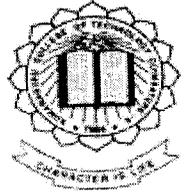
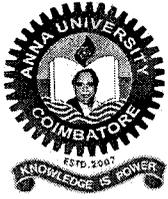


k-3263



# ASYNCHRONOUS INSTITUTE SERVICE

(STUDENT INFORMATION SYSTEM)

PROJECT REPORT

*Submitted By*

**T.SOWMYAMOORTHY**

**Register No.: 0720300046**

*in partial fulfillment for the award of the degree*

*of*

**MASTER OF COMPUTER APPLICATIONS**

*in*

**COMPUTER APPLICATIONS**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

# KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

**COIMBATORE – 641 006.**

Department of Computer Applications

**PROJECT WORK**

**MAY 2010**

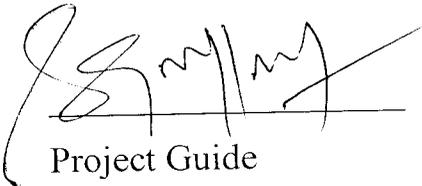
This is to certify that the project entitled  
**ASYNCHRONOUS INSTITUTE SERVICE**  
**(STUDENT INFORMATION SYSTEM)**

is the bonafide record of project work done by

**T.SOWMYAMOORTHY**

**Register No: 0720300046**

of MCA (Computer Applications) during the year 2009-2010.



Project Guide



Head of the Department

Submitted for the Project Viva-Voce examination held on 17.05.2010



Internal Examiner



External Examiner

## DECLARATION

I affirm that the project work titled **ASYNCHRONOUS INSTITUTE SERVICE** being submitted in partial fulfillment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

*T. Sowmya Meethy*

(Signature of the Candidate)

*T. Sowmya Meethy*

Name of the Candidate

*0720300046*

Register Number

I certify that the declaration made above by the candidate is true

*S. Ganesan*  
05/05/17

Signature of the Guide,

*S. GANESAN*  
SL/MC

With Name & Designation



www.GuruG.net

Educational environment for all

---

## **Project Completion Certificate**

We are providing this to Mr.T.Sowmyamoorthy (Reg No 0720300046) doing Final Year MCA in "Kumaraguru College Of Technology" for the Project "Asynchronous Institute Service" completed with our extreme Organization.

**Project duration Dec 2009 to May 2010.**

The developed software have been tested and forwarded to User Acceptance testing .After successful implementation of the application we are issuing the certificate.

  
Managing Director

# ASYNCHRONOUS INSTITUTE SERVICE

## (STUDENT INFORMATION SYSTEM)

### ABSTRACT

The Asynchronous Institute Service is one of the rapid growing applications which are used in every School in different forms. This particular system which is based on client- server architecture provides few of the administration tasks and also tracks the details of each student and staff in the department.

This client – server based application can be loaded at server and can be accessed from a different machine in a network (intranet) through a portal named ‘Browser’. This allows a high portability for the application which need not be installed in every machine where it can be used.

The multi-user application can be accessed through the browser from client end and the server takes up the responsibilities to perform a particular operation based upon the type of user logged in. It also handles an unauthenticated user or operation which is performed on the application by returning back an appropriate warning message.

The system identifies the tasks to be performed based upon the authenticated user.

The modules used this system are Admin, Staff, Student.

- **Admin:** Admin user can add a student or staff. A profile search also can be done along with other administrative tasks like maintaining staff allocation for each class and students.
- **Staff:** Staff can maintain their personal details along with viewing staff and student’s details. Teaching and non-teaching staffs can maintain their own details. Teaching staffs can add attendance for students and enter marks for each exam.
- **Student:** Students can view their personal details, attendance on the basis of date or subjects, and mark details.

This web-based application is developed using Flex, PHP with HTML, MySQL, and WampServer.

## ACKNOWLEDGEMENT

I wish to express sincerest thanks to **Dr. J. Shanmugam** , Director-Kumaraguru College of Technology, **Dr. S. Ramachandran**, Principal-Kumaraguru College of technology and **Dr. S. Thangasamy** – Dean , Department of Computer Applications for providing necessary facilities in carrying out my project work

I wish to thank my Project coordinator **Mr. S. Hameed Ibrahim, Sr. Lecturer** for his sincere advice, thought provoking discussions and immense help throughout the project and encouragement given by him.

I wish to thank my Project guide **Mr. S. Ganesh Babu, Sr. Lecturer** for his sincere advice, thought provoking discussions and immense help throughout the project and encouragement given by him.

I wish to thank all my staff members for their timely help and guidance to complete the project successfully.

Also I would like to thank my parents, friends, and all those who helped me in this project and whose names are leftover.

## TABLE OF CONTENTS

CHAPTER NO	PAGE NO
ABSTRACT	iv
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	x
<b>1. INTRODUCTION</b>	
1.1 ORGANIZATION PROFILE	1
1.2 PROJECT OVERVIEW	2
<b>2. SYSTEM STUDY AND ANALYSIS</b>	
2.1 EXISTING SYSTEM	3
2.1.1 DISADVANTAGES OF EXISTING SYSTEM	3
2.2 PROPOSED SYSTEM	4
2.2.1 ADVANTAGES OF PROPOSED SYSTEM	4
<b>3. SYSTEM CONFIGURATION</b>	
3.1 HARDWARE REQUIREMENTS	5
3.2 SOFTWARE REQUIREMENTS	5
3.3 SOFTWARE OVERVIEW	6
3.3.1 PHP(Hypertext Pre-Processor)	6
3.3.2 MYSQL	7

3.3.3	WAMPSEVER	8
3.3.4	FLEX	9
<b>4</b>	<b>SYSTEM DESIGN</b>	
4.1	USE CASE DIAGRAM	11
4.1.1	System Environment Use Case Diagram	11
4.1.2.	Login System Use Case Diagram	12
4.1.3	Admin Use Case Diagram	13
4.1.4	Staff Use Case Diagram	13
4.1.5	Student Use Case Diagram	14
4.2	ELEMENTS OF DESIGN	
4.2.1	Input Design	15
4.2.2	Output Design	17
4.2.3	Database Design	18
4.2.4	Module Design	23
<b>5.</b>	<b>SYSTEM TESTING AND IMPLEMENTATION</b>	
5.1	IMPLEMENTATION	24
5.2	SYSTEM VERIFICATION	24
5.3	SYSTEM VALIDATION	25
5.4	TESTING	25

<b>6.</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	
	6.1 CONCLUSION	27
	6.2 FUTURE ENHANCEMENT	28
<b>7.</b>	<b>APPENDICES</b>	
	SCREEN SHOTS	29
<b>8.</b>	<b>REFERENCES</b>	

## LIST OF FIGURES

Figure No.	Description	Page No.
4.1(a)	System Environment Use Case Diagram	11
4.1(b)	Login System Use Case Diagram	12
4.1(c)	Admin Use Case Diagram	12
4.1(d)	Staff Use Case Diagram	13
4.1(e)	Student Use Case Diagram	14
A.1	Home Page	29
A.2	Admin Home Page	30
A.3	Student Window	31
A.4	Login page	32
A.5	Attendance Window	32
A.6	Attendance Report	33
A.7	Attendance Chart Report	33
A.8	Mark Report	34
A.9	Mark Report Chart	34

**LIST OF TABLES**

<b>Table No.</b>	<b>Description</b>	<b>Page No.</b>
4.3(a)	Student_Details	18
4.3(b)	Staff_Details	19
4.3(c)	Class_Details	19
4.3(d)	Staff_Allocation	20
4.3(e)	Overall_attendance	20
4.3(f)	Holiday_Specification	20
4.3(g)	Marks_details	21

## CHAPTER 1

### INTRODUCTION

This chapter is organized into two parts. The first part deals with the organization profile. It provides a brief insight into the history of the organization and the products. The second part gives an introduction about the project.

