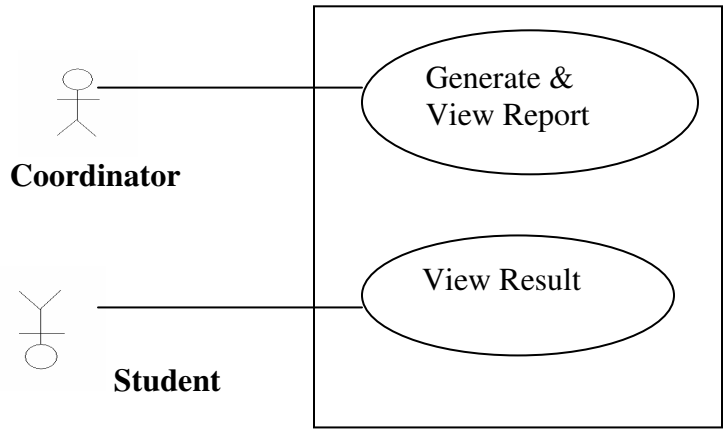
Scenario 2: At step 2 of the mainline sequence

2. System: Display the message "User does not exist."

2.1.3 Practice Sessions

2.1.3.1 Examination Review and Report

The system generates the examination review report. The report shows the number of correct answers with their explanations.



Exam Review and Report

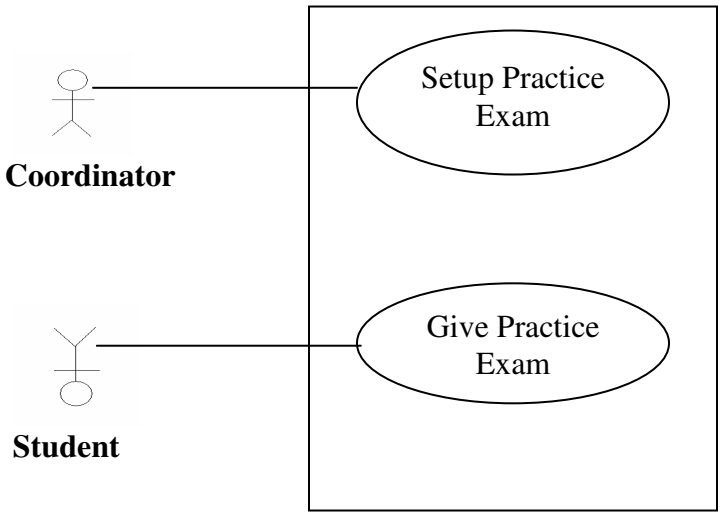
U1: Display & Generate Report
Secenario1: Mainline Sequence

1. Coordinator: -He can choose the options to generate the report according to CDAC pattern and monthly or quarterly basic.
2. System: - Generate and display the report accordingly.

U2: Review his/her result
Secenario1: Main Line Sequence

1. Student: - Login First.
2. System: - Validate the student's Login and Display Prompt To Select Monthly & Quarterly result.
3. Student: -Select which result he wants to see.
4. System: - Display it accordingly.

2.1.3.2 Practice Session



U1: Setup Examination Pattern

Practice session consist of practice mock examination s in which a student can select the subject, topic and/or the level of difficulties of examination.

- Both the practice session and online examination can be resumed in case of any failure with the help of unique examination ID.

Secenario1: Mainline Sequence

1. Coordinator Setup Module ID, Module Name, Start Date, End Date.
2. System generates Examination Paper.

Secenario2:

At Step1 of Main Line Sequence:

1. Display The Message That Some Information Has Not Be Entered.

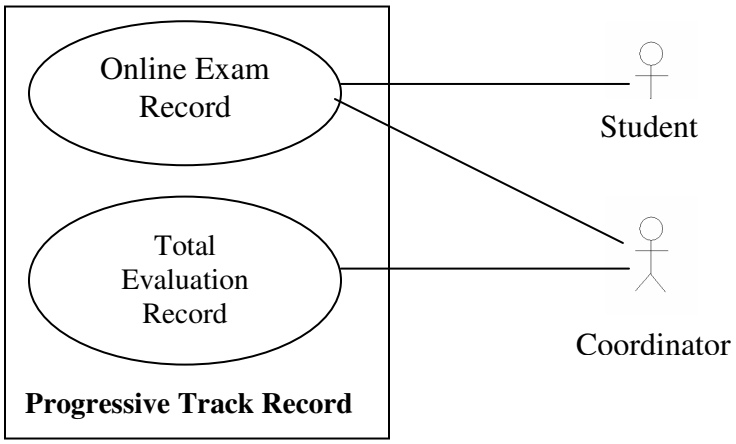
U2: Give Practice Examination

Secenario1: Main Line Sequence

1.Student Logins.
2.System Validates The Login and Display Prompt To Select Subject and Topic.
3.Student selects Subject and Topic.

4.System: System displays The Question Paper, Result and Explanation it Student Want.

2.1.4 Progressive Track Record



U1: Online Examination Record

The system maintains the students history of online examination.

Scenario 1: Mainline Sequence

1. User: Enter login ID & password.
2. System: Authenticate user.
3. System: Provide exam report.

Scenario 2: At step2 of the mainline system

System: Display the message "User doesn't exist."

U2: Total Evaluation Record

The total evaluation record consists of the sum of all evaluations based on online examination evaluation, internal lab evaluation and evaluation based on other academic activities.

Scenario 1: Mainline Sequence

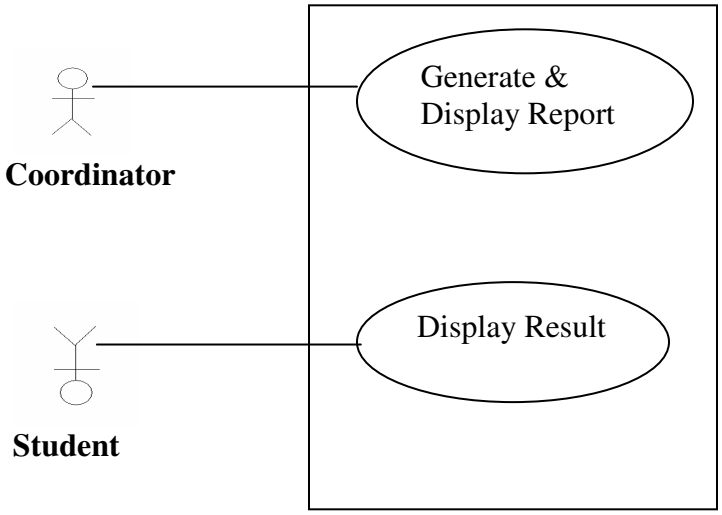
1. User: Enter login ID & password.
2. System: Authenticate user.
3. System: Provide final evaluation report.

