|  |
| --- |
| THAPAR UNIVERSITY |
| Library Content Management System |
| Software engineering project report |
|  |
| **Gursimran singh, Yuvraj singla** |
| **12/1/2011** |

|  |
| --- |
|  |

Table of Contents

[Problem statement 1](#_Toc310501034)

[DFD Diagrams 1](#_Toc310501035)

[CFD 2](#_Toc310501036)

[DFD Level1: 2](#_Toc310501037)

[Software specification diagram 4](#_Toc310501038)

[Introduction 4](#_Toc310501039)

[Purpose 4](#_Toc310501040)

[Document Conventions 4](#_Toc310501041)

[Intended Audience and Reading Suggestions 4](#_Toc310501042)

[Project Scope 4](#_Toc310501043)

[References 4](#_Toc310501044)

[Overall Description 4](#_Toc310501045)

[Product Perspective 4](#_Toc310501046)

[Product Features 5](#_Toc310501047)

[User Classes and Characteristics 5](#_Toc310501048)

[Operating Environment 5](#_Toc310501049)

[Design and Implementation Constraints 5](#_Toc310501050)

[User Documentation 5](#_Toc310501051)

[System Features 6](#_Toc310501052)

[Upload ‘Question paper’ 6](#_Toc310501053)

[Description and priority 6](#_Toc310501054)

[Delete/ update question paper 6](#_Toc310501055)

[Search question paper 6](#_Toc310501056)

[Download question paper 7](#_Toc310501057)

[External Interface Requirements 7](#_Toc310501058)

[User Interfaces 7](#_Toc310501059)

[Hardware Interfaces 8](#_Toc310501060)

[Communications Interfaces 8](#_Toc310501061)

[Other Nonfunctional Requirements 8](#_Toc310501062)

[Performance Requirements 8](#_Toc310501063)

[Safety Requirements 8](#_Toc310501064)

[Software Quality Attributes 8](#_Toc310501065)

[Other Requirements 8](#_Toc310501066)

[Appendix A: Glossary 9](#_Toc310501067)

# Problem statement

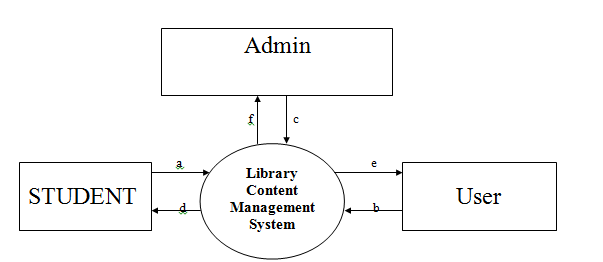
The quality of service that a library provides depends greatly on its content management system. As the libraries are growing every day, it has become practically impossible to manage content manually. So it is the need of the hour to fully automate the library content management system and make it efficient. Larger volume of books and material doesn’t always ensure that library is reader friendly. On the contrary, it can become more chaotic if the books are not managed well. So content management system is one of the most crucial part of library.

Our major concern of content management system is the ‘*management of question papers*’ and ‘*class notes*’ in our Nava Nalanda Central library of Thapar university, Patiala. This must be of major concern to any university or organisation, as this is the part of ‘knowledge management’, which is very crucial to any organisation. The present system of managing and storing question papers has only a browsing interface and is not searchable. Moreover question papers are arranged by codes, which change very frequently and are very difficult to identify. So students/ users face a lot of difficulty in locating question papers on the website. We aim to design a database and make the system searchable with a variety of different queries.

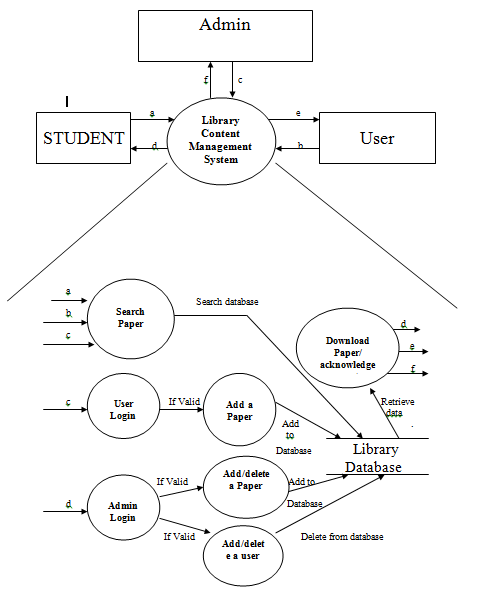
Moreover care has been taken the proposed system is compatible with the existing infrastructure so minimal change is required to update the present semi-automated system to our fully automatic solution. Many ‘out of box’ ideas have been accommodated to make the system indigenous and efficient.

# DFD Diagrams

## CFD



## DFD Level1:



# Software specification diagram

## Introduction

### Purpose

The product “Library content management system” is a product that is focussed for managing information and knowledge in the library. The information and knowledge that the product proposes to manage, includes student study notes and previous year question papers.

### Document Conventions

The document divides different sections by numbers in font ‘Times New Roman’ of size ‘18’ .Sub-sections are numbered in further numbering of size ‘14’ and the text-size in paragraphs is ‘12’.

### Intended Audience and Reading Suggestions

The document includes generic use cases for different types of users that can access the software. Apart from that system requirements and minimum system requirements are covered for installing the product. This document can be used by developers/ users to get an overview of the product and its main features.

This document is intended for:

* Developers
* Programmers
* Users
* Managers

### Project Scope

Library content management system” is designed to fit in the existing infrastructure of the Nava Nalanda Central library and will include participation ranging from faculty, library staff and students. These ‘actors’ will have different access rights according to their roles, so that coordination can be achieved and knowledge maintained without any hassle.

