

1. INTRODUCTION

An individual report card of each student has to be displayed and printed at a keystroke according to any selected format. An important aid for teachers and students to judge their performance. Merit list printing by totals for a class by individual subject marks for a class. Student performance in a particular subject or all the subjects must be expressed. Performance of teachers of various classes can be easily compared.

1.1 Purpose

The system displays the list of all issues that are open, closed, in progress. If the user can get registered by clicking on the logon button and provide the required information as specified. Each time the registered customer come on to the site he can makes use of the user name and the password that is allocated to him.

1.2 Key features

- User configurable grading system.
- User configurable examination pattern.
- Examinations-weight age handling.
- Calculated / Average Column handling.
- Grace marks handling.
- Special Analysis section.
- Sub-subjects marks entry handling.

2. SOFTWARE AND HARDWARE SPECIFICATIONS

2.1 Software requirements

Operating System	:	Window 2000, XP
User interface	:	Java, Servlets, JSP
Database	:	My SQL
Documentation Tool	:	Ms Office

2.2 Hardware requirements

Processor	:	Standard processor with a speed of 1.6 GHz or more
RAM	:	256 MB RAM or more
Hard Disk	:	20 GB or more
Monitor	:	Standard color monitor
Keyboard	:	Standard keyboard
Mouse	:	Standard mouse

3. LITERATURE SURVEY

3.1. Java Server Pages

JSP not only enjoys cross-platform and cross-Web-server support, but effectively melds the power of server-side Java technology with features of static HTML pages.

JSP pages typically comprise of: Static HTML / XML components.

Special JSP tags.

Optionally, snippets of code written in the java programming language called “script lets.”

JSP Architecture

The purpose of JSP is to provide a declarative, presentation-centric method of developing servlets. JSP pages are subject to a translation phase and a request-processing phase. The translation phase is carried out only once, unless the JSP page changes, in which case it is repeated. The JSP engine itself typically carries out the translation phase, when it receives a request for the JSP page for the first time.

Life Cycle of A JSP: Life cycle of a JSP consists of the following three methods:

`_jspInit`

`_jspService`

`_jspDestroy`

3.2 Servlets

A servlet is a java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming mode. Servlets are Java technology's answer to Common Gateway Interface (CGI) Programming. They are programs that run on a Web server, acting as middle layer between request coming from a Web browser or other HTTP client and databases of applications on the HTTP server.

Servlet Life Cycle: The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

1. If an instance of the servlet does not exist, the Web container:

Loads the servlet class.

Creates an instance of the Servlet class.

Initializes the servlet instance by calling the init method.

2. Invokes the service method, passing request and response objects.

If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

Session Management

Many applications require that a series of requests from a client be associated with one another. Sessions are represented by an Http Session object. A session can be accessed by calling the get Session () method of a request object. This method returns the current session associated with this request, or, if the request does not have a session, it creates one. The timeout period can be accessed by using a session's [get\set] Max Inactive Interval methods.

Session Tracking

A Web container can use several methods to associate a session with a user, all of which involve passing an identifier between the client and the server. The identifier can be

maintained on the client as a cookie, or the Web component can include the identifier in every URL that is returned to the client.

In fact, on many servers, they use cookies if the browser supports them, but automatically revert to URL-rewriting when cookies are unsupported or explicitly disabled.

4. SOFTWARE REQUIREMENTS ANALYSIS

Requirement analysis for web applications encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals for the web application are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user's point-of-view are developed. This intent is to establish a basic understanding of why the web application is built, who will use it, and what problems it will solve for its users.

4.1 Scope

4.1.1 Existing System with Limitations

- It is time consuming process as the user has to type the dbase commands. He has to remember all the commands which are difficult.
- It is limited to a single system.
- A user who wants only to have some information has to contact the administrator every time.

4.1.2 Proposed System Features

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the internet.