Scenario 2: At step2 of the mainline system

System: Display the message "User doesn't exist."

2.1.5 Periodic Performance Statements

The student has been provided the facility to see his performance statements on monthly or quarterly basis.



Test Description:

U1: Coordinators: Using this Use Case Coordinator Displays & Generates Report by providing the necessary details.

Secenario1: Mainline Sequence

3. Coordinator: He can generate result He can choose the options to generate the report according to monthly or quarterly basic.
4. System: - Generate and display the report accordingly.

U2: Student: Review his/her result

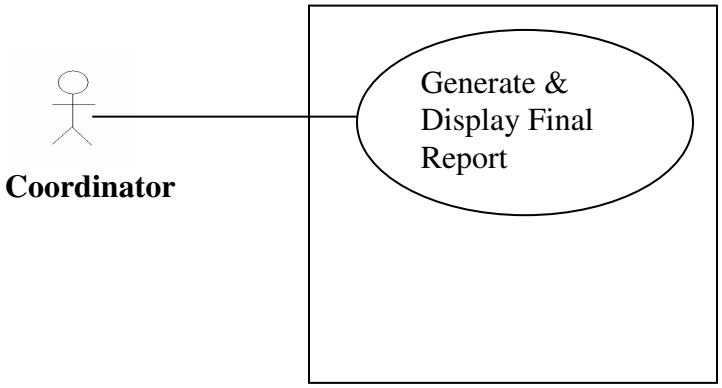
Secenario1: Main Line Sequence

1.Student: Login First.
2.System: Validate the student's Login and Display Prompt To Select

monthly & quarterly result.
3.Student: Select which result he wants to see.
4.System: Display it accordingly.

2.1.6 C-DAC Evaluation Record Generation

The system generates the finally evaluated record that is generated as per the format specified by CDAC (for CCME, CCEE).



C-DAC Evaluation Record Generation

Test Description:

U1: Coordinator: - Using this Use Case Coordinator Displays & Generates Final Report As per CDAC format.

Secenario1: Mainline Sequence
1.Coordinator: He enters the necessary values to generate result
2.System: - Generate and display the record as per CDAC format.

Secenario2: At step 2 of mainline sequence
1.System: Displays the message that some input information has not been entered.
The system displays the prompt to enter the missing values.

2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface

Go to Appendix B for user interfaces

2.2.2 Performance

- **Number of Concurrent Users:**
  The system should support at least 200 concurrent users.
- **Continuation of Examination:**
  The system is susceptible to any temporary server failure since it uses the strong feature of ADO.NET, Disconnected Architecture. Hence the examination will be continued even if the sever gets disconnected in between the examination.

2.2.3 Constraint

In this version of the software, it has a limitation that maximum 100 questions can be set for the online examination.

2.2.4 Other Requirements:

- **Hardware Interfaces**

The SPMS is expected to function on Intel PIII 900 MHz Processor equivalent or above, 128 MB RAM, 20 GB HDD.

- **Software Interfaces**

The SPMS shall work on MS Windows operating systems family (MS Windows 98, MS Windows NT Workstation, MS Windows 2000, MS Windows XP). It configures to work with Oracle database. This System works on .Net Framework. It uses browser IE 5.0 & above. It uses IIS 5.0 server.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: T_Faculties

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	FacultyID	varchar	50	0
0	Password	varchar	50	1
0	DesigID	varchar	50	1
0	FalcultyName	varchar	100	1
0	Address1	varchar	50	1
0	City	varchar	50	1
0	State	varchar	50	1
0	HireDate	date	8	1
0	Qualification	varchar	50	1
0	Lock	number	4	1

Table2: T_Modules

3	ModuleID	varchar	16	0
0	CourseID	varchar	50	1
0	ModuleName	varchar	75	1
0	Duration	varchar	50	1
0	FacultyID	varchar	50	1
0	StartDate	date	8	1
0	EndDate	date	8	1
0	LabSessions	number	4	1
0	Lock	number	4	1

Table3: T_Questions

3	QuestionID	number	4	0
0	ModuleID	varchar	16	1
0	TopicID	number	4	1
0	DifficultyLevel	char	1	1
0	Question	varchar	3000	1
0	Option1	varchar	500	1
0	Option2	varchar	500	1
0	Option3	varchar	500	1
0	Option4	varchar	500	1
0	Option5	varchar	500	1
0	CorrectAnswer	char	1	1
0	ImageSrc	text	16	1
0	Explaination	varchar	3000	1

Table4: T_Results

3	SessionID	varchar	10	0
0	TestID	varchar	5	1
0	RollNo	varchar	6	1
0	ExamStartDate	Date	8	1
0	ExamEndDate	date	8	1
0	Total_Marks	number	4	1
0	Scored_Marks	number	4	1
0	Machine_Code	varchar	16	1
0	Question_Code	varchar	16	1
0	Answer_Code	varchar	16	1
0	Candidate_Answers	varchar	16	1

Table5: T_Students

3	RollNo	varchar	6	0
0	Password	varchar	20	1
0	BatchID	varchar	50	1
0	FirstName	varchar	50	1
0	MiddleName	varchar	50	1
0	LastName	varchar	50	1
0	Address1	varchar	255	1
0	City	varchar	50	1
0	State	varchar	50	1
0	Dob	date	7	1
0	Phone	varchar	50	1
0	Gender	varchar	1	1
0	BloodGroup	varchar	50	1
0	F_PWD	varchar	50	1
0	Net_Folder	varchar	50	1
0	ProjectGroupId	varchar	50	1
0	SubBatch	varchar	1	1

Table6: T_TempQuestion

3	QuestionId	Number	4	0
0	ModuleId	varchar	16	1
0	TopicId	Number	4	1
0	DifficultyLevel	char	1	1
0	Question	varchar	3000	1
0	Option1	varchar	500	1
0	Option2	varchar	500	1
0	Option3	varchar	500	1
0	Option4	varchar	500	1
0	Option5	varchar	500	1
0	CorrectAnswer	char	1	1
0	Imgsrc	varchar	16	1
0	Explaination	varchar	3000	1