### References

1. *Problem statement*
2. [*http://cl.thapar.edu/*](http://cl.thapar.edu/)
3. [*http://cl.thapar.edu/library\_qp.html*](http://cl.thapar.edu/library_qp.html)

## Overall Description

### Product Perspective

The major concern of ‘Library Content Management System’ is the ‘management of question papers’ in our Nava Nalanda Central library of Thapar university, Patiala. This must be of major concern to any university or organisation, as this is the part of ‘knowledge management’, which is very crucial to any organisation. Because already a semi-automatic system exist, care has been taken the proposed system is compatible with the existing infrastructure so minimal change is required to update the present system to our fully automatic solution.

### Product Features

1. LCMS provides controlled access with appropriate rights to students who want to download question papers and library staff who can download/ upload or delete existing question papers.
2. Librarian administrator (super admin) has rights to add a new administrators and ban/ unban existing administrators.
3. Question papers can be added/ updated by the library staff on the fly, remotely from anywhere, without having actual access to the webserver.
4. Similarly questions papers can be deleted remotely from anywhere.
5. Students have multiple options of searching question papers by:
   1. Name of subject
   2. Semester (Even/ Odd)
   3. Time of subject taught.
      1. Year
      2. MST/ EST
   4. Faculty coordinator.
   5. Code of Subject.

### User Classes and Characteristics

The following types of users will use the interface

* Students
* Library staff

### Operating Environment

The product is designed to run in a webserver, supporting PHP scripting and a database management system like MySQL. The servers should provide a robust and fast environment for the product. It should cater the needs of its many users both faculty and users and support parallel access without any lag and avoid downtime.

These features are required in the webserver where the product is supposed to be hosted

* Servers with Top Line Hardware specs.
* Dedicated MySQL server to handle SQL Queries load.
* Remote weekly backups of server’s data for safety of data.

### Design and Implementation Constraints

* **Memory Constraint**: As mobiles have limited amount of RAM so while developing mobo apps developer has to take that into account.
* **Display Constraint:** Screen size of mobiles as compared to desktops so developer has to take care of it while developing the GUI..
* **Processor Constraints**: Processor is not that fast as that of computer.
* **Software Constraints**: Java installed on the device.
* **Device Constraint:** Android

### User Documentation

For making experience better, the following documentation is supplied along with the product:

* Operator side help.
* Operator side troubleshooting.
* Instruction for Students.

## System Features

### Upload ‘Question paper’

### Description and priority

This is used to upload the question paper whose various details are provided from the interface. The details include the following items. Various security and data integrity features has been embedded to maintain correctness of data being entered. The question paper has to be uploaded in *pdf* format.

* Name of subject
* Year of exam
* Type of Exam – [EST|MST]
* Semester of question paper
* Faculty coordinator
* Code of Subject

#### Stimulus/Response Sequences

The server will receive data along with *pdf* file of question paper and execute a INSERT INTO query for updating the data in the question paper database. Following this data is entered and the file is available for students to download.

### Delete/ update question paper

#### Description and priority

The library staff user has privileges for updating/ deleting a question paper. For that library staff has to login in the backend where the user can view the list view of all question papers whose details can be updated or the whole record including file can be deleted.

#### Stimulus/Response Sequences

The server will take the request and from the backend web interface and run a DELETE FROM query to remove the record from the database and delete the corresponding file from the file system.

### Search question paper

#### Description and priority

The users can use one of the various ways (listed below) to search a question paper from the database and get the results from the server in a list view form. The question papers relevant to search can be downloaded by clicking on the download button.

* Name of subject
* Year of exam
* Type of Exam – [EST|MST]
* Semester of exam
* Faculty coordinator
* Code of Subject

#### Stimulus/Response Sequences

The server processes the request for searching the question papers using the various attributes in the question paper databases using the SELECT query. The results are sent to the client for downloading.

### Download question paper

#### Description and priority

Students can download the question paper by clicking a download link and save the file in their own file system. After clicking the download button a window will appear asking the user where he/she wants to save the file in the file system. Having located the correct location the user can click on the save button to save the file.

#### Stimulus/Response Sequences

The server will calculate the address of the file using the various details in the database. Having done so, the server presents the file to the downloading program in the user’s computer system which will download the file over HTTP protocol.

## External Interface Requirements

### User Interfaces

The main page of product will contain user interfaces for

* Searching a question paper
* Logging on to backend

#### Searching a question paper

The user looking for a question paper can search using various options. So the user is required to select one of the available ways as listed below:

* Name of subject
* Year of exam
* Type of Exam – [EST|MST]
* Semester
* Faculty coordinator
* Code of Subject

Having selected one of the ways and entered the required information, the user can issue the search command using the Search button.

#### Search results interface

The server will receive the user’s request and will process it to get a list of results. This list of results is displayed in a list view along with various attributes like year/ type/ faculty coordinator etc with a Download button alongside for downloading a question paper.

#### Logging on to backend

The main page of the product will contain a column from where the library staff can log-in to backend for administration purposes like

* Uploading a question paper
* Updating/ Deleting a question paper

The library staff has to provide its credentials and get validated using the Log-in button provided. Having got validated the backend is provided to the library staff for administration purposes.

#### Backend interface

The backend will contain options for uploading and updating/deleting a question paper. The user is expected to provide information (attributes) of the file while preforming the administrative tasks.

### Hardware Interfaces

The server running web-server software is required to run the product properly. The server should have sufficient hardware resources like primary memory, secondary memory to satisfy the needs of the server software and the database. The server should be equipped with a network interface from which it should be able to communicate using TCP/IP within the LAN/ WAN.

#### Software Interfaces

The web-server software should support the following features for running the product.

* PHP scripting support
* Dedicated MySQL server to handle SQL Queries load.
* Remote weekly backups of server’s data for safety of data.

### Communications Interfaces

The server should be equipped with ethernet interface for cat-5 or faster cable such as RS-232. The product should be able to communicate using TCP/IP protocol with the clients.

