PROJECT REPORT

ON

ACCOUNT MANAGEMENT SYSTEM

FOR

PAARIJAT SOFTWARE SOLUTIONS Pvt. Ltd.

BY

RESHU

UNIVERSITY OF PUNE

MASTER OF COMPUTER APPLICATION

M.E.S's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411029

2012-13

Acknowledgement

I have an immense pleasure to present my project work entitled "Account Management System" done for "**Paarijat Software Pvt. Ltd**. ". The Project gave me a chance to get first hand industry experiences and at the same time helped me to enrich my skills and abilities. I am more than pleasure to thank **Dr. V. H. Inamdar (DirectorIMCC)**,

Mrs. ManasiBhate (Training & Placement Officer IMCC),

Dr. SantoshDeshpande (Head Of Department, IMCC)and

Mrs. AshwiniPatil (Co-coordinator, MCA) for giving me such animportant opportunity.

I thank all the teaching and non - teaching staff of IMCC without whom the completion of this project in time would have been a remote possibility.

Reshu

Index

	Contents	Page No.
	Chapter 1	
	INTRODUCTION	
1.1	Company Profile	1-2
1.2	Existing System and Need for System	3-4
1.3	Scope of Work	5-8
1.4	Operating Environment- Hardware and software	9-10
1.5	Details Description of Technology Used	11-16
	Chapter 2	
	PROPOSED SYSTEM	
2.1	Proposed System	17-18
2.2	Objective of System	19
2.3	User Requirements	20
	Chapter 3	
	ANALYSIS & DESIGN	
3.1	Object Diagram	21-22
3.2	Class Diagram	23-24
3.3	Use Case Diagram	25-30
3.4	Activity Diagram	31-35
3.5	Sequence Diagram	36-39
3.6	E-R Diagram	40
3.7	Module Hierarchy	41

3.8	Component Diagram	42
3.9	Deployment Diagram	43
3.10	Module Specification	44-48
3.11	Website Map	49
3.12	User Interface Design	50-59
3.13	Table Specification	60-67
3.14	Test Procedures and Implementation	68-81
	Chapter 4	
	USER MANUAL	
4.1	User Manual	82-96
4.2	Operations Manual	97-101
4.3	Program Specifications	102-105
	Drawbacks and Limitations	106
	Proposed Enhancements	107
	Conclusions	108
	Bibliography	
	Annexure 1:User Interface Screen	
	Annexure 2:Output Report	
	Annexure 3:Sample Code	

CHAPTER 1:

INTRODUCTION

1.1Company Profile

PaarijatSoftware'sis a leading solution provider for Internet based applications as well as standaloneapplications. The company has been promoted by some highly experienced Professionals dedicated to provide total IT solutions under one roof. It possesses not only the latest technology gadgets but also the most knowledgeableand experience hands to offer most user friendly customized solutions. PaarijatSoftware's provides high quality onsite services for software development and the end users on a broad range of hardware and software platforms and latest technologies.

AtPaarijatSoftware's with Technical proficiency and expertise, we cohesively integrate graphic design with web page layout, with interactive programming, with database driven content, to plan, build and deploy e-business and to emerge as one of the top IT serviceprovider.

PaarijatSoftware'sprovides services like Web Application C.M.S, Development,Mobile Development, Testing Q.A., e-Commerce, Off-Shore Outsourcing, Website/Server Maintenance Services.

The business philosophy of the company is to lay emphasis on Human Values and Personal Relations. 'At Paarijat Software's – end to end solutions for you'. Greatstress is laidon proper communication, transparency and human relations, whichforms an integral part of thecorporate culture. AtPaarijatSoftware's, we not only develop products but we develop relationships. We at PaarijatSoftware's believe in teamwork. With every new day the quest for acquiring new competencies continues. Forever searching, experimenting, innovating, learning, moving ahead with our sincere efforts and dedications, shaping the future, and challenging our competencies to create new opportunities, is a never-ending process in the company.

1.2 Existing system and Need for system

- The existing system is a manual system. Here the employees needs to save the information in the form of excel sheets or Disk Drives.
- There is no sharingis possible if the data is in the form of paper or Disk Drives.
- The manual system gives us very less security for saving,datasome data may be lost due to mismanagement.
- It's a limited system and fewer users friendly.
- Searching of particular information is very critical it takes lot of time.
- It is very critical to maintain manually fee details of student and salary details of employee.
- Searching fee details of student and payroll detailsemployee details in account management system is a tedious job.
- This system may not fulfill the requirements managing of account system properly.

- The current system may not organize students and employees details efficiently.
- In existing system there is no knowledge repository.
- Manual report generation is time taking.

1.3 Scope of Work

TheMain aim of Online"Account Management System"is to automate the process of Accounting transaction and provide the efficiency of administrative system.Reduce the tedious work of store and save the information of student fee details and salary of school staffs.It also provide facility for guardian to see online fee structure of their child.

Following are the option of fee module:--

Fee Period:-

Fee Period decided by admin whether it is monthly,quarterlyetc. eg-(jan-mar,apr-july).

Fee Structure:-

Admin define fee types like admission fee,tuitionfee,computer fee etcand define amount corresponding to fee type.Fee type define according to classes in the school.

Fee Concession:-

Admin define different Type Concessions to student on the basis of their category,cast,management quota etc.

Other Fee Collection:-

It manages the fee collection of other events and courses conducted during curriculum in the school.eg : computer training for one month, fluteclasses, cultural events etc.

Collect Fee:-

It takes admission number as an input and shows fee types, totalamount, concessionamount, late fine amount etc and calculates the final fee that has to be paid by the student/guardian.

SMS Facility:-

Guardian can receive the SMS from an accountant about the late fee date and amount to be paid.

Late Fine:-

Admin define late fine date and amount on that basis accountant calculate the fee of students.

Payroll System:-

Salary of an employee is calculated per month by an accountant.On the basis of leaves,designation.

It also manages salary of non-teaching staff such as: - accountant,administrator,HR,director etc.

Following are the option of Salary module:--

Deduction Details:-

Admin define deduction type and corresponding amount.

Pay Structure:-

Admin define the pay Structure type for the employee and amount corresponding to pay structure .it is different for teaching and non-teaching employee.

Generate pay slip:-

Accountant generates the pay slip for employee by calculating number of leave taken by employee.And how much deduction provided to employee.

SMS Facility:-

Employee can receive the SMS from an accountant about their salary details .

1.4 Operating Environment - Hardware and Software

Hardware Requirement:

Server Side:-

Processor: -Intel Pentium IV or More.

RAM : - 512 MB.

Hard Disk : - 80 GB.

Client Side:-

Processor: -Intel Pentium4 or More.

RAM : - 512 GB.

Hard Disk : - 80 GB.

Software Requirement:

Server Side:-

Internet Explorer 6.0 or later version.

MYSQL

Windows XP or later

JDK 1.5 or later

Tomcat 7.0

Client Side:-

Internet Explorer 6.0 or later version.

1.5 Detail Description of Technology Used

J2EE

The Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing multitier enterprise applications. The J2EE platform simplifies enterprise applications by basing them on standardize, modular components, by providing a complete set of services to those components, and by handling many details of application behavior automatically, without complex programming.

The J2EE platform takes advantage of many features of the Java 2 Platform, standard edition (J2EE), such as "Write Once Run Anywhere" portability, JDBC API for database access, CORBA technology for interaction with exiting enterprise resources, and a security model that protects data even in internet applications. Building on this base, Java 2 Platform, Enterprise Edition adds full support for enterprise JavaBeans components, java Servlets API, Java Server Pages and XML technology.

The J2EE standard includes complete specifications and compliance tests to ensure probability of applications across the wide range of existing enterprise system capable of supporting the J2EE platform.

JSP

Java Server Pages (JSP) is a java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 as Sun's answer to ASP and PHP, JSP was designed to address the perception that the Java programming equivalent didn't provide developers with enough support for the web.JSP syntax is a fluid mix of two basic content forms: Scriptlet elements and markup. Markup is typically standard HTML or XML, While Scriptlet elements are delimited blocks of java code. This may be intermixed with the markup. When the page is requested the java code is executed and its output is added, in situ, with the surrounding markup to create the final page.JSP pages must be compiled to Java byte code classes before they can be executed, but such compilation is needed only when a change to the source JSP file has occurred.

The JSP syntax adds additional XML-like tags called JSP actions, to invoke built in functionality.

TOMCAT 7.0 (SERVER)

Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, and provides a "pure java" HTTP web server environment for java code to run.

Tomcat should not be confused with the Apache web server, which is a C implementation of HTTP web server; these two web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

Advanced Tomcat Features

- The following subjects are discussed
- Accept Logs
- Single Sign-on
- Request Filtering

- Persistent Session Manager
- Tomcat and JDBC, JNDI
- JAVA Mail Session
- Configuring Lifecycle Listeners

MYSQL

MYSQL is a free, widely used SQL engine. It can be used as a fast database as well as a rock-solid DBMS using modular engine architecture.

The purpose of this wiki book is to provide a practical knowledge on using the database from two points of view :

- Application Developer: learn about SQL basics, MyAdmin, query optimization.
- System Administrator: learn about installation, security, maintenance, failover, high availability.

The reasons for using MYSQL as a backend are

- It is freeware.
- Open source.
- Relational database.
- Compatible with SQL.
- Supports cursors Triggers and Stored procedures.
- Graphical user Interface.
- Speed.
- Platform Independent.
- Performance.

CHAPTER 2:

PROPOSED SYSTEM

2.1 Proposed System

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.

- User Friendliness is provided in the application with various controls provided by system Rich User Interface.
- The system makes the overall project management much easier and flexible.
- The user information can be stored in centralized database which can be maintained by the system.
- This can give the good security for user information because data is not in client machine.
- Authentication is provided for this application only registered Users can access.
- There is no risk of data management at any level while the project development is under process.

- The speed and accuracy of this system will improve more and more.
- The system will provide reports in a formal way.

2.2 Objective of the System

- The main Objective of this system is to maintain considerable information of the students and employees of school.
- Maintain the fee and salary details of students and employees.
- Query handling in robust way.
- Maintains the different types of concession fee, other fee, deduction type, fee structure,pay structure etc.
- Calculating the late fine of students.
- Calculating the fee and salary details of students and employees of school.
- To maintain the knowledge repository in an efficient manner so that the search for query results become faster and easier.
- To generate all kinds of reports related to the organization.

2.3 User Requirement

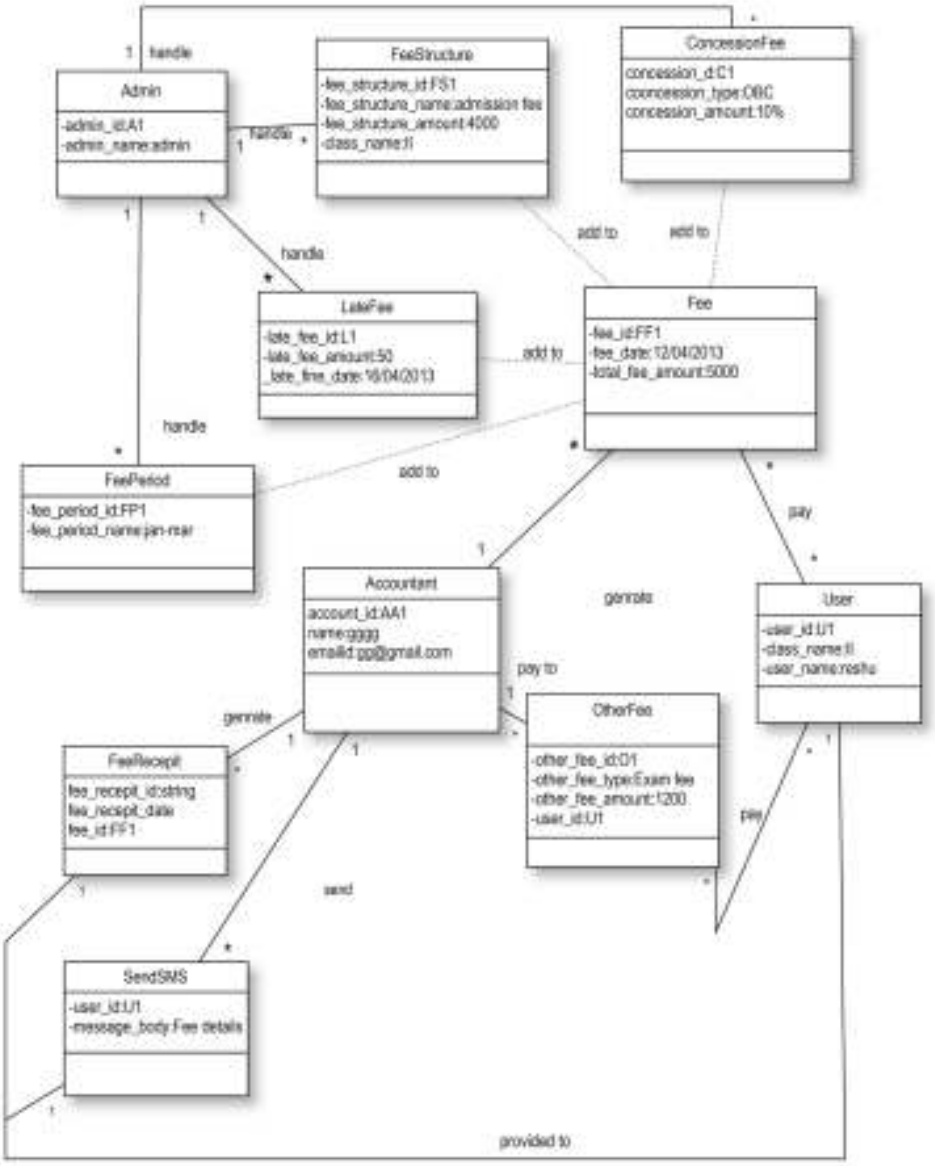
Sr No.	Requirement	Description
1	Automatic	System should contain more Of automatic work rather than manual work.
2	Record Searching	Searching of records should Be done easily.
3	Reflection of changes	If changes are to be made in one record then they have to be reflected in other required fields.
4	Report Generation	Report should be generated As per their requirement and in short duration of time.
5	Storing of transactions	All the transactions should be stored in the database in the proper sequence.
6	Interactivity	The system should be interactive.
7	Details of previous transactions	There should be easy way to retrieve the details of previous transactions.
8	Comparison between the records	There should be a convenient way for comparison to be made between the records.
9	Moderation of the Work	Moderation of the Work should be easily made.

CHAPTER 3:

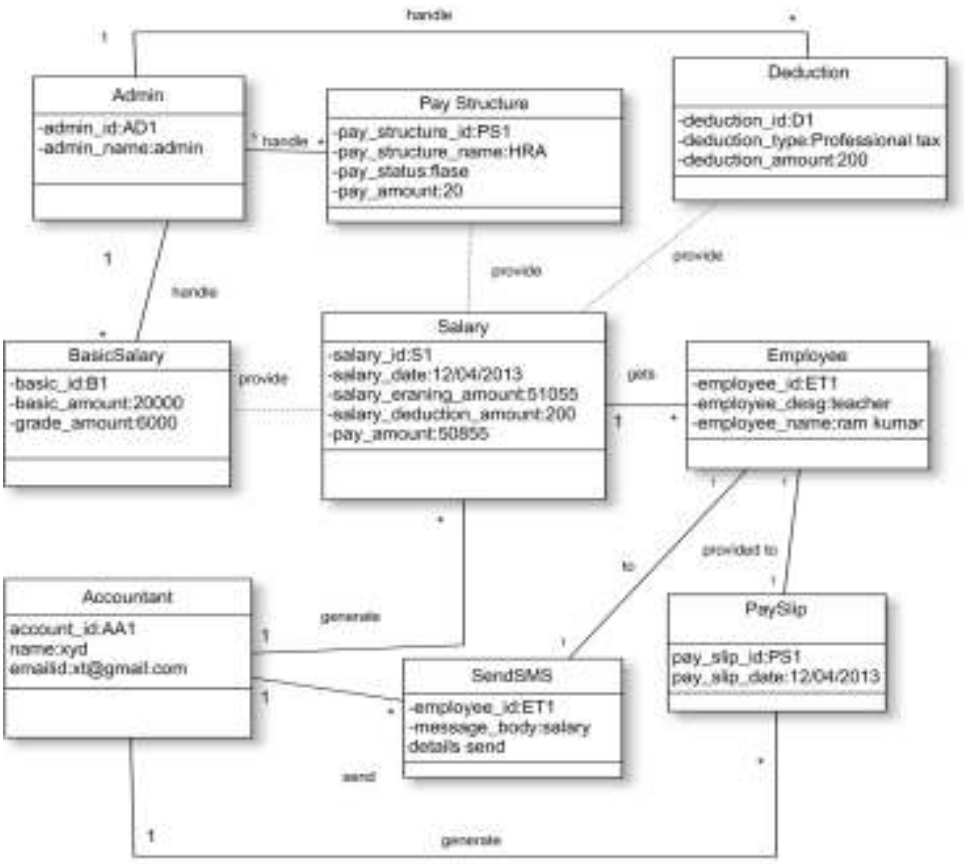
ANALYSIS & DESIGN

3.1Object Diagram

Object Diagram for Fee

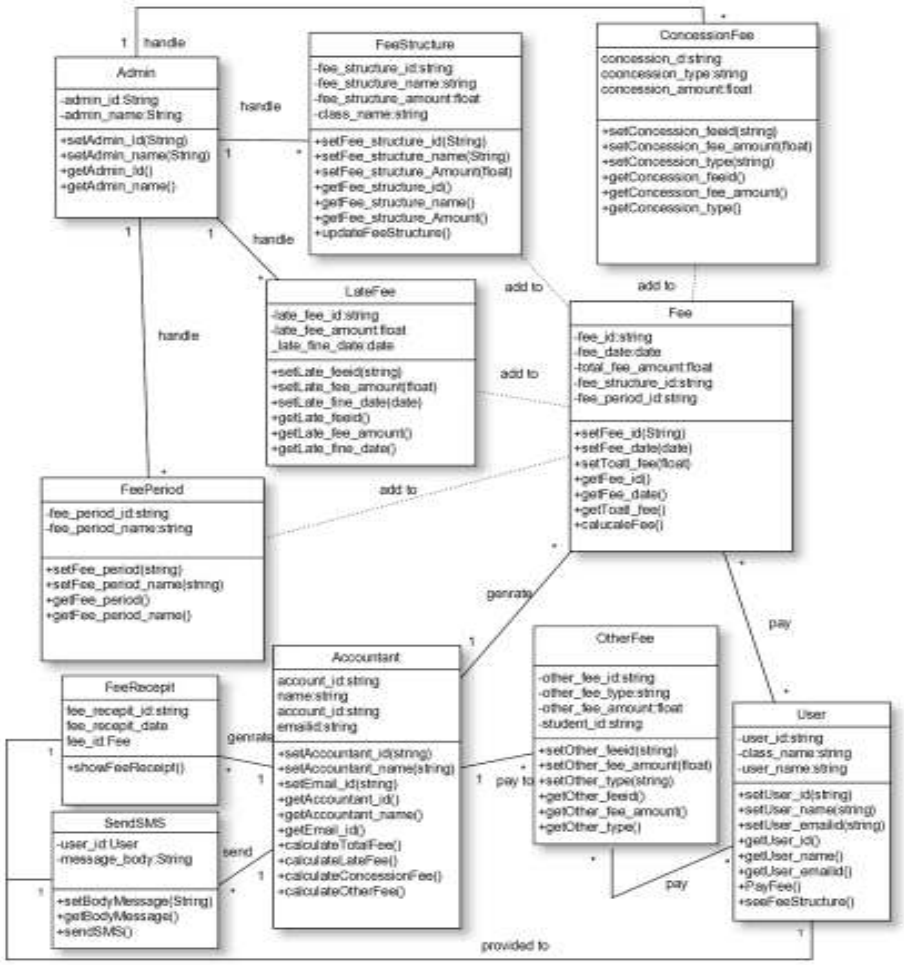


Object Diagram for Payroll

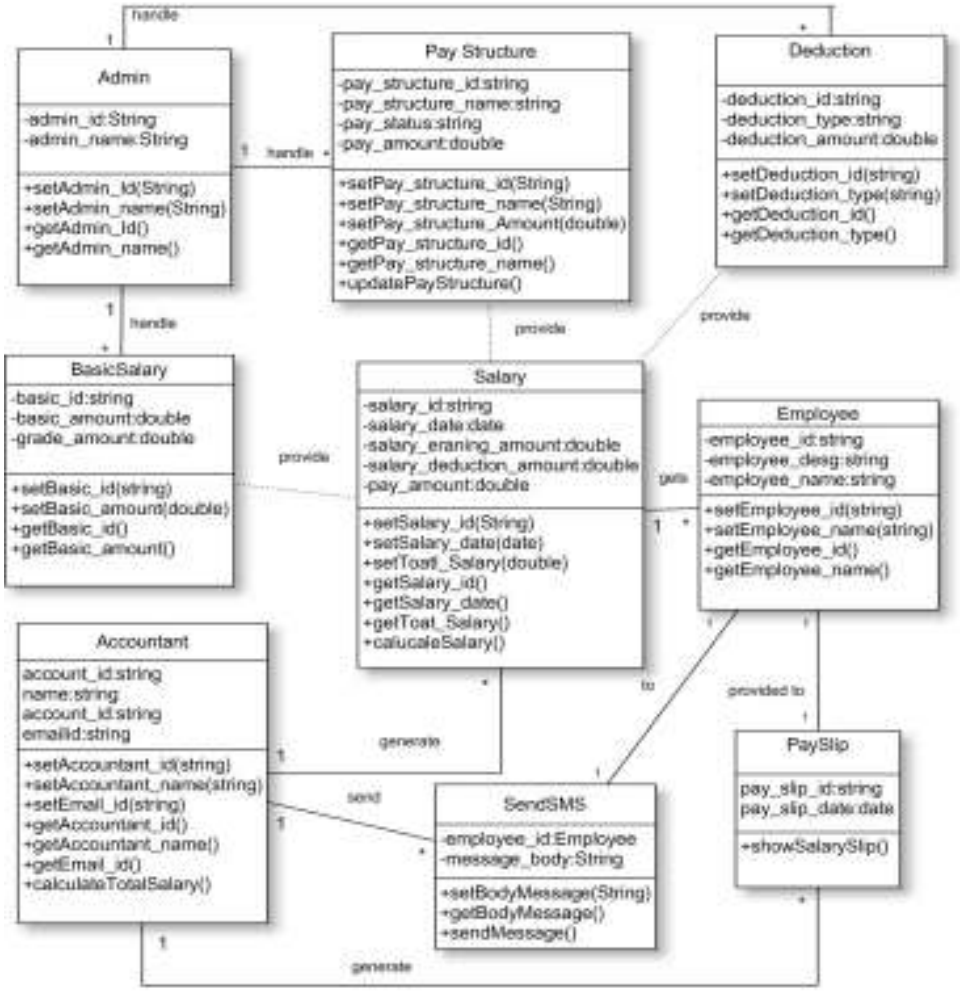


3.2Class Diagram

Class Diagram for Fee

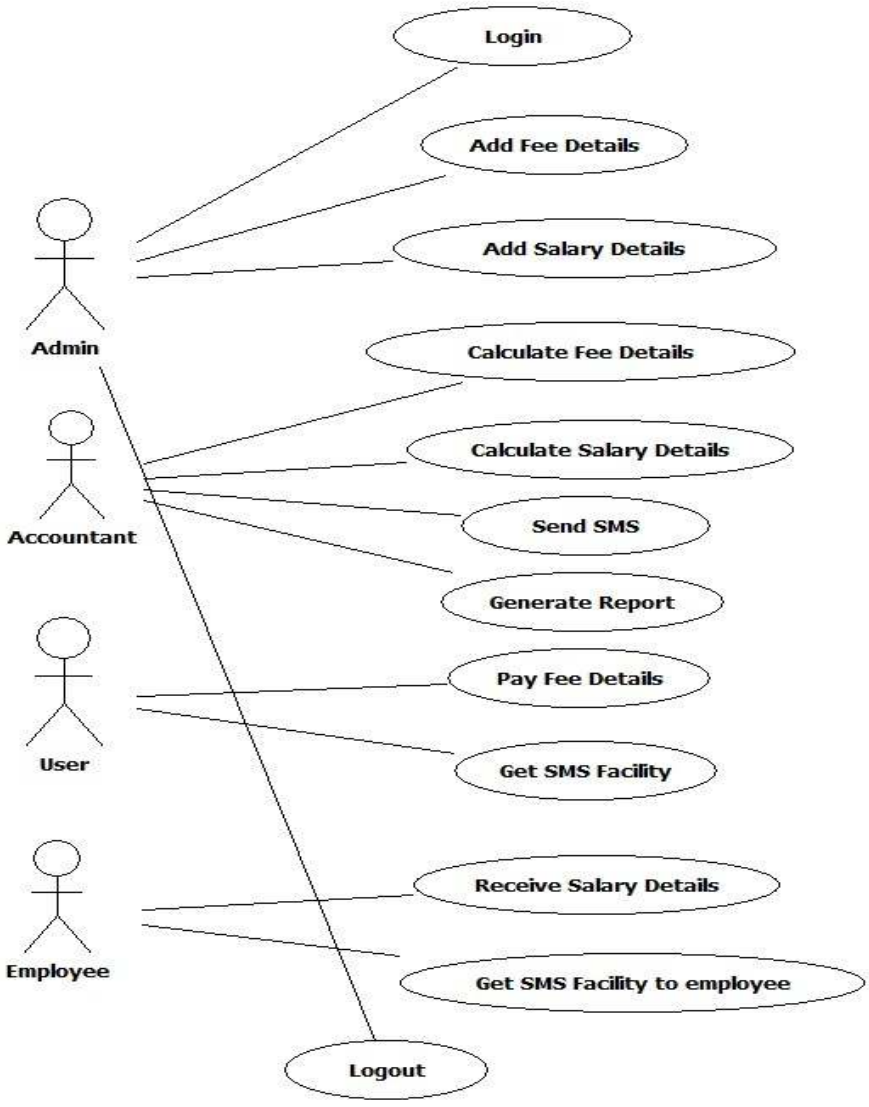


Class Diagram for Payroll

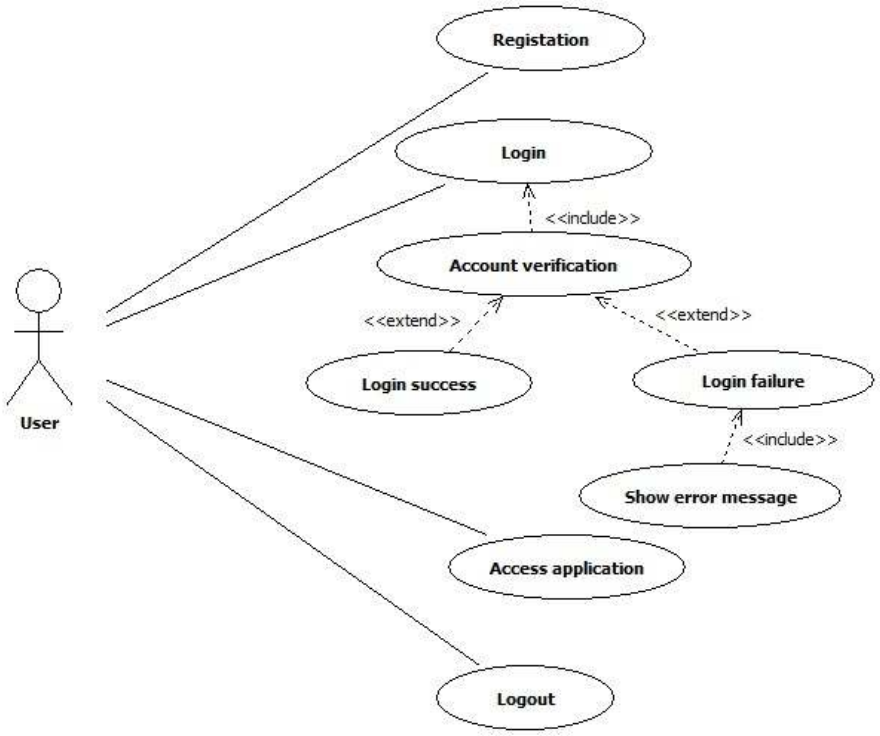


3.3 UseCase Diagram

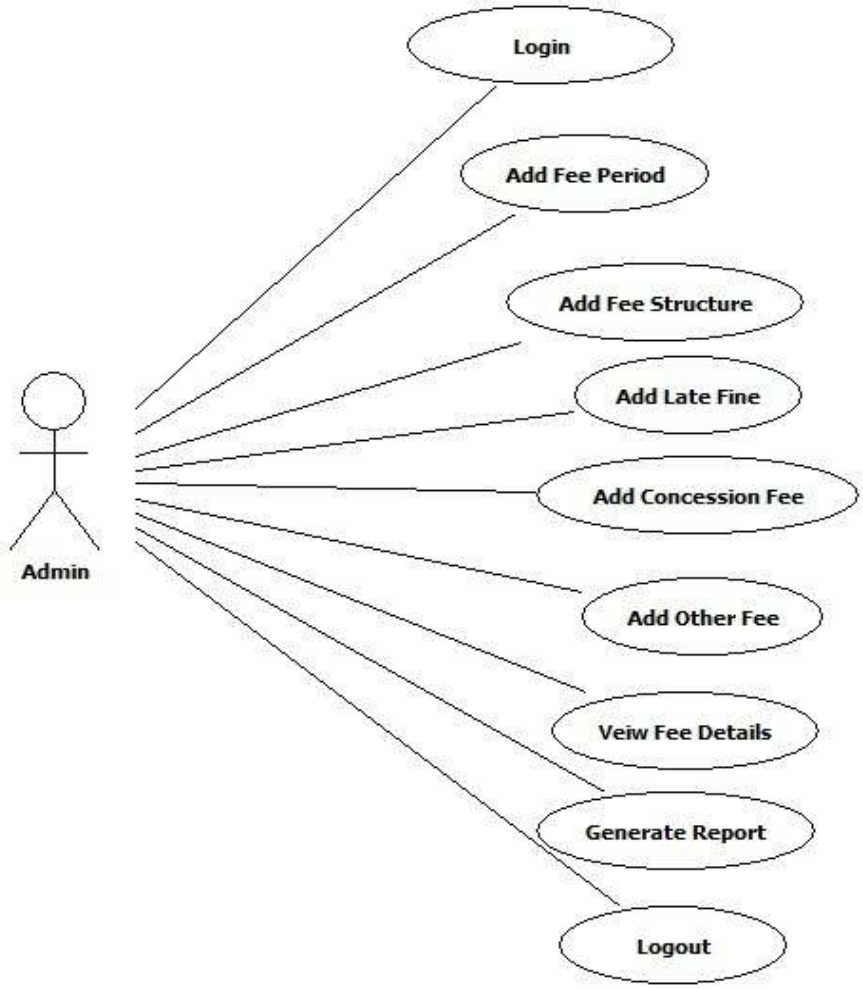
UseCaseDiagram for Account Management System



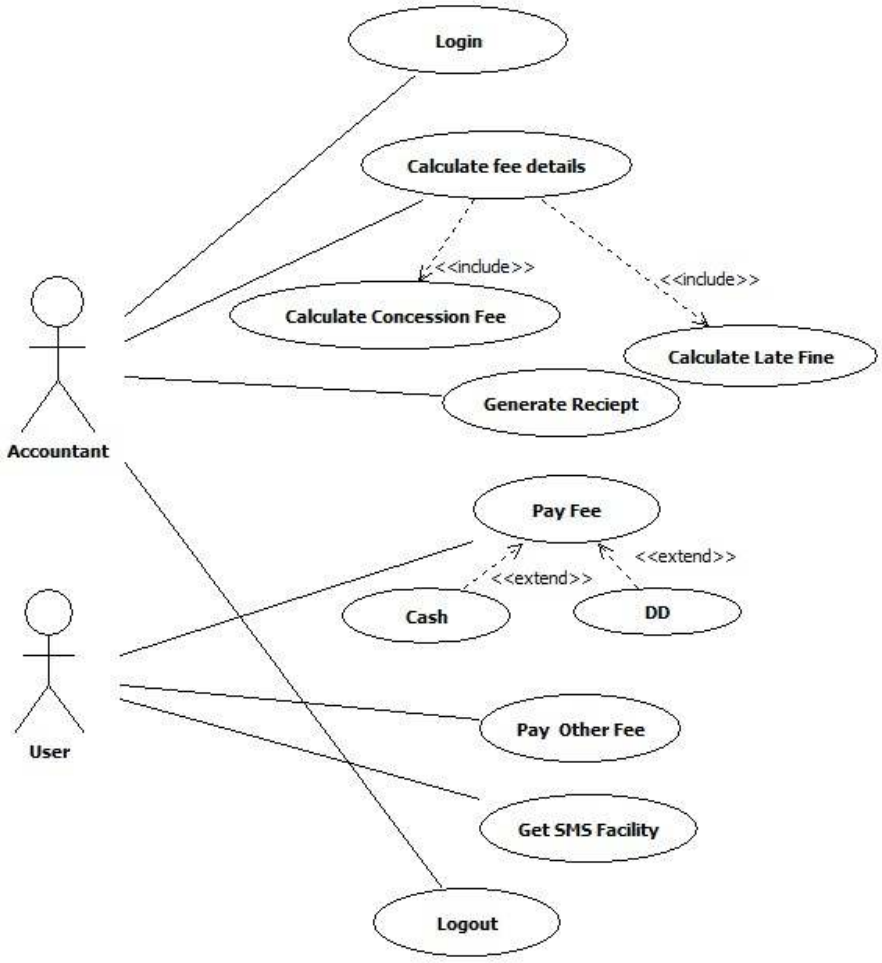
UseCase Diagram for user authentication and authorization process of Account Management System