Table7: T_ TestConfiguration

2	TestID	varchar	5	0
0	TestName	varchar	16	1
0	Duration	number	4	1
0	NoOfQuestions	number	4	1
0	Marks	real	4	1
0	Easy	number	2	1
0	Average	number	2	1
0	Difficult	number	2	1
0	Lock	char	1	1
0	ModuleID	varchar	16	1
0	NegativeMarking	number	9	1
0	ConfigType	char	1	1

Table8: T_ TestQuestions

3	QuestionId	number	4	0	
0	ModuleID	varchar	16	1	
0	TopicId	number	4	1	
0	DifficultyLevel	char	1	1	
0	Question	varchar	3000	1	
0	Option1	varchar	500	1	
0	Option2	varchar	500	1	
0	Option3	varchar	500	1	
0	Option4	varchar	500	1	
0	Option5	varchar	500	1	
0	CorrectAnswer	char	1	1	
0	Imgsrc	varchar	16	1	
0	Explaination	varchar	3000	1	

Table9: T_ Topic

2	TopicId	number	4	0
0	ModuleID	varchar	16	1
0	Topic	varchar	255	1
0	Easy	number	2	1
0	Average	number	2	1
0	Difficult	number	2	1

Table10: T_ TopicConfiguration

2	TopicId	varchar	5	1
0	TopicId	number	4	1
0	NoOfQuestions	number	4	1
0	DifficultyLevel	char	1	1

E-R Diagram, Class Diagram and Collaboration Diagram:

Go to Appendix A .

4. CODING STANDARDS IMPLEMENTED

4.1 Coding Standards as provided by the client.

Primitive Type Notation

Data Type	Prefix
sbyte	sy
short	s
int	i
long	l
byte	y
ushort	us
uint	ui
ulong	ul
float	f
double	d
decimal	dec
bool	b
char	c

Type Notations

Type	Prefix
Boolean	bln
Char	ch
Double	dbl
Exception	ex
Integer	int
String	str
StringBuilder	strb
DateTime	date

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in .Net. Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Person, BankVault, SMSMessage, Dept	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Pascal	GetDetails, UpdateStore	Methods should use **verbs** or verb phrases.
Parameter	camel	personName, bankCode	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	IDisposable	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_foreColor, _backColor	Use underscore camel casing for the private member variables
Exception Class	Pascal with "Exception" suffix	WebException, SMSException	
Event	Pascal plus optional "EventHandler" suffix where relevant	btnSubmit_Click, Painting, Click, Clicked, MyEventHandler	VS uses underscores to separate an object from its event. Be careful with tense (pre/past), e.g. a Close event that can be canceled should have a Closing event and a Closed event. MS also recommend adding the "EventHandler" suffix where thought needed.

User Interface Objects and Controls

Control	Prefix	Example
Label	lbl	lblSurname
TextBox	txt	txtSurname
DataGrid	dg	dgResults
Button	btn	btnSave
ImageButton	ibtn	ibtnSave

Hyperlink	lnk	lnkHomePage
DropDownList	ddl	ddlCompany
ListBox	lst	lstCompany
DataList	dlst	dlstAddress
Repeater	rep	repSection
Checkbox	chk	chkMailList
CheckBoxList	chk	chkAddress
RadioButton	rdo	rdoSex
RadioButtonList	rdo	rdoAgeGroup
Image	img	imgLogo
Panel	pan	panSection
PlaceHolder	plh	plhHeader
Calender	cal	calMyDate
Adrotator	adr	adrBanner
Table	tbl	tblResults
[All] Validators	val	valCreditCardNumber
ValidationSummary	vals	valsErrors
ListView	lv	lvStudentList
ImageList	iml	imgPhotoList
DataGridTableStyle	dgts	dgtsTable
DataGridTextBoxColumn	dgtbc	dgtbdName
DataReader	dreader	dreaderValue
DataRow	drow	drowTime
DataSet	dset	dsetValue
DataTable	dtable	dtableStudent
Mainmenu	mm	mmFile
MenuItem	mi	miNew
PictureBox	pbx	pbxPhoto
SDI-Form	form	formEmployee
SqlCommand	sqlcom	sqlcomQuery
SqlCommandBuilder	sqlcomb	sqlcombQuery
SqlConnection	sqlcon	sqlconQracleConnection

SqlDataAdapter	sqlda	sqldaData
StatusBar	stb	stbMyStatusBar
TabControl	tabctrl	tabctrlInfo
TabPage	tabpage	tabpageColor
ToolBar	tbr	tbrMyToolBar
ToolBarButton	tbb	tbbFileNew
Timer	tmr	tmrExamCounter
UserControl	usr	usrControl
HashTable	htbl	htblEmployee
DataGridDateTimePickerColumn	dgdtpc	dgdtpcMyPicker

Namespaces
All namespace should use Pascal casing and be prefixed with your company (and department if you wish). See examples below:
Sample "project level" namespaces:

YourCompany.YourDept.YourProject
Visualize.Blog

Generic reusable routines, classes etc which will be used across projects, can sit at the 'company' or 'dept' level, for example:

Visualize.Utils

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

4.2 Standards used for User Interface Design.

Name of Controls	Size	Font Name
Label	10 pts	Verdana
Textbox	10 pts	Verdana
Button	8 pts	Microsoft Sans Serif (Bold)
Label	18 pts	Verdana (Bold)

5. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.