#### 1.1 ORGANIZATION PROFILE

GURU's @ GURUG.Net is a dedicated team of faculty members provides their excellent teaching capabilities which makes GURUG.Net is a successful portal. The faculty comprises of high qualified and experienced who are distinguished in field of study. GURUG.Net mission is to master all profession intelligence under one umbrella, where students can be trained on essential professional subjects and become industrial ready. Precisely , gurug.net is a forum for knowledge sharing of profession . Other than providing excellent study materials for Anna University B.E , gurug.net also develop and maintain the software's for schools. Gurug.net maintains the student and staff records of Kikani Schools where more than 1500 students are studying.

GURUG.Net was inaugurated by Dr . K . Kulandaivel , the former Chancellor of Avinashilingam University on 2<sup>nd</sup> October 2008. With technology permitting various spheres of live, e-learning is also catching up with the plethora of course online . Technology enhances the effectiveness of the teacher and also provides the effective teaching expenses to the students .GuruG.net is aiming the to serve the students through e-learning.

GuruG.net is an educational resources hub for students and teaching community to advance knowledge and educates students in an effective way. The main purpose of GuruG.net is to provide free and open of quality study materials and educational software's

GuruG.net is launched for prime aspiration to promote education to higher level by assisting the students through providing exposure and employability .

## 1.2 PROJECT OVERVIEW

“**Asynchronous Institute Service** “(Student Management System) focuses on the process carried to fulfil the needs of staff and student in a school by maintaining their details such as profile, attendance, marks . The system identifies user based on their username and provides their authorized menu for further process.

The system is a client-server based application and can be accessed from different machine in a network. It manages staff and student operations carried a school according to their classes. This web-based application is developed using Flex, PHP, MySQL, and WampServer . The system can be used in either internet or intranet environment.

The users of the system are Administrator, Staff, and Student. Modules such as Admin, Student, and Student perform various operations to maintain and access the data in the system.

Administrator can add new student and/or staff of the class. If a profile is added, his/her number is saved in the login table to make him/her as an authorized user of this system.

Staff can view their personal details; can maintain attendance and marks, for various subjects, for all the students in the class. They enter details of attendance and produce reports, visual charts .

Student can view personal details. Attendance can be viewed from start date and end date. Subject wise marks are also displayed in visual charts.

## **CHAPTER 2**

### **SYSTEM STUDY AND ANALYSIS**

A complete understanding of the requirement is essential for the success of software development. The software scope, initially established by the system engineer and refined during the project planning, is refined in detail. Model of the required data, information and control flow, and operational behaviour are created. Alternative solution are analyzed and allocated to various software elements. The feasibility study evaluates the viability of the project and presents the recommended strategy adopted for the development.

#### **2.1 EXISTING SYSTEM**

In existing system, either the system is manual or the computer system is not efficient i.e not user friendly , less graphical interface and low secure .

##### **2.1.1 Drawbacks of the Existing System**

- In existing system, manual work is needed to maintain details of staffs and students.
- Attendance and marks details of student are maintained manually, which needed lot of paper work.
- While entering records, careless mistakes or error might occur.
- Marks allotments for each subject does not reach to all the students.
- No user interface (friendly) or graphical visual, if the system is computer based.

## **2.2 PROPOSED SYSTEM**

In the proposed system, administrator maintains overall process carried in a school especially adding new student or staff, staff allotment for each academic year. Staff will enter attendance and marks for students and those entries are stored in the database. Students can view only the entries entered by staff under their roll number.

### **2.2.1 Advantages of Proposed System**

The expected benefits of the proposed system are

- Based on client- server architecture
- Run as stand alone system, but accessible globally.
- Provides few of the administration tasks and also tracks the details of student and staff in the School.
- Handles an unauthenticated user or invalid operation by returning back an appropriate warning message.
- The system identifies the tasks to be performed based upon the authenticated user.
- This web-based application is developed using Flex, PHP , MySQL and WampServer all are open source ,which is cost effective.

## CHAPTER 3

### DEVELOPMENT ENVIRONMENT

#### 3.1 HARDWARE REQUIREMENTS

The hardware support required for deploying the application

Processor	:	Intel Pentium IV
Speed	:	3.1 GHZ
Memory	:	1 GB RAM
Hard Disk Capacity	:	80 GB
Monitor	:	15" inch SVGA
Mouse	:	Logitech Mouse (Scroll)
Keyboard	:	108 Keys

#### 3.2 SOFTWARE REQUIREMENTS

The software support required for deployment is

Operating System	:	Windows 7
Designing Tool	:	Zend Studio, Flex Builder 3.0
Scripting Language	:	PHP 5.3.0
Web Server	:	WampServer2.0i
Database	:	MY SQL 5.1

### 3.3 SOFTWARE OVERVIEW

#### PHP:

**PHP: Hypertext Preprocessor** is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems' JavaServer Pages, and mod\_perl.

PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not processed by PHP (although non-PHP text is still subject to control structures described within PHP code). The most common delimiters are `<?php` to open and `?>` to close PHP sections. The first form of delimiters, `<?php` and `?>`, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted (") and heredoc strings allow the ability to embed a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: `/* */` marks block and inline comments; `//` as well as `#` are used for one-line comments. The echo statement is one of several facilities PHP provides to output text (e.g. to a web browser).

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. If conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

**MYSQL:**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Sun Microsystems, Inc.

**MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

**MySQL is a relational database management system.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

**MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License, to define what you may and may not do with the software in different situations.

**The MySQL Database Server is very fast, reliable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

**MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

**A large amount of contributed MySQL software is available.**

It is very likely that your favorite application or language supports the MySQL Database Server.

**WAMP SERVER:**

WAMP5 (WAMP means Windows Apache Mysql PHP) is a platform of Web development under Windows. WAMP is a form of mini-server that can run on almost any Windows Operating System. WAMP includes Apache 2, PHP 5 (SMTP ports are disabled), and MySQL (phpMyAdmin and SQLitemanager are installed to manage your databases) preinstalled.

WampServer is a Windows web development environment. It allows you to create web applications with Apache, PHP and the MySQL database. It also comes with PHPMYAdmin and SQLiteManager to easily manage your databases.

An icon on the taskbar tray displays the status of WAMP, letting you know if; a) WAMP is running but no services are opened (the icon will appear red), b) WAMP is running and one service is opened (the icon will appear yellow) or c) WAMP is running with all

services (they can be disabled by left-clicking on the taskbar icon, guiding your cursor over the service you wish to disable and selecting "Stop Service").

The files/web pages that are hosted on your WAMP server can be accessed by typing *http://localhost/* or *http://127.0.0.1/* in the address bar of your web browser. WAMP must be running in order to access either of the above addresses.

## **FLEX:**

Flex is a highly productive, free, open source framework for building expressive web applications that deploy consistently across browsers, desktops, and operating systems by leveraging the Adobe® Flash® Player and Adobe AIR® runtimes. While Flex applications can be built using only the Flex framework, Adobe Flash Builder™ (formerly Adobe Flex® Builder™) software can accelerate development through features like intelligent coding, interactive step-through debugging, and visual design of the user interface layout.

Flex is the way to make rich Internet applications (RIAs) quickly and easily. At its basic level, it's a framework for creating RIAs based on Flash Player. Along with being a framework, Flex is also a new language. At its heart is MXML, a markup language based on Extensible Markup Language (XML) that makes it really easy and efficient to create applications. Unlike developing for some desktop platforms requiring a proprietary binary file format, MXML is just text, so it's easy to read and modify using just a text editor. Therefore, sharing code is as easy as sharing a simple text file.

MXML is an XML language that you use to lay out user interface components for Adobe® Flex® applications. You also use MXML to declaratively define nonvisual aspects of an application, such as access to server-side data sources and data bindings between user interface components and server-side data sources.

MXML development is based on the same iterative process used for other types of web application files such as HTML, JavaServer Pages (JSP), Active Server Pages (ASP), and ColdFusion Markup Language (CFML). Developing a useful Flex application is as easy as opening your favorite text editor, typing some XML tags, saving the file, requesting the file's URL in a web browser, and then repeating the same process.

In the Flex model-view design pattern, user interface components represent the view. The MXML language supports two types of user interface components: controls and containers. Controls are form elements, such as buttons, text fields, and list boxes. Containers are rectangular regions of the screen that contain controls and other containers.

