DESIGN AND IMPLEMENTATION OF A QUESTION BANK

BY

WAHIDA HOSSAIN
ID: 113-25-227

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Masters of Computer Science and Engineering

Supervised By

Dr. Yousuf Mahbubul Islam
Professor
Department of CSE, CIS and CS
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
20TH JANUARY 2013

© Daffodil International University
APPROVAL

This Project titled “Design and Implementation of a Question Bank”, submitted by Wahida Hossain to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 20th January, 2013.

BOARD OF EXAMINERS

______________________________
Dr Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________________
Dr Yousuf Mahbubul Islam
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________________
Dr Md Kabirul Islam
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

______________________________
Dr Mohammad Shorif Uddin
Professor
Department of Computer Science and Engineering
Jahangirnagar University

© Daffodil International University
DECLARATION

We hereby declare that, this project has been done by us under the supervision of Dr Yousuf Mahbubul Islam, Professor, Department of Computer Science and Engineering Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

______________________________
Dr Yousuf Mahbubul Islam
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Submitted by:

______________________________
(Wahida Hossain)
ID: 113-25-227
Department of Computer Science and Engineering
Daffodil International University
ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

We would like to express our heartiest gratitude to Dr Syed Akhter Hossain, Professor and Head of Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We fell grateful to and wish our profound our indebtedness to Dr Yousuf Mahbubul Islam, Professor, Department of Computer Science and Engineering, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in Question bank influenced us to carry out this project .His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.
ABSTRACT

Teachers periodically need to set questions for tests, exams or assignments. When they sit down to compose a question, often they need to refer to previous questions. In addition, it is important that questions or question sets are not duplicated. The project “Design and Implementation of a Question Bank” is to help teachers maintain a question bank of their own subjects. The web-based software will allow viewing/importing questions and will also check for duplicate sets of questions.

The aim of the Question Bank is to automate the entire process under the observation of Top management panel. After using this system people can easily sort out the record of individual questions along with the previously organized list of question sets.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board of examiners</td>
<td>i</td>
</tr>
<tr>
<td>Declaration</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
</tbody>
</table>

# CHAPTER

## CHAPTER 1: INTRODUCTION

1.1 Knowledge evaluation
1.2 Communication
1.3 Questions-Bank database
1.4 User Panel Management
1.5 Problems of current system
1.6 Advantages of Automated Question Bank
1.7 Limitations of Automated Question Bank
1.8 Approach
1.9 Conclusion
CHAPTER 2: REQUIREMENT ANALYSIS

2.1 Introduction

2 Architecture

2.3 Implementation, Testing and Documenting

2.4 Deployment and maintenance

2.5 Overview of user panels

2.6 System Requirement Specification
   2.6.1 Operating System
   2.6.2 Web server
   2.6.3 Web programming language
   2.6.4 Web application Database
   2.6.5 Web browser
   2.6.6 IDE/Web development tools/Editor
   2.6.7 Graphics work

CHAPTER 3: MATERIALS AND METHODS

3.1 System overview

3.2 Objective

3.3 System Development Life Cycle

3.4 Applicability

3.5 Phases of system development issues

3.6 System study / initial idea

3.7 Feasibility study
3.8 Requirement analysis
3.9 System Analysis
3.10 System Design
3.11 Entity Relationship Diagram
3.12 The Administrator Application
3.13 The Teacher Panel Application
3.14 The Moderator Panel Application
3.15 Development and coding
3.16 Testing
3.17 Practice
3.18 Maintenance

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Use case Model
4.2 Use case model template
   4.2.1 Login
   4.2.2 Create user
   4.2.3 Add new subject
   4.2.4 Add new semester
   4.2.5 Add Question Category Name
   4.2.6 Add Profile
   4.2.7 Publish Question
   4.2.8 Add Question
   4.2.9 Activity model

© Daffodil International University
CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation of Data Model

5.2 Tables and Columns in a Database Diagram

5.3 Proposed Application Layout

   5.3.1 User Login Panel
   5.3.2 Super Admin Application Detail
   5.3.3 Main Dashboard
   5.3.4 Profile Entry Page
   5.3.5 Profile View Page
   5.3.6 Profile Update Page
   5.3.7 New User Entry Form Page
   5.3.8 List View of Users
   5.3.9 Add Subject Page
   5.3.10 List of Subjects
   5.3.11 Edit Subject Page
   5.3.12 Add Subject Navigation
   5.3.13 Add Semester Entry Form with List View
   5.3.14 Template Theme Selector
   5.3.15 Question Template Page
   5.3.16 Question Template Update
   5.3.17 Account Tracking
   5.3.18 Top User Tools
   5.3.19 Question Category Add and List View
CHAPTER 1
INTRODUCTION

Introducing Web based solution such as “Question bank automation” at educational institutes has encouraged the development of new methods and systems within the scope of education. Teachers, at the same time, could manage their regular question work through “on-line” setting and could arrange them so that whenever they want to use those questions no need to recreate. It will release a huge paper work effort and time.