- Vast amount of data can be stored.
- There is no risk of data mismanagement at any level while the project development is under process.
- Relationship between the administrator, owner/developer and subcontractor can be maintained very easily.
- It provides high level of security using different protocols like https etc.

The Student Result Processing consists of 3 users or modules, they are:

- Administrator
- Student
- Staff

4.2 Administrator Module

The functionalities of Administrator are

1. The Administrator should Login into the system with unique his/her username and password.
2. If the username and password is valid then he can gain the access to the system.
3. Admin views staff Personal details.
4. Admin updates staff Personal details.
5. Admin views student Personal details.
6. Admin updates student Personal details.
7. Admin views his/her own Personal details.
8. Admin updates his/her Personal details.
9. Admin views attendance of students.
10. Admin updates attendance of students
11. Admin views Results of students.
12. Admin updates Results of students.

13. Admin views Schedules.
14. Admin updates Schedules.
15. Admin sends emails to students once marks have been entered.

The Administrator can do the following actions

1. Login
2. Admin Actions
 1. Views personal details
 2. updates schedules
 3. views reports
 4. updates attendance of students
3. Logout

4.3 Staff Module

The functionalities of Staff are

1. The Staff should login into the system with unique her/his username and password.
2. If the user name and password are valid then he can gain access to the system.
3. Staff views students Personal details.
4. Staff views his/her own Personal details.
5. Staff updates his/her Personal details.
6. Staff views attendance of students.
7. Staff views Results of students.
8. Staff views Schedules.

The Staff can do the following actions

1. Login
2. Staff Actions
 1. views personal details
 2. views attendance
 3. view results
3. Log out

4.4 Student Module

The functionalities of Student are

1. The Student should login into the system with unique her/his username and password.
2. If the user name and password are valid then he can gain access to the system.
3. Student views his/her own Personal details.
4. Student updates his/her Personal details.
5. Student views attendance of students.
6. Student views Results.
7. Student views Schedules

The Student can do the following actions:

1. Login
2. Student Actions
 1. views attendance
 2. views results

3. View Schedules

3. Log out

2. SYSTEM DESIGN

5.1 UML DIAGRAMS

5.1.1 Class diagram

A class diagram represents the structure of the system. It shows set of classes, interfaces, and relationships between them.

