EXAM MANAGEMENT SYSTEM USING WEB SERVICE

PROJECT REPORT

Submitted by

S.DINESH

Reg.No: 0710108010

G.S.SARAVANAKUMAR

Reg.No: 0710108046

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

APRIL 2011

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)

COIMBATORE - 641049

Department of Computer Science and Engineering

PROJECT WORK, April 2011

This is to certify that the project entitled **"EXAM MANAGEMENT SYSTEM USING WEB SERVICE"**, the bonafide work of **S.DINESH, G.S.SARAVANAKUMAR** who carried out the project work under my supervision.

[Mrs.M.S.Hema M.E.,]
Project Guide

[Mrs.S.Devaki M.E., (Ph.D)]
Head of Department

The candidates with **University Register Nos. 0710108010 & 0710108046** was examined by us in project viva-voce examination held on 20\4\11.

Internal Examiner

External Examiner

DECLARATION

We,

S.DINESH	**Reg.No: 0710108010**
G.S.SARAVANAKUMAR	**Reg.No: 0710108046**

Hereby declare that the project entitled **"EXAM MANAGEMENT SYSTEM USING WEB SERVICE"**, submitted in partial fulfillment to Anna University as the project work of Bachelor of Engineering (Computer Science and Engineering) degree, is record of original work done by us under the supervision and guidance of Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore.

Place: Coimbatore
Date: 19\4\11

[S.Dinesh]

[G.S.Saravanakumar]

Project Guided by,

[Mrs.M.S.Hema M.E.,]
Assistant Professor,
Department of Computer Science and Engineering,
Kumaraguru College of technology,
Coimbatore - 641049

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for enabling us to complete this project.

We express our profound gratitude to our Chairman **Padmabhusan Arutselvar Dr.N.Mahalingam, B.Sc., F.I.E.,** for giving this opportunity to pursue this course.

We would like to thank **Dr.J.Shanmugam M.E., Ph.D.,** *Director* and **Dr.S.Ramachandran, Ph.D.,** *Principal* for providing the necessary facilities to complete our project.

We take this opportunity to thank **Dr.S.Thangasamy Ph.D.,** Professor and *Dean, Research and Development,* for his precious suggestions. We also thank **Mrs.S.Devaki M.E.,(Ph.D)** *Head of Department*, Department of Computer Science and Engineering, for her support and timely motivation.

We register our hearty appreciation to the Guide **Mrs.M.S.Hema M.E.,** *Assistant Professor,* Department of Computer Science and Engineering, our Project advisor. We thank for her support, encouragement and ideas. We thank her for the countless hours she has spent with us, discussing everything from research to academic choices.

We would like to convey our honest thanks to all **Teaching** staff members and **Non Teaching** staff of the department for their support. We would like to thank all our classmates who gave us a proper light moments and study breaks apart from extending some technical support whenever we needed them most.

TABLE OF CONTENTS

ABSTRACT

The project "**Exam Management System using Web Service**" is developed for commercial purpose to maintain exam department records and also for developing the web service useful for implementing EMS. The objective of this project is to provide the Exam Management System service useful for implementing EMS.

Now available or existing system doesn't have the web service functionality. So, reusing the functionality is not possible. If there are any changes in functionality it should be changed in every place the function exists.

We developed the EMS with the Web service. The function is developed as the web service and that service are available for reusing and its accessed anywhere. And any changes in the service are made at service only.

In this project the web service client module are accessing this service and having the Module such as Mark Updation, Result Calculation and Admission.

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

The project "Exam Management System using Web Service" rovides software as a service. This provides the information about he student and the staffs belong to that organization. The staffs have ewer privileges than the administrator.

The work of administrator in this project is to verify the new user registered with this organization. Administrator also allocates staff for the subject and the credits for the subject and the subject for the semester. Administrator has rights to publish or block the result.

The work of the staff in this project is to add or update the both internal and external mark of the student for the subject that staff handles. The result also viewed by the staff.

Examinations are an important issue while proving one's ability in the education line. To show case the talent present in the student, examinations play an important role. Waiting for the result gives a tension in the minds of the student. The student can view the result at any time. The student can also view the personal information that given during the time of admission.

CHAPTER-2

SYSTEM DEVELOPMENT

.1 SYSTEM ANALYSIS

.1.1 Existing System

The current system of Exam management has a long process. 'he student wants to choose the class details, department details, ubject details, semester details and user details etc., the mark alculations are not done internally. The subject and staff are llocated in the department at a particular time and place. Also, the tudent will not able to view their result immediately. It takes more ime to process.

Drawbacks of Existing system

- Too hard to make changes.
- Not component based model.
- Not user centric model.
- Process time is high.
- Does not follow modularized design

.1.2 Proposed System

The modern computerized system is developed with the aim to overcome the drawbacks of existing system. The proposed system has got many advantages. In the proposed project of an Exam management System using web service, the web service are created for the main functionality such as allocating roll number, calculating result ,cgpa , sgpa and etc.,

Features of Proposed System

- The system is more flexible in which changes can be done easily.
- It is a web based system which makes web interaction easy.
- Maintain the user related data in centralized database.

Advantage of Web Service

In existing system the new user get registered without any verification from the administrator. The proposed system replaces this process with web services administrator can verify the new user registered on that system.

These smart web services will understand the context of each request and produce dynamic result based on each specific situation. The services will adapt their process based on the user ID, subject ID, and class ID. Multiple services will combined and produce a customized solution. The user will experience only the collective

2.1.3 SOA

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable *services* that can be used within multiple, separate systems from several business domains.

SOA also generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. For example, several disparate departments within a company may develop and deploy SOA services in different implementation languages; their respective clients will benefit from a well understood, well defined interface to access them. XML is commonly used for interfacing with SOA services, though this is not required.

Each SOA building block can play one or both of two roles:

Service Provider - The service provider creates a web service and possibly publishes its interface and access information to the service registry. Each provider must decide which services to expose, how to make trade-offs between security and easy availability, how to price the services, or (if no charges apply) how/whether to exploit them for other value. The provider also has to decide what category the service should be listed in for a given broker service and what sort egisters what services are available within it, and lists all the otential service recipients.

Service consumer - The service consumer or web service client locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its web services. Whichever service the service-consumers need, they have to take it into the brokers, then bind it with respective service and then use it. They can access multiple services if the service provides multiple services.

WSDL

The **Web Services Description Language** (WSDL, pronounced 'wiz-del') is an XML-based language that provides a model for describing Web services.

The WSDL defines services as collections of network endpoints, or ports. The WSDL specification provides an XML format for documents for this purpose. The abstract definitions of ports and messages are separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service.

WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program what operations are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL file.

UDDI

Universal Description, Discovery and Integration (**UDDI**, pronounced Yu-di) is a platform-independent, Extensible Mark up Language (XML)-based registry for businesses worldwide to list themselves on the Internet and a mechanism to register and locate web service applications. UDDI is an open industry initiative, sponsored by the Organization for the Advancement of Structured Information Standards (OASIS), enabling businesses to publish service listings and discover each other and define how the services or software applications **interact** over the Internet.

2 DEVELOPMENT ENVIRONMENT

2.1 Hardware requirements

Operating system	: Windows XP, VISTA, 7
Processor	: Intel Pentium 4, 2.79GHz
RAM	: 2 GB
Hard disk capacity	: 80 GB

.2.2 Software requirements

Programming language	: PHP
SOAP – toolkit	: NuSoap
Database	: MYSQL
Domain Service	: IndiaInternet.in
Host Service	: Freehostia.com
Web Service site	: emsservice.net
Web Client site	: xctems.com
UDDI Registry	: xmethods.net

.2.3 Software Description

HP

PHP stands for **P**HP: **H**ypertext **P**re-processor. It is a server-ide scripting language, like ASP. It scripts are executed on the erver. It supports many databases (MySQL, Informix, Oracle, ybase, Solid, PostgreSQL, Generic ODBC, etc.). It is open source oftware. PHP files can contain text, HTML tags and scripts. It files re returned to the browser as plain HTML.

OAP toolkit

NuSOAP is a group of PHP classes that allow developers to reate and consume SOAP web services. It does not require any pecial PHP extension.

MySQL

MySQL is a database. The data in the MySQL is stored in latabase objects called tables. A table is a collection of related data entries and it consists of columns and rows. Databases are useful in storing information categorically. An Examination department in the organization have a database with following tables: user information, staff allocation, subject allocation, student marks etc..,

CHAPTER - 3

PROJECT PLAN AND FEASIBILITY STUDY

.1 TEAM STRATERGY AND WORK

The team consist of 2 members all are qualified for the system) be developed. The system was first designed, coded and nplemented in the local host using XAMPP server and testing rocess was done.

.2 DEVELOPMENT SCHEDULE

.2.1 Milestones

Milestones are being established for each and every module to mprove the product visibility. It enhances the development process to ecome more tangible. It exposes errors, which help in improving the roduct quality and increase project communication. In our pplication it has been done sub-module wise.

.2.2 Reviews

Review issues lists, are prepared to identify problem area within he product. As a programmer we do the following reviews.

- Critical Design Review
- Source code Review
- Acceptance Test Review

.3 FEASIBILITY ANALYSIS

The feasibility study is very rough analysis of the viability of a roject. It is however a highly desirable checkpoint that should be ompleted before committing to more resources. Feasibility study is onducted to obtain an overview of the problem and to roughly vhether feasible solutions exist prior to committing substantial esources to a project.

The primary objective of a feasibility study is to assess three ypes of feasibility.

- Operational feasibility
- Technical feasibility
- Economical feasibility

3.3.1 Operational feasibility

Operational feasibility study is must, because it ensures that hat the project implements in the organization work the feasibility should be high.

The Operational feasibility is high in this project as it allows to register for mark updating, result viewing and provides good interface, which is easy and friendly for the user to use it.

.3.2 Technical feasibility

Technical feasibility analysis makes a comparison of the level f technology available and the same is required for the development f the product. The level of technology accounts for factors such as he programming language, the machine environment, the rogramming practices and the software tools.

Resource availability such as Pentium 3 processor with 128MB RAM, software and tools required for the project are available at the organization. Hence it is technical feasible.

3.3.3 Economical feasibility

This is the most important aspects that has to be critically evaluated. This includes the feasibility study of cost - benefit analysis. This is an assessment of the economic justification for a computer based system project. Most of the software are available in the web. Hence the threat of financial non-feasibility does not exist. It is determined that benefits out beat the cost of implementation and the system is considered to be economically feasible.