In this project we have introduced the web-based "questions-bank" system, which can be used with any educational institute and may save hours in preparing exams questions as well as may save resources like, hard copy record keeping and file management etc. The proposed system has the following Functionalities and features:

1.1 Knowledge evaluation:
Teachers/Lecturers can evaluate their Question creating understanding levels and top management/institute head can track the activities and database of their teachers and can guide them to reach the pre-determined level of the questions.

1.2 Communication:
The system has a communication tool that allows internal management and teaches/lecturer to interact together and discuss their activities using feedback.

© Daffodil International University
1.3 Questions-Bank database:
The system allows the lecturers/teachers to create a Questions-Bank database that stores the previous exams, the reviews and if necessary update options.

1.4 Problems of current system:
- The current system has many problems. In this section we discuss those problems-
- Each and every question is currently creating manually.
- All the question making works are carried out in pen and paper.
- Record keeping is a general problem.
- File management according to year/semester/courses needs a lot of work.
- The current system is not efficient and takes long time.

*1.5 Advantages of Automated Question Bank:
- In absence of teacher/lecturer through online question submission is possible.
- Top level management/dept. head can track question creating and publishing system.

1.6 Limitations of Automated Question Bank:
- Student involvement is not present.
- Each layer of user panel, each user privilege/access level is different.
- Internet connection is a factor.
- Must take minimum training to operate the system perfectly.

1.7 Approach:
We had used the V-model to develop the system, this model started by specifying the system requirements, analyzed and classified the requirements into functional and nonfunctional requirements, constructed the use cases and domain model, the system design was started by determining the database tables and appropriate interfaces and finally, the implementation phase was launched by implementing the functionalities of the system.

1.8 Organization:

© Daffodil International University
Once the general requirements are gleaned from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

1.9 Conclusion:

Implementing this system will allow the institutions to create a Questions-Bank database. The system allowed the teachers/lectures to manage their records of questions in an effective way.

CHAPTER 2
REQUIREMENT ANALYSIS

2.1 Introduction:

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.

Specifying the system requirements using different gathering approaches the requirements were analyzed and classified into functional and nonfunctional requirements and use cases. The system design was started by determining the database tables, appropriate interfaces, class diagram: To show a static view of used classes and interaction diagrams: To show the collaborations among the objects of the system to complete an action The implementation phase was launched by implementing the functionalities of the system.

The important task in creating a software product is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result, but not what software should do. Incomplete, ambiguous, or even contradictory requirements are recognized by skilled and experienced software engineers at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Once the general requirements are gleaned from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document. Certain functionality may be out of scope of the project as a function of cost or as a result of unclear requirements at the start of development. If the development is done externally, this document can be considered a legal document so that if there are ever disputes, any ambiguity of what was promised to the client can be clarified.
2.2 Architecture

The architecture of a software system or software architecture refers to an abstract representation of that system. Architecture is concerned with making sure the software system will meet the requirements of the product, as well as ensuring that future requirements can be addressed. The architecture step also addresses interfaces between the software system and other software products, as well as the underlying hardware or the host operating system.

2.3 Implementation, Testing and Documenting

Implementation is the part of the process where software engineers actually program the code for the project.

Software testing is an integral and important part of the software development process. This part of the process ensures that bugs are recognized as early as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the authoring of an API, be it external or internal.

2.4 Deployment and maintenance

Deployment starts after the code is appropriately tested, is approved for release and sold or otherwise distributed into a production environment.

Software Training and Support is important because a large percentage of software projects fail because the developers fail to realize that it doesn't matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are often resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, it is very important to have training classes for new clients of your software.