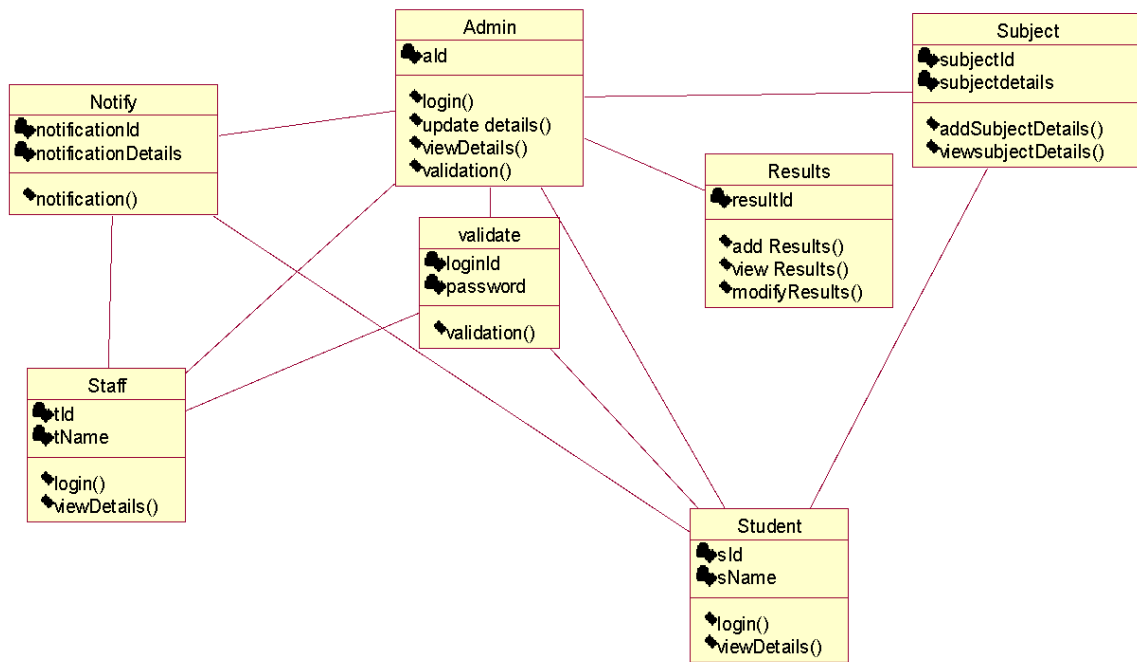


Fig. 5.1

5.1.2 Sequence and collaboration diagrams

Sequence Diagram

- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.