## Other Nonfunctional Requirements

### Performance Requirements

1. The product should handle the various requests from different users trying to download question papers all at once.
2. These product should be intelligent and robust to handle parallel accesses should be properly handled by the product so that it can withstand without failing or generating unnecessary lag.
3. The product should deliver its services without any downtime and should have self-recovery functionality.

### Safety Requirements

Not applicable.

### Software Quality Attributes

* Interoperability
* Maintainability
* Portability
* Reliability
* reusability

## Other Requirements

1. The user must be agreed with all term and conditions that have provided by the System Administrator.
2. Legal requirements to get the **copyright** of the project

## Appendix A: Glossary

**SDK** - Software Development Kit

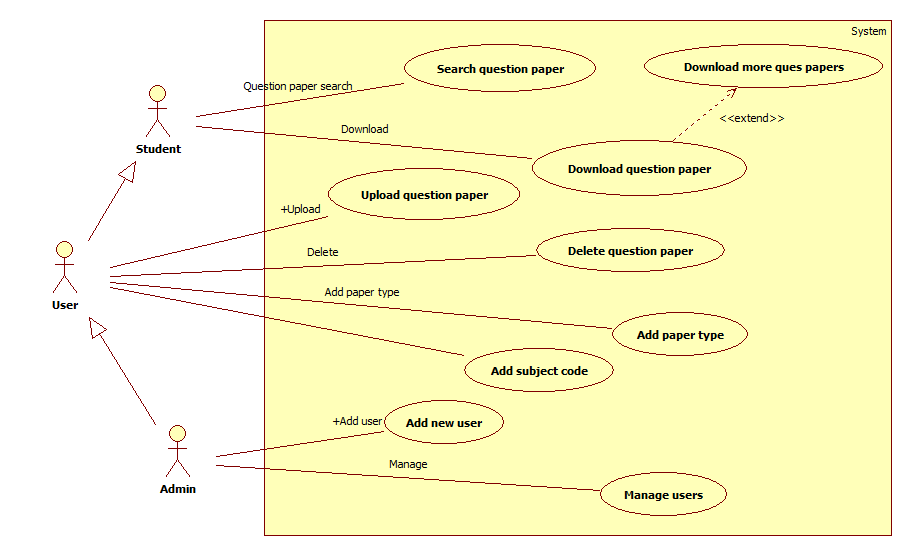
I**DE** - Integrated Development Kit

**PHP -** Hypertext Preprocessor

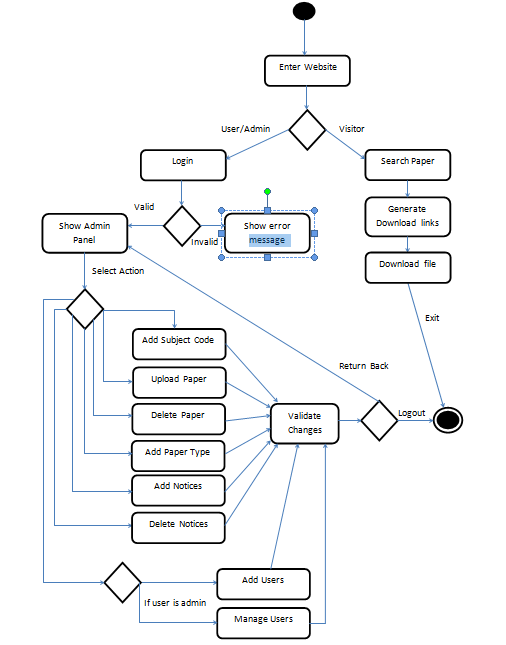
**SQL** – Structured Query Language

# Unified modelling language diagrams

## Use case diagram

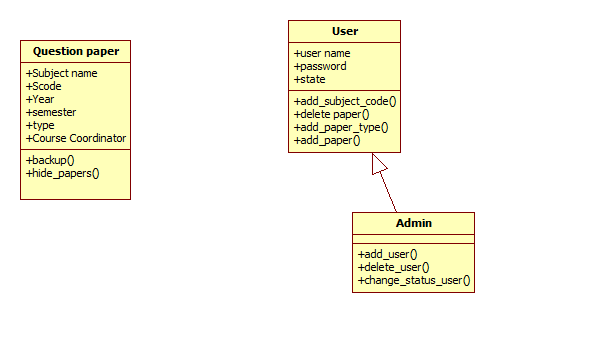