CHAPTER – 4

MODULE DESCRIPTION

.1 Introduction

The exam Management system has three modules with Web ervice which is accessed through HTTP using NuSoap Toolkit. The Web Service available is

- Alc_reg
- Percent.
- Total.
- Chk_id.
- Subj_tot.
- Subj_grade_l.
- Subj_grade_p.
- Sgpa.
- Cgpa.

4.1.1 Web Service

4.1.1.1 Alc_reg

This service takes input as array of String and Returns it in

.1.1.2 Percent

This service takes input as array of integer and returns the ercentage value for the given value.

.1.1.3 Total

This service takes input as array of integer and returns the utput as total of all integers given.

4.1.1.4 Chk_id.

This service takes input as user information such as id and password and return 1 if id and password are correct else return 0.

4.1.1.5 Subj_total

This service takes two integers i.e. internal and external and return total of both marks.

4.1.1.6 Subj_grade_l

This service takes two integers i.e. internal and external and calculates it total and return grade letter according to the mark. Eg. Return S for marks above 90

4.1.1.7 subj_grade_p

This service takes two integers i.e. internal and external and calculates it total and returns grade point according to the mark. Eg. Return 10 for marks above 90

1.1.8 SGPA

This service takes two arrays of integer i.e. credit point id grade point in order and its calculate summation of credit point id summation of multiplication of grade point and credit and its get ivided and quotient is returned as result.

.1.1.9 CGPA

This service takes two arrays of integer i.e. credit point ind grade point in order and its calculate summation of credit point ind summation of multiplication of grade point and credit and its get livided and quotient is returned as result.

.1.1.10 WEBSERVICE WSDL ANALZER RESULT:

VSDL URL: http://emsserver.net/emsws.php?wsdl

/SDL ANALYZER URL: http://www.xmethods.net/ve2/Tools.po

/SDL Analyzer: Operations

or the WSDL file http://emsserver.net/emsws.php?wsdl

'he following table lists the operations for the service. You can drill own into the various messages associated with the operations by licking on the message links.

peration / Method Name	SOAP Action*	Style	Input Message
lelloWorld	http://emsserver.net/emsws.php/HelloWorld	rpc	Input Msg
lc_reg	http://emsserver.net/emsws.php/alc_reg	rpc	Input Msg
ercent	http://emsserver.net/emsws.php/percent	rpc	Input Msg
otal	http://emsserver.net/emsws.php/total	rpc	Input Msg
:hk_id	http://emsserver.net/emsws.php/chk_id	rpc	Input Msg
subj_tot	http://emsserver.net/emsws.php/subj_tot	rpc	Input Msg
subj_grade_l	http://emsserver.net/emsws.php/subj_grade_l	rpc	Input Msg
subj_grade_p	http://emsserver.net/emsws.php/subj_grade_p	rpc	Input Msg
Sgpa	http://emsserver.net/emsws.php/sgpa	rpc	Input Msg
Cgpa	http://emsserver.net/emsws.php/cgpa	rpc	Input Msg

SDL Of WebService:

```
xml version="1.0" encoding="ISO-8859-1"?>
efinitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
lns:xsd="http://www.w3.org/2001/XMLSchema"
lns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
IC="http://schemas.xmlsoap.org/soap/encoding/"
lns:tns="http://emsserver.net/soap/EMSSERVICE"
lns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
lns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
getNamespace="http://emsserver.net/soap/EMSSERVICE">
ypes>
sd:schema targetNamespace="http://emsserver.net/soap/EMSSERVICE"

xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
/xsd:schema>
/types>
message name="HelloWorldRequest">
<part name="name" type="xsd:string" /></message>
message name="HelloWorldResponse">
<part name="return" type="xsd:string" /></message>
message name="alc_regRequest">
<part name="name" type="xsd:string" /></message>
message name="alc_regResponse">
 <part name="return" type="xsd:string" /></message>
message name="percentRequest">
 <part name="name" type="xsd:string" /></message>
<message name="percentResponse">
 <part name="return" type="xsd:string" /></message>
<message name="totalRequest">
 <part name="name" type="xsd:string" /></message>
<message name="totalResponse">
 <part name="return" type="xsd:string" /></message>
<message name="chk_idRequest">
 <part name="name1" type="xsd:string" />
 <part name="name2" type="xsd:string" />
 <part name="name3" type="xsd:string" />
 <part name="name4" type="xsd:string" /></message>
<message name="chk_idResponse">
 <part name="return" type="xsd:string" /></message>
<message name="subj_totRequest">
 <part name="name1" type="xsd:string" />
 <part name="name2" type="xsd:string" /></message>
<message name="subj_totResponse">
 <part name="return" type="xsd:string" /></message>
<message name="subj_grade_lRequest">
 <part name="name1" type="xsd:string" />
 <part name="name2" type="xsd:string" /></message>
<message name="subj_grade_lResponse">
 <part name="return" type="xsd:string" /></message>
```

```
oart name="name2" type="xsd:string" /></message>
essage name="subj_grade_pResponse">
part name="return" type="xsd:string" /></message>
essage name="sgpaRequest">
part name="name1" type="xsd:string" />
part name="name2" type="xsd:string" /></message>
essage name="sgpaResponse">
part name="return" type="xsd:string" /></message>
nessage name="cgpaRequest">
<part name="name1" type="xsd:string" />
<part name="name2" type="xsd:string" /></message>
nessage name="cgpaResponse">
<part name="return" type="xsd:string" /></message>
portType name="EMSSERVICEPortType">
<operation name="HelloWorld">
 <documentation>Simple Hello World Method</documentation>
 <input message="tns:HelloWorldRequest"/>
 <output message="tns:HelloWorldResponse"/>
</operation>
<operation name="alc_reg">
 <documentation>Allocating roll number</documentation>
 <input message="tns:alc_regRequest"/>
 <output message="tns:alc_regResponse"/>
</operation>
<operation name="percent">
 <documentation>Return percentage of given marks</documentation>
 <input message="tns:percentRequest"/>
 <output message="tns:percentResponse"/>
</operation>
<operation name="total">
 <documentation>Return total of given marks</documentation>
 <input message="tns:totalRequest"/>
 <output message="tns:totalResponse"/>
</operation>
<operation name="chk_id">
 <documentation>Return whether login inforamtion correct or not</documentation>
 <input message="tns:chk_idRequest"/>
 <output message="tns:chk_idResponse"/>
</operation>
<operation name="subj_tot">
 <documentation>return total for single subject by providing internal and external
marks</documentation>
 <input message="tns:subj_totRequest"/>
 <output message="tns:subj_totResponse"/>
</operation>
<operation name="subj_grade_l">
 <documentation>return grade letter for single subject by providing internal and external
marks</documentation>
 <input message="tns:subj_grade_lRequest"/>
 <output message="tns:subj_grade_lResponse"/>
</operation>
```

```
   <input message="tns:subj_grade_pRequest"/>
   <output message="tns:subj_grade_pResponse"/>
  </operation>
  <operation name="sgpa">
   <documentation>Return sgpa of given marks</documentation>
   <input message="tns:sgpaRequest"/>
   <output message="tns:sgpaResponse"/>
  </operation>
  <operation name="cgpa">
   <documentation>Return cgpa of given marks</documentation>
   <input message="tns:cgpaRequest"/>
   <output message="tns:cgpaResponse"/>
  </operation>
</portType>
<binding name="EMSSERVICEBinding" type="tns:EMSSERVICEPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="HelloWorld">
   <soap:operation soapAction="http://emsserver.net/emsws.php/HelloWorld" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="alc_reg">
   <soap:operation soapAction="http://emsserver.net/emsws.php/alc_reg" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="percent">
   <soap:operation soapAction="http://emsserver.net/emsws.php/percent" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="total">
   <soap:operation soapAction="http://emsserver.net/emsws.php/total" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="chk_id">
   <soap:operation soapAction="http://emsserver.net/emsws.php/chk_id" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
```

```
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="subj_grade_l">
   <soap:operation soapAction="http://emsserver.net/emsws.php/subj_grade_l" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="subj_grade_p">
   <soap:operation soapAction="http://emsserver.net/emsws.php/subj_grade_p" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="sgpa">
   <soap:operation soapAction="http://emsserver.net/emsws.php/sgpa" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
  <operation name="cgpa">
   <soap:operation soapAction="http://emsserver.net/emsws.php/cgpa" style="rpc"/>
   <input><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input>
   <output><soap:body use="encoded" namespace="http://emsserver.net/emsws.php"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output>
  </operation>
 </binding>
 <service name="EMSSERVICE">
  <port name="EMSSERVICEPort" binding="tns:EMSSERVICEBinding">
   <soap:address location="http://emsserver.net/emsws.php"/>
  </port>
 </service>
 </definitions>
```

UDDI Registry

Xmethod.net

4.2 Client Modules

The Module in the EMS Client is:

- Administrator module
- Staff module
- Student module

4.2.1 Administrator module

The administrator has more privileges on that

Verify user

- Department
- Staff
- Subject
- Class
- Admission
- Generate roll number
- Staff and Subject allocation
- Mark update
- Semester wise result
- Semester status
- result

4.2.1.1 Verify user

The administrator needs to verify the user newly registered with this organization. This form contains information like user ID, user type, status and unique ID. It has two options one is to verify and the other is to reject, this option that depends on the administrator by viewing the information.

4.2.1.2 Department

It registers the department ID for the newly registered batch with their department name.

4.2.1.3 Staff

It registers the staff for the department with their user ID and designation. This form contains Staff ID, staff name, designation and department.

4.2.1.4 Subject

It registers the subject for the department for the particular semester with subject ID, subject name and credit points.

4.2.1.5 Class

It registers the class for the department for newly admitted students and the number of semester for the particular class.

4.2.1.6 Admission

It contains admission number and the department the student belongs and the student information like student first name and last name. It registers the student information at the time of admission.

4.2.1.7 Generate roll number

It generates the roll number for the students by the information that given at the time of admission. By using the department ID and class ID the roll number is generated.

4.2.1.8 Staff and Subject allocation

It allocates the staff for the particular subject and the mode of the subject they going to handle. The allocation process done using the department, class and staff ID.

4.2.1.9 Mark Update

It updates the mark of the student by selecting the class ID and roll number and the semester the subjects in the semester will appear and it selected. The both internal and external marks are updated.

4.2.1.10 Semester wise result

It calculates the total and calculates cgpa and sgpa for the student for the particular semester.

2.1.11 Semester Status

The status is made to display the result. If status is 1, it displays sult. If status is 0, it does not display the result.

2.1.12 Result

It used to view the result. By giving the department ID and the oll number and the semester is selected the result is displayed.