Remote-procedure-call (RPC) services let your application interact with remote servers to provide data to your applications, or for your application to send data to a server. Flex is designed to interact with several types of RPC services that provide access to local and remote server-side logic.

The MXML components that provide data access are called RPC components. MXML includes the following types of RPC components:

- WebService provides access to SOAP-based web services.
- HTTPService provides access to HTTP URLs that return data.
- RemoteObject provides access to Java objects using the AMF protocol (Adobe LiveCycle Data Services ES only).

## CHAPTER 4 SYSTEM DESIGN

### 4.1 DIAGRAMS

#### 4.1.1 USE-CASE DIAGRAMS

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**Fig 4.1(a): System Environment Use Case Diagram**

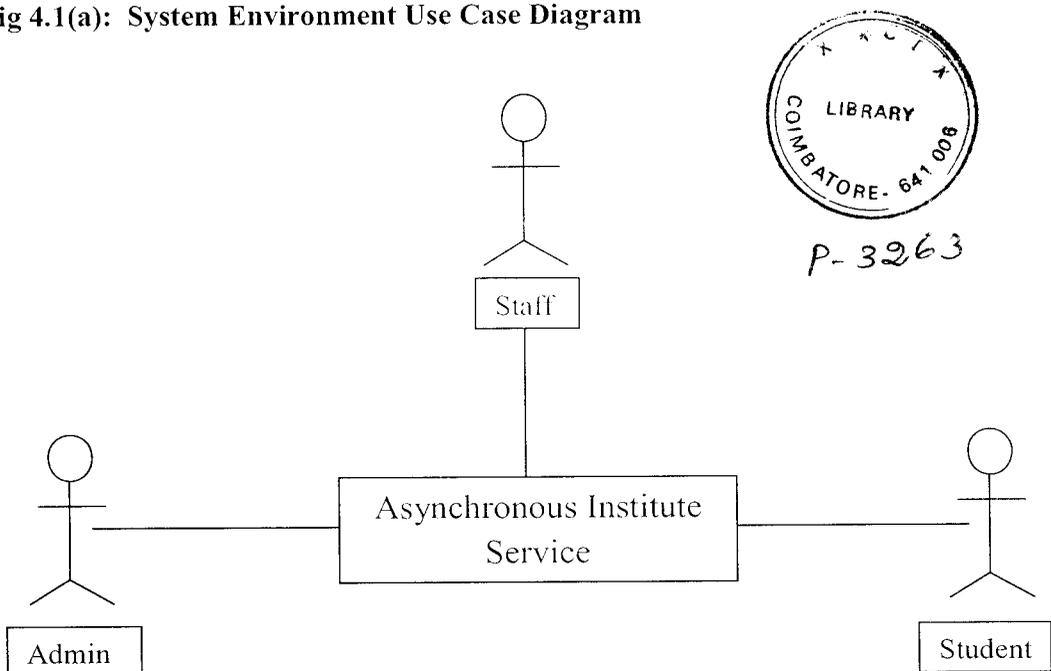


Fig 4.1(b): Login System Use Case Diagram

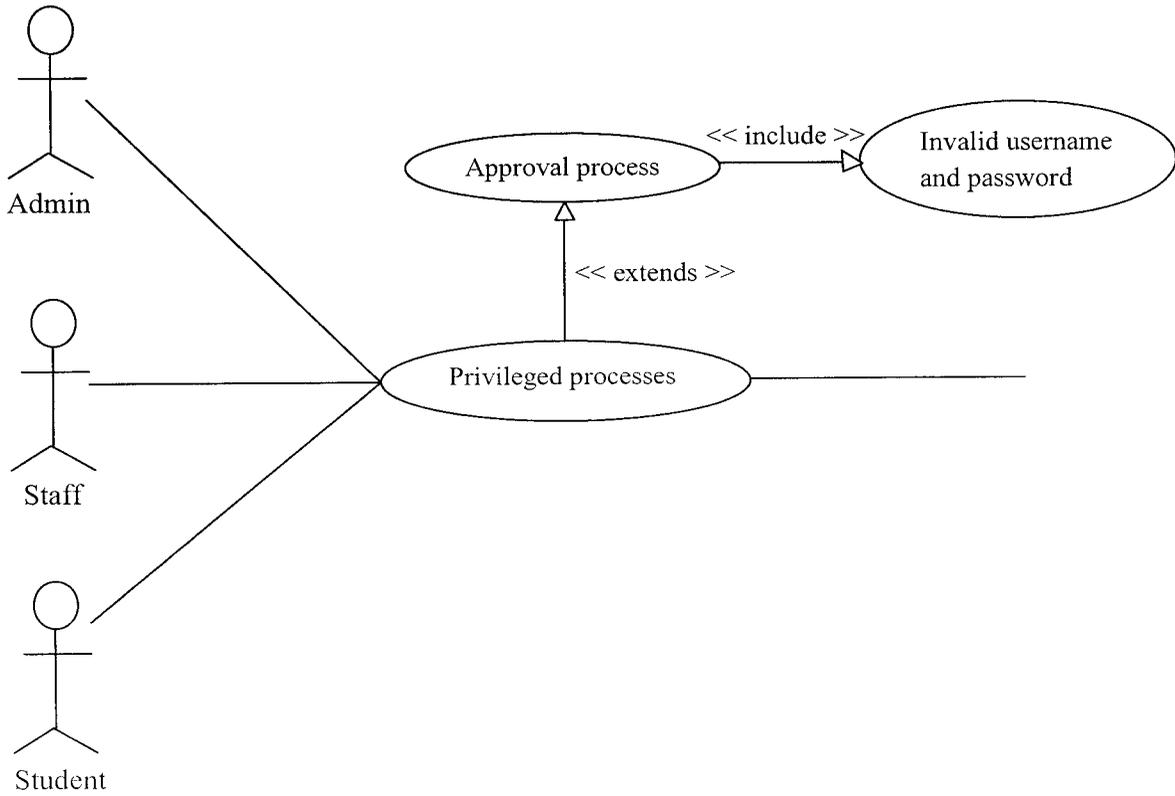
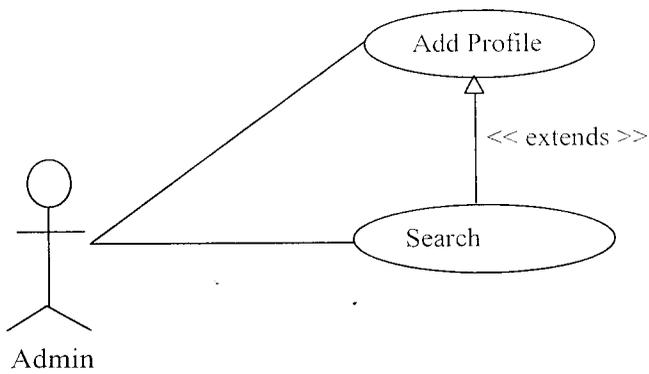
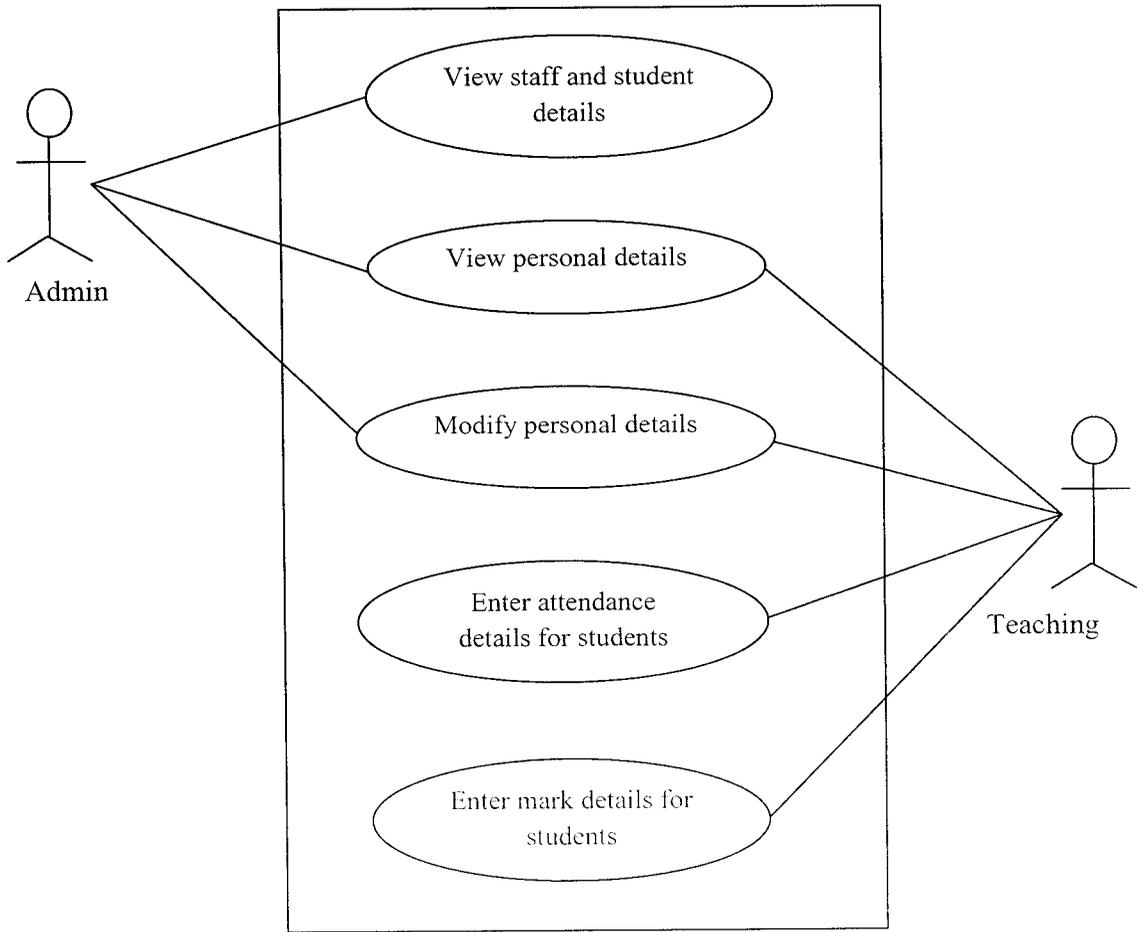
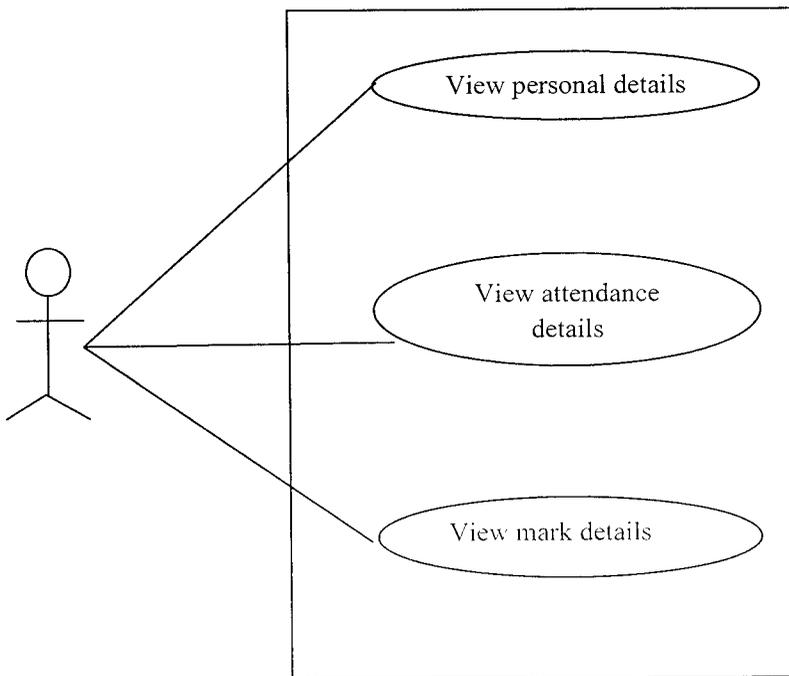