	GENERAL TESTING	
SR-NO	**TEST CASE**	**EXPECTED RESULT**
1	Login Page	Redirected to Next page
2	Login Page/Reset Button	All Fields should be cleared
3	Questions Entry Page (Student)	All the fields should be filled for submission
4	Questions Entry Page (Student)	Log out
5	Questions Entry Page (Faculty)	All the fields should be filled for submission
6	Questions Entry Page (Faculty)	Log out
7	Set Pattern	Correct Pattern should be set
8	View Set Result	Set Result
9	Online Exam	By default no option should be set on to and fro motion
10	StaffQuestion Entry	On back it should be reverted to previous page
	STATIC TESTING	
SR-NO	**Deviation**	**Program**
1	Commenting not followed	All Web Application

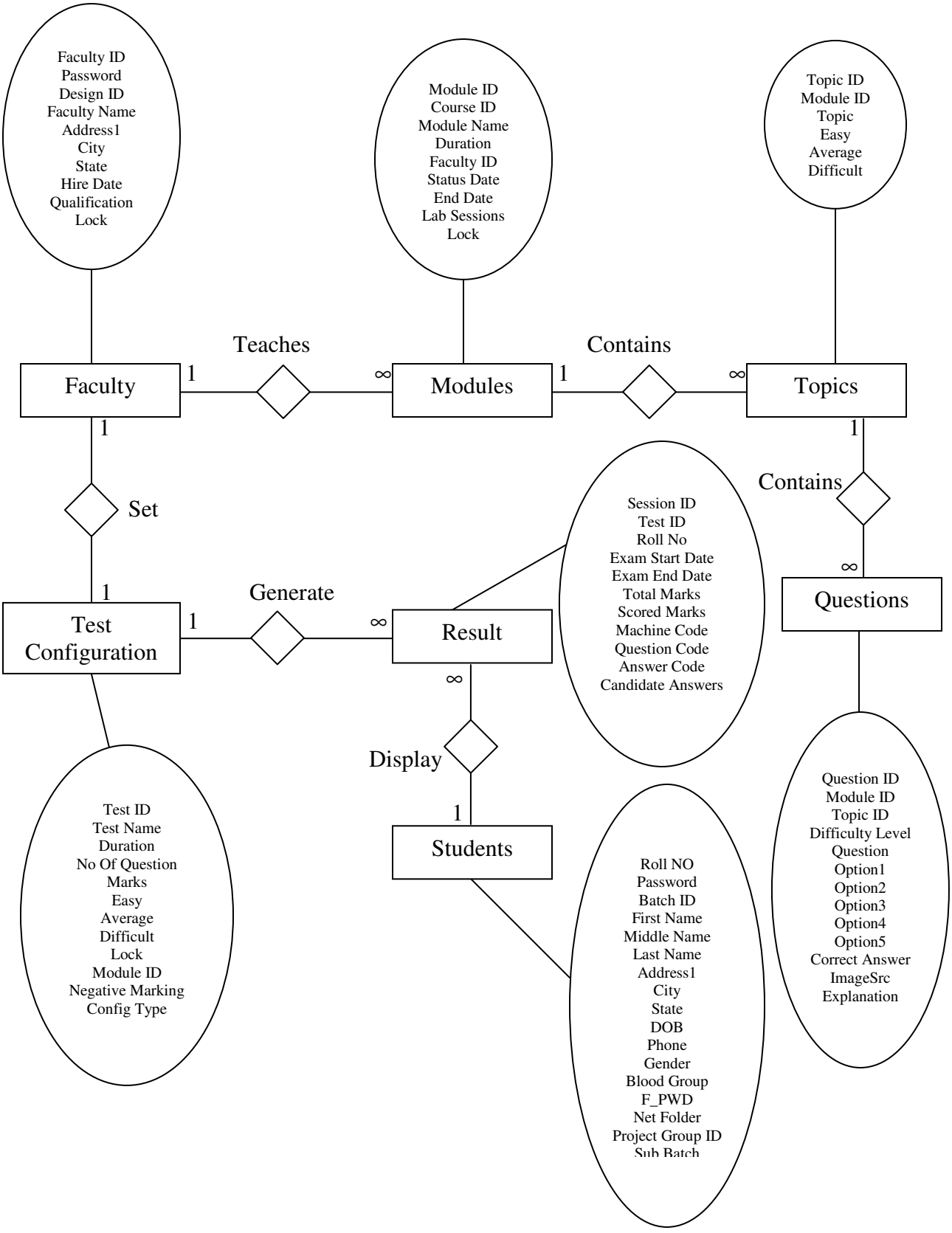
6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	SLC Phase	Additional Notes
JAN 16,2006	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Sarang Patil (Director, SIIT Pune and Karad) to know his requirements.
JAN 17,2006	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
JAN 18,2006	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
JAN 19,2006	Business Logic Component design Started	Design Phase	----------
JAN 20,2006	Coding Phase Started	Coding Phase	70% of Class Library implemented.
JAN 21,2006	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
JAN 22, 2006	Off	Off	Off
JAN 23, 2006	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.