Administrator Sequence

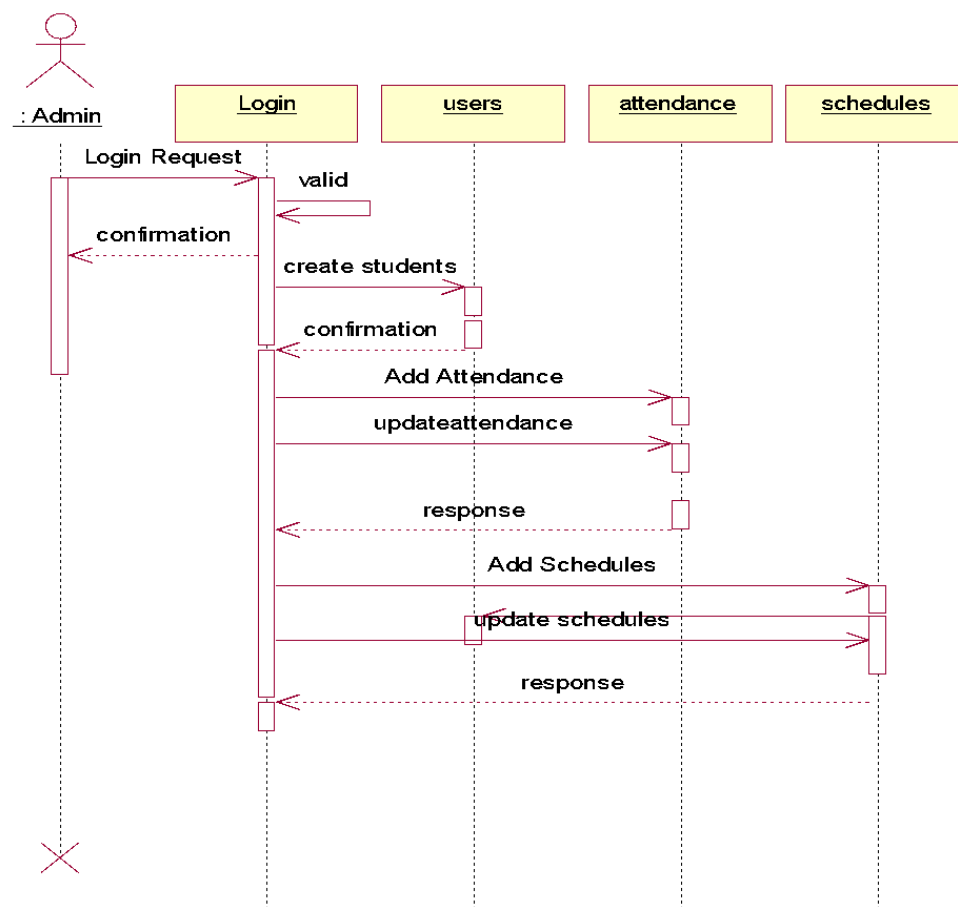


Fig. 5.2

Staff Sequence

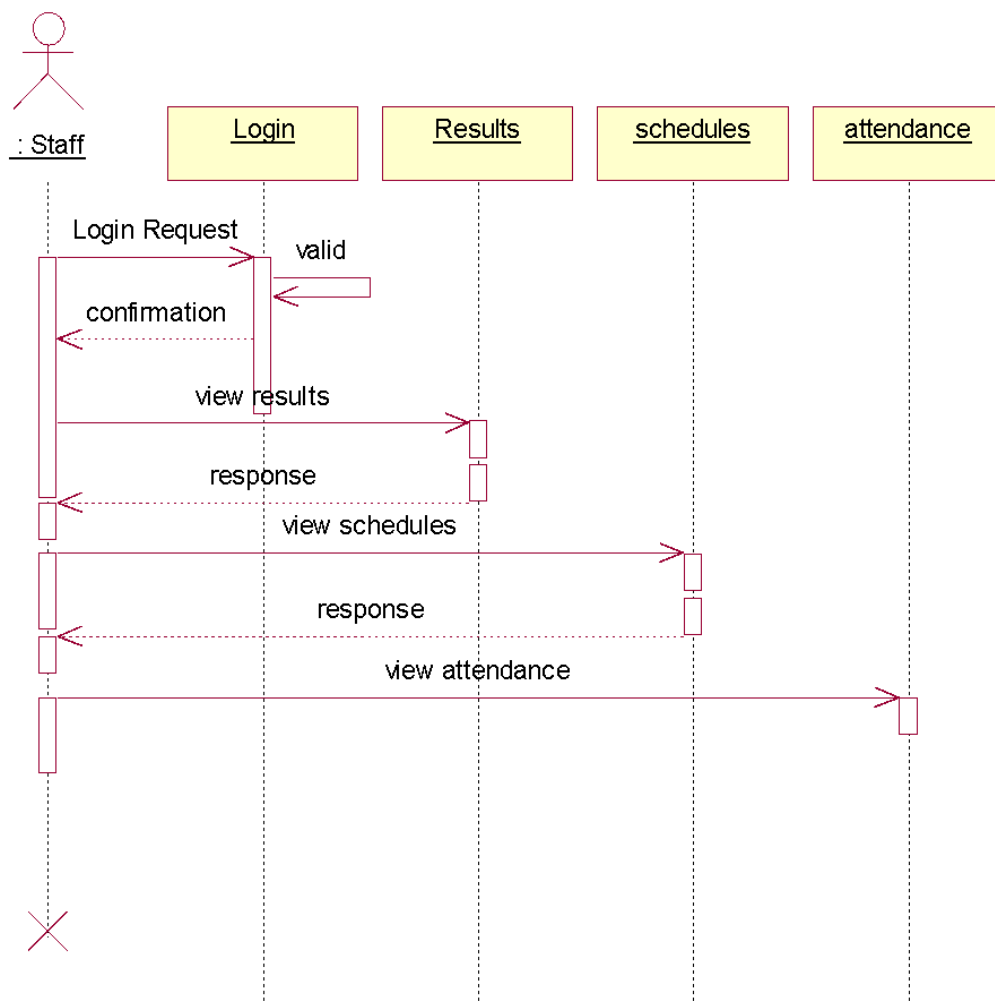


Fig. 5.3

Student Sequence

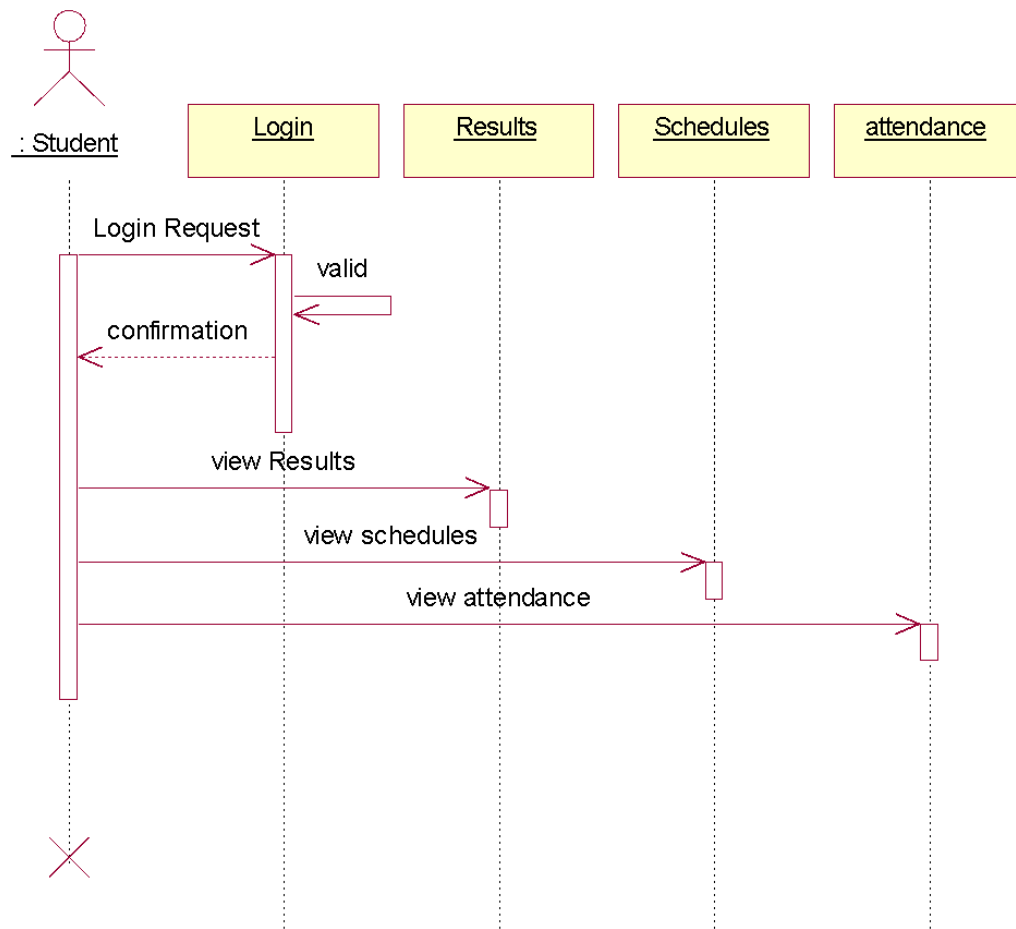


Fig. 5.4

Collaboration Diagram

- Collaboration is a society of classes, interfaces, and other elements that work together to provide some cooperative behavior that's bigger than the sum of all its parts.
- Collaboration is also the specification of how an element, such as a classifier or an operation, is realized by a set of classifiers and associations playing specific roles used in a specific way

Administrator Collaboration:

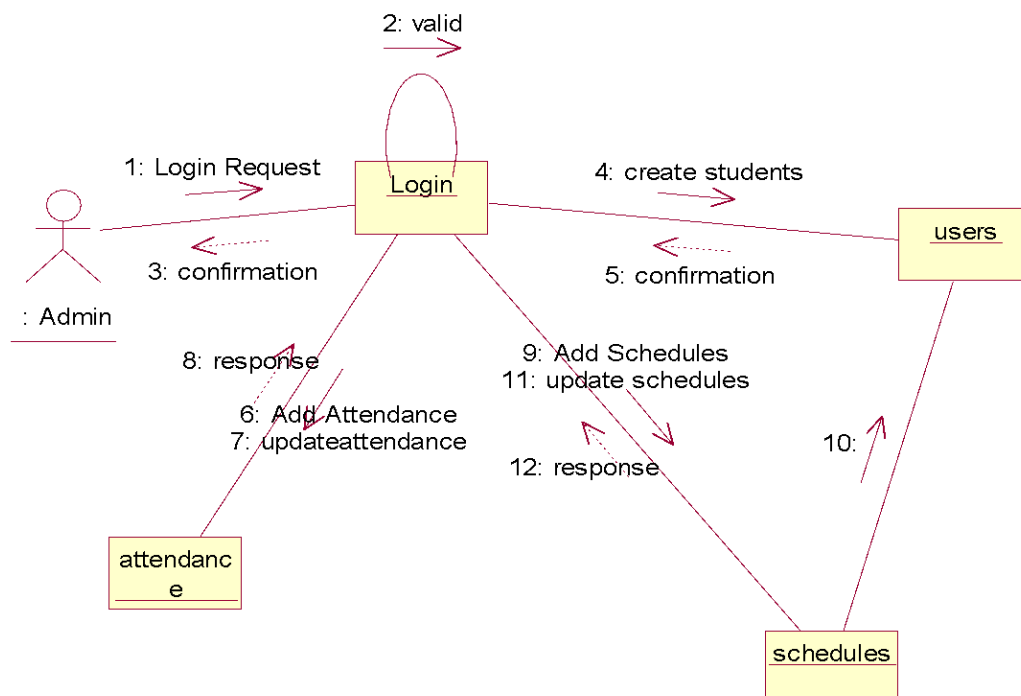


Fig. 5.5

Staff Collaboration

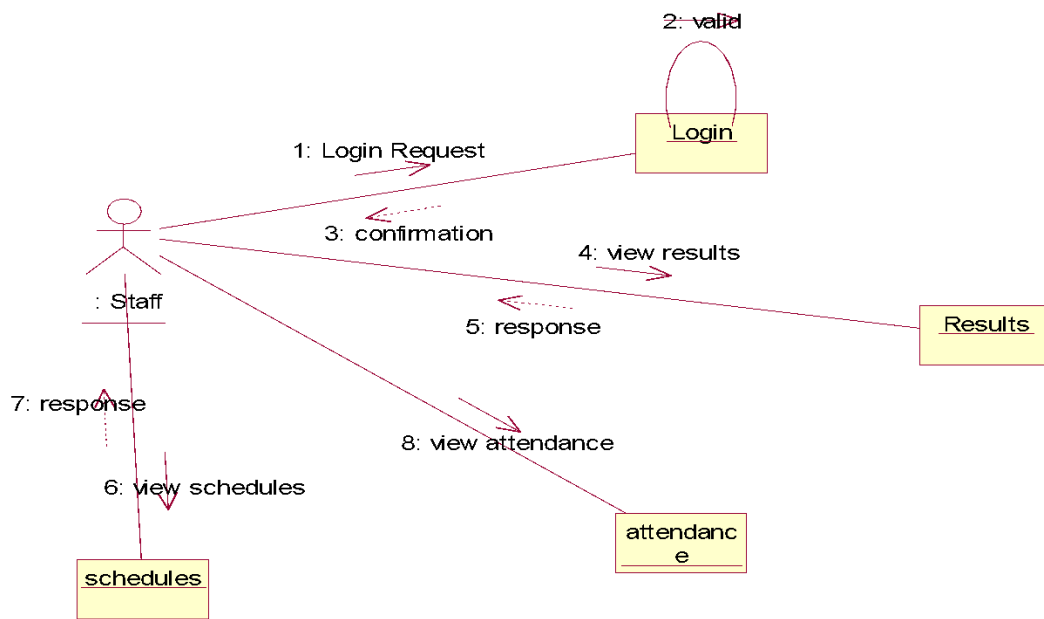


Fig. 5.6

Student Collaboration

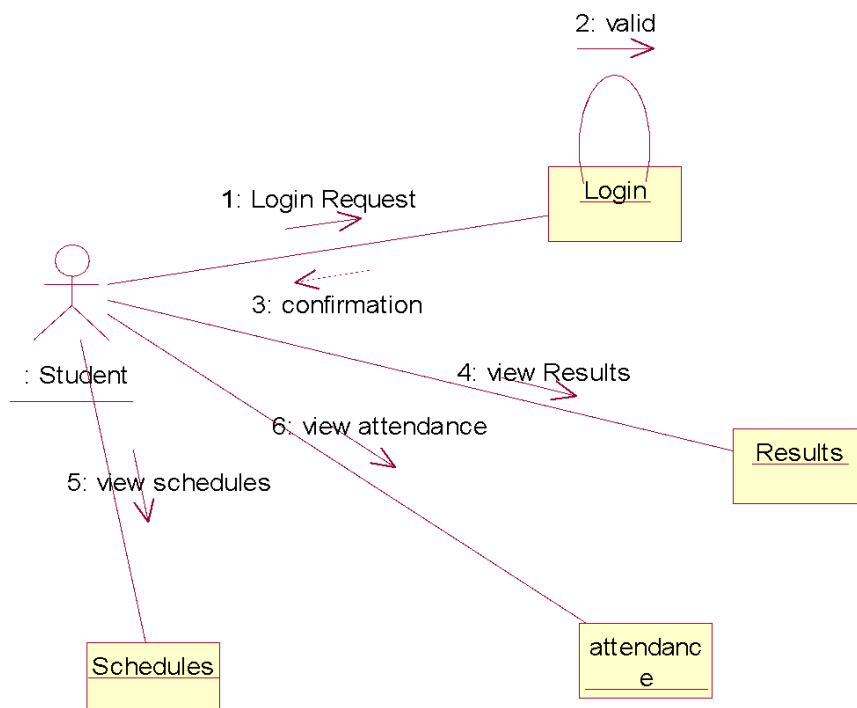