## Activity diagrams



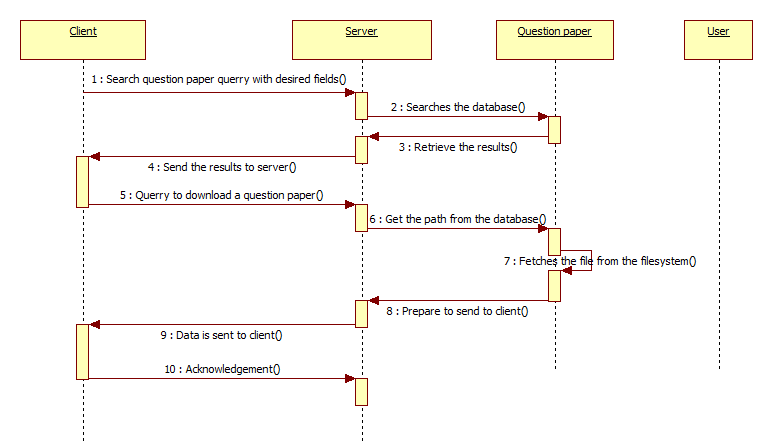
## Swimlane diagram

## Class Diagram

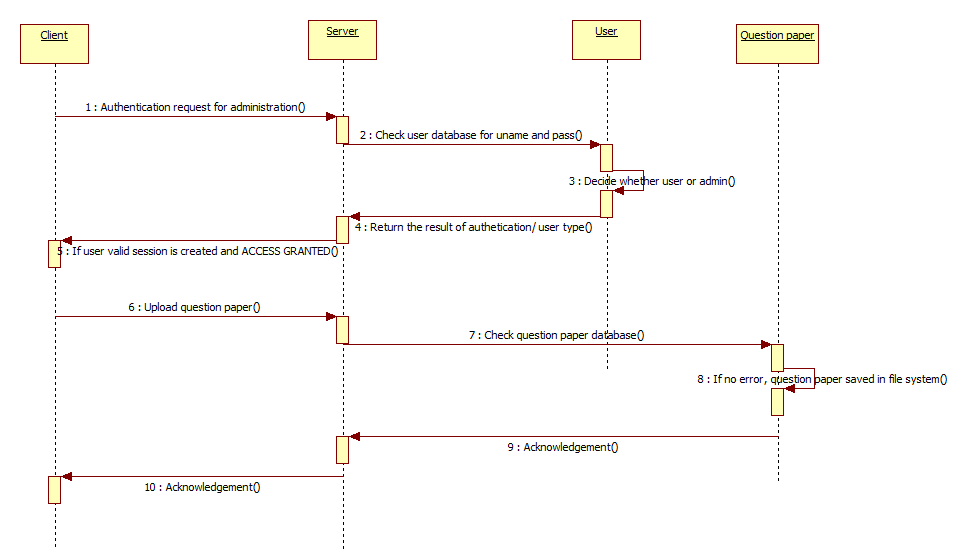


## Sequence Diagram:

### Sequence diagram – Download question paper

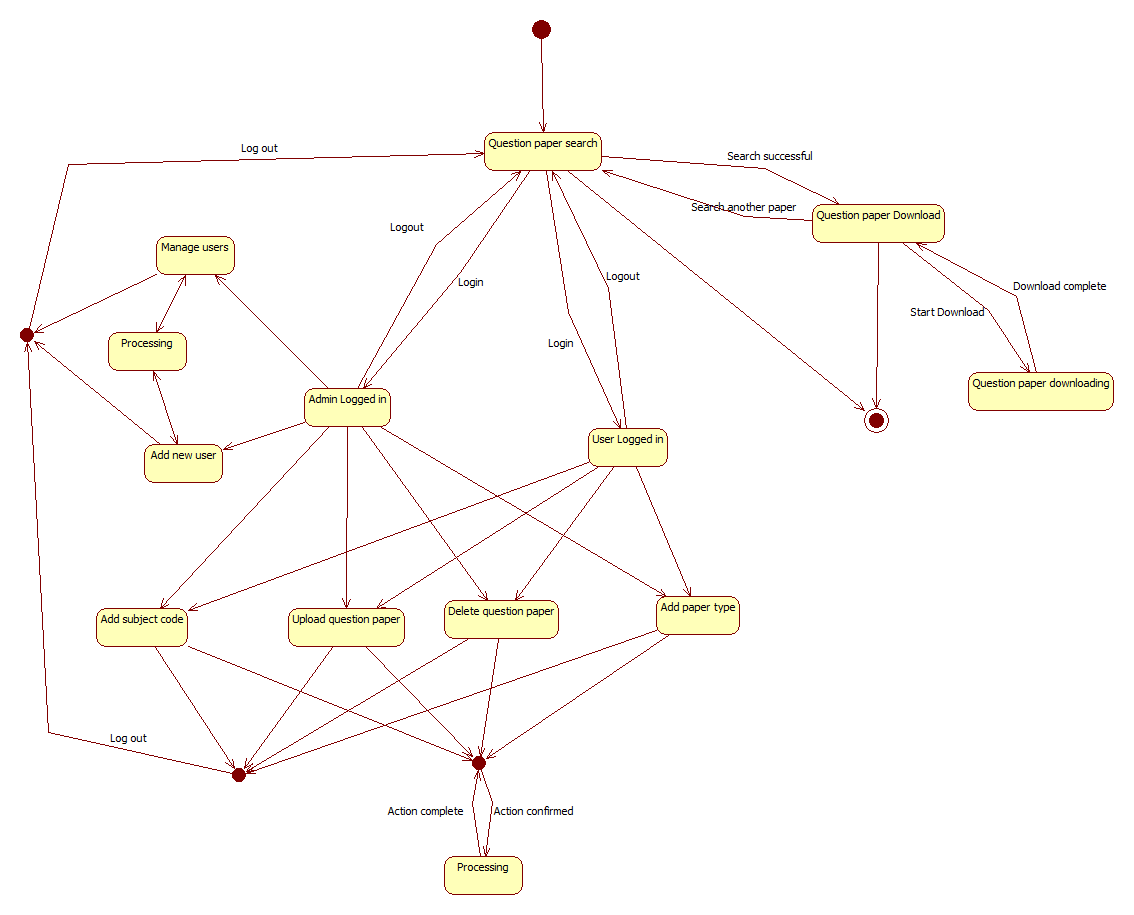


### Sequence diagram – Upload question paper

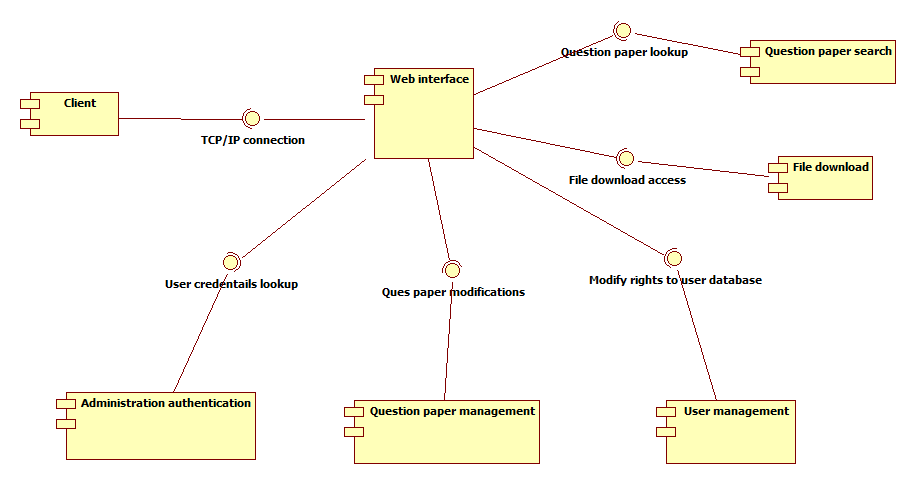


## Collaboration Diagram:

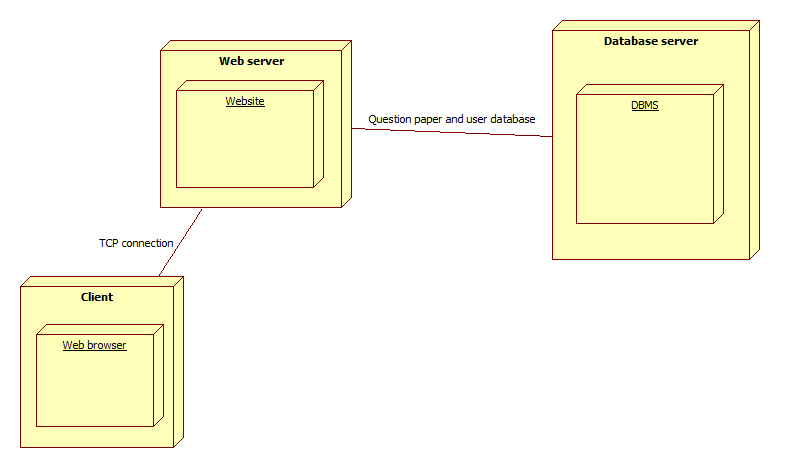
## State Transition Diagram:



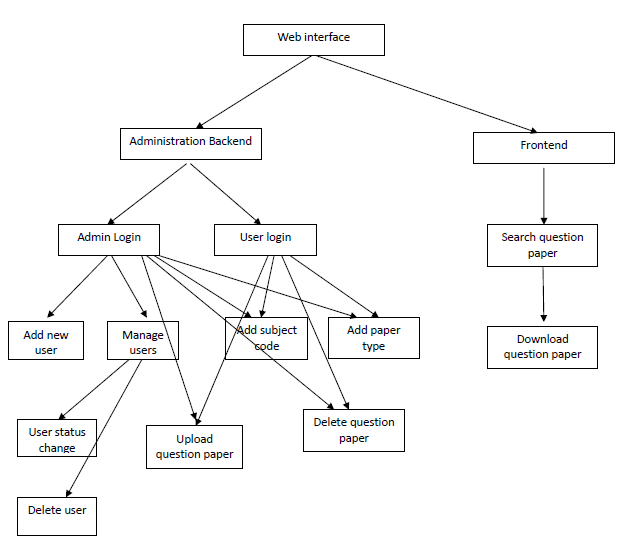
## Component Diagram:



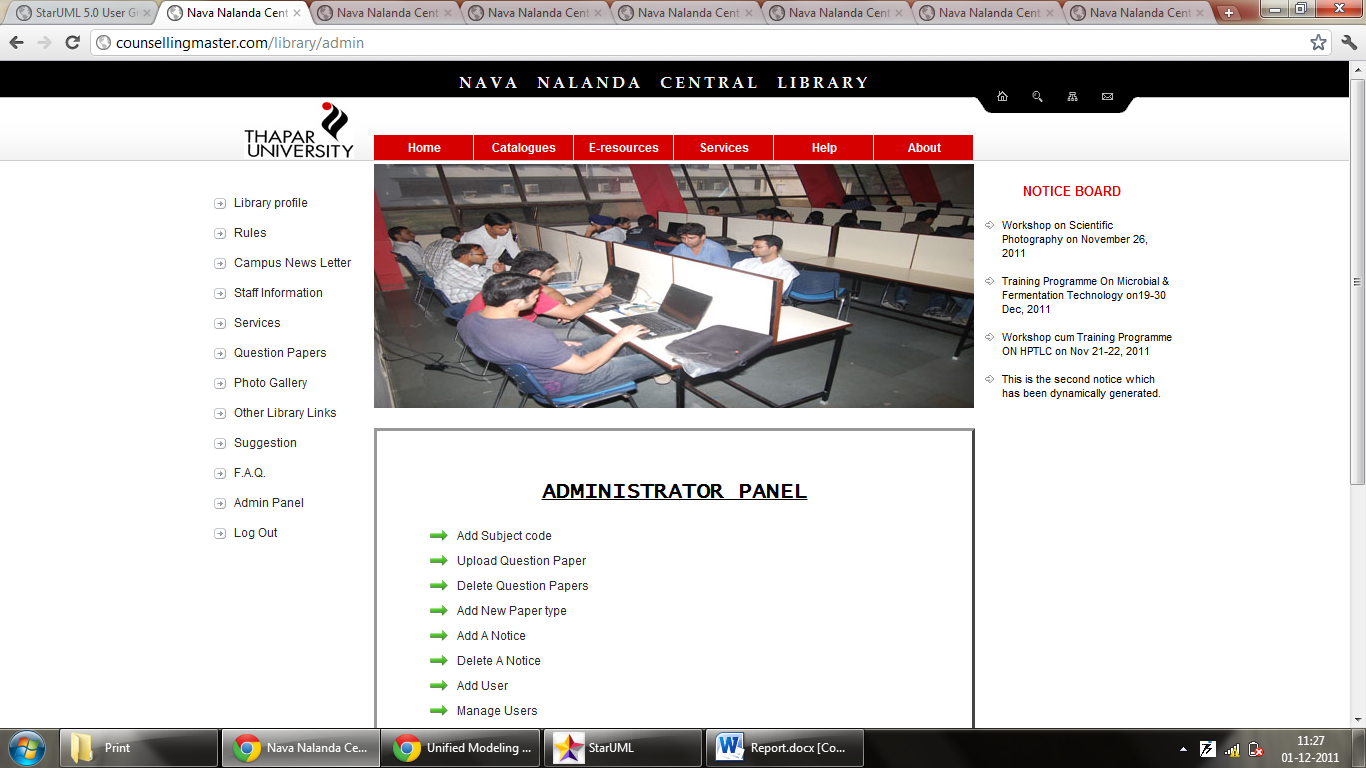
## Deployment Diagram:

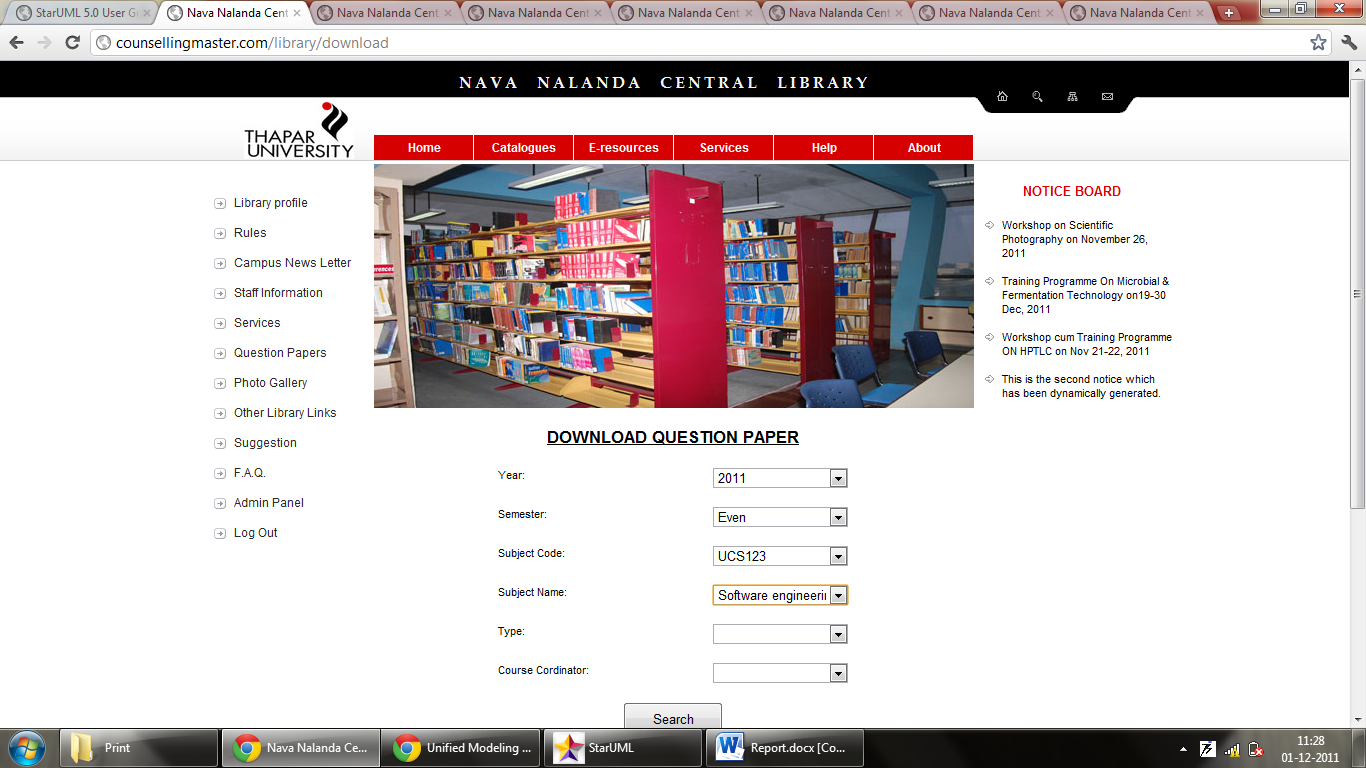


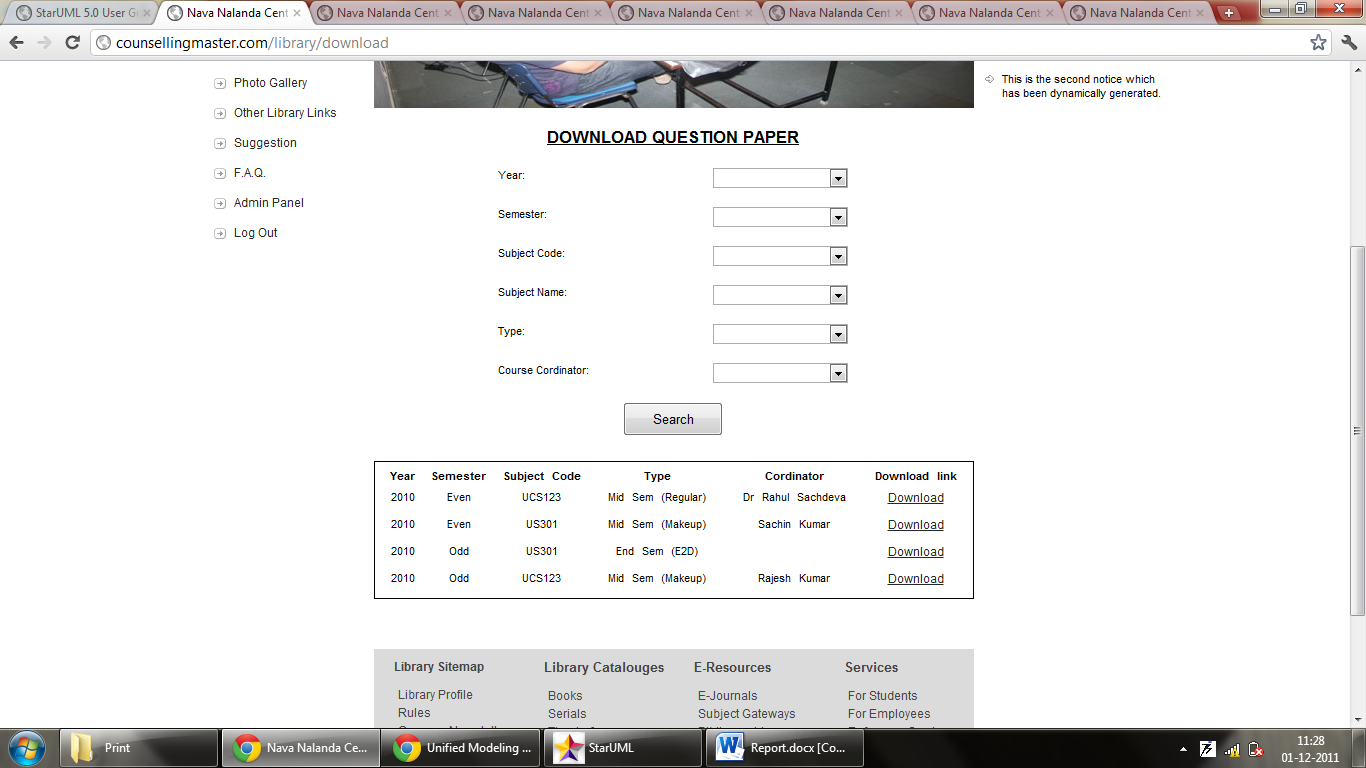
## Program Structure Chart

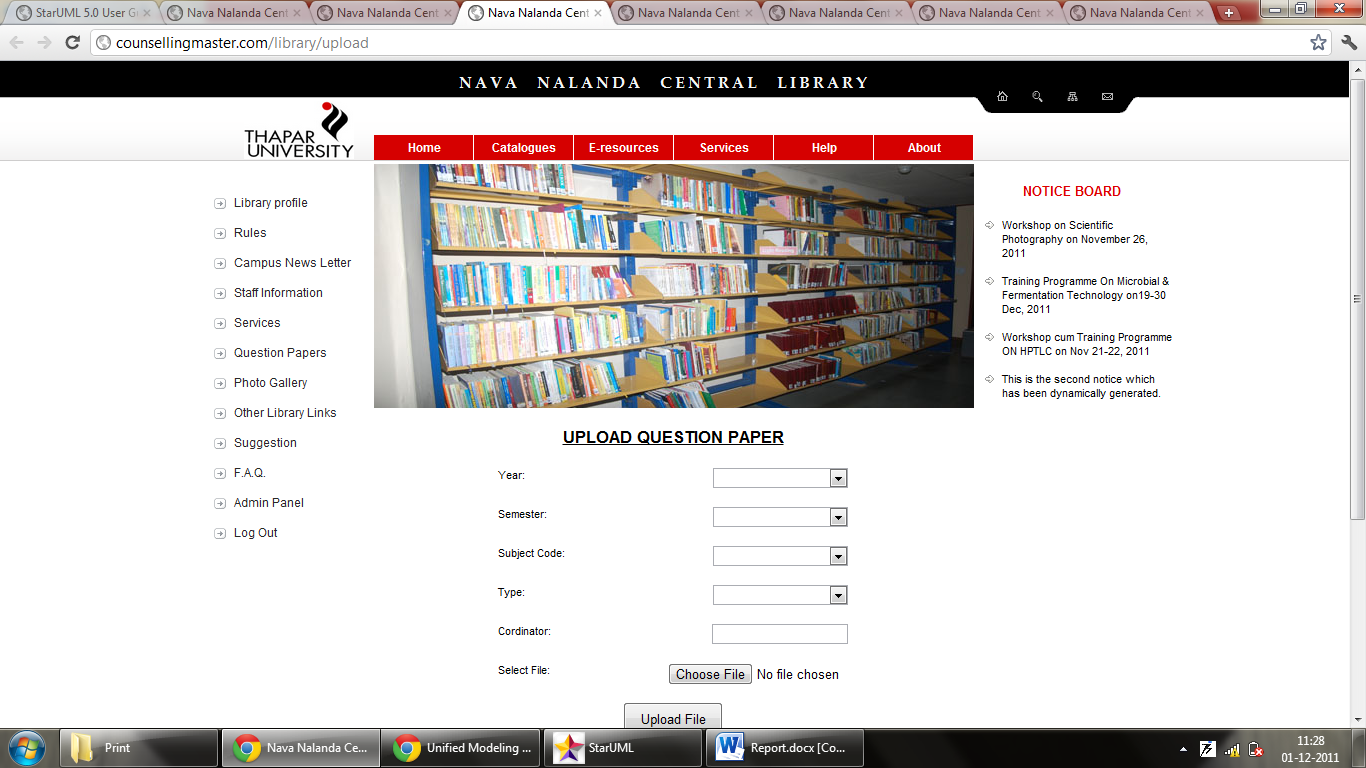


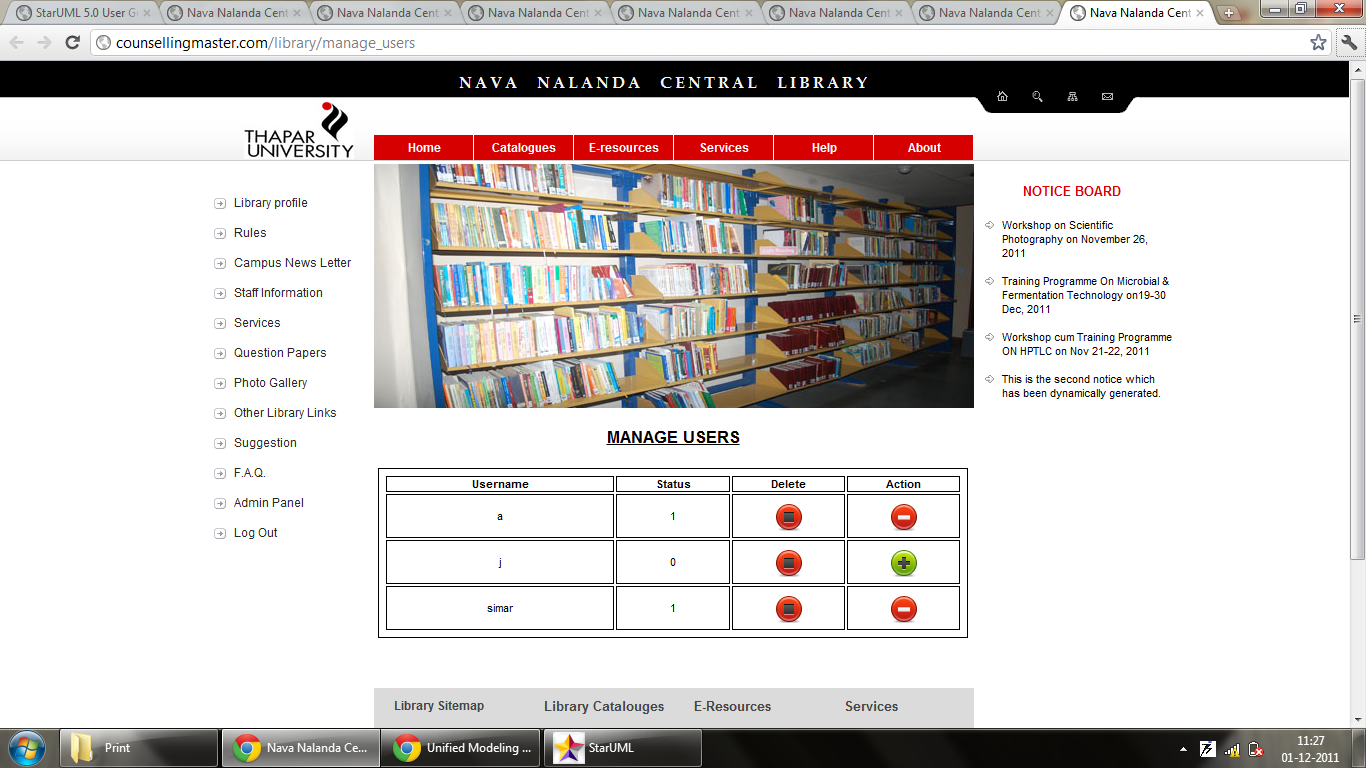
## Snapshots:











# Test Plan

## Unit Testing

For this we took the smallest piece of testable software in the application, isolated it from the remainder of the code, and determined whether it behaved exactly as we expected. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects were identified during its use.

* Different Units tested in this phase were:
* Administrator Login Panel
* Add Subject Code Form
* Upload Paper Form
* Delete Paper Form
* Add Paper Type Form
* Add Notice Form
* Delete Notice Form
* Add User Form
* Manage User Form (Ban/Unban/Delete Users)
* Download Question Paper Form

## Integration testing

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. Eventually all the modules making up a process were tested together. Integration testing helped in identifying problems that occurred when units were combined. This method reduced the number of possibilities to a far simpler level of analysis.