.2.2 Staff module

The staffs have only few privileges like admission and mark pdating.

- Admission
- Mark update
- Result

l.2.2.1 Admission

It contains admission number and the department the student belongs and the student information like student first name and last name. It registers the student information at the time of admission.

4.2.2.2 Mark update

It updates the mark of the student by selecting the class ID and roll number and the semester the subjects in the semester will appear

2.2.3 Result

It used to view the result. By giving the department ID and the ll number and the semester is selected the result is displayed.

.2.3 Student

Student only views the result and the personal information.

- Result
- Personal information

.2.3.1 Result

It used to view the result. By giving the department ID and the oll number and the semester is selected the result is displayed.

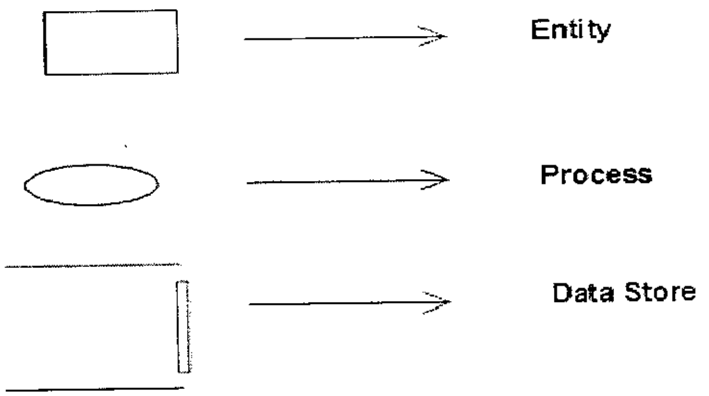
.2.3.2 Personal information

It contains the personal information about the student like user ame, user ID, Email address.