Maintenance and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software. It may be necessary to add code that does not fit the original design to correct an unforeseen problem or it may be that a customer is requesting more functionality and code can be added to accommodate their requests. It is during this phase that customer calls come in and you see whether your testing was extensive enough to uncover the problems before customers do. If the labor cost of the maintenance phase exceeds 25% of the prior-phases'
labor cost, then it is likely that the overall quality, of at least one prior phase, is poor. In that case, management

Requirements analysis is critical to the success of a systems or software project. The steps that we follow for requirement analysis are:

- Research on Application Requirements,
- Find out Technical Issues,
- Application Basic Features,
- Use Cases,
- User Interface Specification,
- User Requirements Specification,
- Architecture,
- Data Fields,
- Time Estimation,

2.5 Overview of user panels:

1. Super admin / Top level Management (Single)
2. Exam moderator/Mid level Management – (Unlimited)
3. Teacher / Lecturer - (Unlimited)

![Question Bank System Diagram]

Fig 1: System user panel

2.6 System Requirement Specification:

2.6.1 Operating System:

© Daffodil International University
Windows XP Professional: XP is a client operating system, it can function as a server as a server peer to peer network.

Windows NT: Its designed to control the site is deployed. NT server is widely used to support commercial Web sites.

2.6.2 Web server:

- **Personal web server**: Personal web server is an abbreviated version of internet information server and was made available as an NT options.

- **Apache server**: Apache server is an open source HTTP server for windows, UNIX system and other platforms. Apache features highly configurable error messages.

2.6.3 Web programming language:

- **PHP**: It is an open source server-side scripting language designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data.

   PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

2.6.4 Web application Database:

- **MySQL**: It is the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is a relational database management system. SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard.

2.6.5 Web browser

- Internet explorer
Netscape
Opera
Mozilla Firefox
Google chrome

2.6.6 IDE/Web development tools/Editor:

➢ Backend coding-
  ▪ PHP storm
  ▪ Hypedit

➢ GUI or front end design-
  ▪ Adobe Dreamweaver

2.6.7 Graphics work:

➢ Adobe photoshop
➢ CorelDRAW X5
CHAPTER 3
MATERIALS AND METHODS

3.1 System overview:

The system interface is entirely web-based and doesn’t necessitate any technical skills from the potential users. The system is divided into three applications, called administrator, teacher and moderator application.
3.2 Objective:

- Ability to capture aspect of question papers previously impossible to capture.
- Reduced workload.
- Timely feedback/evaluations.
- Improved quality of feedback/evaluations.
- Multiple question making are possible.
- Making our educational system more digital.

### 3.3 System Development Life Cycle

The **waterfall model** is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design (validation), Construction, Testing and maintenance.

The first formal description of the waterfall model is often cited to be an article published in 1970 by Winston W. Royce (1929–1995), although Royce did not use the term "waterfall" in this article. Ironically, Royce was presenting this model as an example of a flawed, non-working model (Royce 1970). This is in fact the way the term has generally been used in writing about software development—as a way to criticize a commonly used software practice.

In Waterfall model, the following phases are followed in order:

1. Requirements specification
2. Design
3. Construction (implementation or coding)
4. Integration
5. Testing and debugging (Validation)
6. Installation
7. Maintenance
3.4 Applicability

Many traditional software development projects follow the waterfall model. In this case projects are similar to the Software Development Life Cycle (SDLC) which includes 5 main phases: Requirements Phase, Design Phase, Development Phase, System Testing, and Release to Production.

3.5 Phases of system development issues

- System Development life cycle
- Prototyping
- Forming the development
- Technical issues

3.6 System study / initial idea

System study is the first stage of system development life cycle. This gives a clear picture of what actually the physical system is?
To describe the system study phase more analytically, we would say that system study phase passes through the following steps:

- Problem identification
- Background analysis
- Inference or findings

3.7 Feasibility study

Operational feasibility study tests the operational scope of the software to be developed. The proposed software must have high operational feasibility.

3.8 Requirement analysis

Specifying the system requirements using different gathering approaches the requirements were analyzed and classified into functional and nonfunctional requirements and use cases.

3.9 System Analysis

System analysis involved a detailed study of the current system, leading to specification of a new system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files and transactions handled by the present system. Interviews, on site observation and questionnaire are the tools used for system analysis.

3.10 System design

Based on the user requirements and the detailed analysis of a system, the new system is designed. Normally design proceeds in two stages:

- Preliminary or general design
- Structure or detailed design

3.11 Entity Relationship Diagram

In software engineering, an **Entity-Relationship Model** (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* or *ERDs* for short.
3.12 The Administrator Application

This application provides the Super administrator with simple tools to manage the information of the Questions-Bank system.