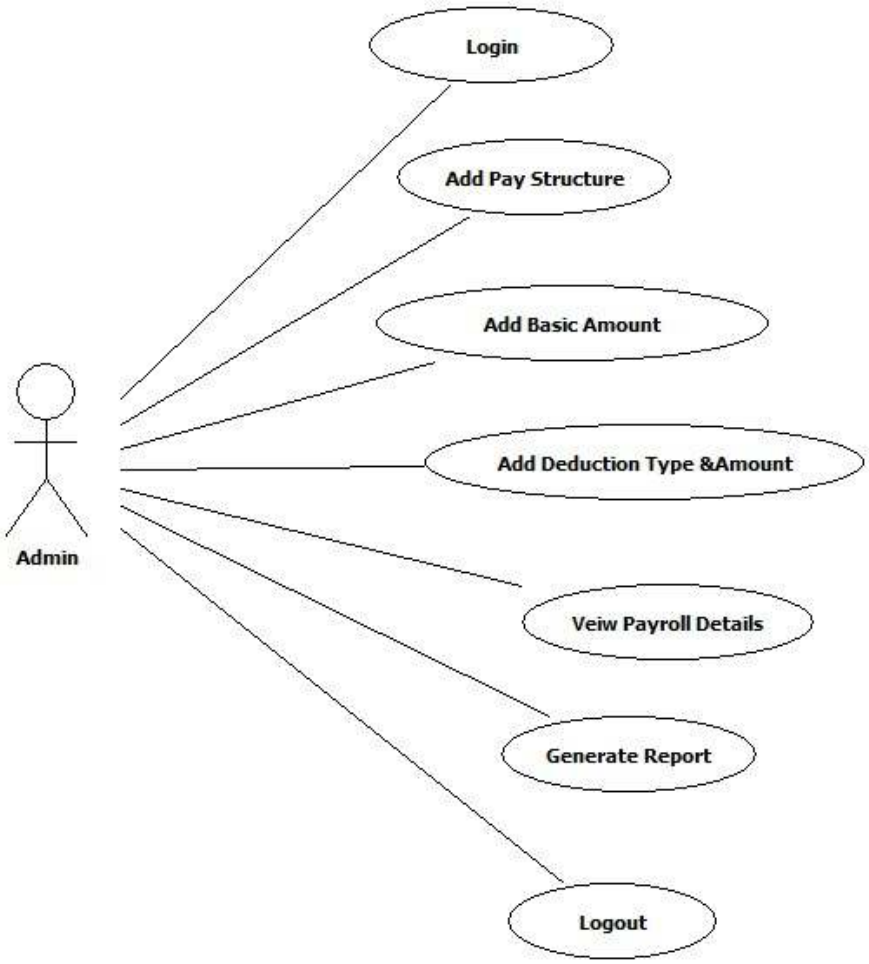
UseCaseDiagram for Fee Admin



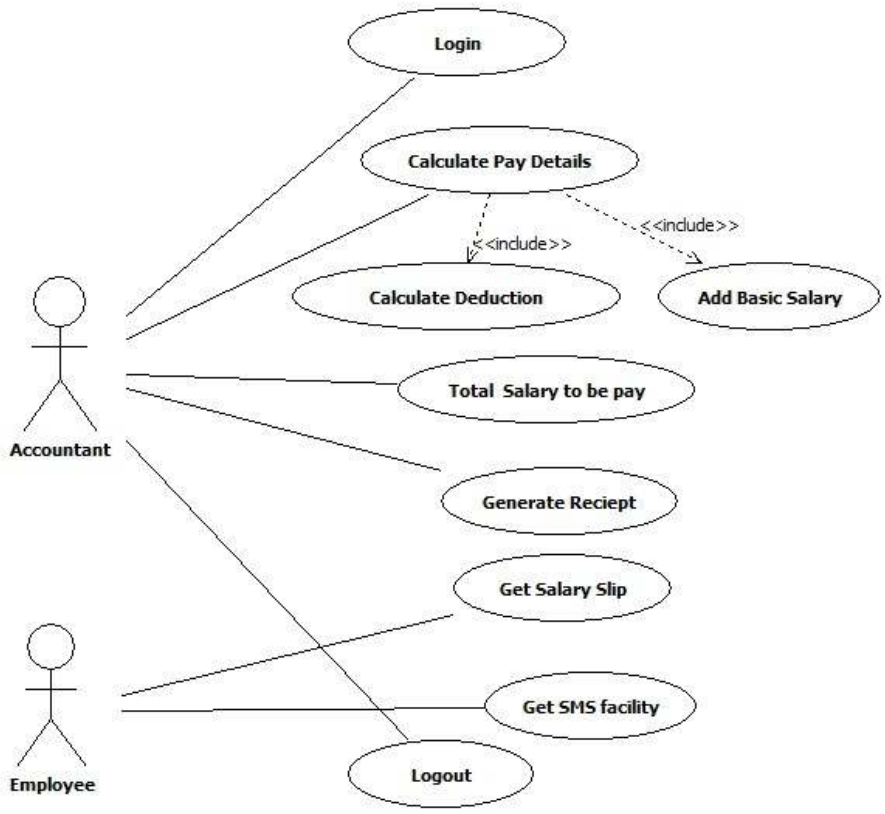
UseCaseDiagram for Fee Accountant



UseCaseDiagramFor Payroll Admin

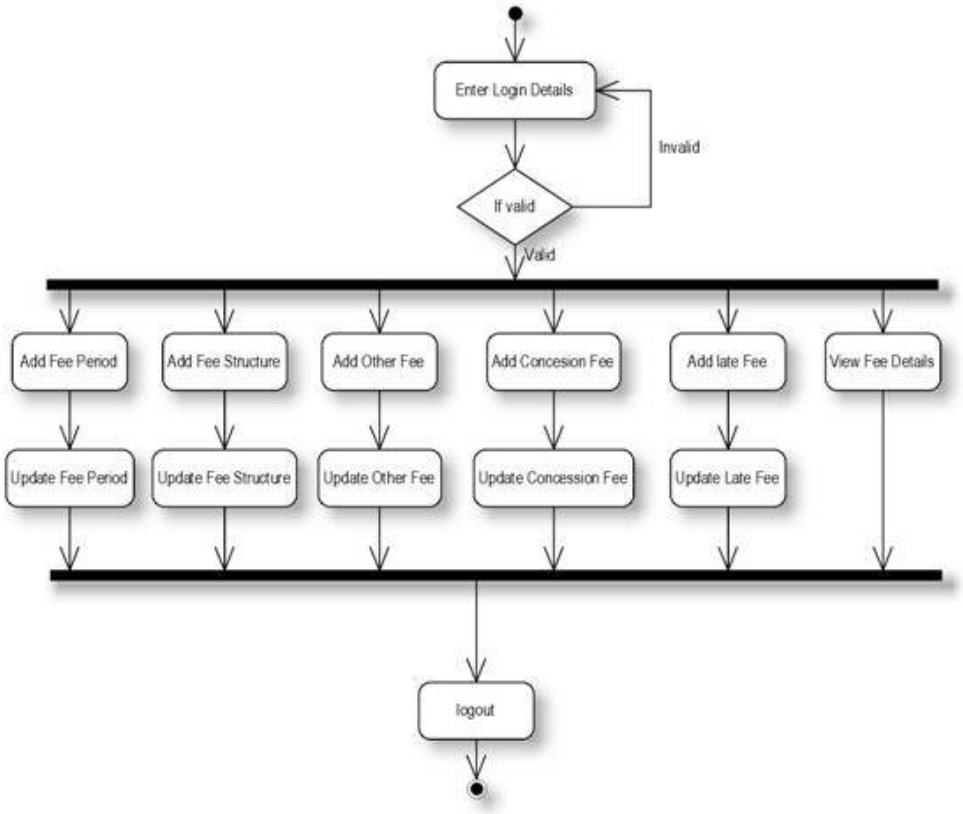


UseCaseDiagramFor Payroll

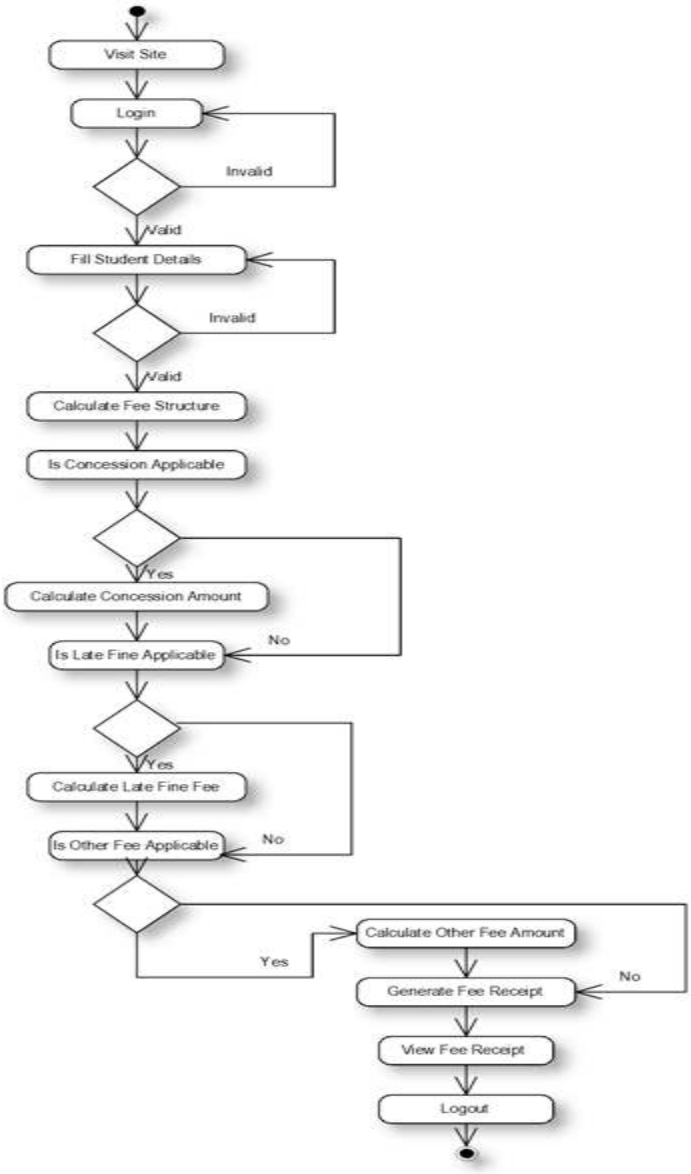


3.4Activity Diagram

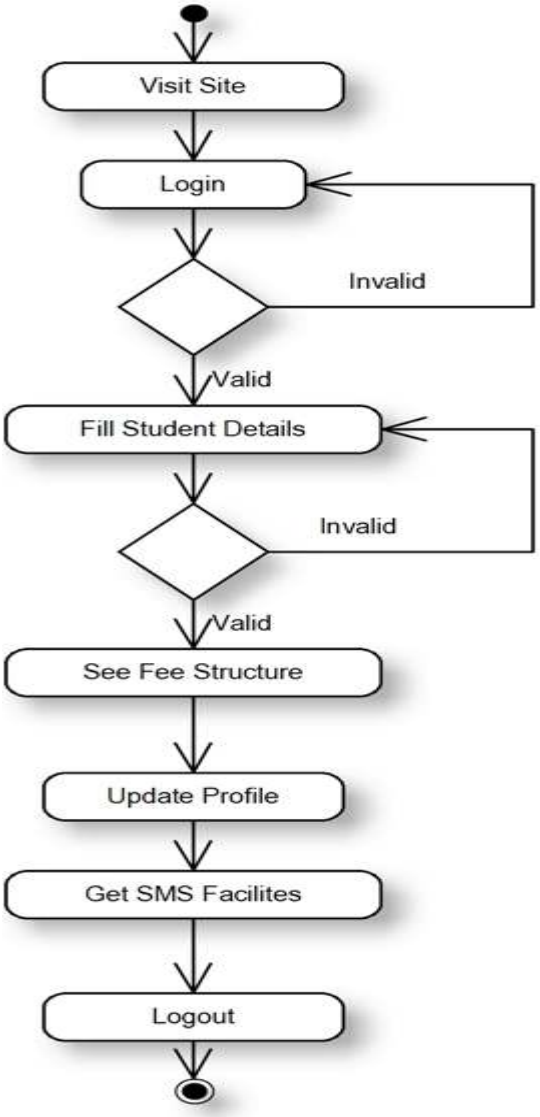
Activity Diagram for Fee Admin



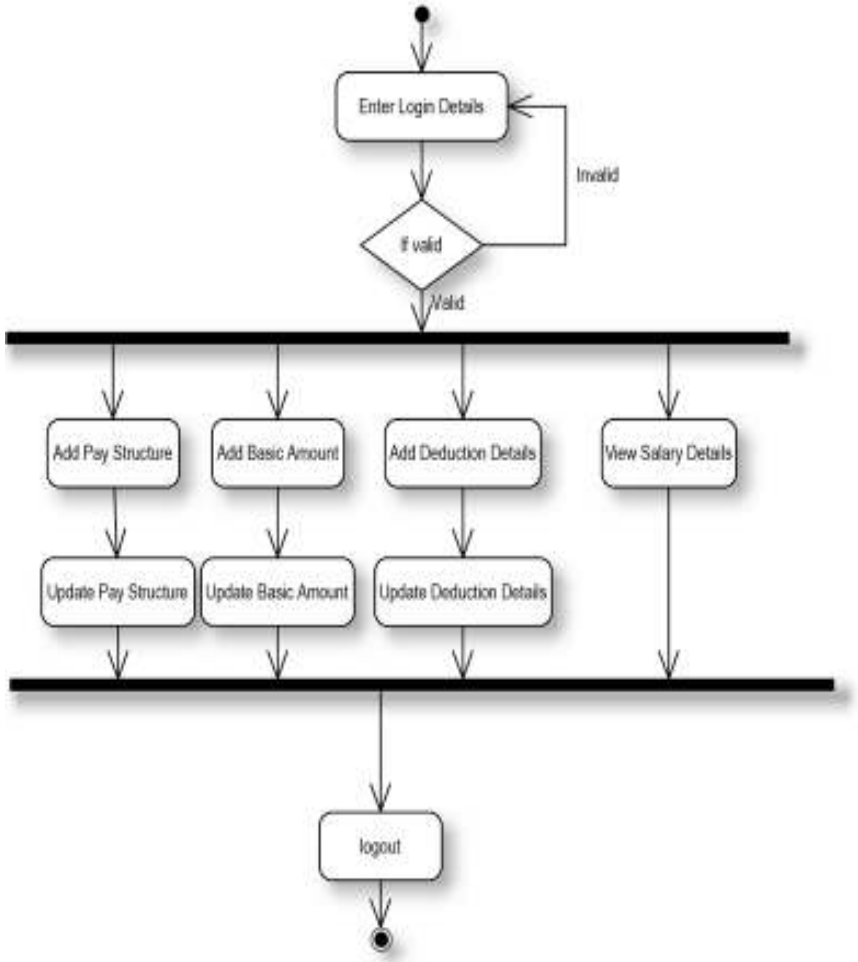
Activity Diagram for Fee Accountant



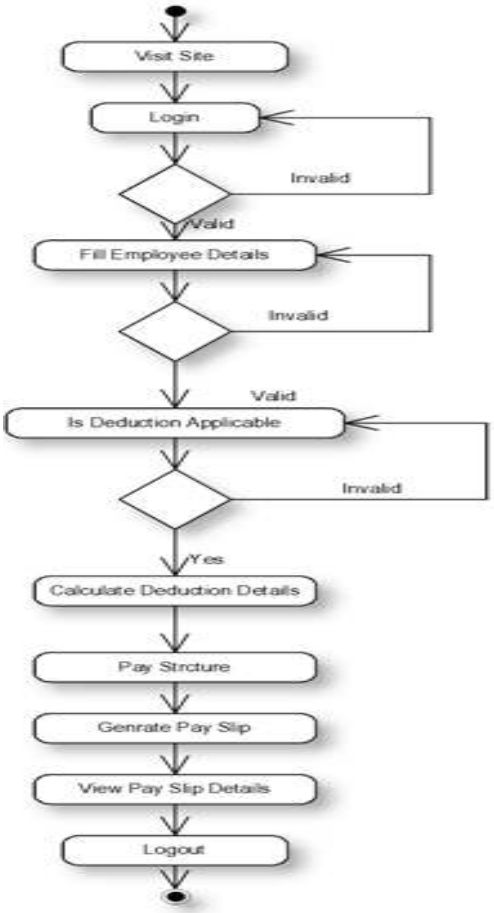
Activity Diagram for User



Activity Diagram Payroll Admin

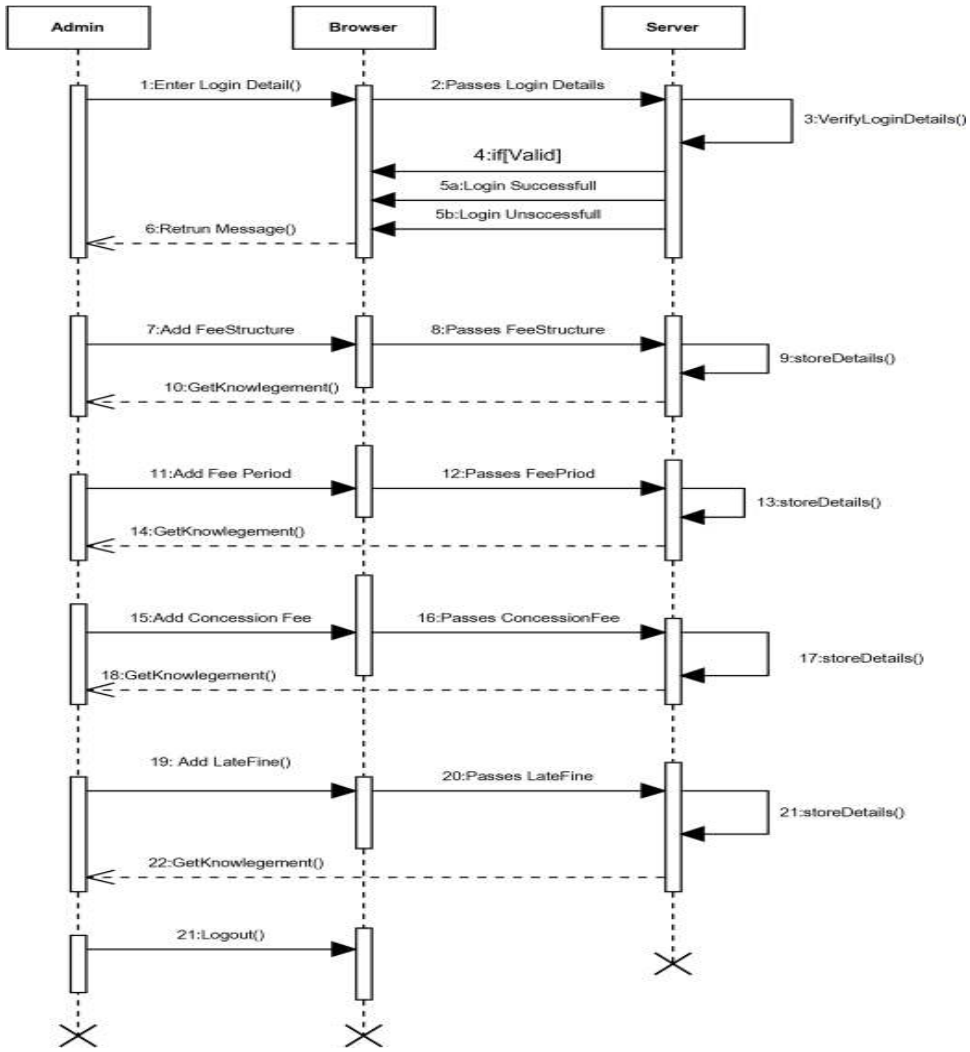


Activity Diagram for Payroll Accountant

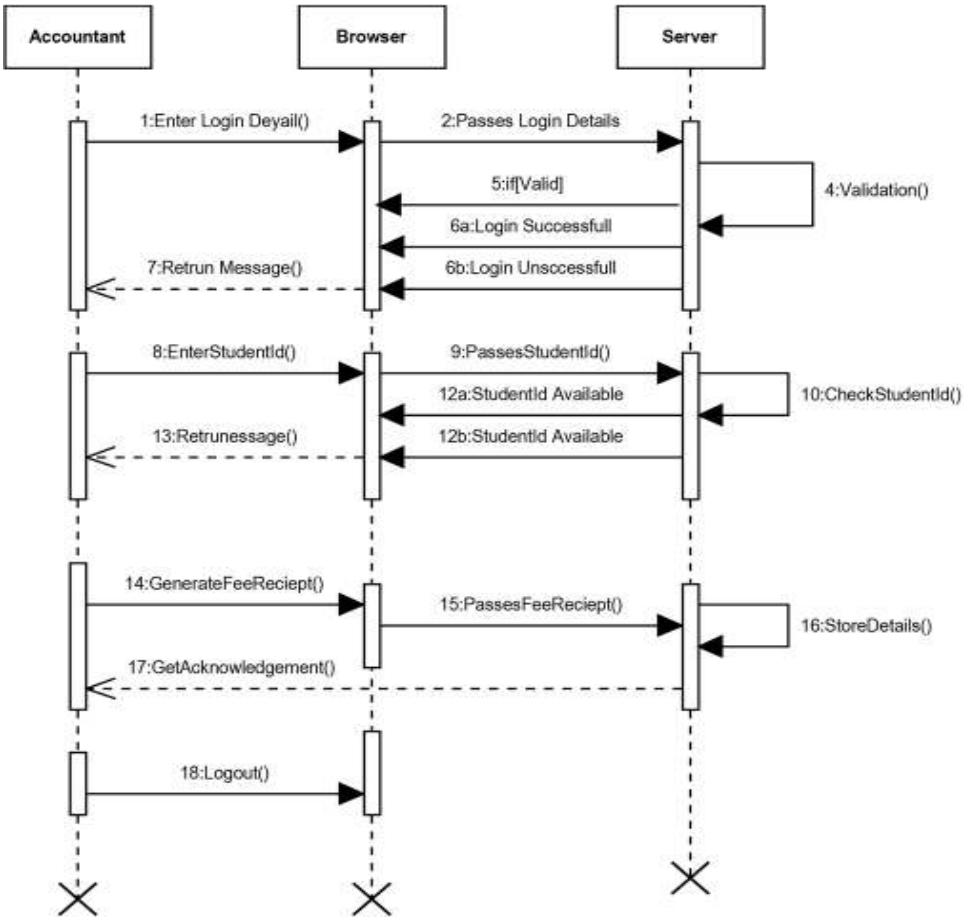


3.5Sequence Diagram

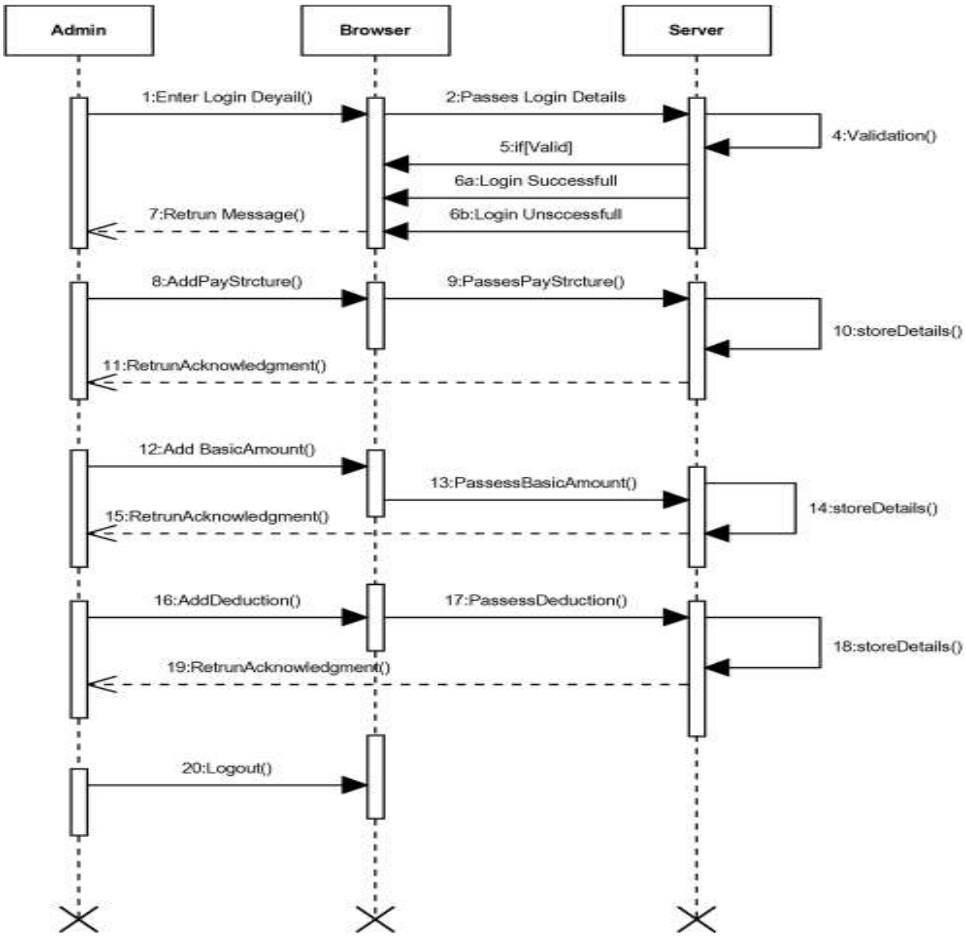