Fig 4.1(c): Admin Use Case Diagram



**Fig 4.1(d): Staff Use Case Diagram**



**Fig 4.1(e): Student Use Case Diagram**

## 4.2 ELEMENTS OF DESIGN

System Design is the most creative and challenging phase in the development of a software system. The first step is to determine what input data is needed for the system and then to design a database that will meet the requirements of the proposed system. The next step is to determine what outputs are needed from the system and the format of the output to be produced.

During the design of the proposed system some areas where attention is required are:

- ⌚ How are the inputs required and the outputs produced?
- ⌚ How should the data be organized?
- ⌚ What will be the processes involved in the system?
- ⌚ How should the screen look?

The steps carried out in the design phase are as follows:

- ✓ Input Design
- ✓ Output Design
- ✓ Database Design

### 4.2.1 INPUT DESIGN

Input design is the process of correcting a user-oriented description of the inputs to a computer based one. Inaccurate data is one of the most common causes of data processing errors. If poor input design, particularly where operators enter data from source documents permit wrong data to enter into a computer system, then it will change the entire process in an unpleasant way consists of,

- The sequence of field matches the sequence of data or type of data which is going to be entered.
- The data format is well identified for entering or specifying the data

In this system, the screen includes appropriate labels or prompts for data entry which

1. The screen is not over crowded. It facilitates the user to identify the labels easily and enter the data.
2. Provides validation when user enters a text in numeric field such as phone number, pin code, marks, etc or a number in a character field like name, title and organization name of the student's project, staff's designation, etc., or alphanumeric character in roll number, staff number, etc or invalid format entry in email id.
3. Tool tips are provided which can make clear when user is confused about the option to choose. For instance, in the menu provided to admin, staff, or student, the buttons shows appropriate tip when the mouse pointer is hovered to make the user to identify the menu easily.
4. The input designs are designed in such a way that the user can follow easily. The text boxes are arranged such that maximum characters it accepts does not exceed the need. For example, the pincode text box's maximum character size is set as 6 digits, phone number text box is arranged to accept 10 digits.
5. The values for drop-down boxes for degree and branch are dynamically updated from database when the page is called from the server. This makes view of updates made in the table when the page is loaded from the server.
6. Different categories of elements are grouped under <fieldset> tag. For example, in add student form, separate divisions are made between address and pincode.
7. The menu item for the authorized user is made visible in all the forms they choose to make navigation easy and friendly.

### 4.2.2 OUTPUT DESIGN

Output generally refers to the results and information that are generated by the system. For many end-users, output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Most end-users will not actually operate the information system or enter data through workstations, but they will use the output from the system. When designing output, system analysis must accomplish the following.

- ☞ Determine what information to present
- ☞ Decide whether to display, print or speak the information and select the output medium
- ☞ Arrange the presentation of information in an acceptable format.
- ☞ Decide how to distribute the output to intended recipients.

The arrangement of information on a display or printed document is termed as layout. Accomplish the general activities listed above will require specific decisions, such as whether to use pre-printed forms when preparing reports and documents, how many lines to plan on a printed page or whether to use graphics and colour.

The output design is specified on layout performs, sheets that describe the location characteristics, and format of the column headings and pagination.

The output must be provided in a format easily understandable even by a novice user. After analyzing the operations of the system, output information required for each jobs are determined. In addition to this, these outputs may be in format suitable as inputs for subsequent processing.

### 4.2.3 DATABASE DESIGN

A database is a collection of inter-related data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. An elegantly designed database can play a strong foundation for the whole system.

The details about the relevant data for the system are first identified. According to their relationship, tables are designed through the following method.

- The data type for each data item in the table is decided.
- The tables are then normalized.

The tables are normalized so that they can provide better response time, have data integrity, avoid redundancy and be secure.

#### DATABASE STRUCTURE:

This system uses many numbers of tables to store the details of agents and clients. It also contains the table about transactions.

#### 4.3(a): Table Name: Student\_Details

This table contains the details about the Student.