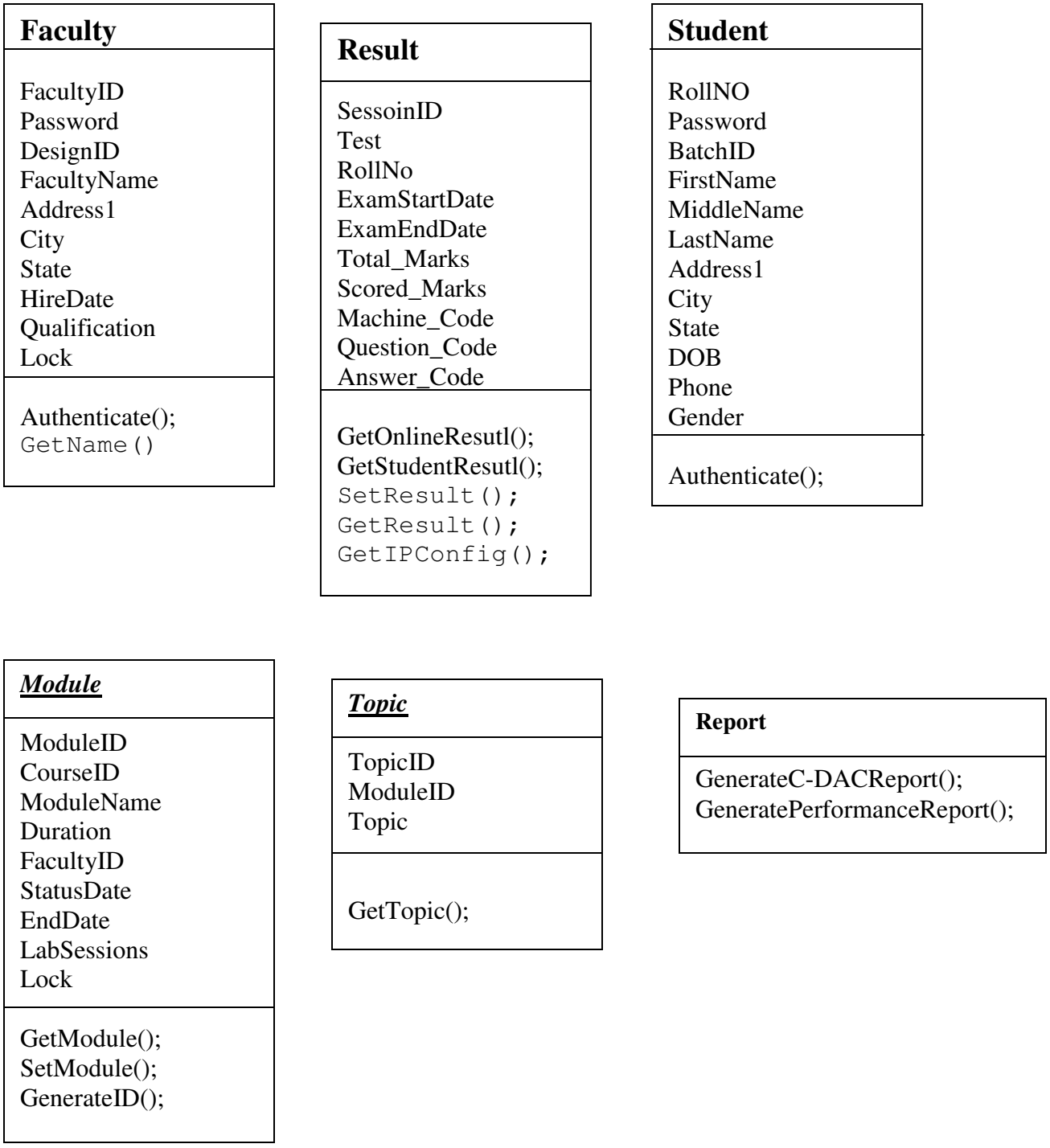
JAN 24, 2006	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--
JAN 25, 2006	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
JAN 26, 2006	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
JAN 27, 2006	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
JAN 28-30, 2006	The Errors Found were Removed	Debugging	The Project was complete for submission
JAN 31, 2006	Final Submission of Project		

Appendix A

E-R Diagram



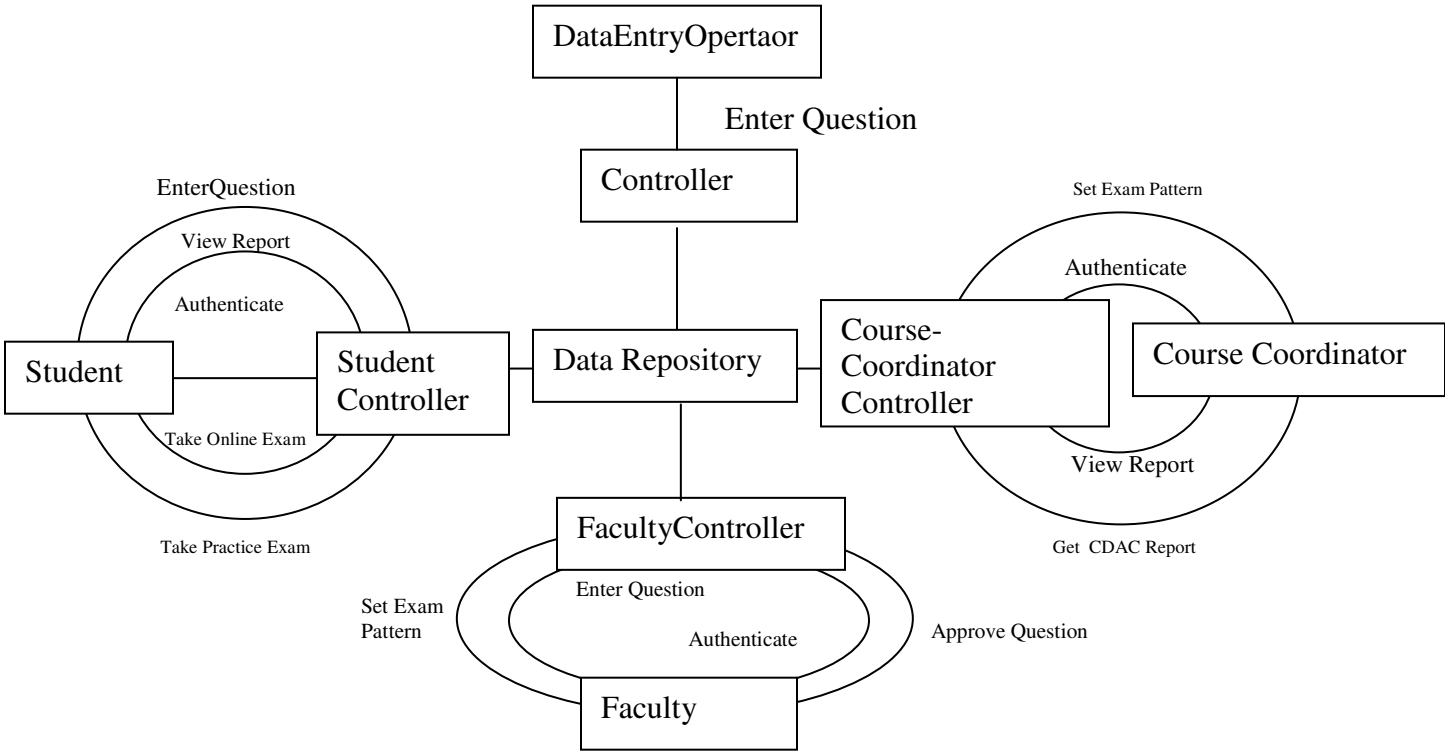
Class Diagram



TestConfiguration
TestID TestName Duration NoOfQuestion Marks Easy Average Difficult Lock
GetRandomQuestion(); SetPattern() GetRandomQuestionID(); SetCustom2Configuration(); GetTestPattern(); SetCustom1Configuration(); SetTopicConfiguration(); SetRandomConfiguration(); SetCustom3Configuration(); UndoTestConfigurationUpdate(); UndoTopicConfigurationUpdate(); GetLastTestID(); GetConfigurationType(); GetPattern(); GenerateTestConfigurationID(); GetRandomQuestionIDForEasy(); GetRandomQuestionIDForAverage(); GetRandomQuestionIDForDifficult(); GetRandomEasyQuestion(); GetRandomAverageQuestion(); GetRandomDifficultQuestion(); GetCustom1Questions(); GetCustom4Questions();