Sequence Diagram for Fee Admin



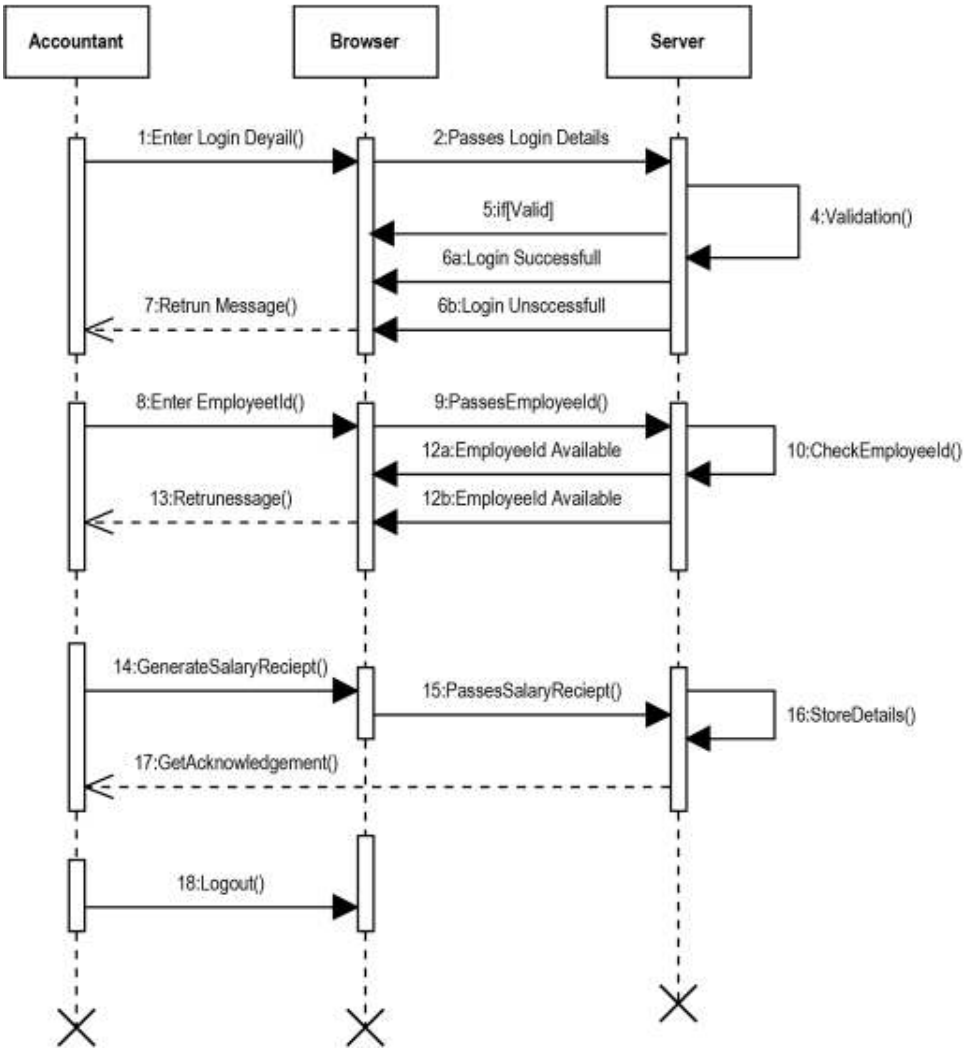
Sequence Diagram for Fee Accountant



Sequence Diagram for Payroll Admin

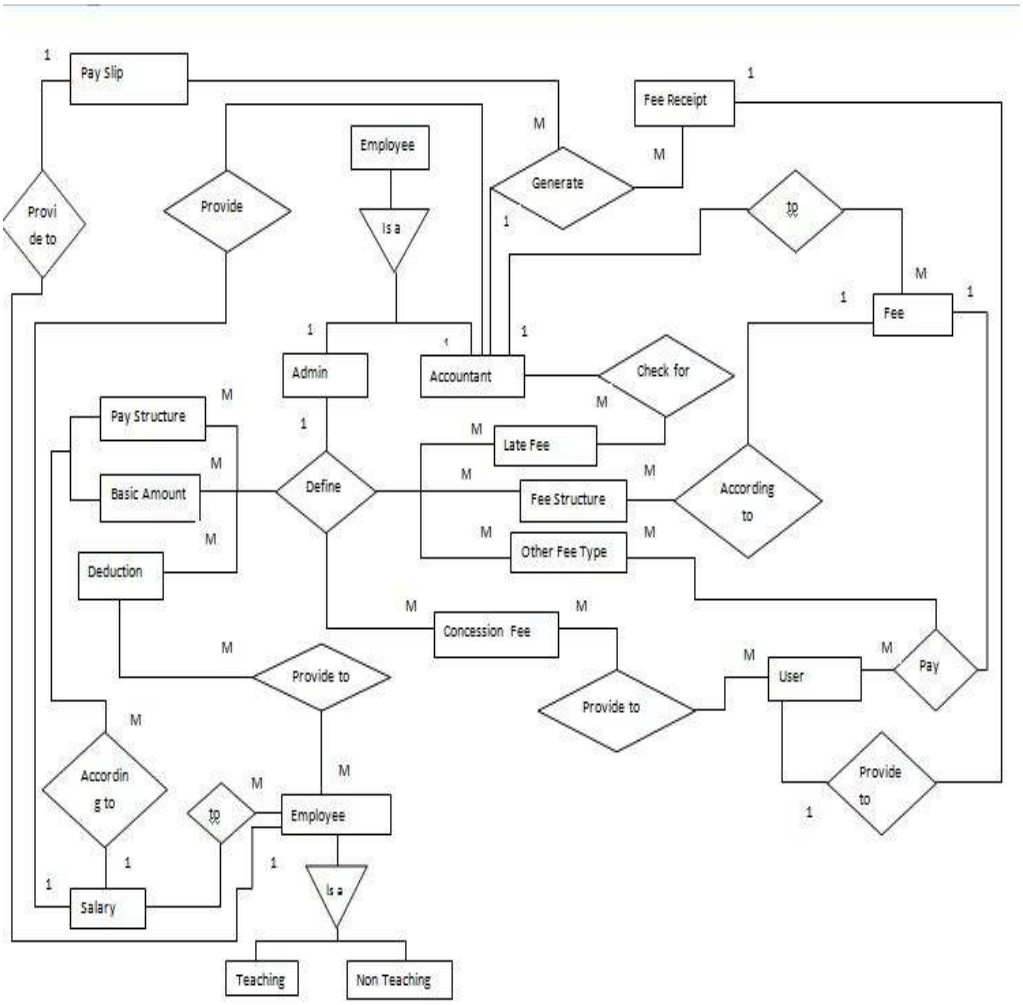


Sequence Diagram for Payroll Accountant

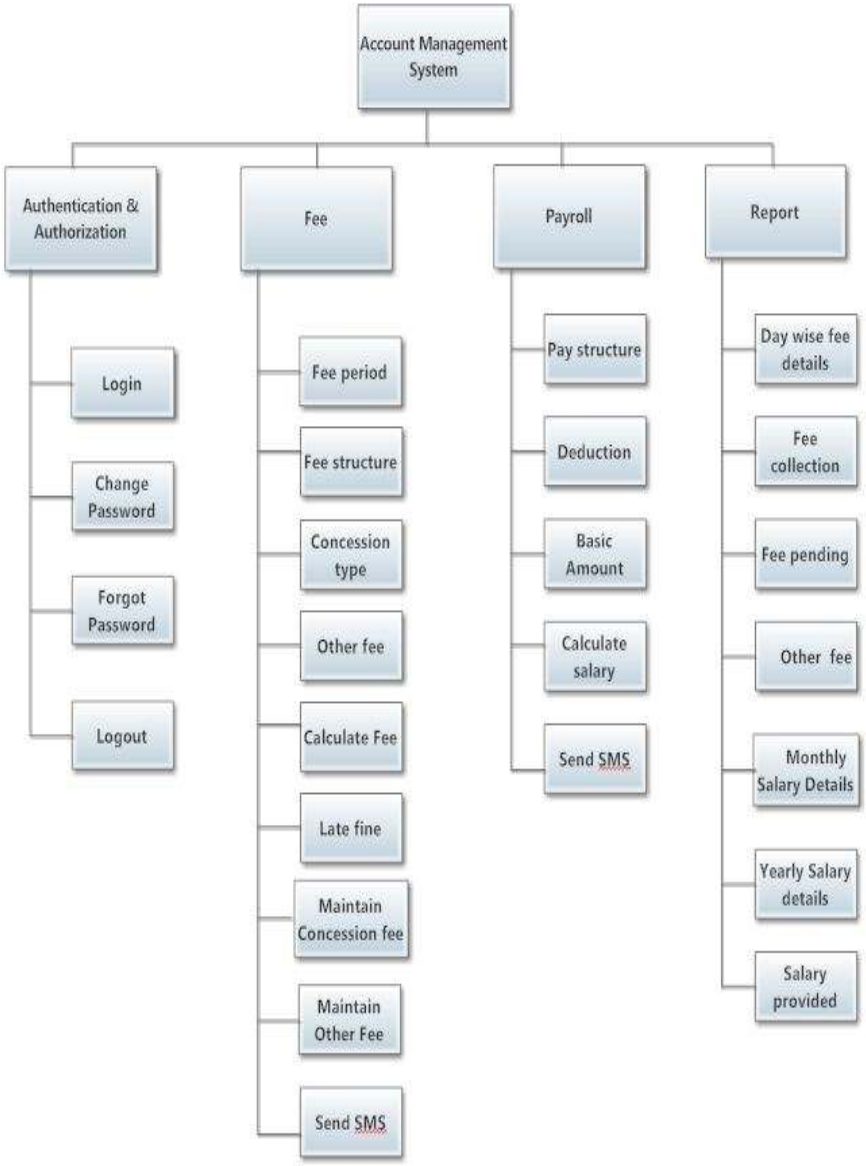


3.6E-R Diagram

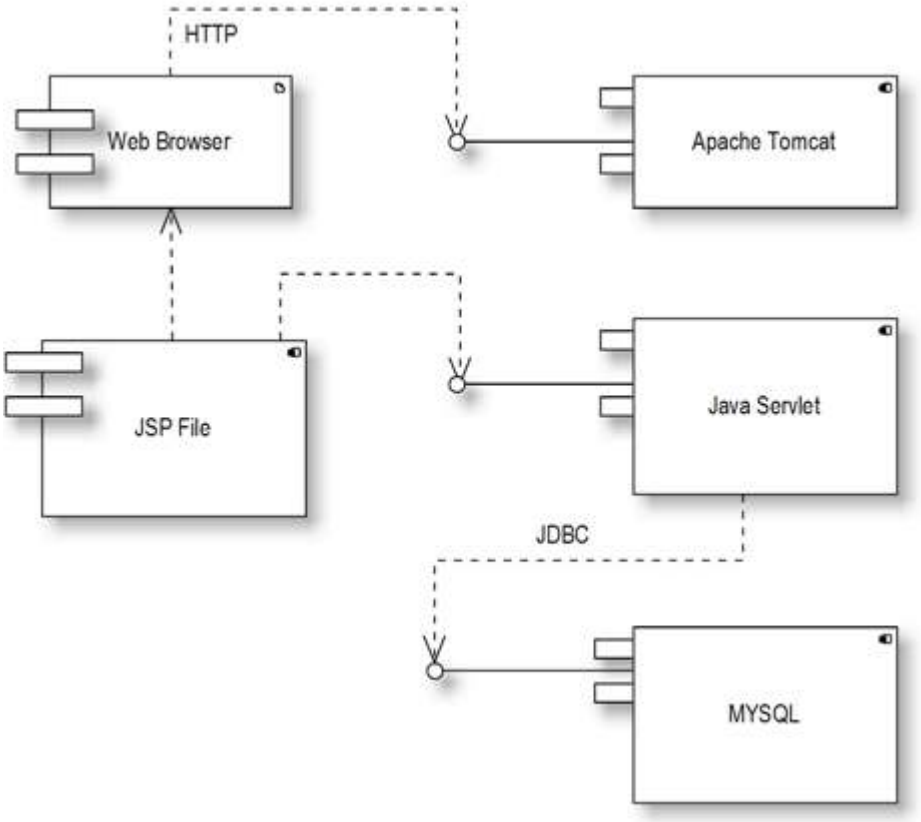
E-R Diagram for Accountant Management System



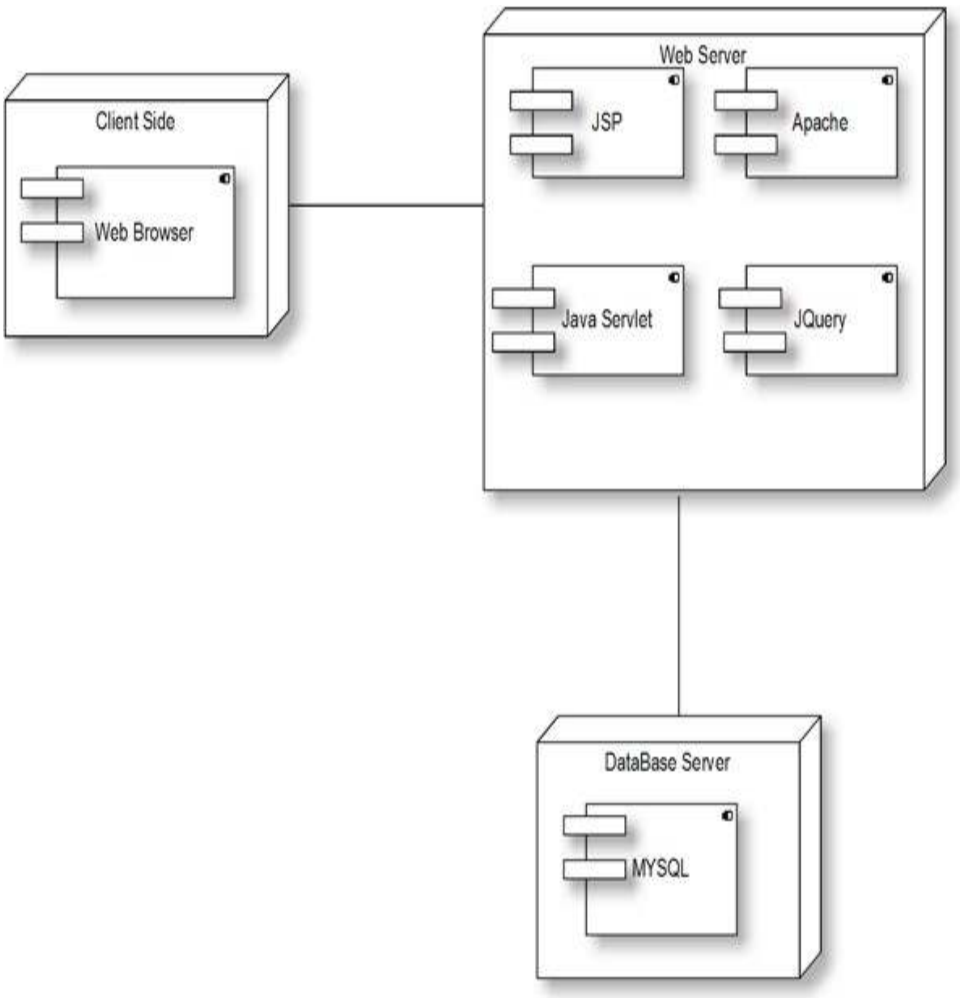
3.7Module Hierarchy



3.8Component Diagram



3.9Deployment Diagram



3.10Module Specification

This project contains following Modules:-

- Authentication and authorization
- Fee Module
- Payroll Module
- Report Module

Authentication and authorization

1. This module deals with Authentication and authorization of the user of

   Account management system.