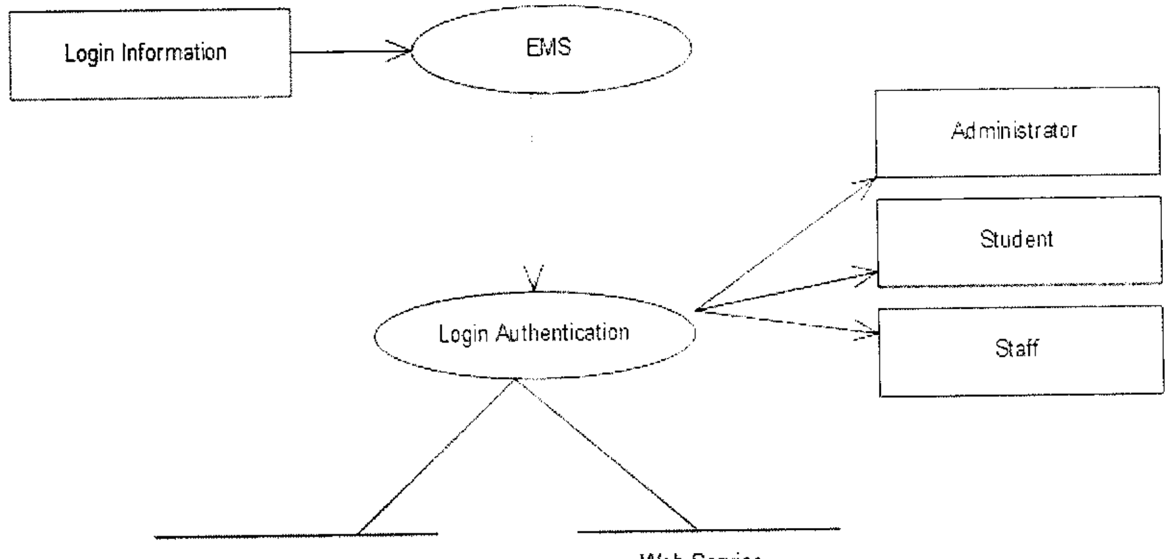
CHAPTER – 5

SYSTEM DESIGN AND IMPLEMENTATION

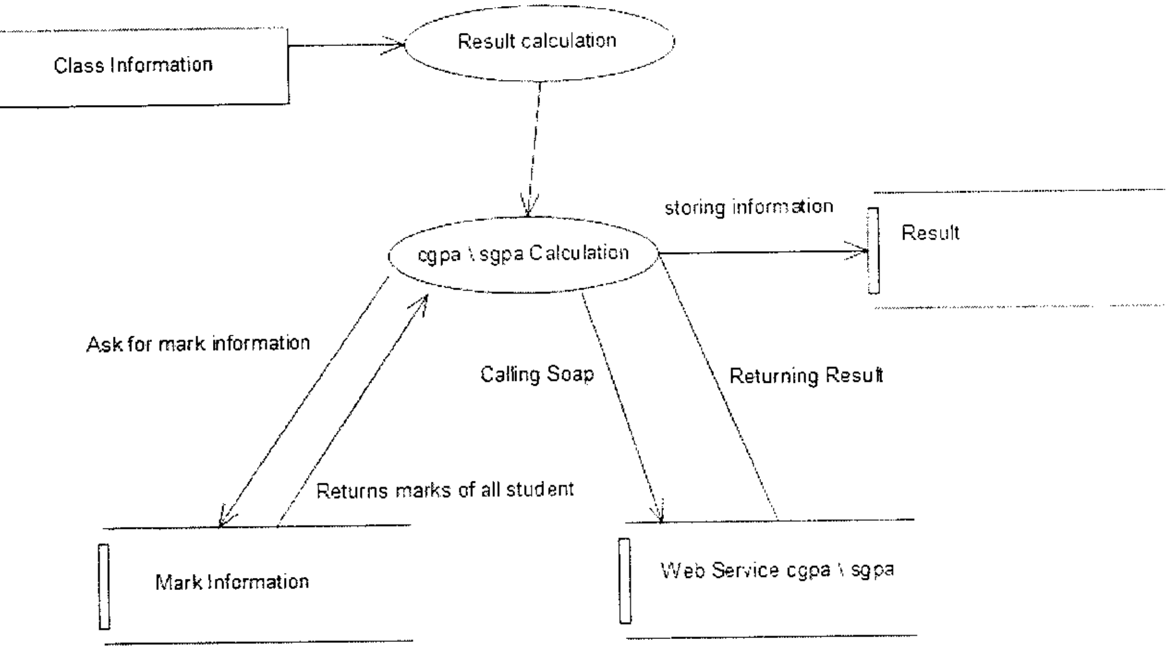
.1 SYSTEM DESIGN



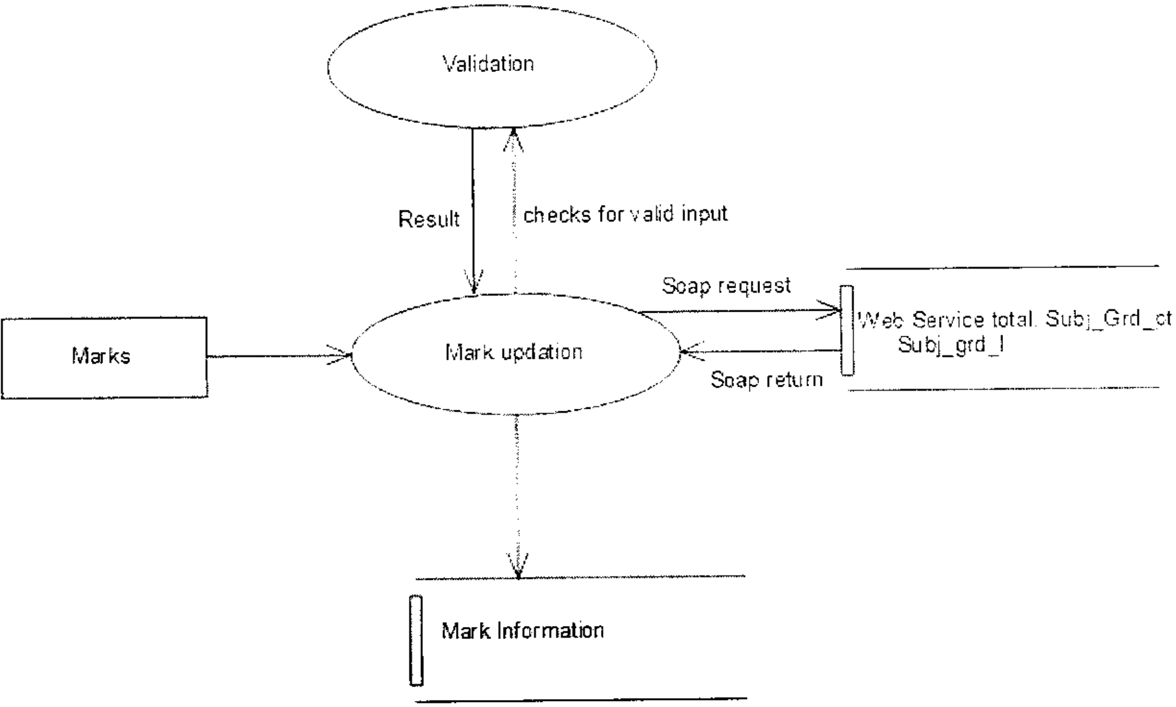
D 1.1 Student Registration



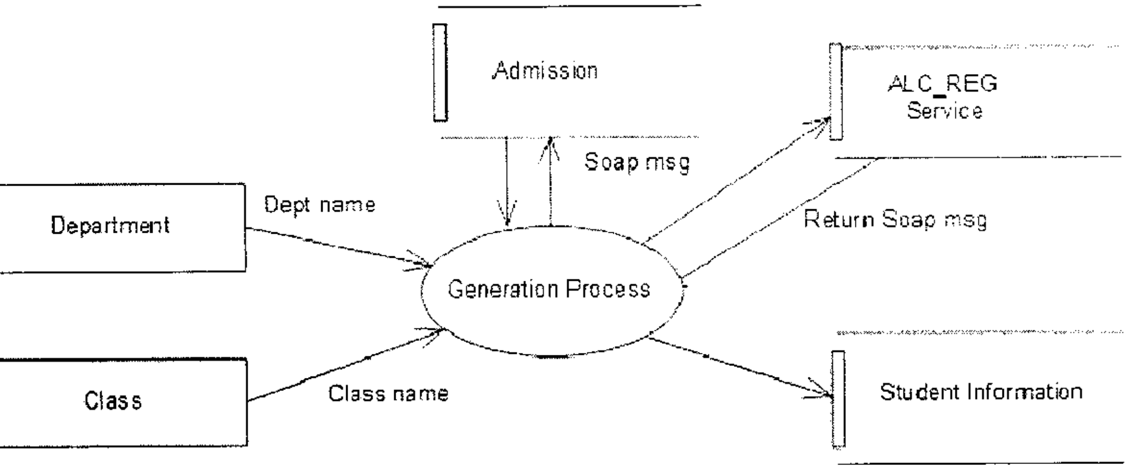
1.2 Result Calculation



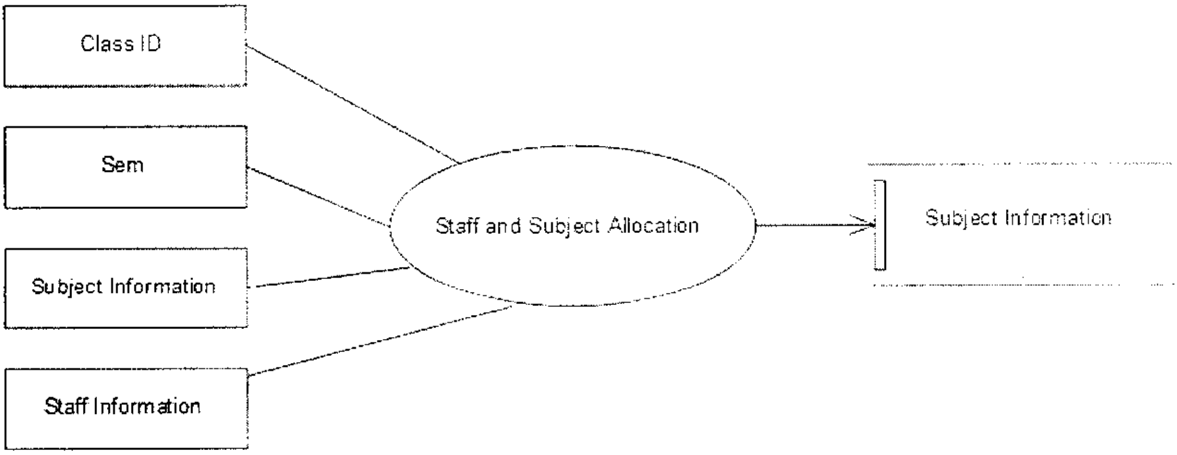
) 1.3 Mark updating



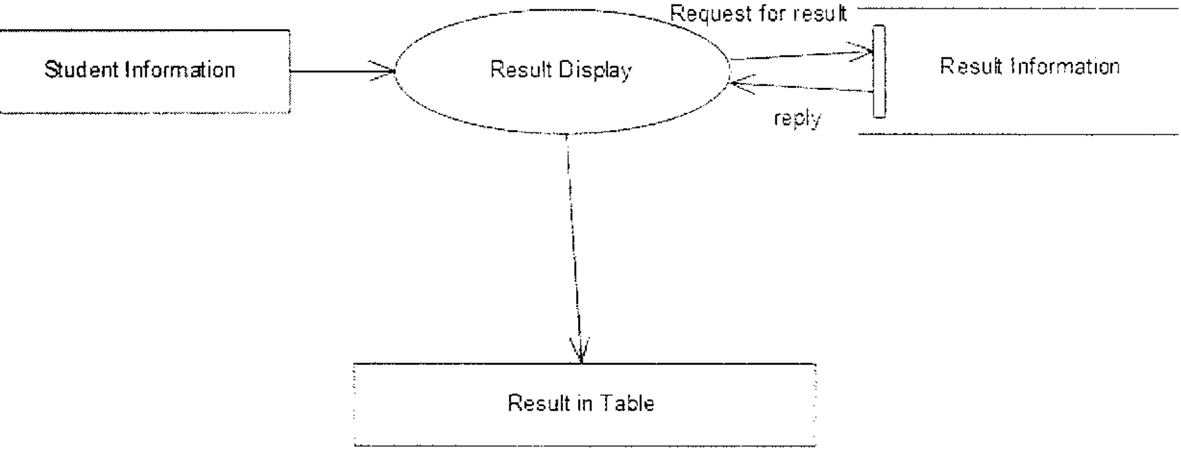
) 1.4 Generate Roll number



D 1.5 Staff and Subject allocation



) 1.6 Result display



.2 SYSTEM TESTING

Testing of software extends throughout the coding phase and t represents the ultimate review of specification, design and coding. \ series of test case are created with an intention of testing of the oftware. Based on the way the software reacts to these tests, we can lecide whether the product that has been built is robust or not.

.2.1 Verification and Validation

Verification refers to the set of activities that ensures that ystem correctly implements a specific function. Validation refers to lifferent set of activities that ensure that the system has been built is raceable to customer requirements. Verification and validation encompass a wide array of software quality assurance(SQA) activities hat include formal reviews, quality, performance monitoring, locumentation review, database review.

Numeric field:

The numeric field must not contain any alphabets or special characters. In mark updating process the numeric types are marked validated.

In our web site during Mark Updating phase, It ask for both nternal and external mark. In that it checks whether given data is number or not, and then it check whether both marks are less than or equal to 50.

Character field

The character field must not contain any special characters. The character field in user name,student name and subject name are validated.

In our Project character in user id field, check whether the given user id starts with character or not.

And in the email id the valid email id is checked and subject name and student name it checks whether all character or not.

5.2.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of software unit of software design, the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

Test 1:

Procedure:

The mandatory fields have to be filled before procedding to next process.

In our project in the user registration process all values should be enter and if the all information are not present then it alert to enter all information.

olution:

The alert message has to be displayed to fill the mandatory etails when information required for the registration is not present.

'est 2:

'rocedure:

The EMS involves adding users, marks, subjects, staff and nark calculation involving various operations.

Solution:

The problem is solved by using session tracking.

5.2.3 Integration Testing

Once the modules are tested individually under the testing strategy, it is necessary to put all these modules together-interfacing. It is here that the data can be lost across the interface, one module can have an inadvertent, adverse effect on another.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit testing modules and build a program structure that has been dictated by design.

In integrating system web service integration with the project

In our project integration of web service are checked, that in calculating the total using web service, other two web service,subject total and subject grade point service are called and that result are given to this webservice.

5.3 IMPLEMENTATION

Implementation means the process of converting a new or revised system design into an operational one. It is the most crucial stage in achieving a new successful system and in giving a confidence on the new system for the users that it will work efficiently and effectively. In this phase, we build the components either from scratch or by composition. Given the architecture document from the design phase and requirement document from the analysis phase, we can build exactly what has been requested.

This phase deals with issues of quality, performance, baselines, libraries and debugging. The end deliverable is the product itself. There are three types of implementation:

1. Implementation of a computer system to replace a manual system.
2. Implementation of new computer system to replace an existing one.
3. Implementation of a modified application to replace an existing one, using the same computer.

Implementation of Web service required the NuSoap toolkit, which is open source tool kit and using this toolkit the Web service are created and hosted in the web site.

And client software is also hosted in other web site which uses the web service for this project.

5.4 FUTURE ENHANCEMENT

The software is developed in such a way that any kind of further changes or modifications to the system get themselves updated at regular intervals. Some enhancement can be made like exam scheduling, hall arrangements, hall ticket printing and certificate verifications.

CHAPTER – 6

CONCLUSION

Working over the project "Exam management System" has been a great experience with lot of exposure to various evolving software trends both in developing web service and web service client. The project has been found to work effectively and efficiently. It clearly gives the client a competitive advantage tool that would help to improve the system at the bottom line.

The application is formulated by analyzing the requirements of the user. Each and every module has undergone various test conditions. With a full stretch testing, it has been ensured that the system can enhance ideally without any bugs or crashes, which will make the end user more compatible with the project.

The application is designed as user friendly and all the options available are clear and self explanatory so that the user can understand the system easily.

APPENDIX 1

TABLE DESIGN

Table name: Admission Detail table

Description: This Table contains the information about the student at that used at the time of admission.

Fields	**Data type**	**Description**
Adm_no	Varchar(20)	The unique admission number.
Dept_id	Varchar(20)	Department name.
Cls_id	Varchar(20)	The class ID of the department.
Stu_fname	Varchar(100)	This field contains the first name of the user.
Stu_lname	Varchar(100)	This field contains the last name of the user.

Table T 1.1 – Admission Detail table

Table name: Class registration table

Description: This table contains the information about the class for the particular semester.

Fields	**Data Type**	**Description**
Cls_id	Varchar(20)	The class ID of the department.
Dept_id	Varchar(20)	Department name.
No_sem	Varchar(2)	Number of semesters for the particular batch.

Table T 1.2 - Class registration table

Table name: Department registration table

Description: This table contains the information about the registered departments in the system.

Fields	**Data Type**	**Description**
Dept_id	Varchar(5)	Department name.
Dept_name	Varchar(100)	It contains the information about the department.

Table T 1.3 – Department registration table

Table name: Elective allocation table

Description: This contains information about the elective subjects and the allocation details.

Fields	**Data Type**	**Description**
Cls_id	Varchar(10)	The class ID of the department.
Subj_id	Varchar(50)	The subject ID for that semester.
Stu_id	Varchar(25)	Student ID of the user.

Table T 1.4 – Elective allocation table

Table name: Semester wise mark table

Description: This table contains Semester wise mark of the student.

Fields	**Data Type**	**Description**
Stu_id	Varchar(50)	Student ID of the user
Sem_id	Varchar(5)	The semester which the mark is to be added.
Total	Varchar(5)	Total of the student.
Percent	Varchar(5)	Percentage of the stuednt.
Sgpa	Varchar(4)	Sgpa of the student.
Cgpa	Varchar(4)	Cgpa of the student.

Table name: Subject mark table

Description: It contains the internal mark, external mark, total and grade points of the student.

Fields	**Data Type**	**Description**
Sem_id	Varchar(3)	Mark of the semester.
Cls_id	Varchar(50)	The class ID of the department.
Stu_id	Varchar(50)	Student ID of the user.
Sub_id	Varchar(50)	Subject ID for that semester.
Int_mark	Varchar(3)	Internal mark of the student.
Ext_mark	Varchar(3)	External mark of the student.
Total	Varchar(3)	Total of marks in that semester.
Grd_p	Varchar(3)	Grade point.
Grd_l	Varchar(3)	Grade letter.

Table T 1.6 – Subject mark table

Table: subject allocation table

Description: It allocates the subject for the staff in that semester.

Fields	**Data Type**	**Description**
Cls_id	Varchar(10)	The class ID of the department.
Dept_id	Varchar(10)	Department name.
Sem_id	Varchar(3)	The semester which it allocates.
Subj_id	Varchar(50)	Subject ID for that subject.
Stf_id	Varchar(50)	Staff ID for the staff.
Mode	Varchar(10)	Type of Selection.