Field Name	Data type	Size	Constraint
Student_id	Varchar	20	Primary
Student_name	Varchar	50	
Class_id	Varchar	20	Foreign key References Class_details
Father_name	Varchar	50	
Mother_name	Varchar	50	
Dob	Date	-	
Phone	Big Int	12	
Email_id	Varchar	50	
Address	Varchar	100	
Pincode	Int	6	

**4.3(b): Table Name: Staff\_Details**

This table contains the details about the Staff's.

Field Name	Data type	Size	Constraint
Staff_id	Varchar	50	Primary
Staff_name	Varchar	50	
Dob	Date	-	
Sex	Varchar	6	
Degree	Varchar	50	
Email_id	Varchar	50	
Phone	Big Int	12	
Address	Number	100	
Pincode	Int	6	
State	Varchar	50	

**4.3(c): Table name: Class\_details**

This table contains the list of Classes.

Field Name	Data type	Size	Constraint
Class_id	Varchar	20	Primary Key
Class_name	Varchar	50	

**4.3(d): Table name: Staff\_allocation**

This table consists of the staff details which specifies to what classes there are responsible to mark attendance and marks.

Field Name	Data type	Size	Constraint
Staff_id	Varchar	50	Foreign Key references Staff_details
Class_id	Varchar	20	Foreign Key references Class_details
Passwrđ	Date		

**4.3(e): Table Name: Overall\_attendance**

This table contains the list of students entry added into attendance sheet.

Field Name	Data type	Size	Constraint
Student_id	Varchar	20	Foreign key referenced from Student_details
Class_id	Varchar	20	Foreign key referenced from Class_Details

**4.3(f): Table Name: Holiday\_specification**

This table contains the list of holidays.

Field Name	Data type	Size	Constraint
Holiday_id	Varchar	10	Primary key
Holiday_date	date		

**4.3(g): Table Name: marks\_Quarterly**

This table contains the policy details of each client who owned the policies.

Field Name	Data type	Size	Constraint
Student_id	Varchar	20	Foreign key referenced from Student_details
Class_id	Varchar	20	Foreign key referenced from Class_details
Tamil	Int	3	
English	Int	3	
Math	Int	3	
Science	Int	3	
Social_Science	Int	3	

**4.3(h): Table Name: marks\_Halfyearly**

This table contains the policy details of each client who owned the policies.

Field Name	Data type	Size	Constraint
Student_id	Varchar	20	Foreign key referenced from Student_details
Class_id	Varchar	20	Foreign key referenced from Class_details
Tamil	Int	3	
English	Int	3	
Math	Int	3	
Science	Int	3	
Social_Science	Int	3	

**4.3(i): Table Name: marks\_Annual**

This table contains the policy details of each client who owned the policies.

Field Name	Data type	Size	Constraint
Student_id	Varchar	20	Foreign key referenced from Student_Details
Class_id	Varchar	20	Foreign key referenced from Class_Details
Tamil	Int	3	
English	Int	3	
Math	Int	3	
Science	Int	3	
Social_Science	Int	3	

#### 4.2.4 MODULAR DESIGN

**Modular design** — or "modularity in design" — is an approach that subdivides a system into smaller parts (modules) that can be independently created and then used in different systems to drive multiple functionalities. Besides reduction in cost (due to lesser customization, and less learning time), and flexibility in design, modularity offers other benefits such as augmentation (adding new solution by merely plugging in a new module), and exclusion.

This system is also modularized to reduce the complexity of the system. This contains various modules.

##### A) User Modules:

User modules are modelled based on the users of the system. This module contains three users:

###### i) Admin

- ✓ add a student or staff
- ✓ profile search (student, staff)
- ✓ Holiday specification and disabling the dates

###### ii) Staff

- ✓ view and maintain their own details
- ✓ add attendance for students
- ✓ enter marks for each exam
- ✓ Send sms or mail to **Students** and **Parents**

###### iii) Student

- ✓ view their personal details
- ✓ view attendance on the basis of date
- ✓ view mark details

##### B) Reports Module

Reports forms the main output of any system. This system too has many reports which will help admin and agent to take many important decisions. This system contains many reports based on agent id, client id, dates, etc.

##### C) Charts Module

Reports is generated based on 2D or 3D charts for more visual and user friendly data generations based on attendance and marks details which consist of Total number days percent, attendance percentage, marks in each exam ,comparative analysis in all

## **CHAPTER 5**

### **SYSTEM IMPLEMENTATION AND TESTING**

#### **5.1 IMPLEMENTATION**

The system is implemented using Zend Studio- Flex builder, PHP and MYSQL.

##### **5.1.1 Implementation of Business Logic**

The business logic is implemented using Flex Builder. Flex builder contains many controls which make the website very user friendly. The data are entered using the controls and the output is displayed very effectively. This flex builder is very helpful in the way that it doesn't need to navigate through various pages. The loading of different pages for every function call is omitted which helps in improving performance. This IDE uses only one HTML page. The controls are made visible and invisible at needed times. The communication between the IDE and Database is through the Scripting language named PHP. The communication is in the form of XML data. This IDE uses an HTTP Object to send the data to the Scripting language. After processing of PHP file, the results are returned to IDE in the XML format which can be displayed using various controls and containers.

##### **5.1.2 Implementation of Database Communication**

For database communication, MYSQL is used along with PHP. MYSQL is the most popular Open Source SQL database management system. MYSQL has many inbuilt functions to carry out the operations with database. These functions are used as a part of PHP file. Using these functions the queries are executed and the operations on database are carried out. The inputs are received from action script file of IDE. The operations are performed according to the command received from IDE. Those operations are performed and the results are returned to the IDE using XML files.

#### **5.2 SYSTEM VERIFICATION**

**System Verification** is the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Verification is ensuring that the product has been built according to the requirements and design specifications- i.e. you built it right. Verification is the assurance

that the products of a particular development phase are consistent with the requirements of that phase and preceding phase(s).

In this website, review of interim work steps is done to ensure they are acceptable. In data access, it verifies whether the right data is being accessed in terms of the right place and in the right way.

### **5.3 SYSTEM VALIDATION**

**System Validation** is the process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. Validation checks that the product design satisfies or fits the intended usage (high-level checking) — i.e., you built the right product. This is done through dynamic testing and other forms of review. Validation ensures that the product actually meets the user's needs, and that the specifications were correct in the first place.

In this project, validation checks whether the developer is moving towards the right product. Validation coding is written using pre-defined validators available in Flex 3. Each field in registration form are validated such that the right username, password, date etc., is added. Any wrong entry display error messages or warnings. The login form is validated such that the valid registered user only can login to new page. Fields such as e-mail id and website are checked for its format. Validation also determines if this project complies with the requirements and performs functions for which it is intended and meets the organization's goal and user needs.

### **5.4 TESTING**

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The main type of test carried out in Asynchronous Institute Service is Unit Testing and Integration Testing.

#### **5.4.1 Unit Testing**

A series of stand-alone tests are conducted during Unit Testing. Each test examines an individual component that is new or has been modified. A unit test is also called a module test because it tests the individual units of code that comprise the application. Unit tests focus

module or specific to a particular module. Unit testing is done in a test environment prior to system integration. If a defect is discovered during a unit test, the severity of the defect will dictate whether or not it will be fixed before the module is approved.

In Asynchronous Institute Service each component i.e. each form is tested individually to verify that the detailed design for unit has been correctly implemented. Initially the flow of control and data through that page is checked. In a page, each control is further tested in unit testing. The process is done in all the forms of the system.

#### **5.4.2 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. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process are tested together. Beyond that, if the program is composed of more than one process, they should be tested in pairs rather than all at once. Integration testing identifies problems that occur when units are combined.

Many forms in the system have communication between each other. This helps in testing integration testing. For ex: if the staff logs in, first system should check whether he is a valid user and then it should move to the staff page and display some details of that staff. Here more than a single process is involved and so it needs integration testing.