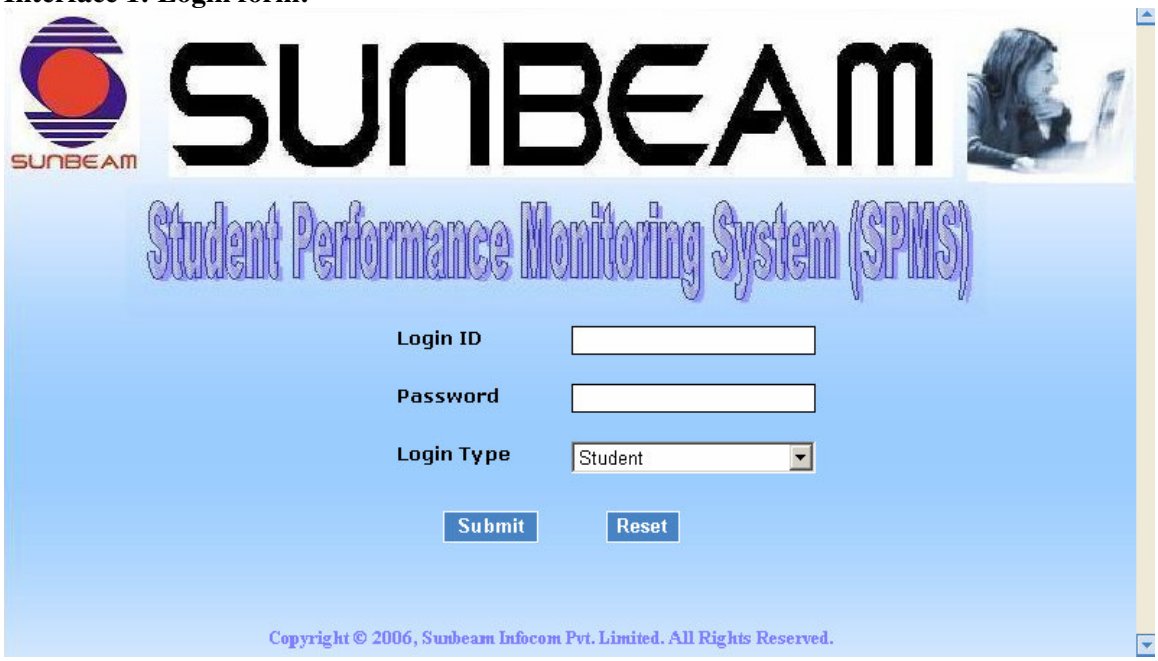
Question
QuestionID ModuleID TopicID DifficuktyLevel Question Option1 Option2 Option3 Option4 Option5 CorrectAnswer ImageSrc
SetQuestion(); GetQuestion(); GetQuestionTemp(); GetImpQuestion(); UpdateTopicTable(); DeleteQuestion(); SetQuestionTemp(); Approve(); DeletQuestion(); UpdateQuestion(); DeleteTempQuestion() SetImpQuestion(); GenerateImpQuestionID(); GenerateQuestionID(); GenerateTempQuestionID();