Table T 1.7 – Subject allocation table

Table: Semester status table

Description: This contains information that particular semester result is to be released or not.

Fields	**Data Type**	**Description**
Cls_id	Varchar(50)	The class ID of the department.
Sem_id	Varchar(50)	The semester which the status to set.
Status	Varchar(3)	Type of selection.

Table: Staff registration table

Description: It registers the staff for the department using the staff ID.

Fields	**Data Type**	**Description**
Staff_id	Varchar(20)	Staff ID for the staff.
Staff_name	Varchar(100)	Staff name.
Desg	Varchar(50)	Designation of the staff.
Dept_id	Varchar(20)	Department name.

Table T 1.9 – Staff registration table

Table: Student Information table

Description: It contains the student information that given during the time of admission.

Fields	**Data Type**	**Description**
Stu_id	Varchar(100)	Student ID of the user.
Adm_no	Varchar(100)	Admission number of the student.
Stu_name	Varchar(100)	Student name.
Stu_cl_id	Varchar(100)	Class ID of the student.

Table T 1.10 – Student information table

Table: Subject Registration table

Description: To register the subject for the semester with its credits.

Fields	**Data Type**	**Description**
Sub_id	Varchar(20)	Subject ID for the subject.
Sub_name	Varchar(100)	Subject name.
Crd_pt	Varchar(2)	Credits for the subject.
Dept_id	Varchar(20)	Department name.

Table T 1.11 – Subject Registration table

Table: Student mark table

Description: It contains the mark of the students.

Fields	**Data Type**	**Description**
Year_id	Varchar(20)	Year the student belongs.
Stu_id	Varchar(20)	Student ID of the user.
Sub_id	Varchar(20)	Subject ID of the subject.
Sem_id	Varchar(20)	Semester
Mark	Varchar(3)	Mark of the student.

Table T 1.12 – Student mark table

'able: User authentication table

)escription: It contains the information about the user like user ID, ›assword and user type.

'ields	Data Type	Description
Jser_id	Varchar(20)	User ID of the user.
'asswd	Varchar(20)	Password for login.
User_type	Varchar(20)	Type of user.
Status	Varchar(3)	Status for the type.
Uni_id	Varchar(20)	Unique ID for the user.

Table T 1.13 – User authentication table

Table: User Information Table

Description: It contains the information about the user like user ID and E- mail address.

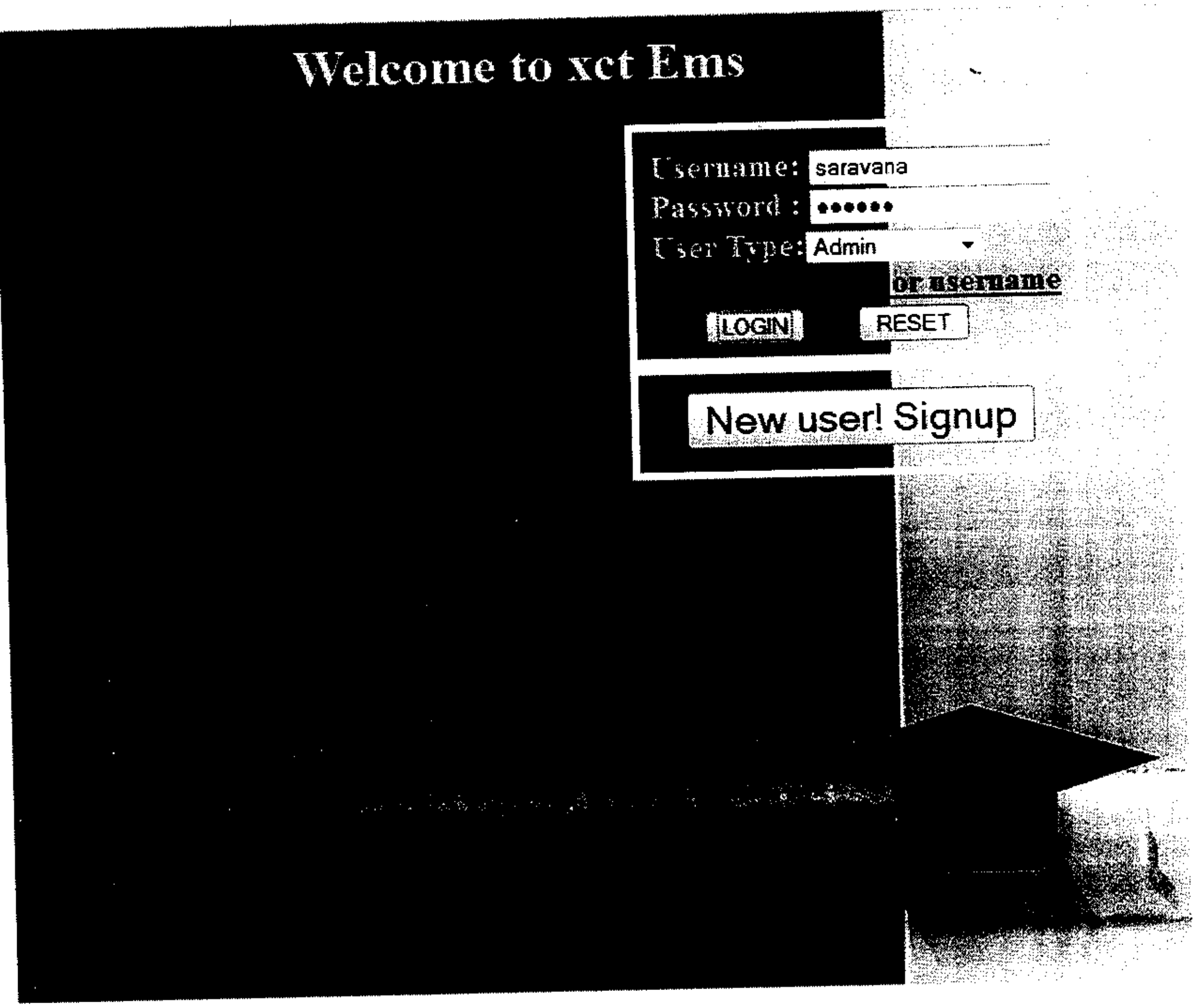
Fields	Data Type	Description
User_id	Varchar(20)	User ID of the user.
Email	Varchar(100)	Email address of the user.