   Authentication: the user is valid or not.

   Authorization: what type of access privilege the user has.

2. There are 3 types of user in this system

   i) Admin

   ii) Accountant

   iii) User

3. Admin can perform these all functionality

   i) Add Fee Period

   ii) Add Fee Structure

   iii) Add Concession Fee

   iv) Add Other Fee

   v) Add Late Fine

   vi) Add Pay Structure

   vii) Add Deduction Details

   viii) Add Basic Amount

4. Accountant can perform these all functionality

   i) Generate Fee receipt

   ii) Generate Salary Slip

   iii) Maintain Concession Fee

   iv) Maintain Other Fee

   v) Send SMS

5. User can perform these all functionality

   i) View Fee Structure

   ii) Provide SMS Details about Student Fee and last date to be pay

Fee Module

1) Calculate fee details of student according to fee period and according to their class.
2) Calculate late fine also of particular student.
3) Calculate concession details that provided to students.
4) Provide Concession to students.
5) Maintain Other fee details of students.
6) Generate Fee receipt.
7) Send SMS to parent.

Payroll Module

1) Calculate pay structure of employee according to designation .
2) Calculate deduction details of employee provided to employee.
3) Generate Salary Slip.
4) Send SMS to Employee.

Report Module

1) This module deals with all type of report generated by the admin as well as accountant of the school.
2) Admin can generated report on following
    i) Report related to fee details.
    ii) Report related to payroll details.

3) Accountant generated report on following
    i) Day wise fee details

ii) Fee collection

iii) Fee pending

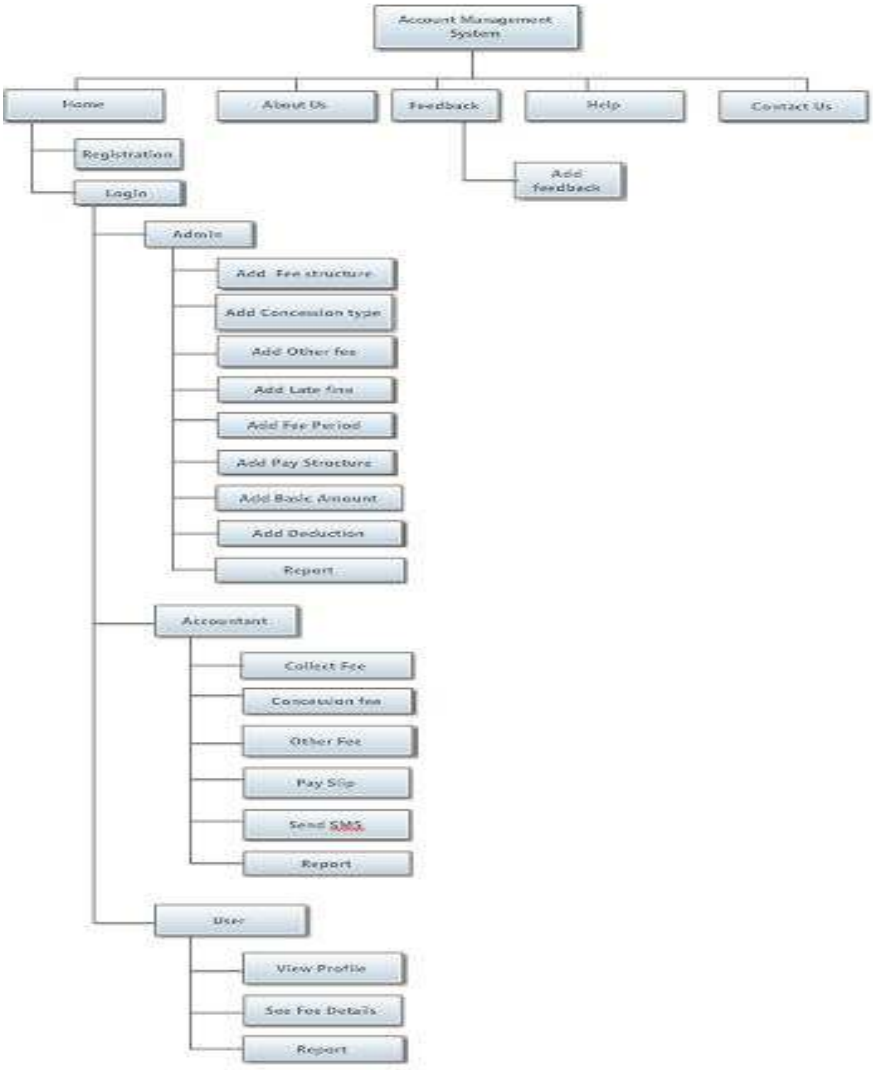
iv) Concession fee

v) Other fee

vi) Monthly and yearly Salary details

vii) Salary provided

3.11Website Map



3.12User Interface Design

Home Page

User Registration

Login Page

Account Management System
For School
HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
ADMIN LOGIN

Admin Page

Add Fee Period

Add Fee Structure

Accountant Page

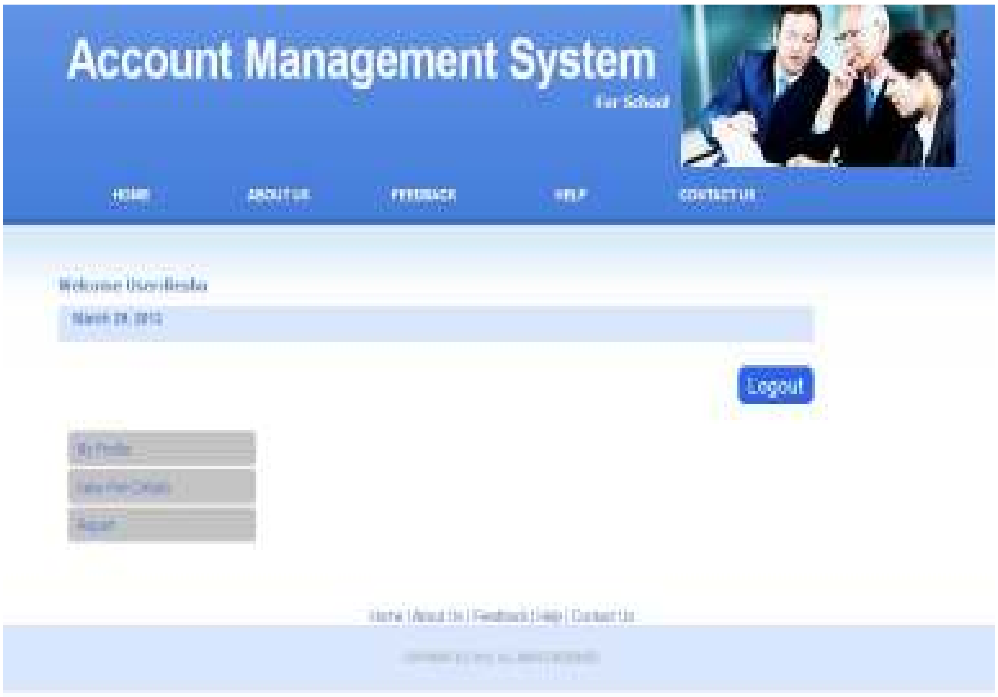
Collect Fee

Account Management System
For School
HOME
ABOUT US
FEEDBACK
HELP
CONTACT US

Pay Slip

Account Management System
For School
HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
Pay Slip
Search
Select
Ok
Logout

User Page



3.13Table Specification

Fee Module:-

Fee_structure_master:-

Field	Data Type	Length	Constraint
Fee_structure_id	Varchar	10	Primary Key
Fee_structure_name	Varchar	30	Not Null
class_id	Varchar	10	Foreign Key of Class_master
Amount	Double		Not Null
fee_session	Varchar	10	Not Null

Fee_period_master :-

Field	Data Type	Length	Constraint
fee_period_Id	Varchar	10	Primary Key
fee_period_type	Varchar	30	Not Null
fee_session	Varchar	10	Not Null

Other_fee_master:-

Field	Data Type	Length	Constraint
Other_fee_Id	Varchar	10	Primary Key
Other_fee_type	Varchar	30	Not Null
Other_fee_amount	Double		Not Null
Other_fee_session	Varchar	10	Not Null

Concession_fee_master:-

Field	Data Type	Length	Constraint
Concessionfee_Id	Varchar	10	Primary Key
Concessionfee _type	Varchar	30	Not Null
Concession_fee_amount	Double		Not Null
Concessionfee_session	Varchar	10	Not Null

Late_fee_master:-

Field	Data Type	Length	Constraint
Late_fee_Id	Varchar	10	Primary Key
Late_fee_date	Date		Not Null
Late_fee_amount	Double		Not Null
Late_fee_session	Varchar	10	Not Null

Other_fee_details:-

Field	Data Type	Length	Constraint
Other_fee_type_id	Varchar	10	Primary Key
Admission_no	Varchar	15	Foreign Key of StudentDetails
Pay_date	Date		Not Null
Other_fee_Id	Varchar	10	Foreign Key of Other_fee_master
Remark	Varchar	30	Not Null

Concession_fee_details:-

Field	Data Type	Length	Constraint
Concession_fee_type_id	Varchar	10	Primary Key
Admission_no	Varchar	10	Foreign Key of Student_details
Concessionfee_Id	Varchar	10	Foreign Key of Concession_fee_master
fee_period_Id	Varchar	10	Foreign Key of Fee_period_master

Collect_fee:-

Field	Data Type	Length	Constraint
Collect_fee_id	Varchar	10	Primary Key
Admission_no	Varchar	10	Foreign Key of Student_details
Pay_date	Date		Not Null
Last_fee_amount	Double		Not Null
Late_fee_id	Varchar	10	Foreign Key of late_fee_master
Fee_period_id	Varchar	10	Foreign Key of fee_period_master
Status	Varchar	8	Default 'Not Paid'
Concession_fee_type_id	Varchar	10	Foreign Key of Concession_fee_details
Pervious_fee	Double		Not Null
Total_fee	Double		Not Null
Final_fee	Double		Not Null

Payment_details:-

Field	Data Type	Length	Constraint
payment_id	Varchar	10	Primary Key
Collect_fee_id	Varchar	10	Foreign Key of Collect_fee
Payment_mode	Varchar	10	Not Null
dd_date	Date		Not Null
dd_no	Varchar	20	Not Null
Bank_name	Varchar	20	Not Null
Branch_name	Varchar	20	Not Null
Amount	Double		Not Null

Student_details:-

Field	Data Type	Length	Constraint
Admission_no	Varchar	10	Primary Key
User_id	Varchar	10	Foreign Key of User
Class_Section__Strength_id	Varchar	10	Foreign Key of ClassSectionStrength_master
Roll_no	Int	5	Not Null
Session	Varchar	10	Not Null

Class_master:-

Field	Data Type	Length	Constraint
Class_id	Varchar	10	Primary Key
Class_name	Varchar	10	Not Null
Session	Varchar	10	Not Null

ClassSectionStrength_master:-

Field	Data Type	Length	Constraint
Class_Section__Strength_id	Varchar	10	Primary Key
Class_id	Varchar	10	Foreign keyofClass_master
Section_name	Varchar	10	Not Null
Strength	Int	10	Not Null
Session	Varchar	10	Not Null

User:-

Field	Data Type	Length	Constraint
User_id	Varchar	10	Primary Key
First_name	Varchar	20	Not Null
Last_name	Varchar	20	Not Null
User_name	Varchar	20	Not Null
Password	Varchar	20	Not Null
Address	Varchar	50	Not Null
City	Varchar	20	Not Null
State	Varchar	20	Not Null
Country	Varchar	20	Not Null
Pincode	Int	6	Not Null
Email_id	Varchar	20	Not Null
Mobile_number	Int	10	Not Null
Security_question	Varchar	20	Not Null
Security_answer	Varchar	20	Not Null
User_status	Varchar	5	Default'False'

Payroll:-

Employee_details:-

Field	Data Type	Length	Constraint
Employee_id	Varchar	10	Primary Key
user_name	Varchar	30	Not Null
Password	Varchar	10	Not Null
Mobile_no	Int	10	Not Null
Name	Varchar	30	Not Null
Address	Varchar	30	Not Null
City	Varchar	20	Not Null
State	Varchar	20	Not Null
Country	Varchar	20	Not Null
Pin_code	Int	6	Not Null
Qualification	Varchar	10	Not Null
Marital_status	Varchar	10	Not Null
Account_number_pf	Varchar	20	Not Null
PAN_no	Varchar	20	Not Null
Date_of_joining	Date		Not Null
Gender	Varchar	10	Not Null
Date_of_birth	Date		Not Null
Department_id	Int	10	Foreign Key of department_master
Employee_type	Varchar	15	Not Null
Designation	Varchar	30	Not Null
Employee_status	Varchar	6	Default'False'
Grade	Varchar	10	Not Null

Department_master:-

Field	Data Type	Length	Constraint
Department_id	varchar	10	Primary Key
Department_name	varchar	30	Not Null

Pay_Structure_master:-

Field	Data Type	Length	Constraint
Pay_Structure _id	Int	10	Primary Key
Pay_Structure _type	Varchar	30	Not Null
Amount	Double		Not Null
Pay_Status	Varchar	6	Default 'False'

Basic_amount:-

Field	Data Type	Length	Constraint
Basic_id	Varchar	10	Primary Key
Employee_type	Varchar	10	Not Null
Grade	Varchar	10	Not Null
Basic_amount	Double		Not Null
Pay_Grade	Double		Not Null

Deduction_master:-

Field	Data Type	Length	Constraint
Deduction _id	Varchar	10	Primary Key
Deduction_type	Varchar	10	Not Null
Deduction_amount	Double		Not Null
Deduction _Status	Varchar	6	Default 'False'

Pay_slip:-

Field	Data Type	Length	Constraint
Salary _id	Varchar	10	Primary Key
Employee_id	Varchar	10	Foreign Key of Employee_details
Pay_date	Date		Not Null
Month	Varchar	10	Not Null
Pay_Structure_Amount _id	Varchar	10	Foreign Key of Pay_Structure_Amount_master
Basic_id	Varchar	10	Foreign Key of Basic_amount
Gross_pay	Double		Not Null
Total_deduction	Double		Not Null
Net_pay	Double		Not Null

3.14Test Procedures and Implementation

What is software testing?

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. It is a process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specifications and in the way the end user expects. This can be done by various software testing techniques which provide a systematic guidance for designing tests that exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

The following software testing techniques were used in order to uncover errors in the system:

- Unit testing
- Integration testing

- White box testing
- Black box testing
- Acceptance tests (Alpha and Beta testing)

1: Unit Testing

Unit testing is normally considered as an adjunct to the coding step. It is the test for the small units of code, e.g. programs, modules or procedures, in order to ensure that they perform their intended functions. All possible paths through the control structure are exercised to ensure that all statements in a program are executed at least once. Unit testing is also done to test the data flow across a module interface.

The following errors are uncovered during unit testing:

- Comparison of different data types.
- Incorrect logical operators or precedence.
- Incorrect comparison of variables.
- Improper or nonexistent loop termination.

- Improperly modified loop variable.

2: Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. During this activity, unit tested components are taken and a program structure is built as per the design. Then incremental integration is performed on the system. This means that programs are constructed and tested in small increments instead of testing the entire program as a whole. This is done because correction of errors becomes difficult in case of whole program testing as many errors were detected and it is not easy to correct them at one go. Thus, through incremental integration testing, any error uncovered could be easily noted and corrected and interfaces are tested completely.

3: White Box Testing

White box testing is also called as glass box testing. It is related with the structure (internal logic) of the program. It helps in uncovering

many errors that black box testing cannot. During white box testing activity, every statement of programs is executed at least once. All independent paths are also executed. Every logical decision is executed to check both true and false conditions. All loops are executed at their boundaries and within their operational bounds. Validation checks are also done during this process.

4: Black Box Testing

Black box testing, also known as behavioral testing, focuses on the functional requirements of the software. It is related to input and output only and not related with the internal structure of the program. This testing is also done so as to find errors such as:

- Initialization and termination errors
- Behavior and performance errors
- Incorrect or missing functions
- Interface errors
- Errors in data structures and external database access
- Performance errors

5: Acceptance Testing (Alpha & Beta Testing)

An acceptance test is a test carried out by the customer or end user rather than the developer in order to enable the customer to validate all requirements. Alpha testing and beta testing are two types of acceptance tests that are conducted.