The following are some features provided by the administrator application:

- Export the questions-bank database
- User panel create
- User management
- Access level define
- Question template management
- User record export
- View questions, feedback and level of standard
- Own profile management
- Login information change/update
- Track own account
- Add new/ Delete semester
- Add new /Delete subjects
3.13 The Teacher Panel Application:

This application provides the teacher application with simple tools to handle the information of the Questions-Bank system.

The following are some features provided by the teacher application:

- Create Question and add to Question Bank
- Export the Question Bank Record
- Account profile management
- Question template view
- Give questions the level of standard
- Update levels
- Edit/change existing questions
- View questions, feedback and level of standard
- Add account profile
➢ Login information change/update
➢ Track own account
➢ View semester list
➢ View subject list

Fig 5: Teacher application Flow chart

3.14 The Moderator Panel Application

This application provides the moderator application with simple tools to handle the information of the Questions-Bank system.

The following are some features provided by the moderator application:

➢ Give feedback based on questions
➢ Print questions
➢ View list of questions
➢ Add account profile

© Daffodil International University
➢ Login information change/update
➢ Track own account
➢ Account profile management

![Moderator panel flow chart]

**Fig 6: Moderator application Flow chart**

### 3.15 Development and coding

After designing the new system, the whole system is required to be converted into computer understanding language. Coding the new system into computer programming language does this. It is an important stage where the defined procedures are transformed into control specification by the help of a computer language. This is also called programming phase.

### 3.16 Testing

Before actually implementation the new system into operations, a test run of the system is done to remove all bugs, if any. It is an important phase for a successful system. It is run on a given set of data; the output of the test run should match the expected results.
3.16 Practice

After having the user acceptance of the new system developed, the Practice phase begins. After loading the system training of the users starts. Main topics of such type of training are:

- How to execute the program
- How to enter the data
- How to process the data
- How to take out the reports

3.17 Maintenance

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected.
CHAPTER 4
ANALYSIS AND DESIGN

4.1 Use case Model

A Use Case Model describes the proposed functionality of a new system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system. This interaction is a single unit of meaningful work, such as Create Account or View Account Details.

Each Use Case describes the functionality to be built in the proposed system, which can include another Use Case's functionality or extend another Use Case with its own behavior.

A Use Case description will generally includes:

- General comments and notes describing the use case.
- Requirements - The formal functional requirements of things that a Use Case must provide to the end user, such as <ability to update order>. These correspond to the functional specifications found in structured methodologies, and form a contract that the Use Case performs some action or provides some value to the system.
- Constraints - The formal rules and limitations a Use Case operates under, defining what can and cannot be done. These include:
  - Pre-conditions that must have already occurred or be in place before the use case is run; for example, <create order> must precede <modify order>
  - Post-conditions that must be true once the Use Case is complete; for example, <order is modified and consistent>
  - Invariants that must always be true throughout the time the Use Case operates; for example, an order must always have a customer number.
- Scenarios – Formal, sequential descriptions of the steps taken to carry out the use case, or the flow of events that occur during a Use Case instance. These can include multiple scenarios, to cater for exceptional circumstances and alternative processing paths. These are usually created in text and correspond to a textual representation of the Sequence Diagram.
- Scenario diagrams - Sequence diagrams to depict the workflow; similar to Scenarios but graphically portrayed.
- Additional attributes, such as implementation phase, version number, complexity rating, stereotype and status.
Actors

Use Cases are typically related to 'actors', which are human or machine entities that use or interact with the system to perform a piece of meaningful work that helps them to achieve a goal. The set of Use Cases an actor has access to defines their overall role in the system and the scope of their action. **Includes and Extends** relationships between Use Cases

One Use Case could include the functionality of another as part of its normal processing. Generally, it is assumed that the included Use Case is called every time the basic path is run. For example, when listing a set of customer orders to choose from before modifying a selected order, the <list orders> Use Case would be included every time the <modify order> Use Case is run.

A Use Case can be included by one or more other Use Cases, so it helps to reduce duplication of functionality by factoring out common behavior into Use Cases that are reused many times.

One Use Case can extend the behavior of another, typically when exceptional circumstances are encountered. For example, if a user must get approval from some higher authority before modifying a particular type of customer order, then the <get approval> Use Case could optionally extend the regular <modify order> Use Case.