Table T 1.14 – User Information table

APPENDIX 2

SCREEN SHOTS

Login form



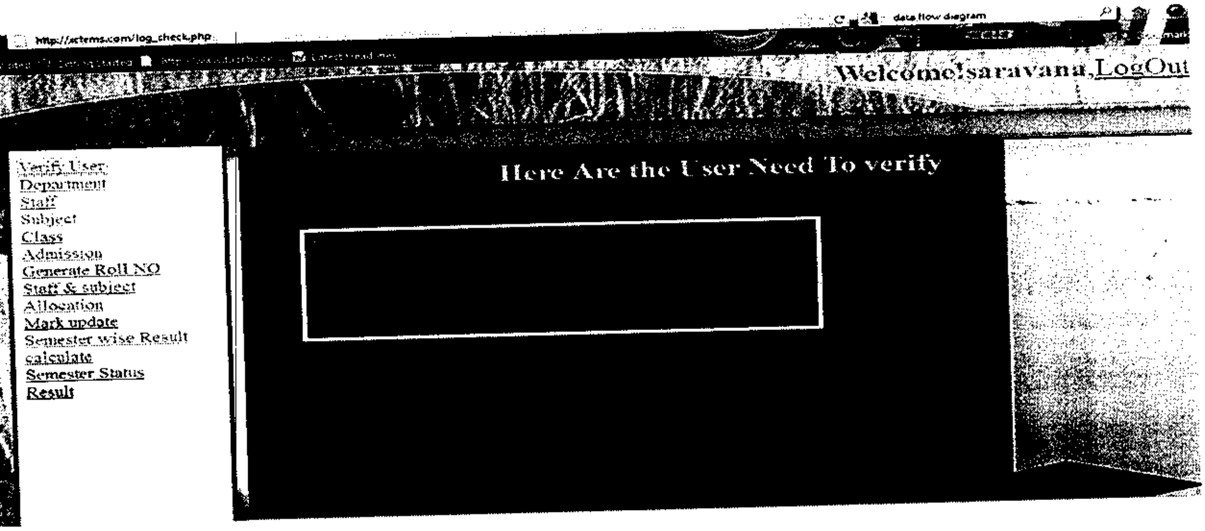
Welcome to xct Ems

Username: saravana

Password : ••••••

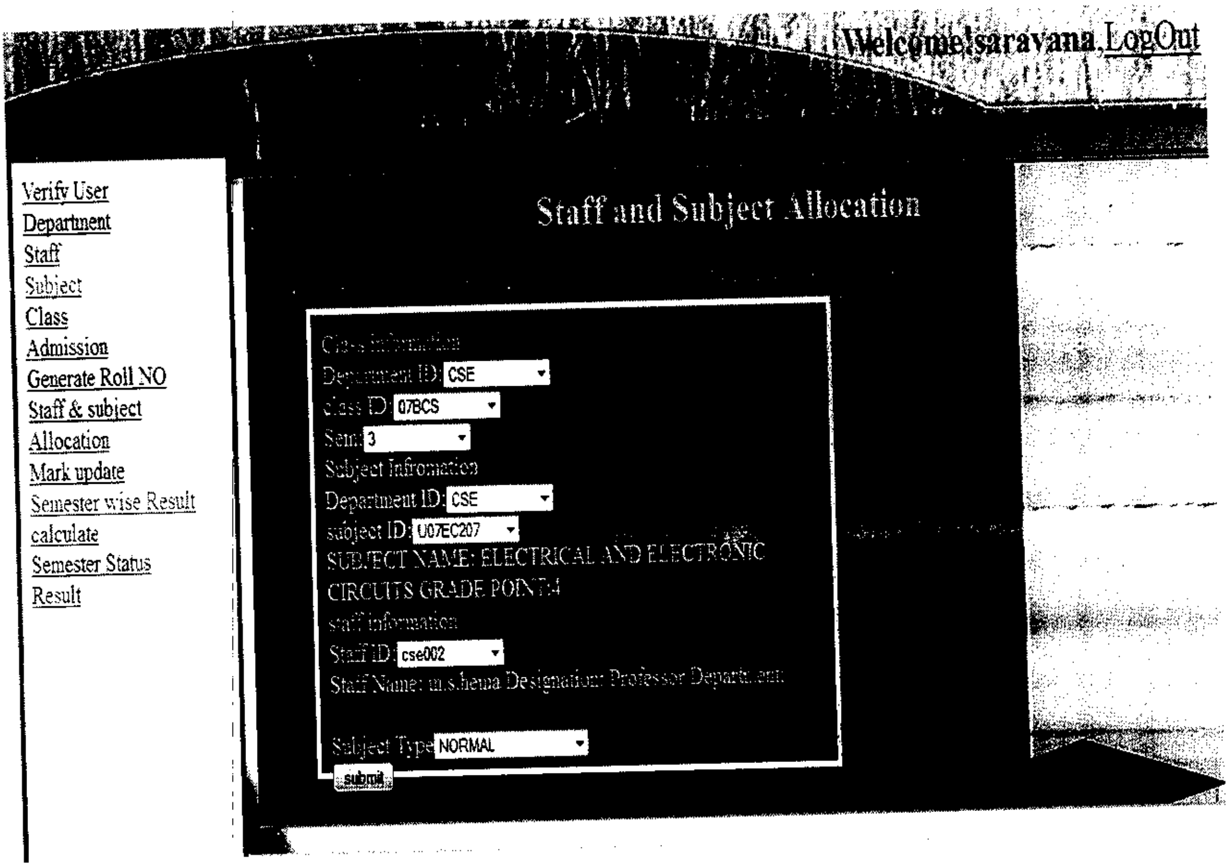
User Type: Admin

Administrator verification form



Here Are the User Need To verify

Staff and Subject allocation



Staff and Subject Allocation

Class information

Department ID: CSE

class ID: 07BCS

Sem: 3

Subject Infromation

Department ID: CSE

subject ID: U07EC207

SUBJECT NAME: ELECTRICAL AND ELECTRONIC CIRCUITS GRADE POINT:4

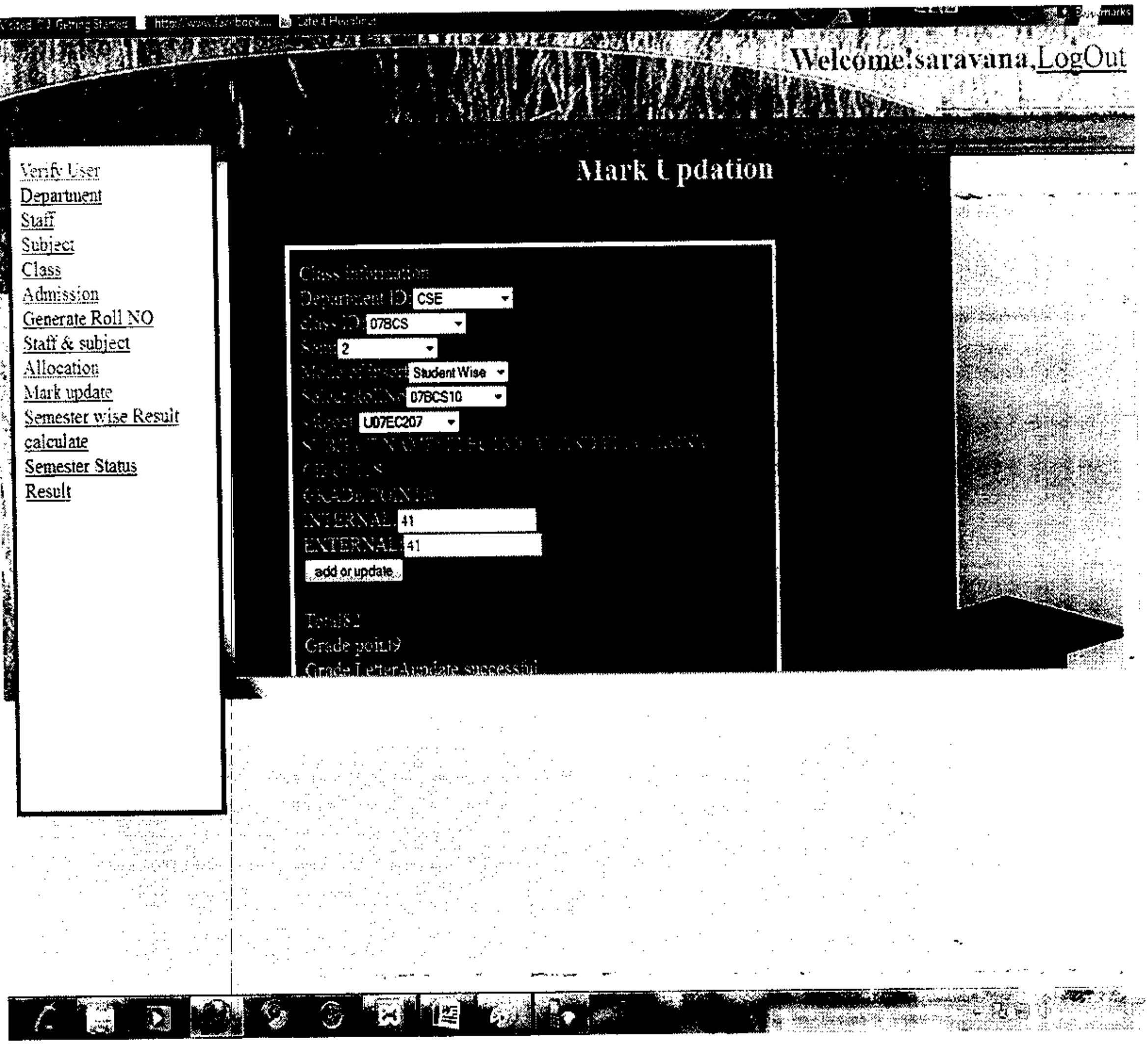
staff information

Staff ID: cse002

Staff Name: m.s.hema Designation: Professor Department:

Subject Type NORMAL

Mark Updating

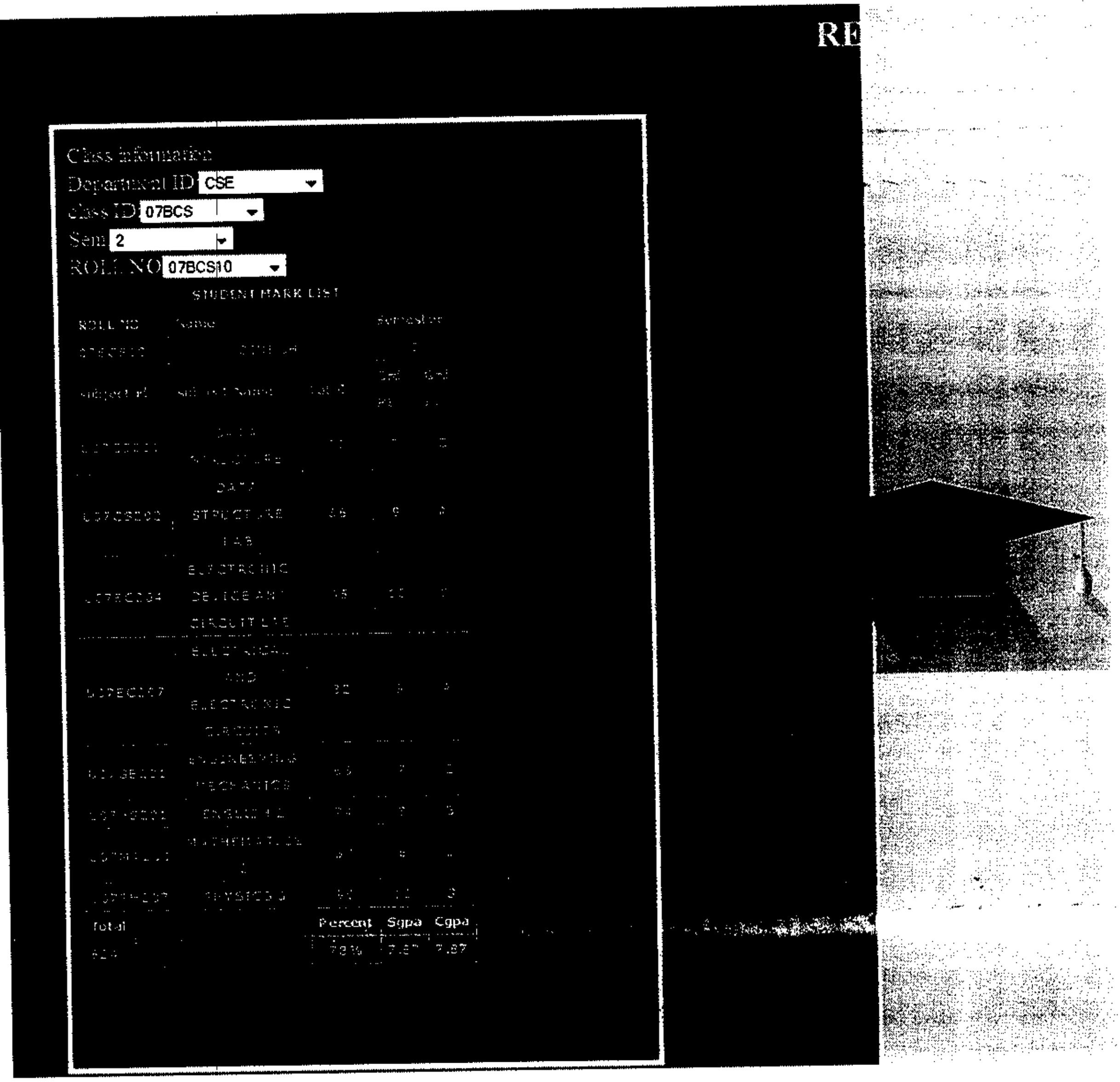


Mark Updation

Class Information
Department ID: CSE
class ID: 07BCS
Sem: 2
Mode of Insert: Student Wise
Select RollNo: 07BCS10
Subject: U07EC207
SUBJECT NAME: [illegible]
CREDITS
GRADE POINTS
INTERNAL: 41
EXTERNAL: 41