### 5.4.3 TEST CASES

Si. No	Test Case	Test Procedure	Pre-Condition	Expected Result	Status
1	Login – valid input	Give valid username, password etc...	None	Should connect with the server and display welcome message.	Pass
2	Login – invalid input	Give invalid username, password etc...	None	Should display appropriate error message.	Pass
3	Email- invalid input	Give valid email i.e. @ symbol, domain etc...	None	Should display appropriate error message.	Pass
4	Date-invalid input	Give invalid date format string DD/MM/YYYY	None	Should display appropriate error message.	Pass
5	Date-valid input	Give valid date format string DD/MM/YYYY	None	Should display appropriate error message.	Pass
6	Data-upload by invalid user	This page is read only. You can view the source, but not change it. Ask your administrator for permission to change it.	None	Should display appropriate error message.	Pass

## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **6.1 CONCLUSION**

The system development titled “**Asynchronous Institute Service**” was designed keeping in mind to provide maximum flexibility to the admin, staff and student in the department. The system was tested with sample data and the performance of the system was found to be efficient. All efforts have been made to understand the system wholly and understand all the details. Much of the process is automated and hence the chance of human errors is limited. More security could have been given to the system by the way authorization and further feature could have been added to make the system more robust.

The system has been developed using Flex, WampServer, PHP and MySQL. The system was able to process and update the database with more cases. It helped in developing a total integrated system.

## 6.2 FUTURE ENHANCEMENT

This project has been developed as a Master's project and is constrained by time. There is scope for extending the system as per the need.

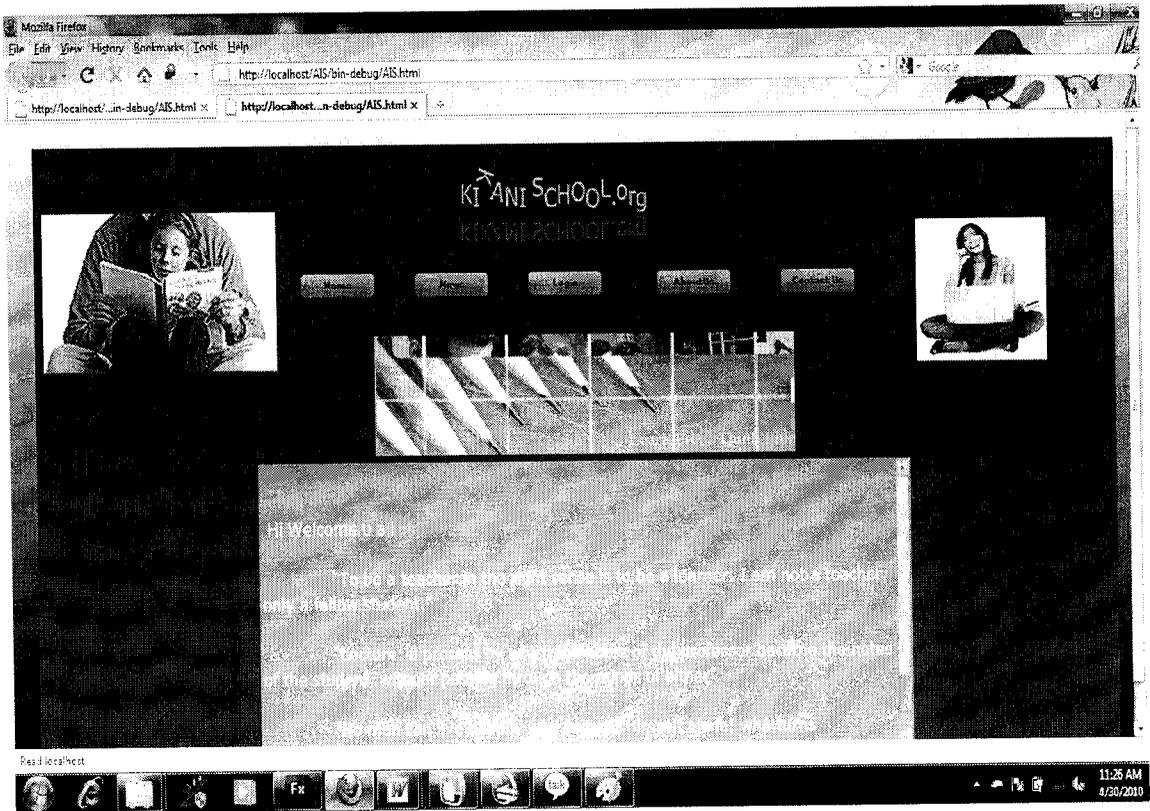
- Alumni module can be developed to maintain alumni details and make them access details of other students and staff.
- Reports of marks can be provided for staff.
- Backup of all the details can be made automatic after each academic year.

Thus the project Asynchronous Institute Service can be enhanced for further future requirements of the college.