Fig 7: Entire use case model

© Daffodil International University
4.2 Use case model template

4.2.1 Login

Use case: login
Actor: super admin
Primary path:
1. key in: username.
2. key in: password.
3. Click login.
Alternative path:
Wrong username and password, try again!

4.2.2 Create user

Use case: Add new user
Actor: super admin

Primary path:
1. key in: Name.
2. key in: Username.
3. key in: password.
4. key in: Confirm Password.
5. key in: Status.
6. key in: Type.
7. Click Add New.
Alternative path:
Same username already exist, try again!

4.2.3 Add new subject

Use case: Add Subject
Actor: super admin

Primary path:
1. key in: Credit Hr.
2. key in: Subject Code.
3. key in: Subject Name.
4. key in: Subject Type.
5. Click Add New.
Alternative path:
Same Subject Code already exists, try again!

4.2.4 Add new semester

Use case: Add new semester
Actor: super admin

Primary path:
1. key in: Semester Name
2. Click Add New.

4.2.5 Add Question Category Name

Use case: Category Name
Actor: Teacher

Primary path:
1. key in: Category Name.
2. Click Add New.

4.2.6 Add Profile

Use case: Add my profile
Actor: Teacher, Moderator, Super admin/ dept. head

Primary path:
1. Key in: Name.
2. Key in: Designation.
3. Key in: E-mail address.
4. Key in: Contact number.
5. Key in: Address.
6. Click Add New.

4.2.7 Publish Question

Use case: Publish question
Actor: Teacher

Primary path:
1. key in: Year
2. key in: Semester
3. Key in: Exam Type.
4. Key in: Subject Name.
5. key in: Question List
6. Click Add New.

4.2.8 Add Question

Use case: Add question
Actor: Teacher

Primary path:
1. key in: Question Category
2. key in: Question.
3. key in: Mark.
4. key in: Marking Scheme.
5. Click Add New.

4.2.9 Login Activity model

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

User gives user ID and password then click login button, if its correct user id and password then state is full but if user gives invalid information then it will be back to first state. The following figure shows the activity model for login class.
CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation of Data Model
The following figure shows the overall Database Plan of the proposed “Question Bank management system”.

5.2 Tables and Columns in a Database Diagram

© Daffodil International University
Within a database diagram, each table can appear with three distinct features: a title bar, a row selector, and a set of property columns.

**Title Bar**

The title bar shows the name of the table.

**Row**

It displays a key symbol if the column is in the table's primary key.

### 5.3 Proposed Application Layout

#### 5.3.1 User Login Panel

![Login panel](image)

Fig 10: Login panel

#### 5.3.2 Super Admin Application Detail

#### 5.3.3 Main Dashboard:
5.3.4 Profile Entry Page:
If user not yet been created the profile gets this form to fill it up. Otherwise he/she will redirected to the display profile page.
5.3.5 Profile View Page:

Fig 13: Profile View Page

5.3.6 Profile Update Page:

Fig 14: Profile Update

5.3.7 New User Entry Form Page:

© Daffodil International University
5.3.8 List View of Users:

Fig 15: New user form

Fig 16: List of users (Account Enabled)

Fig 17: Confirmation Box

Fig 18: List of users (Account Disabled)
5.3.9 Add Subject Page

Fig 19: Add subject form

5.3.10 List of Subjects

Fig 20: List subjects

5.3.11 Edit Subject Page

© Daffodil International University
Fig 21: Edit subject

5.3.12 Add Subject Navigation

![List of Subjects Table]

Fig 22: Top navigation

5.3.13 Add Semester Entry Form with List View

![Add Semester Form]

Fig 23: Add Semester and list

5.3.14 Template Theme Selector

![Theme Settings]

Fig 24: Utility Settings

Example 1

© Daffodil International University
Example 2
Example 3

Example 4

© Daffodil International University
5.3.15 Question Template Page

Fig 25: Question Template View

5.3.16 Question Template Update

Fig 26: Question Template Update

© Daffodil International University
5.3.17 Account Tracking

Fig 27: Account Pop-up

5.3.18 Top User Tools

Fig 28: Top user tool

5.3.18 Teacher Panel

5.3.19 Question Category Add and List View

© Daffodil International University
5.3.20 Add New Questions

![Fig 30: Add New Question](image)

5.3.21 List View of Question

![Fig 31: List of Question](image)
5.3.22 Question Update Page

![Question Update Form](image)

**Fig 32: Question Update**

5.3.23 Set Level at Questions

![Question Level Settings](image)

**Fig 33: Question Level Settings**
CHAPTER 6
FUTURE SCOPE AND CONCLUSION

Future Scope
Introduction a live chat process may reduce the communication gap between teachers and management. In future advance level I will integrate the AJAX live chat module for faster communication system.