In this project we employed a top down approach to integration testing. First the website interface was used as the test driver, and one by one the low level units were integrated with it using Depth-first approach. Upon every successful integration, tests were conducted to validate the working of each unit in the integrated environment. Regression test was also conducted to make sure that the changes have not propagated unintended side effects.

## Flow Graph for administrator panel

Invalid

Valid

V = 11

E = 16

Cyclomatic complexity = E – V + 2 = 16 – 11 + 2 = 6

## Test Cases

### Unit Testing

* 1. Download Paper form

|  |  |
| --- | --- |
| Input | Select Subject Name = Software Engineering |
| Actual Output | All question papers listed with subject code that of input |
| Expected Output | All question papers listed with subject code that of input |
| Result | Pass |

|  |  |
| --- | --- |
| Input | No selection |
| Actual Output | All question papers list |
| Expected Output | All question papers list |
| Result | Pass |

* 1. Admin Login Panel

|  |  |
| --- | --- |
| Input | Username=library Password=Library |
| Actual Output | Admin Panel |
| Expected Output | Admin Panel |
| Result | Pass |

|  |  |
| --- | --- |
| Input | Username=1’ or ‘1 Password=1’ or ‘1 |
| Actual Output | Invalid Username/Password Combination. |
| Expected Output | Invalid Username/Password Combination. |
| Result | Pass |

* 1. Manage user

|  |  |
| --- | --- |
| Input | Ban user ‘x’ |
| Actual Output | The User has been Banned. |