6: Alpha Testing

Alpha test is conducted in a controlled environment. As a matter of fact, the end user conducts alpha test at the developer's site. During the course of the system development, the end user is operating the software in front of the developer and the errors and other problems are recorded. Rectification is made accordingly.

7: Beta Testing

Beta testing is also conducted by the end user, but in the absence of the developer. Here, the end user himself records all the problems that he encounters during testing the system and then reports them to

the developer at regular intervals. As a result of problems reported during beta testing, modifications are made to overcome the problem

Test Cases

Test Case ID #		1		
Test Case Name		To test functionality of login form.		
Prerequisite		Login form should get loaded.		
Objective		To fine out bugs in login form.		
Sr. No	**Steps be to executed**	**Expected Result**	**Actual Result**	**Pass/Fail Criteria**
	Username Textbox Test cases			
1	1.Enter Username With less than 6 Characters. 2. Enter correct password. 3. Click on Submit button.	It should display error message "Enter username with 6 characters."	It display error message"Enter username with 6 characters."	Pass
2.	1.Enter Username With 6 Characters. 2. Enter correct password. 3. Click on Submit button.	It should display homepage.	It display homepage.	Pass

3.	1. Enter Username with greater than 30 Characters. 2. Enter correct password 3. Click on Submit button.	It should display error message"Enter username with maximum 30 characters."	It does not accept username with greater than 30 characters.	Pass
4.	1.Enter Username with 10 characters. 2. Enter correct password 3. Click on Submit button.	It should display homepage.	It displays homepage.	Pass
5.	1.Enter Username As blank field. 2. Enter correct password 3. Click on Submit button.	It should display error message "Enter username".	It displays error message "Enter username".	Pass

	Password Textbox Test cases			
1.	1. Enter correct username. 2.Enter password with less than 6 Characters. 3. Click on Submit button.	It should display error message "Enter password with minimum 6 characters."	It displays error message "Enter password with minimum 6 characters."	Pass
2.	1. Enter correct username. 2.Enter password with greater than 10 Characters. 3. Click on Submit button.	It should display error message "Enter password with maximum 10 characters."	It should display error message "Enter password with maximum 10 characters."	Pass
3.	1. Enter correct username. 2.Enter password with 10 Characters. 3. Click on Submit button.	It should display Homepage.	It displays homepage.	Pass

4.	1. Enter correct username. 2.Enter password as blank field. 3. Click on Submit button.	It should display Error message" Enter password".	It displays Error message" Enter password".	Pass
5.	1. Enter correct username. 2.Enter password with characters , digits and special Characters. 3. Click on Submit button.	It should display error message "Enter password with alphabets and/or digits only.	It displays error message "Enter password with alphabets and/or digits only.	Pass

Test Case ID#			2	
Test Case Name			To test functionality of user registration form.	
Prerequisite			User Registration form should get loaded.	
Objective			To find out bugs in User Registration form.	
Sr. No	**Steps to executed**	**Expected Result**	**Actual Result**	**Pass/Fail**
	First name Textbox Test cases			
1.	1. Enter first name as blank field. 2. Click on Submit button.	It should display error message "Enter first name".	It displays error message "Enter first name".	Pass
2.	1. Enter first name with digits. 2. Click on Submit button.	It should display error message "Enter alphabets only".	It displays error message "Enter alphabets only".	Pass
3.	1. Enter first name with greater than 50 characters. 2. Click on Submit button.	It should display error message "Enter first name with maximum 50 characters."	It accepts first name with greater than 50 characters.	Pass

4.	1.Enter first name with digits And characters. 2. Click on Submit button.	It should display error message "Enter alphabets only."	It displays error message "Enter alphabets only."	Pass
5.	1.Enter first name as Characters. 2. Click on Submit button.	It should display home page.	It displays home page.	Pass
	Last name Textbox Test cases			
1.	1. Enter last name as blank field. 2. Click on Submit button.	It should display error message "Enter last name".	It displays error message "Enter last name".	Pass
2.	1. Enter Last name with digits. 2. Click on Submit	It should display error message "Enter alphabets	It displays error message "Enter alphabets	Pass

	button.	only".	only".	
3.	1. Enter Last name with greater than 50 characters. 2. Click on Submit button.	It should display error message "Enter last name with maximum 50 characters."	It accepts last name with greater than 50 characters.	Pass
4.	1.Enter last name with digits And characters. 2. Click on Submit button.	It should display error message "Enter alphabets only."	It displays error message "Enter alphabets only."	Pass
5.	1.Enter last name as Characters. 2. Click on Submit button.	It should display home page.	It displays home page.	Pass
	Mobile Number Textbox			

	Test cases			
1.	1.Enter Mobile Number as blank field. 2. Click on Submit button.	It should display error message "Enter Mobile Number."	It displays error message "Enter Mobile Number."	Pass
2.	1. Enter Mobile Number with less than 10 digits. 2. Click on Submit button.	It should display error message "Enter Mobile Number with 10 digits."	It displays error message "Enter Mobile Number with 10 digits."	Pass
3.	1. Enter Mobile Number with 10 digits. 2. Click on Submit button.	It should display home page.	It displays home page.	Pass
4.	1. Enter Mobile Number with alphabets. 2. Click on Submit button.	It should display error message "Enter Mobile Number with digits only."	It displays error message "Enter Mobile Number with digits only."	Pass

CHAPTER 4:

USER MANUAL

4.1 User Manual

User Manual provides user with the ease of operation the system. If intended user of the system feels any problem in operation the system, the can refer to this manual to clear their doubts. This user manual guides the users of the system and provides them the briefing of the system of the system to how to use the system.

Browsing Website

Starts TOMCAT Server and open an instance of IE and type URL(Uniform Recourse Locator)

http://localhost:8080/AccountManagementSystem/on the browser and press enter. You will see the home page on the URL type

Home Page

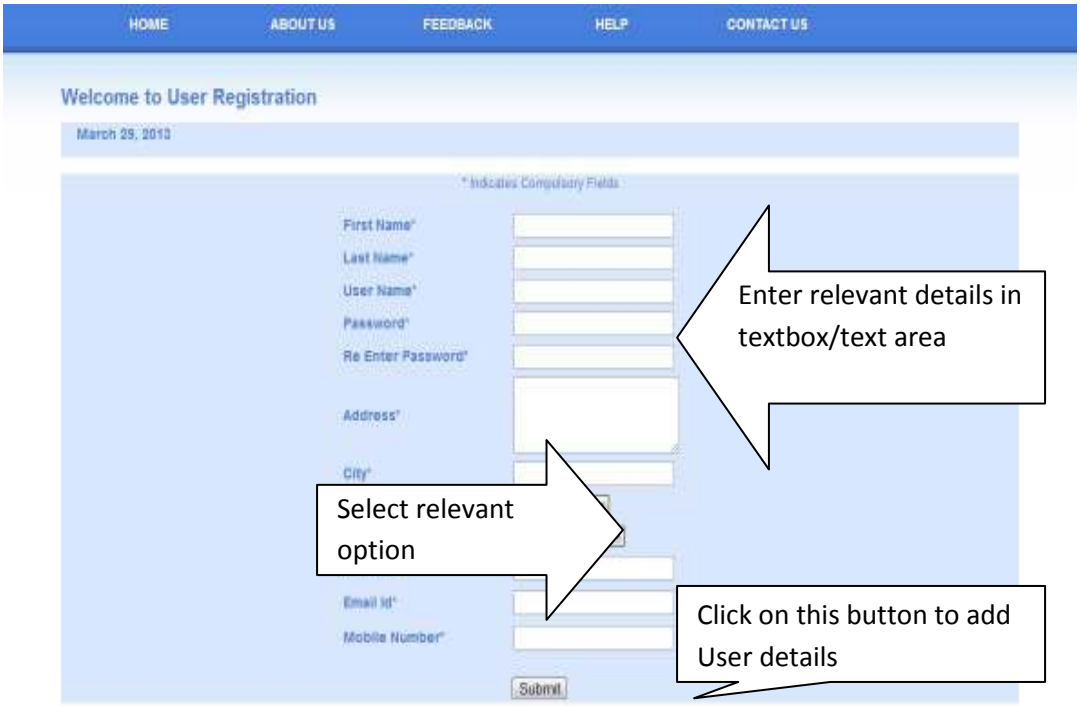
This page contains two buttons

1) Login
2) Registration

Click on registration button to register new parent /guardian.

After clicking on registration,registration page is displayed.

Registration Page for Parent



- Click on Login button to login into the system.

Login Page

Enter E-mail ID and Password to login into the system.

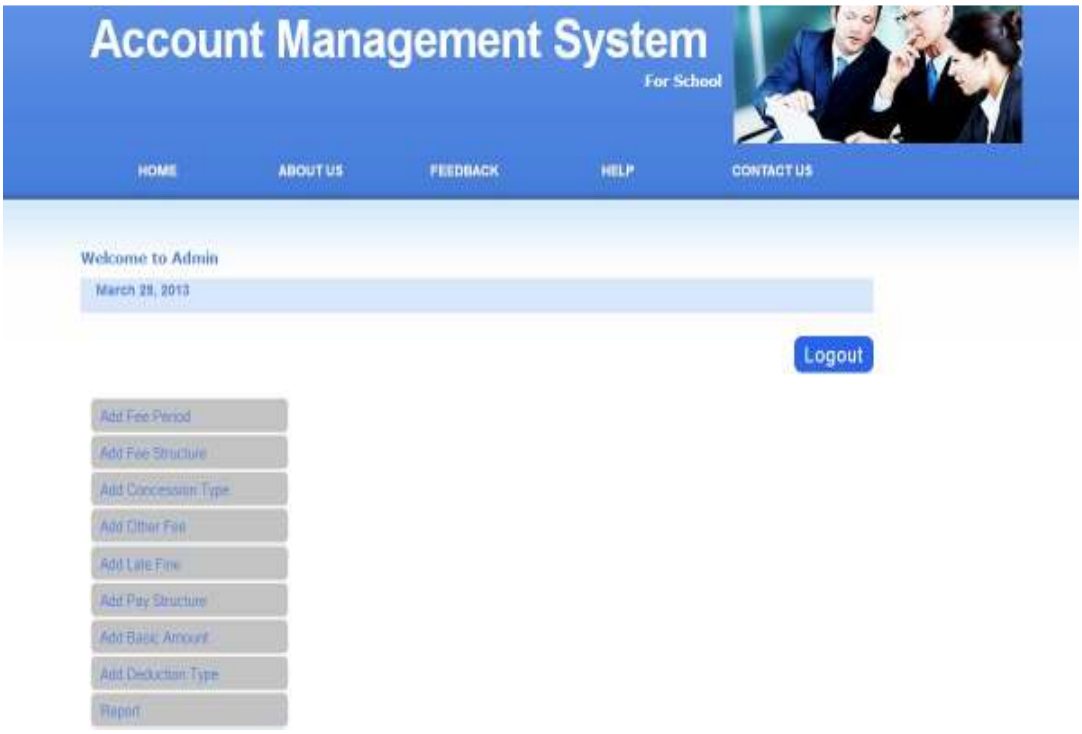
After filling the valid E-mail ID and Password ,based on the type of user home page is displayed.

The user can use the system as per their requirement following their permission.

Change Password

For Admin

Home Page



This page contains all the menus and links that are granted to that user. The menu and linkShown on home page populate dynamically according to user category while the page

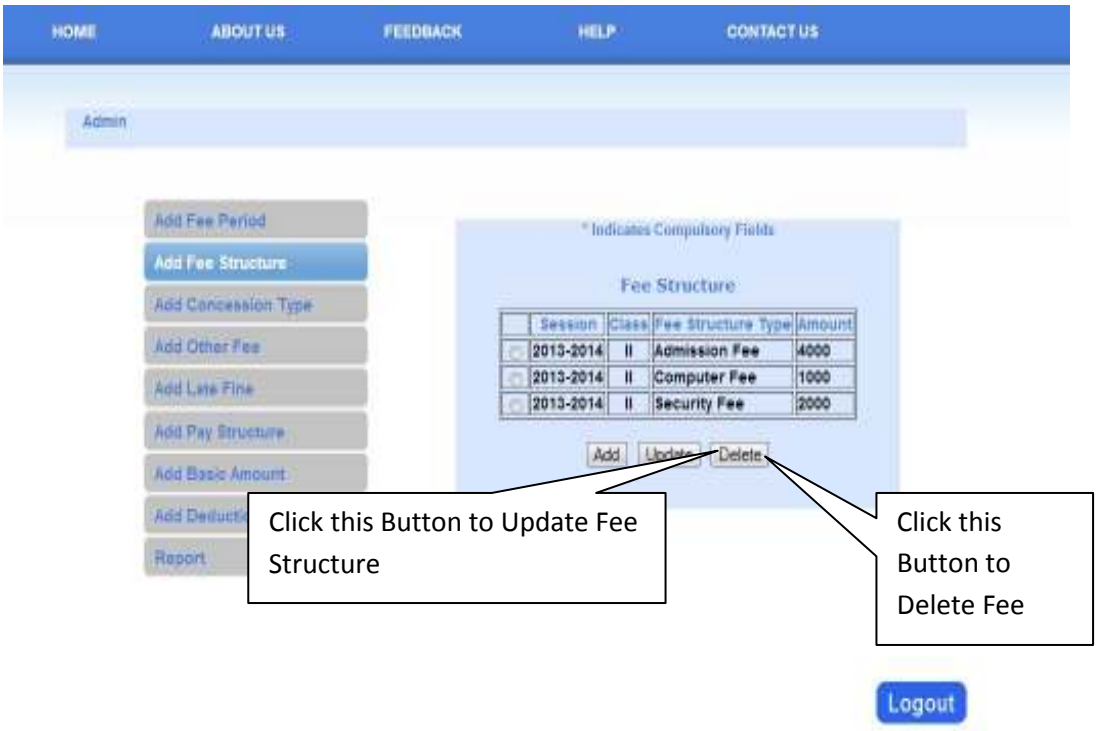
load. A user can select any of the links and menus and starts using the system.

Add Fee Structure

To add new fee structure click on fee structure

View Fee Structure

To View/Update/Delete fee structure click on View Fee structure.



For Accountant

Home Page

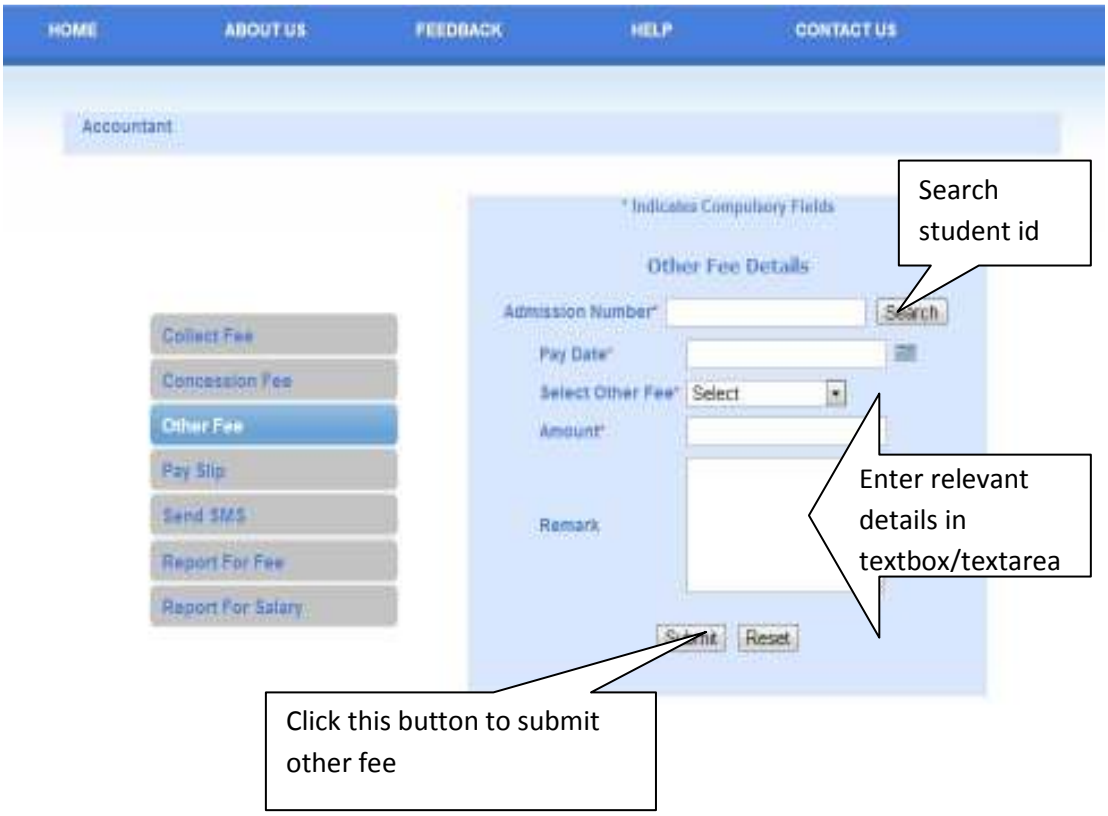
Collect Fee

HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
Accountant
Collect Fee
Concession Fee
Other Fee
Pay Slip
Send SMS
Report For Fee
Report For Salary
* Indicates Compulsory Fields
Collect Fee
Admission Number*
Pay Date*
Fee Period*
New Admission
Click here to search student id
Select fee period
Logout