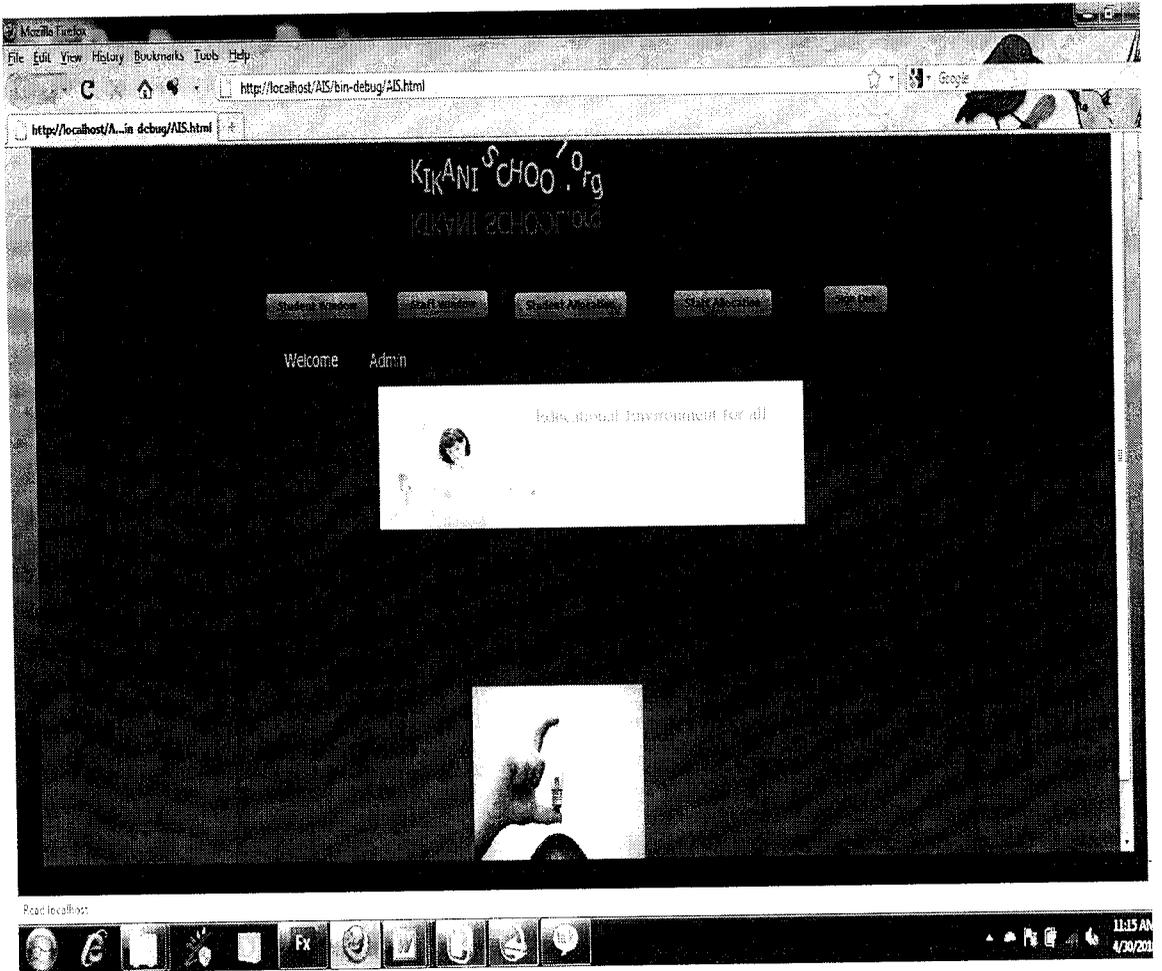
# APPENDIX

## SCREEN SHOTS

### A.1 Home Page



## A.2 Admin Home Page



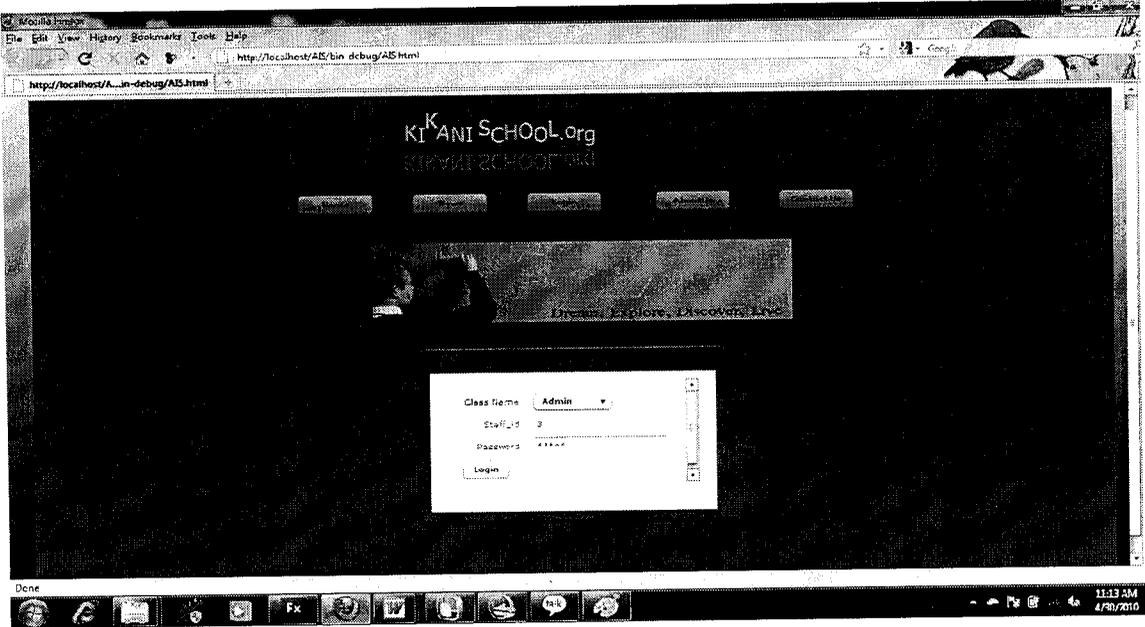
### A.3 Student Window

The screenshot shows a web browser window displaying the KVKANI SCHOOL.org student management interface. The page title is "KVKANI SCHOOL.org" and the URL is "http://localhost/AIS/bin-debug/AIS.html". The interface includes a navigation bar with buttons for "Admin", "Staff", "Students", "Reports", "Settings", and "Logout". Below the navigation bar, there is a "Welcome Admin" message. The main content area displays a table of student records with the following data:

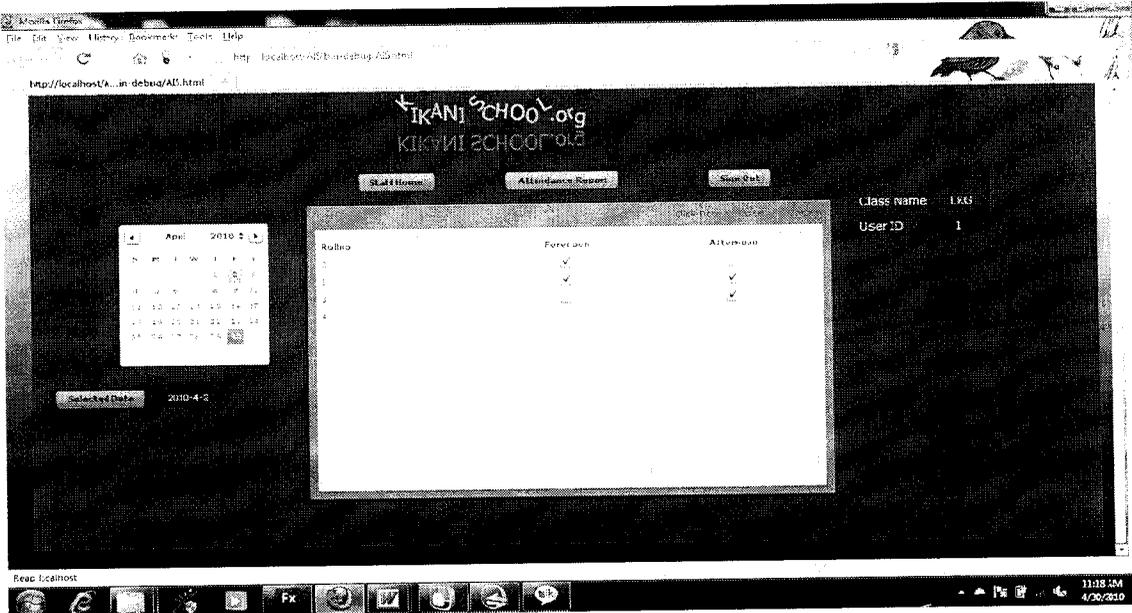
student id	student name	class id	father name	mother name	dob	sex	phone	email	address	pincode	state
1	vareha	UKG	samy	mehesh	2010-04-01	Male	9999222	vareha@gmail.com	omibhatra	641001	Tamilnadu
2	surva	UKG	raj	sheela	2010-04-01	Male	9999222	surva@gmail.com	villivaram	641001	Tamilnadu
3	rayu	UKG	kevi	bharami	2010-04-06	Female	02221234	raj@gmail.com	opp.nikara street	641001	Tamilnadu
4	kishora	UKG	nam	betta	2010-04-06	Female	2147483947	cssd@d.com	2342342	23-22422	Tamilnadu
5	vignesh	UKG	Raja	Mam	0006-09-00	Male	9999766	vign@gmail.com	dasdes	9881988	Tamilnadu
6	Sn	UKG	Dinesh	travisha	0006-09-00	Female	9999766	vign@gmail.com	dasdes	9881988	Tamilnadu

At the bottom of the table, there are buttons for "Add", "Delete", and "Search". The browser's taskbar at the bottom shows the system tray with the time 11:17 AM and date 4/20/2018.