| Expected Output | The User has been Banned. |
| Result | Pass |

|  |  |
| --- | --- |
| Input | Delete user ‘x’ |
| Actual Output | The User has been removed successfully. |
| Expected Output | The User has been removed successfully. |
| Result | Pass |

|  |  |
| --- | --- |
| Input | Unban user ‘x’ |
| Actual Output | The User has been Unbanned. |
| Expected Output | The User has been Unbanned. |
| Result | Pass |

* 1. Add Subject Code Form

|  |  |
| --- | --- |
| Input | UCS301 |
| Actual Output | The Subject Code has been successfully added. |
| Expected Output | The Subject Code has been successfully added. |
| Result | Pass |

|  |  |
| --- | --- |
| Input | 1 |
| Actual Output | Unable to add Subject Code |
| Expected Output | Unable to add Subject Code |
| Result | Pass |

### Integration testing

* 1. Admin panel

|  |  |
| --- | --- |
| Input | Surf through admin panel options |
| Actual Output | Redirected to homepage |
| Expected Output | Session maintained throughout the website |
| Result | Fail |
| Corrected | Yes |

|  |  |
| --- | --- |
| Input | Logout and access admin panel forms. |
| Actual Output | Redirect to homepage |
| Expected Output | Redirect to homepage |
| Result | Pass |

|  |  |
| --- | --- |
| Input |  |
| Actual Output |  |
| Expected Output |  |
| Result |  |