Total82
Grade point9
Grade LetterAupdate successful

Result form



Class information

Department ID: CSE

class ID: 07BCS

Sem: 2

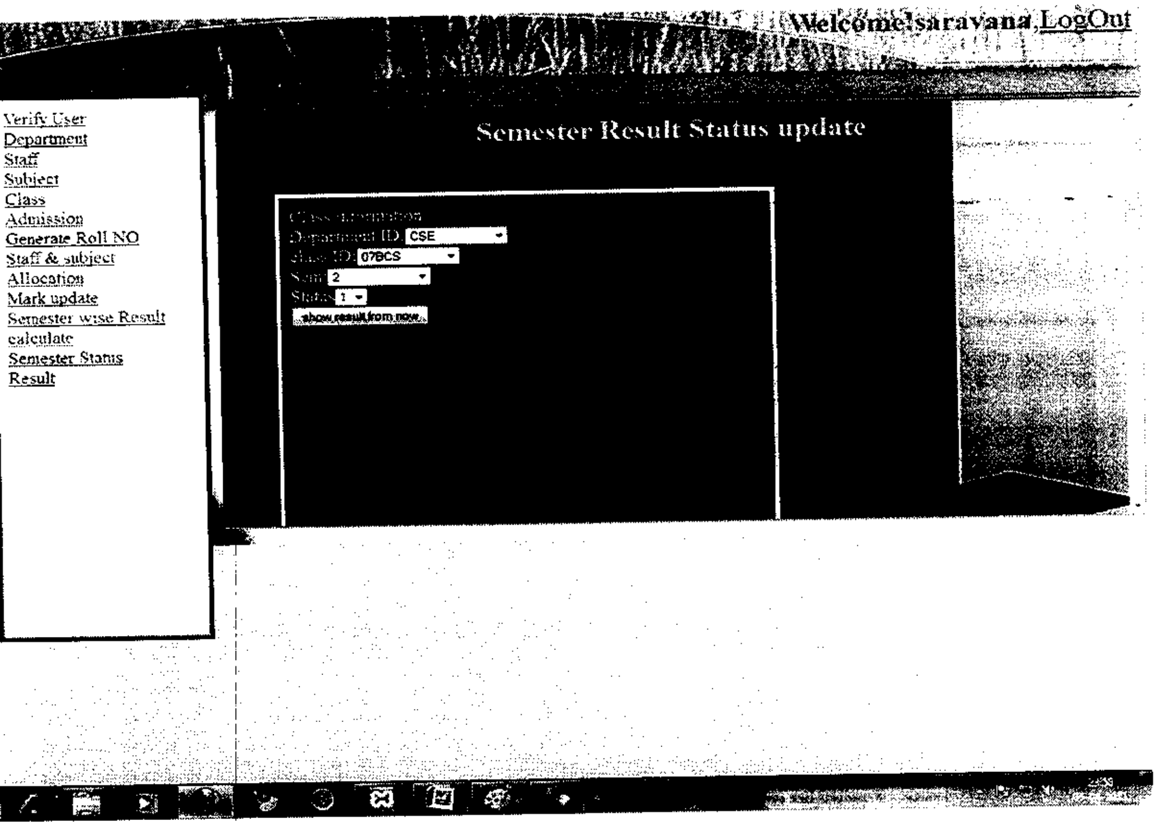
ROLL NO: 07BCS10

STUDENT MARK LIST

ROLL NO	Name		Semester	
07BCS10	[illegible]		2	
subject id	subject Name	total	Grd Pt	Grd
[illegible]	DATA STRUCTURE	[illegible]	[illegible]	[illegible]
U07CS202	DATA STRUCTURE LAB	86	9	A
U07EC204	ELECTRONIC DEVICE AN[illegible] CIRCUIT LAB	95	10	[illegible]
U07EC207	ELECTRICAL AND ELECTRONIC CIRCUITS	82	9	A
[illegible]	ENGINEERING MECHANICS	65	7	[illegible]
[illegible]	ENGLISH II	74	7	B
[illegible]	MATHEMATICS II	[illegible]	8	[illegible]
[illegible]	PHYSICS II	[illegible]	[illegible]	B

Total	Percent	Sgpa	Cgpa
624	78%	7.87	7.87

Status update form



Semester Result Status update

Class information

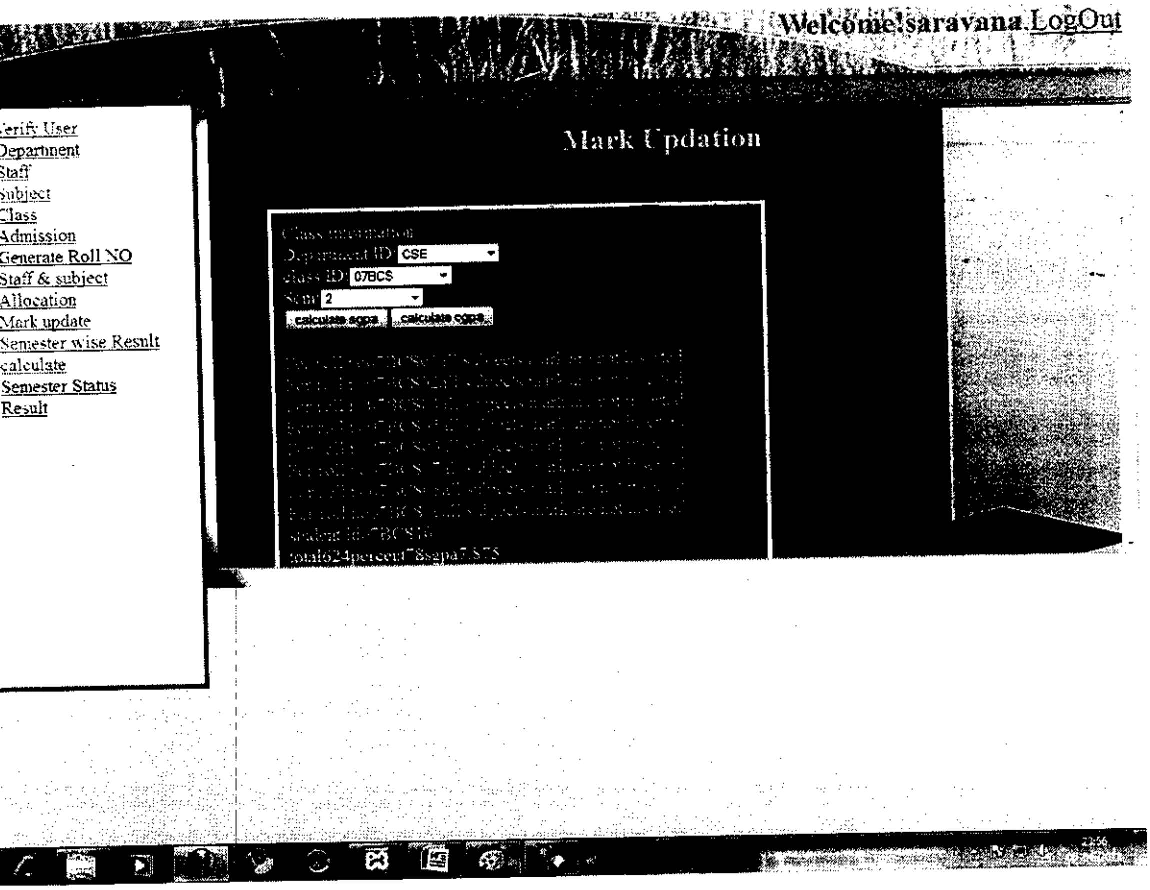
Department ID: CSE

Class ID: 07BCS

Sem: 2

Status: 1

gpa and Cgpa calculation



Mark Updation

Class information

Department ID: CSE

class ID: 07BCS

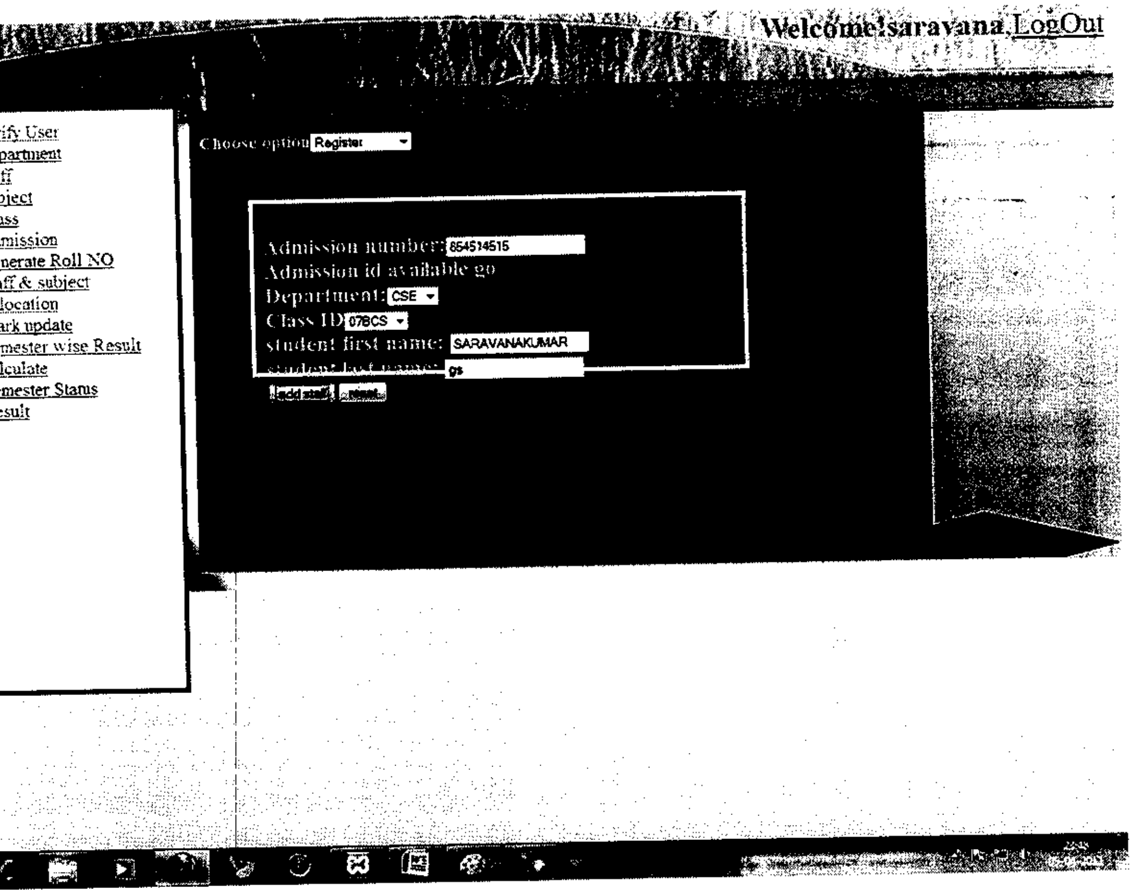
Sem: 2

[illegible]