COLLABORATION DIAGRAM

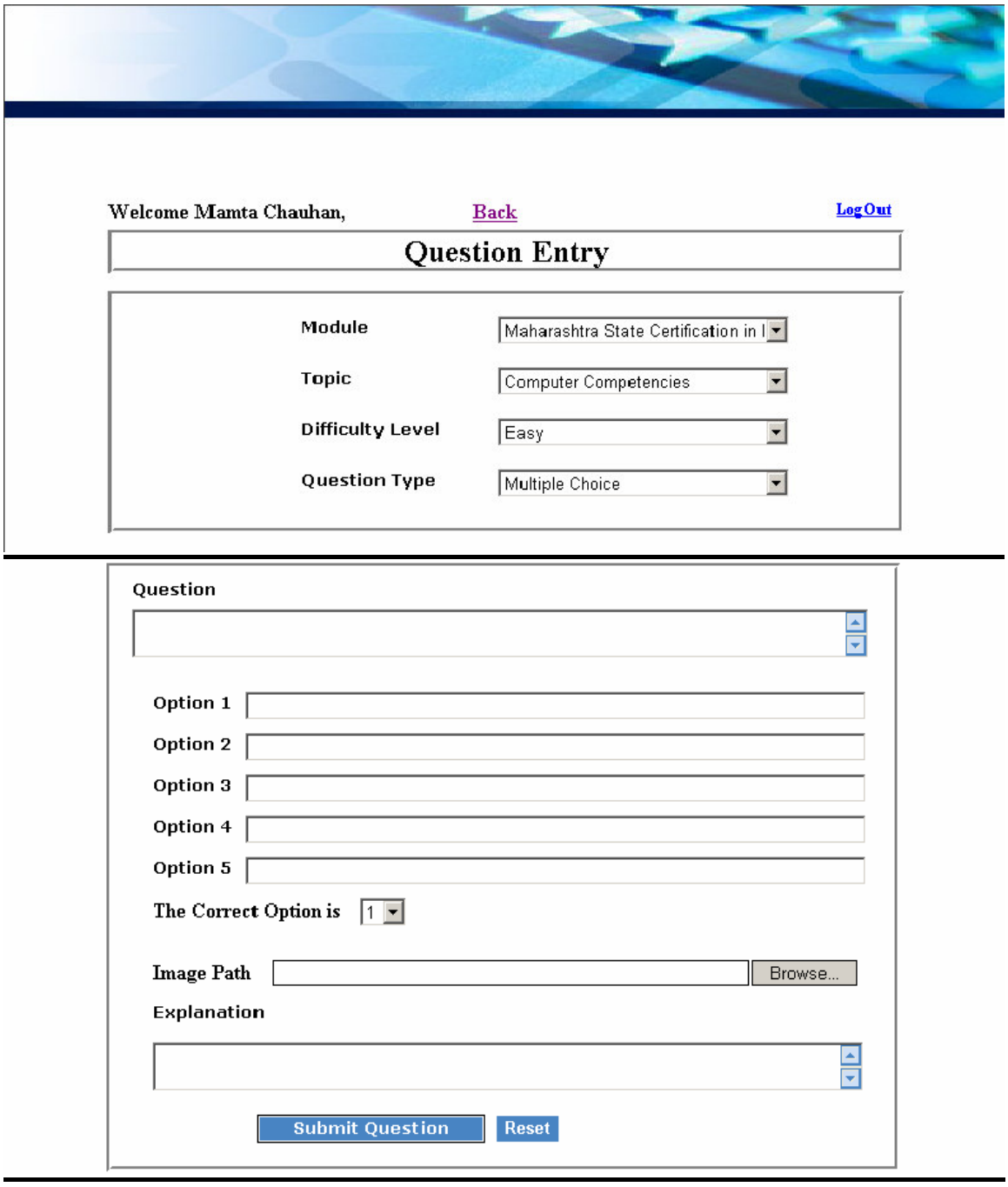


Appendix B

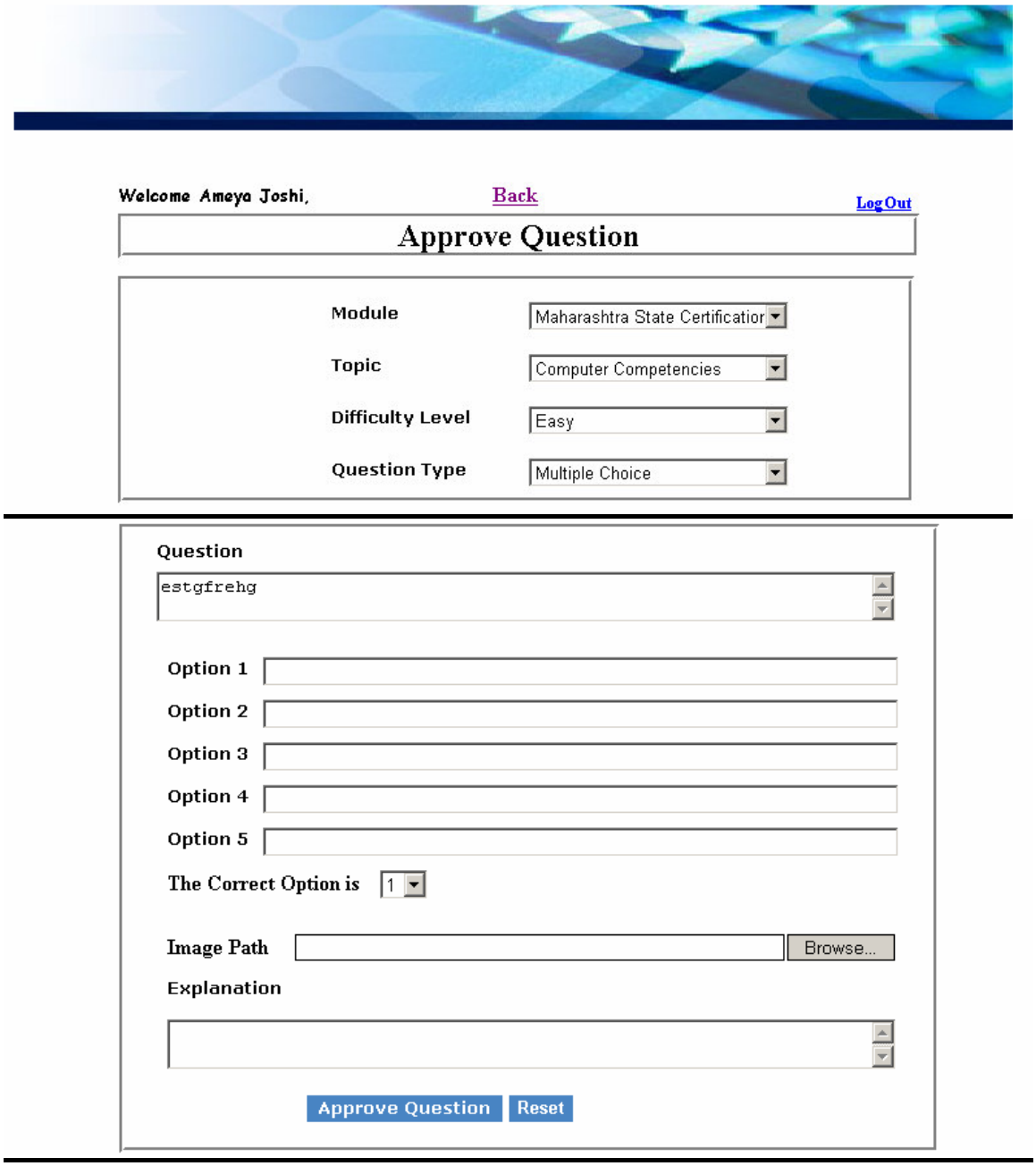
Interface 1: Login form:



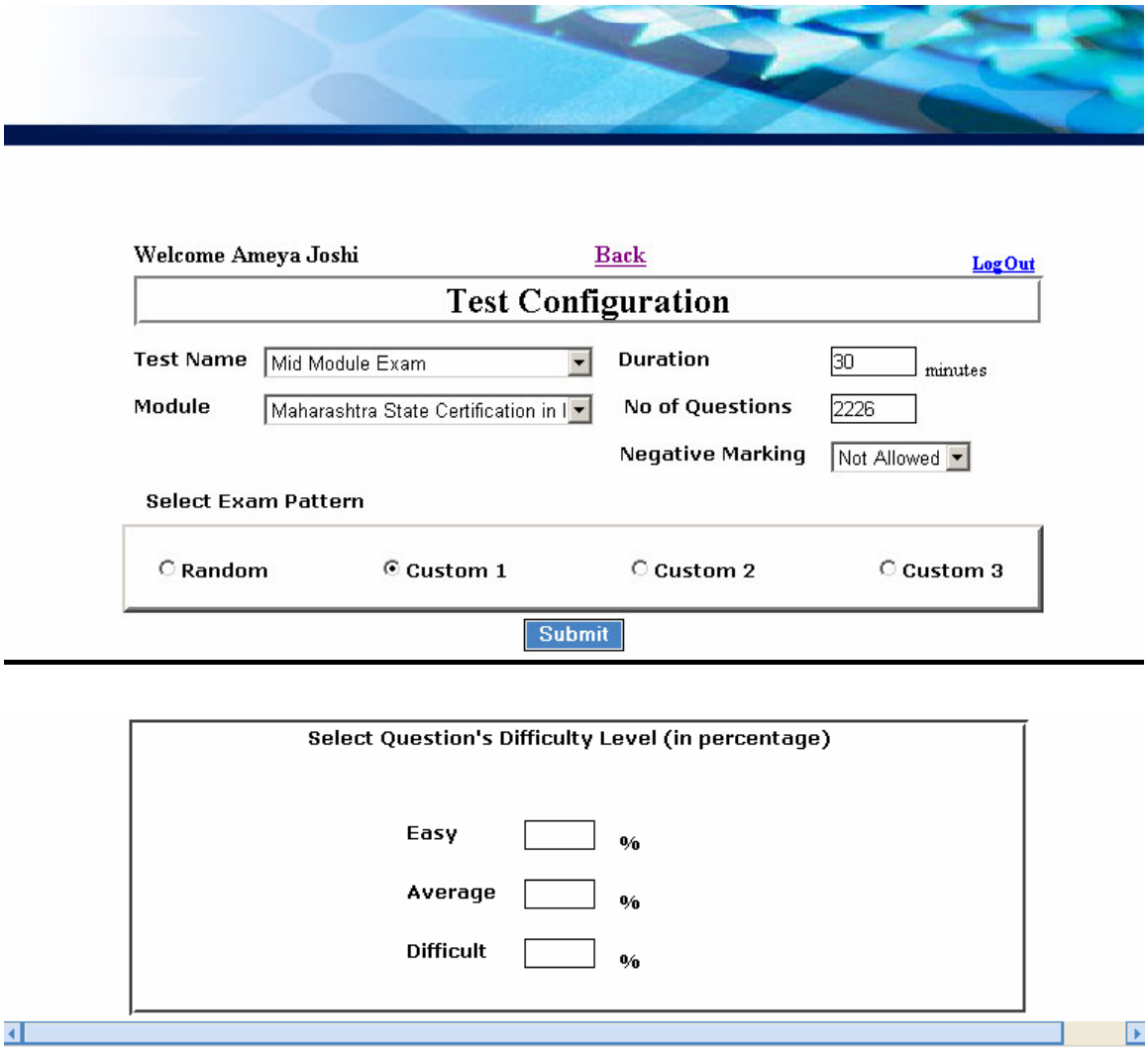
Interface 2:Question Entry



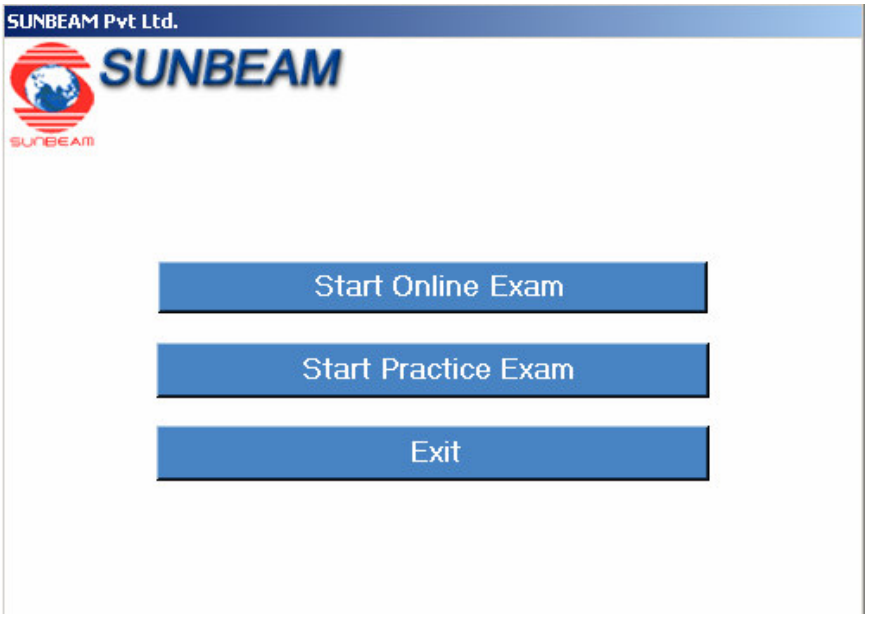
Interface 3: Approve Question



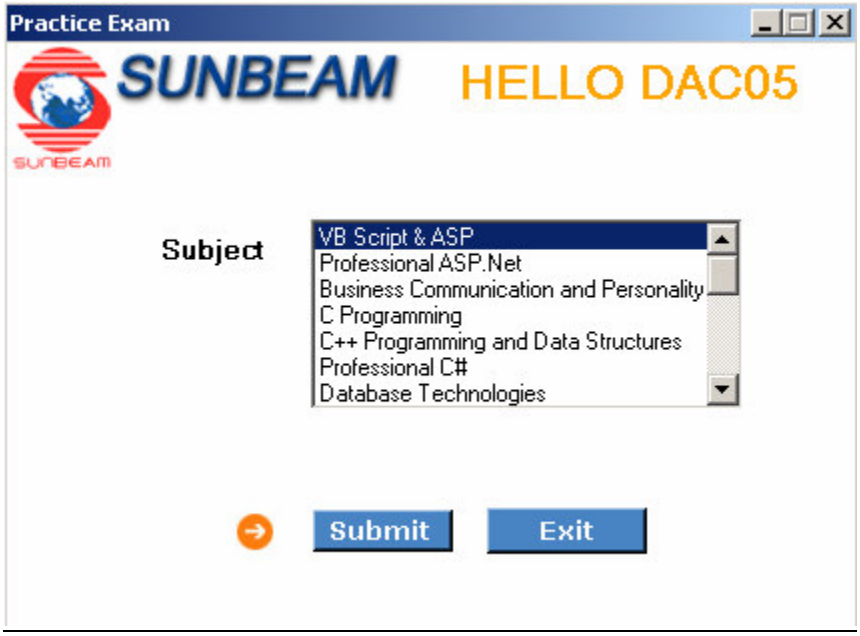
Interface 4:Test Configuration



Interface 5: Select Examination



Interface 6: Select Module for Practice Session



Interface 7: Practice Examination

Interface 8: Online Examination