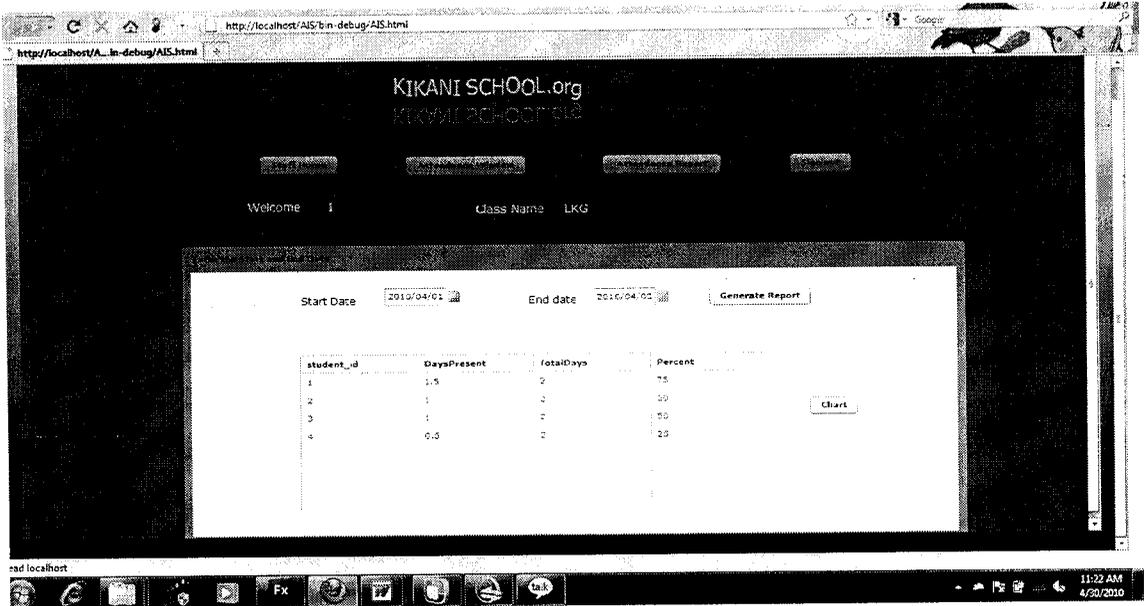
### A.4 Login



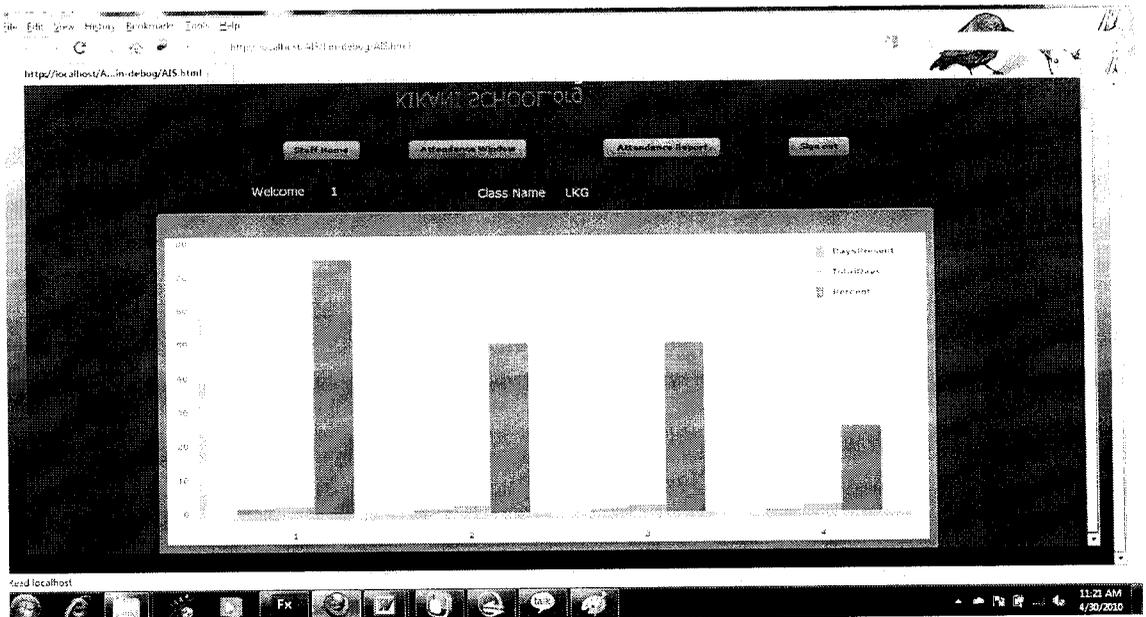
### A.5 Attendance Window



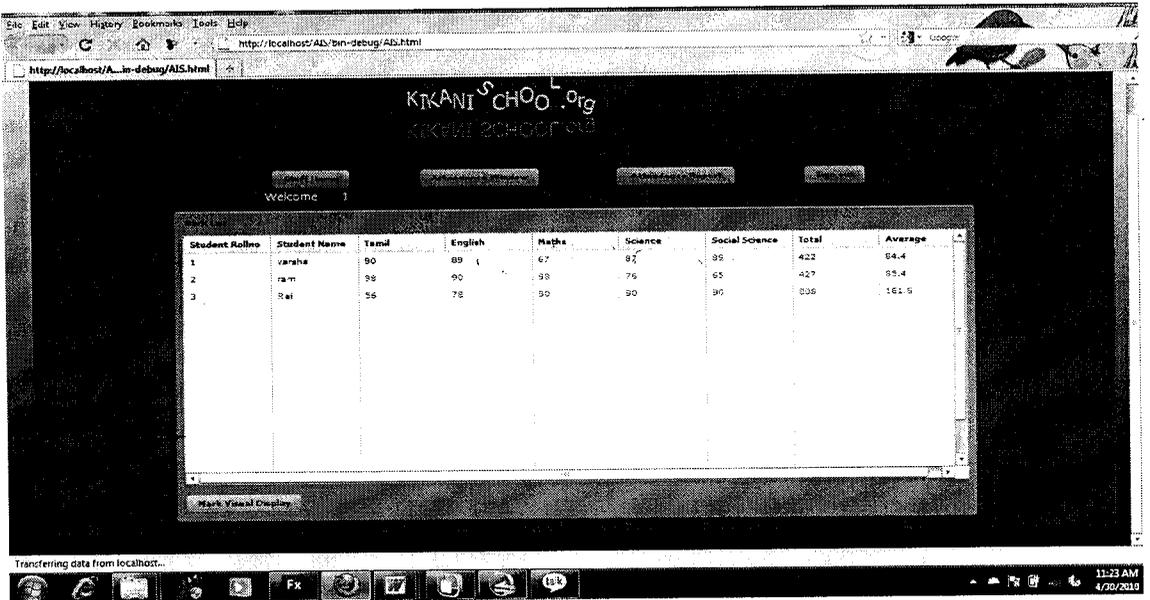
## A.6 Attendance Report



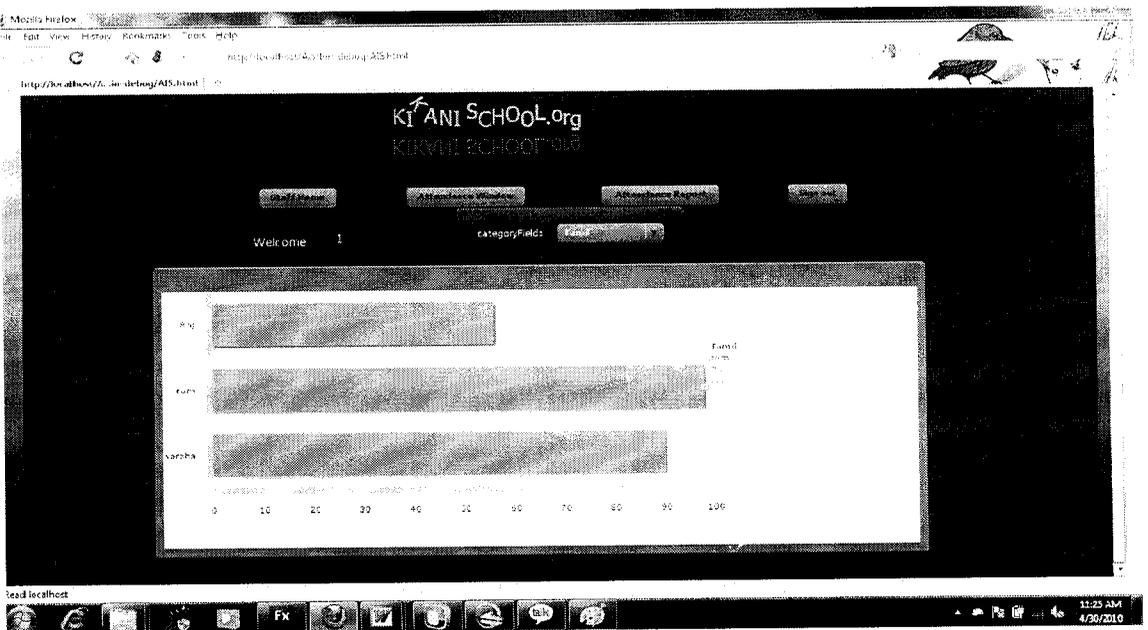
## A.7 Attendance Chart



### A.8 Mark Report



### A.9 Mark Report Chart



## REFERENCES

1. **Learning Flex 3 – Alaric Cole - O'Reilly Media, Inc., June 2008.**
2. **Adobe® Flex® 3 Developer Guide – Adobe Systems, 2008.**
3. **PHP 6/MySQL® Programming for the Absolute Beginner – Andy Harris – Course Technology, 2009.**
4. **<http://livedocs.adobe.com/flex/3>**
5. **<http://www.w3schools.com/PHP>**
6. **<http://devzone.zend.com>**
7. **<http://www.php.net/manual>**
8. **<http://www.actionscript.org>**