Still we are not able to provide a variation of languages, now only English language is available; in future we will provide this facility to our users.
An additional operator panel will be added for record keeping and time by time backup facility.

Conclusion
The system incorporates all the modern features of a smart web site and hence the system is very user friendly. The CSS and template design is also accessible by portable tools such as iPhone, touch pad and other devices.

Summery
In this system we have used several tools and languages that make it efficient one. In future our system will act as a bridge between the manual / traditional format and automated systems in our educational infra structure.

Limitation
In this system student panel is not added. So student involvement with the system is not possible. System is design for a one language, so it can to support only English language
part. Insertion of the images and figures at the question set is not possible at this version. In future extension we will include editor to generate question paper.

**References**

[1] General Understanding of Distance Learning  

http://www.extension.harvard.edu/distance-education/video-distance-education-experience (Accessed on date: 05.01.2013)

[3] Online Examination System in PHP and MySQL with Trends Graph for a Maritime Academy  

[4] Clinical and Basic Science MCQs  

[5] writing across the curriculum from kindergarten to adult  
http://sjsdblogs.com/ (Accessed on date: 09.01.2013)


[7] SQL Server  

[8] Category: Object-oriented programming  
[9] Waterfall paper winston

[10] Use case


[12] Publication of CO600/CO620 Project Reports

[13] The Use Case Model

[14] Edublogs Campus
http://edublogs.org/campus/#TB_inline?width=500&height=290&inlineld=embedded-video
(Accessed on date: 14.01.2013)

[15] Results from ReadWriteThink
http://www.readwritethink.org/search/?strategy-guide-series=30098  (Accessed on date: 15.01.2013)

[16] CEPA Europe Study Abroad Journals
http://studyabroadcepa.blogspot.de/  (Accessed on date: 15.01.2013)

[17] online training courses
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURES</th>
<th>PAGE NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig 1: System user panel</td>
<td>15</td>
</tr>
<tr>
<td>Fig 2: System overview</td>
<td>18</td>
</tr>
<tr>
<td>Fig 3: Different phases of software development life cycle</td>
<td>20</td>
</tr>
<tr>
<td>Fig 4: Super admin application Entity Relationship Diagram</td>
<td>23</td>
</tr>
<tr>
<td>Fig 5: Teacher application Entity Relationship Diagram</td>
<td>24</td>
</tr>
<tr>
<td>Fig 6: Moderator application Entity Relationship Diagram</td>
<td>25</td>
</tr>
<tr>
<td>Fig 7: Entire use case model</td>
<td>28</td>
</tr>
<tr>
<td>Fig 8: Login activity model</td>
<td>31</td>
</tr>
<tr>
<td>Fig 9: Login activity model</td>
<td>32</td>
</tr>
<tr>
<td>Fig 10: Login activity model</td>
<td>33</td>
</tr>
<tr>
<td>Fig 11: Login activity model</td>
<td>33</td>
</tr>
<tr>
<td>Fig 12: Login activity model</td>
<td>33</td>
</tr>
<tr>
<td>Fig 13: Profile View Page</td>
<td>34</td>
</tr>
<tr>
<td>Fig 14: Profile Update</td>
<td>35</td>
</tr>
<tr>
<td>Fig 15: New user form</td>
<td>35</td>
</tr>
<tr>
<td>Fig 16: List of users</td>
<td>35</td>
</tr>
<tr>
<td>Fig 17: Confirmation Box</td>
<td>36</td>
</tr>
<tr>
<td>Fig 18: List of users (Account Disabled)</td>
<td>36</td>
</tr>
<tr>
<td>Fig 19: Add subject form</td>
<td>36</td>
</tr>
</tbody>
</table>

© Daffodil International University
Fig 20: List subjects
Fig 21: Edit subject
Fig 22: Top navigation
Fig 23: Add Semester and list
Fig 24: Utility Settings
Fig 25: Question Template View
Fig 26: Question Template Update
Fig 27: Account Pop-up
Fig 28: Top user tool
Fig 29: Question Category
Fig 30: Add New Question
Fig 31: List of Question
Fig 32: Question Update
Fig 33: Question Level Settings