Pay Slip

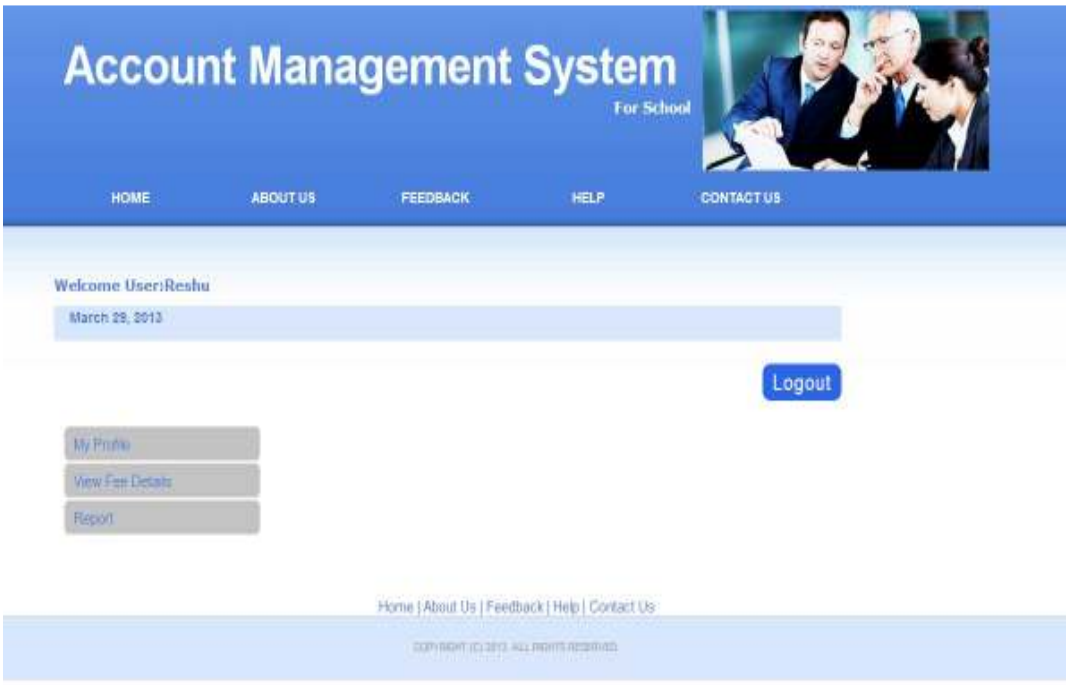
HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
Accountant
Collect Fee
Concession Fee
Other Fee
Pay Slip
Send SMS
Report For Fee
Report For Salary
* Indicates Compulsory Fields
Pay Slip
Employee Id*
Search
Pay Date*
Ok
Click here to search employee id
Select month
Logout

Other Fee Details



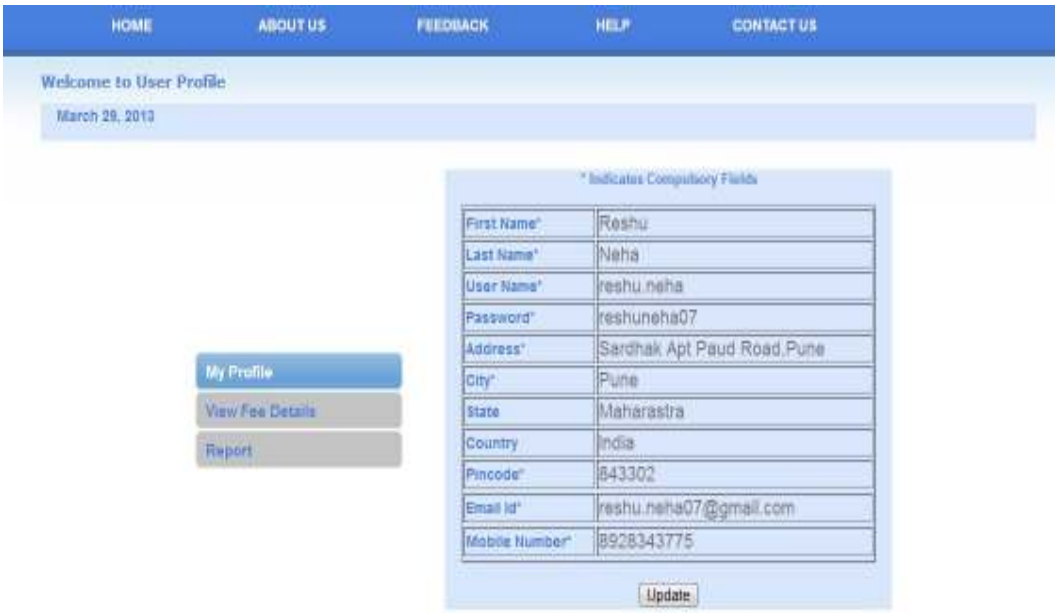
For User

Home Page



User Profile

To display user profile and update their profile



View Fee Structure

To view the fee structure of their child

4.2 Operation Manual/Menu Explanation

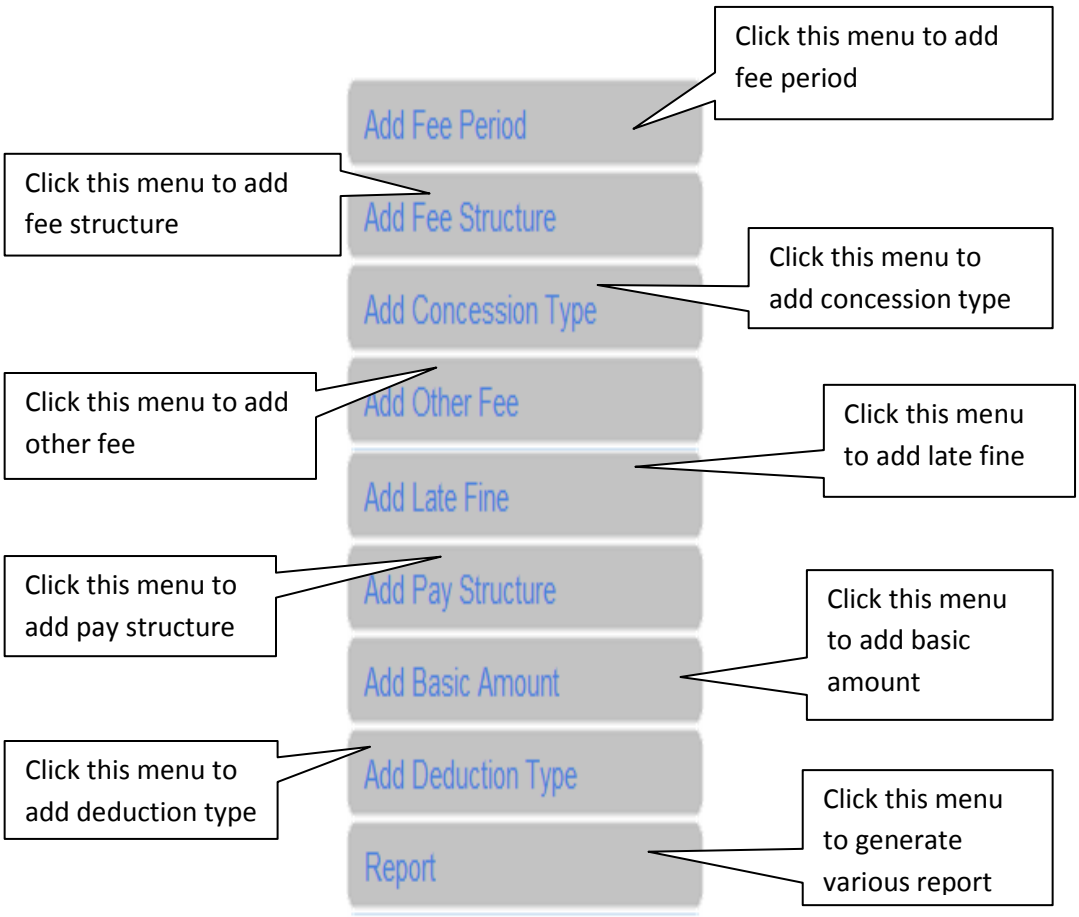
Menu screen provide user with ease of access to various part of the application.

Menu explanation describes the various menus used in the system. Different menus

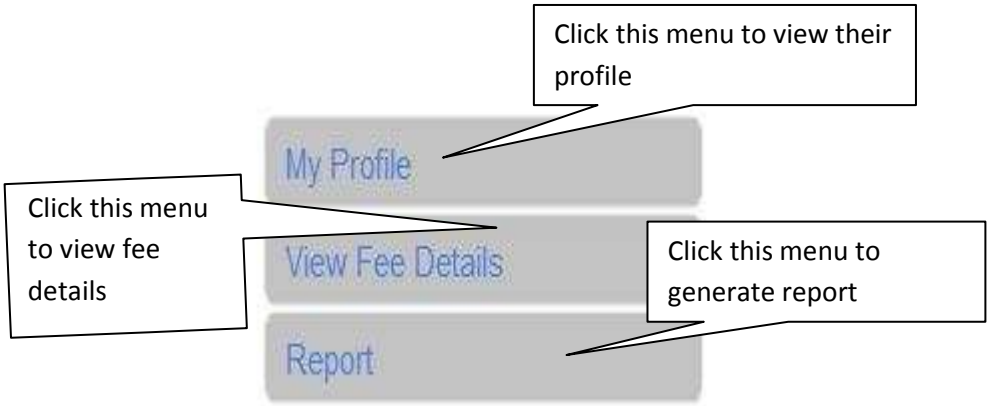
are as follows.

Base on type of user login in to the system these are the menu for different of user.

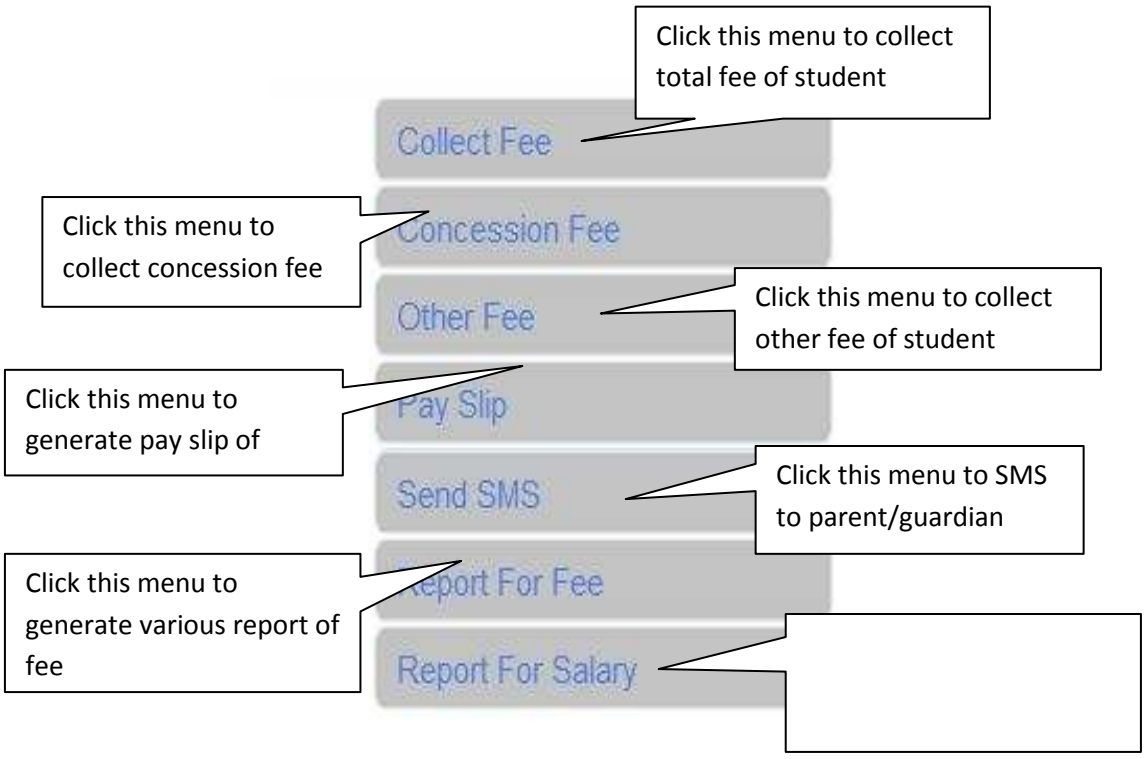
Menu for Admin



Menu for User



Menu for Accountant



4.3 Program Specification/Flow Chart

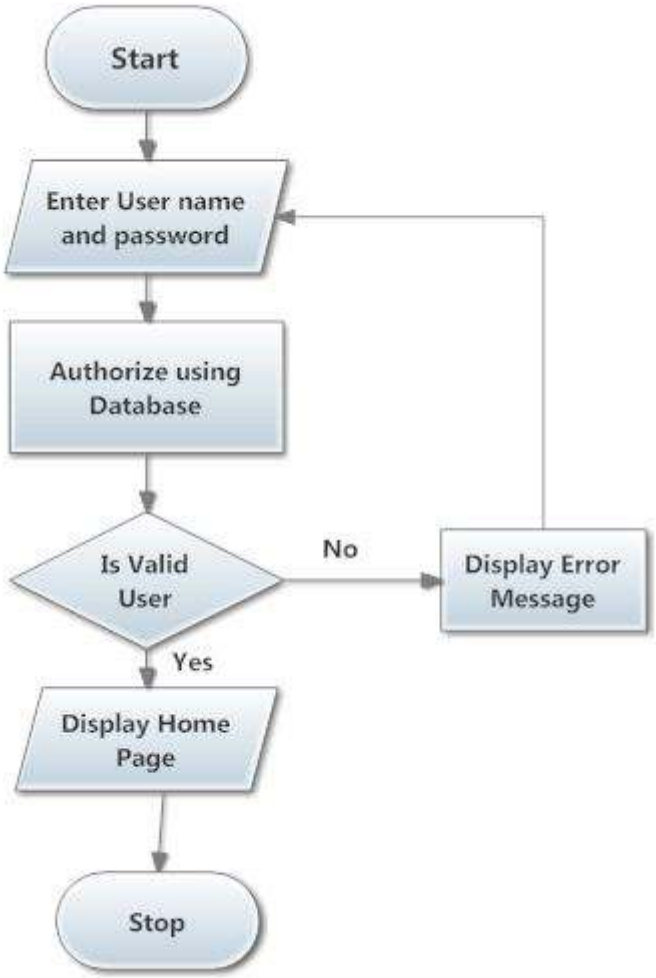
The system is developed in JAVA and use JSP as frontend and MYSQL as backend.

Following are the program specification used in the development process explained with the help of flowcharts.

User Authentication (Login)

Module Name		Authentication and Authorization
Program Name		Login to the System
Purpose		Check Authentication of User
Event		Click on "Login"Button
Input	**Constraint**	**Description**
Login Details (Emails and Password)	The required fields must not be null and input data should be Valid.	Login details gets checked against Database
Change Password	Must enter correct password and also new password twice for verification.	User can change the password his or her current password to new password.
Output	The Login details gets checked against database To check the authentication of user & user gets Notification message of successfully login	

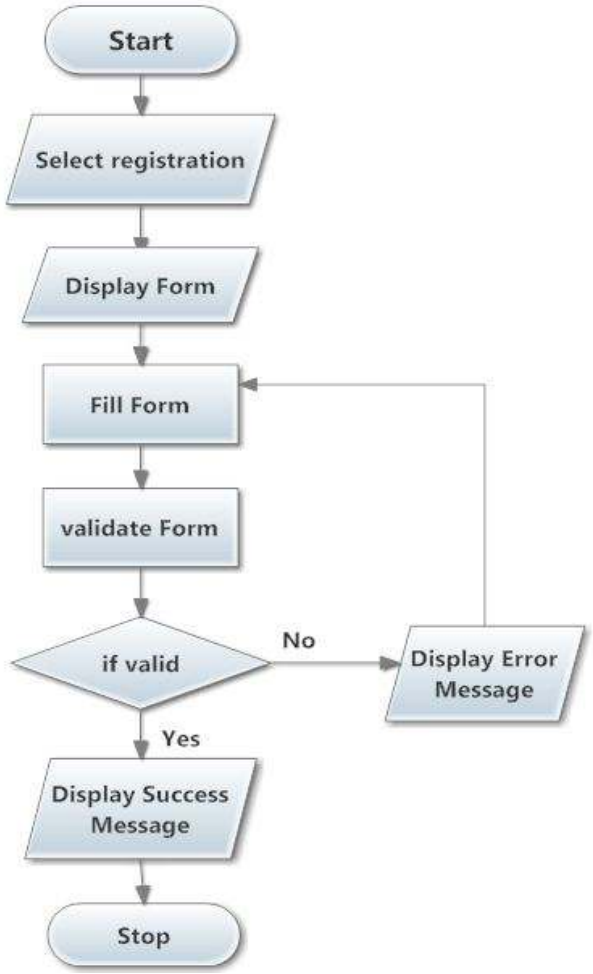
Flow Chart


Start
Enter User name and password
Authorize using Database
Is Valid User
No
Display Error Message
Yes
Display Home Page
Stop

Parent /Guardian Registration

Module Name		Fee Module
Program Name		Add Parent/Guardian Details
Purpose		Add Parent/Guardian Details in database
Event		Click on "Submit"Button
Input	**Constraint**	**Description**
Parent Details	The required fields must not be null and input data should be Valid.	Parent details gets stored into Database
Output	The Login details gets checked against database To check the authentication of user & user gets Notification message of successfully login	

Flow Chart


Start
Select registration
Display Form
Fill Form
validate Form
if valid
No
Display Error Message
Yes
Display Success Message
Stop

DRAWBACKS

&

LIMITATIONS

Drawbacks & Limitations

1) This system is developed for specific account department only.
2) Online Payment is not done.
3) Parent cannot pay fee of their child online .They are supposed to come school to pay the fee.
4) Payroll module is describe briefly .

PROPOSED

ENHANCEMENT

Proposed Enhancement

User requirements keep changing as the system is being used.

Some of the future enhancements that can be done to this system are:-

As the technology emerges, it is possible o upgrade the system and can be adaptable to desired environment.-Because it is based on object-oriented design, any further changes can be easily adaptable.-Based on the future security issues, and security can be improved using emerging technologies.

Payroll module can be enhanced.

CONCLUSIONS

Conclusions

Towards the ends of the project, I would like to say that the target ,which was initially set up, is achieved to a good extent.

The project made me realize the significance of developing software, where the sole aim is to learn.

During the project, the real important for following all principles of System Analysis and Design dawned on me. I felt the necessity of going through the several stages, because only such a process could make one understand the problem in handling the enormous size of data and their manipulations.

Finally I can conclude that this system will eliminate the existing system's drawbacks and limitations to maximum extent and provide the user with a product of high quality, standards and excellence.

BIBLIOGRAPHY

Bibliography

Books:

The Complete Reference J2EE.

JSP Complete Reference.