student id:07BCS16

total624percent78sgpa7.875

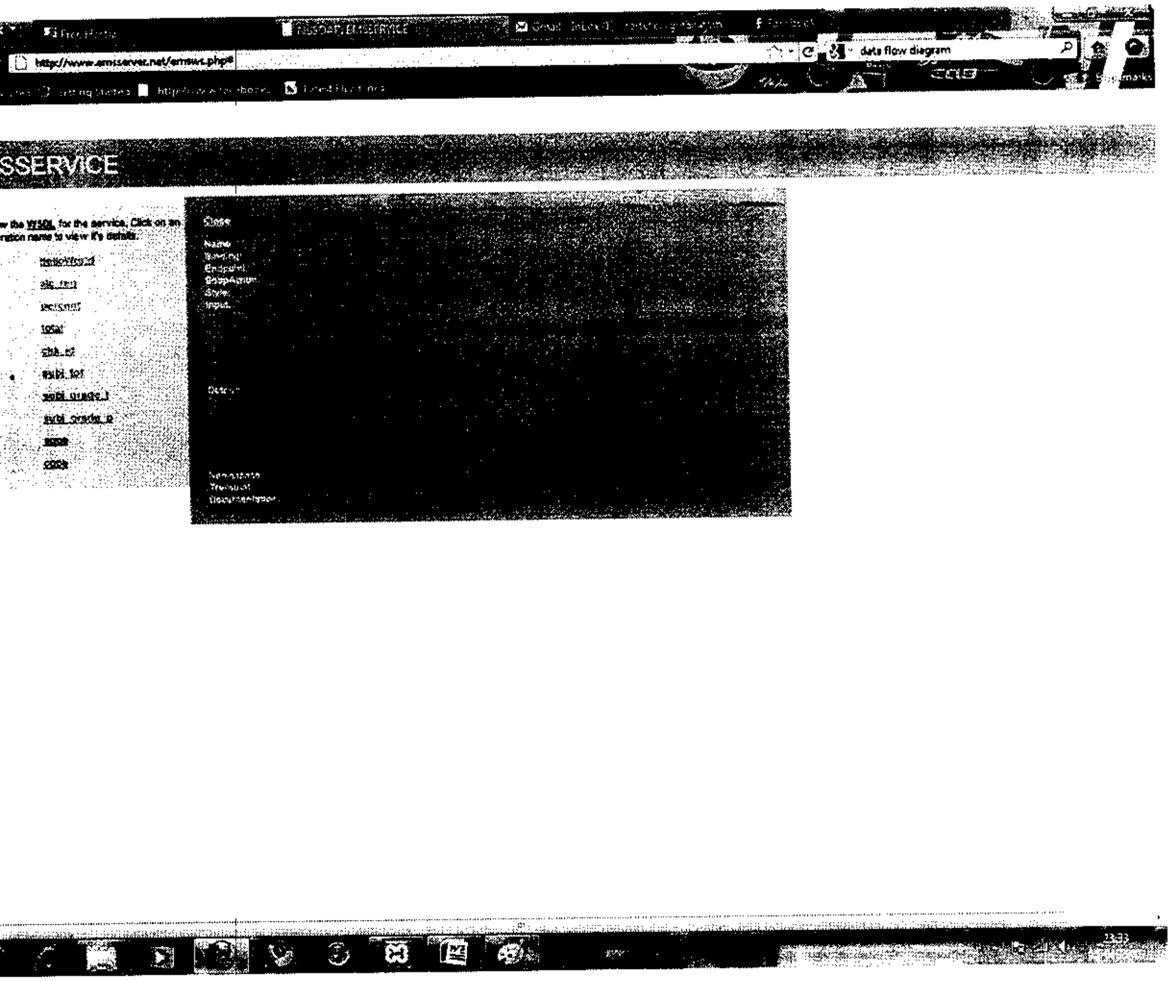
lmission Form



Choose option Register

Admission number: 854514515
Admission id available go
Department: CSE
Class ID 07BCS
student first name: SARAVANAKUMAR
student last name: gs

Soap WSDL template



http://www.amiserver.net/amsws.php#

data flow diagram

SSERVICE

[illegible] the WSDL for the service. Click on an [illegible] name to view it's details.

- HelloWorld
- [illegible]
- [illegible]
- total
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

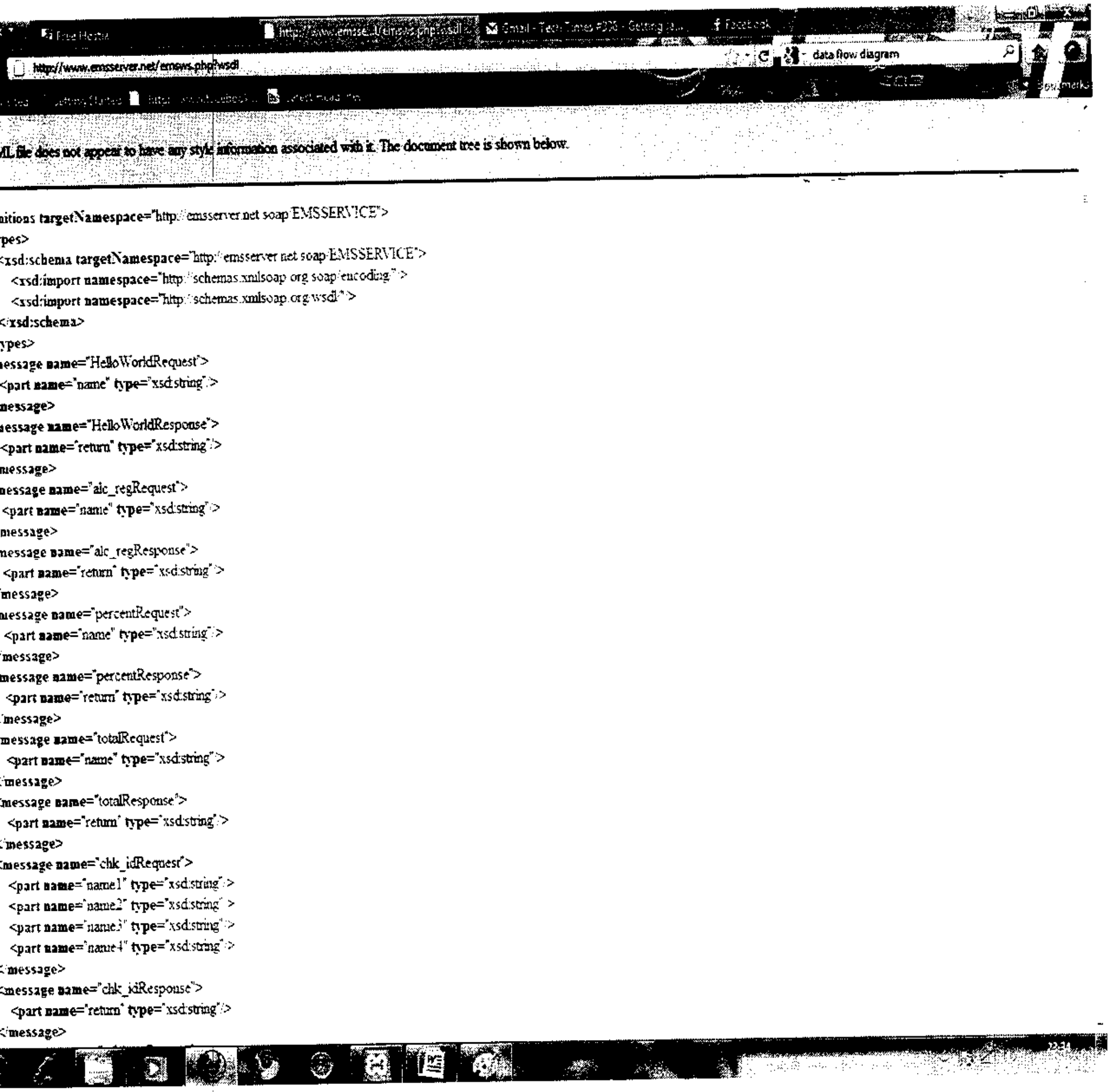
Close

Name:
Binding:
Endpoint:
SoapAction:
Style:
Input:

Output:

Namespace:
Transport:
Documentation:

eb Service WSDL



http://www.emsse...t/emsws.php?wsdl
http://www.emsserver.net/emsws.php?wsdl
data flow diagram

ML file does not appear to have any style information associated with it. The document tree is shown below.

```
nitions targetNamespace="http://emsserver.net/soap/EMSSERVICE">
pes>
<xsd:schema targetNamespace="http://emsserver.net/soap/EMSSERVICE">
  <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
  <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
</xsd:schema>
ypes>
essage name="HelloWorldRequest">
<part name="name" type="xsd:string"/>
nessage>
essage name="HelloWorldResponse">
<part name="return" type="xsd:string"/>
message>
essage name="alc_regRequest">
<part name="name" type="xsd:string"/>
message>
essage name="alc_regResponse">
<part name="return" type="xsd:string"/>
message>
essage name="percentRequest">
 <part name="name" type="xsd:string"/>
message>
message name="percentResponse">
 <part name="return" type="xsd:string"/>
message>
message name="totalRequest">
 <part name="name" type="xsd:string"/>
message>
message name="totalResponse">
 <part name="return" type="xsd:string"/>
/message>
message name="chk_idRequest">
  <part name="name1" type="xsd:string"/>
  <part name="name2" type="xsd:string"/>
  <part name="name3" type="xsd:string"/>
  <part name="name4" type="xsd:string"/>
/message>
message name="chk_idResponse">
  <part name="return" type="xsd:string"/>
/message>
```

REFERENCES

OKS

- Steven Holzner, “The Complete Reference”, Tata McGraw-Hill publications.
- Roger S. Pressman, “Software Engineering”, Tata McGraw-Hill Publications, Sixth Edition, 2004.
- Rasmus Lerdorf, Kevin Tatroe, “Programming PHP”, O'Reilly Media, March 2002.
- Matthias Hertel, “Aspects of AJAX”, 2005.
- Cristian Darie, et al, “AJAX and PHP: Building Responsive Web Applications”, 2006.
- Paul Dubois, “Getting Started with MySQL”, Pearson Education, 2007.
- IEEE Paper , Integrated Examination Management System Based on Computer Supported Cooperative Work

EBSITES

- www.w3schools.com/php
- www.wampserver.com/en
- www.tizag.com/mysqlTutorial
- api.jquery.com/category/Ajax
- www.emsinet.com/
- http://www.mindlogicx.com/intelliEXAMS.html
- http://php.net/manual/en/function.mysql-query.php
- http://www.sourcecodesworld.com/scripts/Professional-PHP-