Java Server Programming J2EE 1.4 Edition

MS SQL Server by Dusanpetkovic

Learning UML

URLs:

www.sun.java.com

www.google.com

ANNEXURES

ANNEXURE 1: USER INTERFACE SCREENS

Home Page

User Registration Page

Login Screen

Change Password

Forgot Password

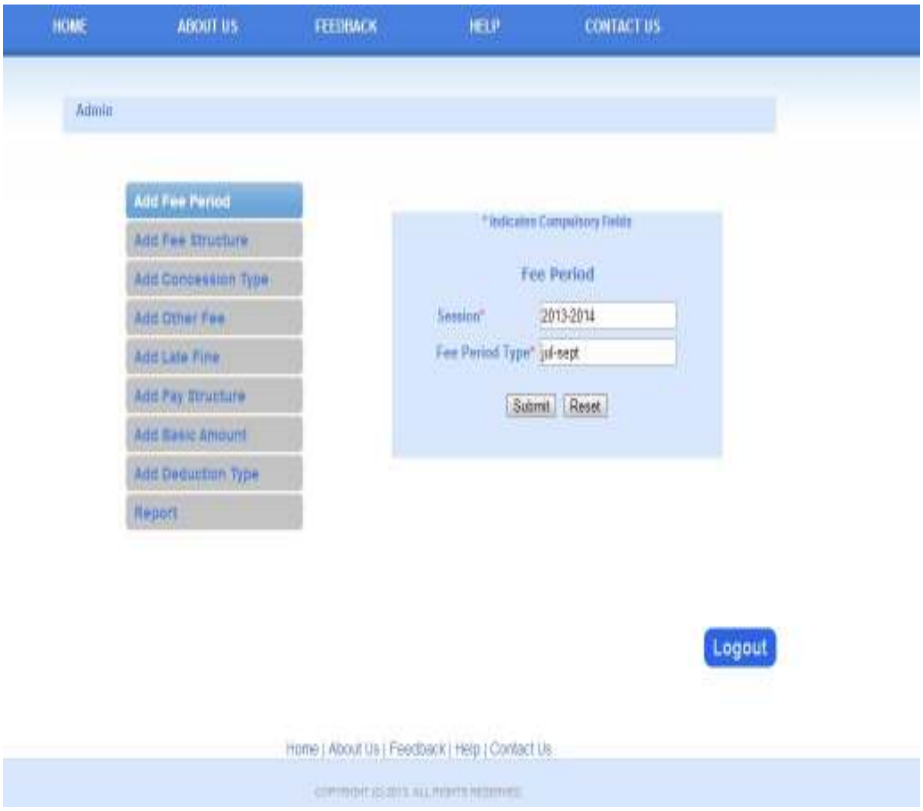
Admin

View Fee Period

Add Fee Period

Update Fee Period

Update Fee Period



View Fee Structure

Add Fee Structure

Update Fee Structure

Update Fee Structure

Concession Fee Type

Other Fee Type

Add Late Fine

Add Pay Structure

Add Basic Amount

Add Deduction Type

Accountant

Fee Collect

Fee Collect

Accountant
Personal Information's
Student Name Reshu
Class II
Section A
Fee Period arp-jun
Pay Date 04/12/2013
Status Not Paid
Last Fee Date 04/16/2013
Previous Amount 7000
Concession Amount 10%
Fee Details
Fee Type
Amount
Admission Fee
4000
Computer Fee
1000
Total Amount
5000
Late Fine Amount
0
Concession Amount
500
Amount To Pay
4500
Payment Details
Pay Mode Cash DD
Amount 4500
DD No.
DD Date
DD Amount
Bank Name
Branch Name
Submit
Reset
Back

Concession Fee

Other Fee

HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
Accountant
Collect Fee
Concession Fee
Other Fee
Pay Slip
Send SMS
Report For Fee
Report For Salary
* Indicates Compulsory Fields
Other Fee Details
Admission Number* A1
Search
Pay Date* 04/12/2013
Select Other Fee* Computer Fee
Amount* 2000
Remark
nothing
Submit
Reset
Logout

Pay Slip

HOME
ABOUT US
FEEDBACK
HELP
CONTACT US
Pay Slip
Search
Apr
Ok
Logout

Pay Slip

Accountant

Personal Information's

Employee Name	Ram Kumar	Designation	Teacher	PF A/c No.	6536453635
Pay Period	Apr	Pay Date	04/13/2013	Basic	20000
PAN	D456464CBF4432	Working Days	25	Paid Days	23

Salary Details

Earning	Amount
Basic	20000
Pay Grade	7000
DA	15930
HRA	5400
Health Allowance	1250
Other Allowance	1275
Eduction Allowance	200
Gross Salary	51055

Deduction Details

Deduction	Amount
Professional Tax	200
Total Deduction	200

Net Pay 50855

Submit Reset Back

Send SMS

Send SMS To Parent

Send SMS To Employee

User

View Profile

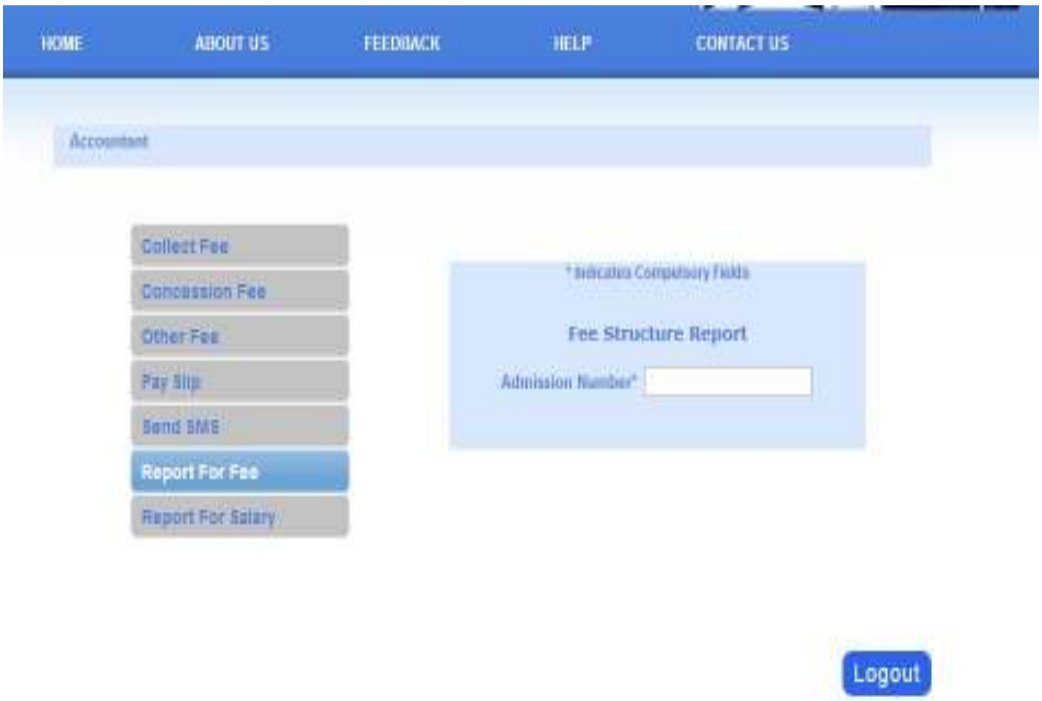
Update Profile

View Fee Structure

View Fee Structure Details

ANNEXURE: 2 OUTPUT REPORTS WITH DATA

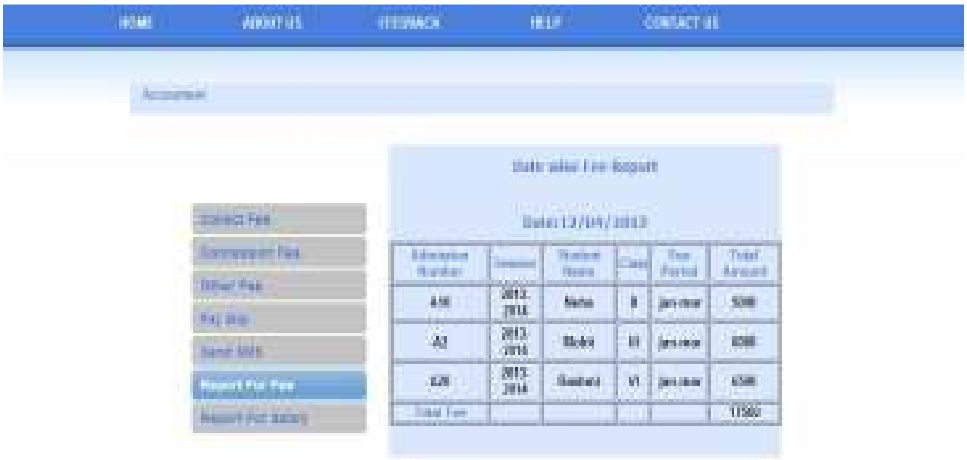
Report on Fee Structure



Fee Details of Student Report

Fee Pending Report

Date wise Fee Details



Report On Payroll

Monthly Salary of Employee

Yearly Salary of Employee

ANNEXURE:3 SAMPLE CODE

Sample code for pay other fee

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"pageEncoding="ISO-8859-1”import="Bean.Fee.*" import="java.sql.*" import="connections.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Other Fee</title>
<script language="javaScript" type="text/javascript" src="JavaScriptFile\calendar.js"></script>
<script language="javaScript" type="text/javascript" src="JavaScriptFile\feevalidation.js"></script>
<link href="CssFile\calendar.css" rel="stylesheet" type="text/css">
<script language="javascript">
var timeout    = 500;
varclosetimer= 0;
varddmenuitem      = 0;
// open hidden layer
functionmopen(id)
{
        // cancel close timer
        mcancelclosetime();
        // close old layer
        if(ddmenuitem) ddmenuitem.style.visibility = 'hidden';
        // get new layer and show it
```

```
        ddmenuitem = document.getElementById(id);
        ddmenuitem.style.visibility = 'visible';
}
// close showed layer
functionmclose()
{
        if(ddmenuitem) ddmenuitem.style.visibility = 'hidden';
}
// go close timer
functionmclosetime()
{
        closetimer = window.setTimeout(mclose, timeout);
}
// cancel close timer
functionmcancelclosetime()
{
        if(closetimer)
        {
                window.clearTimeout(closetimer);
                closetimer = null;
        }
}
// close layer when click-out
document.onclick = mclose;
</script>


<link rel="stylesheet" type="text/css" href="CssFile/menu.css" />
<link                rel="stylesheet"                type="text/css"
href="CssFile/menuCssCode.css" />
</head>
<%
FeeLabelfeeLabelBean=new FeeLabel();
String admissionNo=feeLabelBean.getLblAdmissionId();
String payDate=feeLabelBean.getLblPayDate();
String selectFee=feeLabelBean.getLblSelectOtherFeeType();
```

```
String amount=feeLabelBean.getLblAmount();
String remark=feeLabelBean.getLblRemark();
String regClass=feeLabelBean.getLblClass();
String section=feeLabelBean.getLblSection();
%>
<body>
<!--logo start here=========================--->

        <div>
                        <a
href="HomePage.jsp"><imgsrc="images/logo.png"></a>
        </div>

<!--logo end here=========================--->


<!--menu start here=========================--->

<div align="justify" id="menuhead">
                        <ul>
                                <li                     id="menu_logout"><a
href="HomePage.jsp" >Logout</a></li>

                        </ul></div>

<!--menu end here=========================--->


<!--Title=========================================-
->
<div  >
<table  align="center">
                <tr>
<th class="heading">WELCOME TO ACCOUNTANT</th>
                        </tr>
                </table>
```

```
        </div>
<!--                                    end                                    of
Title==========================================-->
<%
String enquiry = (String)request.getAttribute("Finalmessage");
        // System.out.println("hi"+message);
        if( enquiry != null){
%>
<table align="center" width="50%">
        <tr>
                <td><font
color="red"><b><%=enquiry%></b></font></td>
        </tr>
</table>


<% request.removeAttribute("Finalmessage");
        }//end of if( message != null)
%>
<!--                    start                    of                    main
body==========================================-->
<br><br>
<div>
<table    align="center"    cellpadding="5"    CELLSPACING="5"
height=600 width=960>
<tr>
<th class="t1" valign=top >
<div id="menu4">
                        <ul>
                                <!-- CSS Tabs -->
<li><a href="EditStudent.jsp">Collect Fee</a></li>
<li><a          href="ConcessionAmountPerStudent.jsp">Concession
Fee</a></li>
<li><a id="current" href="OtherFeeType.jsp">Other Fee</a></li>
<li><a  href="FeeReport.jsp">Report</a></li>
```

```
                </ul>
            </div>


</th>
<th class="h">
<p class="body_heading">Concession Fee</p><hr color="#696969">
<div class="note_field">Note: Filled mark with (<span>*</span>) denotes mandatory field</div>

<form name="formOtherFeeType" onSubmit="return formValidationOtherrFee(this)">
<table align="center" cellpadding = "10" border="0" width="490">
<tr>
<td class="parent_infoleft_lable"><%=admissionNo %><font color="red">*</font></td>
<%
                        String regClass1 = "";
                        String section1 = "";
                        String admNo = "";
                        String admissionNumber="";

                        try {
                                Connection con = null;
                                ResultSetrs = null;

        Connections.ConnectionFunctionconnectionFunctionObj = null;
                                connectionFunctionObj = new Connections.ConnectionFunction();
                                con = (Connection) connectionFunctionObj.connectionFunction();
```

```
                                        CallableStatementcalbstmt = null;
                                        ResultSet rs1 = null;
                                        calbstmt = con
                                                .prepareCall("{call
selectAdmissionNumberForReportProcedure(?)}");
                                        calbstmt.setString(1,
request.getParameter("txtAdmissionId"));

        admissionNumber=request.getParameter("txtAdmissionId");
                                        rs = calbstmt.executeQuery();
                                        if (!rs.next()) {
                                                System.out.println("not
found admission");

                                        } else {
                                                admNo = rs.getString(1);

                                                CallableStatement
calbstmt1 = null;
                                                calbstmt1 = con

        .prepareCall("{call
selectClassAndSectionInformationProcedure(?)}");
                                                calbstmt1.setString(1,
admissionNumber);
                                                rs1                     =
calbstmt1.executeQuery();
                                                if (!rs1.next()) {

        System.out.println("not found");
                                                } else {
                                                        regClass1       =
rs1.getString(1);
                                                        section1        =
rs1.getString(2);
                                                }
```

```
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            %>


<td><input type="text" name="txtAdmissionId" maxlength="10" class="student_record_left_box"></td>
<td><input type="submit" name="search" value="Search" class="change_btn2"></td></tr>
</table>


</form>
<hr color="#696969">
<br>
<form name="formOtherFeeType" action="OtherFeeTypeDetailServlet" method="POST" onSubmit="return formValidationOtherFeee(this)">
<table align="center" cellpadding = "10" border="0" width="400">
<tr>
<td class="parent_infoleft_lable"><%=regClass %></td>
<td><input type="text" name="txtClass" value="<%=regClass1%>" class="student_record_left_box" maxlength="10"></td>
</tr>
<tr>
<td class="parent_infoleft_lable"><%=section%></td>
<td><input type="text" name="txtSection" value="<%=section1%>" class="student_record_left_box" maxlength="10"></td>
</tr>
<tr>
<td class="parent_infoleft_lable"><%= payDate%><font color="red">*</font></td>
```

```
<td><input type="text" name="datum1"
class="student_record_left_box" ><a href="#"
onClick="setYears(1990, 2050);showCalender(this, 'datum1');">
<imgsrc="Image\calender.png"></a>



<!--CalenderScript  -->

<table id="calenderTable">
<tbody id="calenderTableHead">
<tr>
<td colspan="4" align="center">
        <select
onChange="showCalenderBody(createCalender(document.getEleme
ntById('selectYear').value,
        this.selectedIndex, false));" id="selectMonth">
        <option value="0">Jan</option>
        <option value="1">Feb</option>
        <option value="2">Mar</option>
        <option value="3">Apr</option>
        <option value="4">May</option>
        <option value="5">Jun</option>
        <option value="6">Jul</option>
        <option value="7">Aug</option>
        <option value="8">Sep</option>
        <option value="9">Oct</option>
        <option value="10">Nov</option>
        <option value="11">Dec</option>
        </select>
</td>
<td colspan="2" align="center">
                        <select
onChange="showCalenderBody(createCalender(this.value,
```

```
        document.getElementById('selectMonth').selectedIndex,
false));" id="selectYear">
                                    </select>
                                </td>
<td align="center">
                                <a  href="#"  onClick="closeCalender();"><font
color="#003333" size="+1">X</font></a>
                                </td>
                            </tr>
</tbody>
<tbody id="calenderTableDays">
<tr style="">
<td>Sun</td><td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</
td><td>Fri</td><td>Sat</td>
</tr>
</tbody>
<tbody id="calender"></tbody>
</table>

</td>
</tr>
<tr>
<td        class="parent_infoleft_lable"><%=        selectFee%><font
color="red">*</font></td>
<td><select                                    name="cmbOtherFeeType"
class="add_infoleft_select2" >
<option value="">Select </option>
<%
        String otherFeeType=null;
  Connection con = null;
ResultSetrs = null;
CallableStatementcalbstmt = null;
Connections.ConnectionFunctionconnectionFunctionObj=new
Connections.ConnectionFunction();
try
```

```
    {
        con=connectionFunctionObj.connectionFunction();
        calbstmt = con.prepareCall("{call
selectOtherFeeTypeProcedure()}");
rs=calbstmt.executeQuery();
if(rs.next())
       {
        do
             {

                otherFeeType= rs.getString(1);
 %>
                <option><%=otherFeeType  %></option>
                <%
                }
while (rs.next());
           }
  }
catch(Exception e1)
  {
        e1.printStackTrace();
  }


%>
</select></td>

</tr>
<tr>
<td class="parent_infoleft_lable"><%=amount %><font
color="red">*</font></td>
<td><input type="text" name="txtAmount" maxlength="20"
class="student_record_left_box"/></td>
</tr>
<tr>
<td class="parent_infoleft_lable"><%= remark%></td>
```

```
<td><textarea name="txtRemark" cols="25" rows="5"
class="new_add_textarea ">
</textarea>
</td>
</tr></table>
<center>
<input type="submit" name="submit" value="Submit"
class="change_btn2"/>
<input type="hidden" name="hidAdmNO"
value="<%=admissionNumber%>"/>
</center>

</form>
</th>
</tr>
</table>
</div>
<!-- end of main
body==========================================-->
<!-- start of
footer==========================================-->

<div id="footer">
        <p>Copyright (c) 2013. All rights reserved.</p>
    </div>


<!-- end of
footer==========================================-->

</body>
</html>
```