Fig. 5.7

5.1.3 Use Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and relationships.

Administrator Use Case

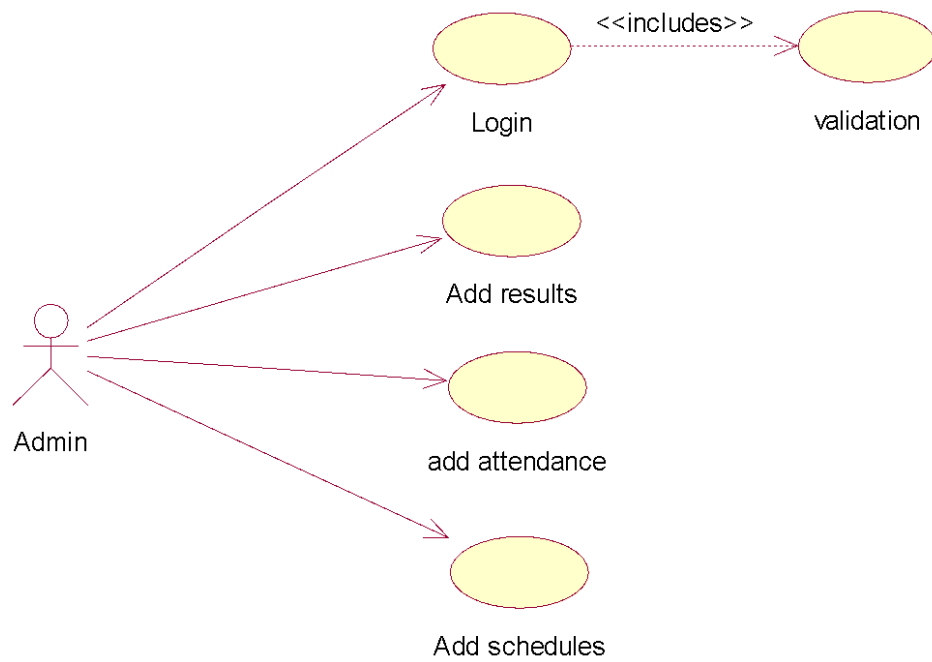


Fig. 5.8

Staff Use Case

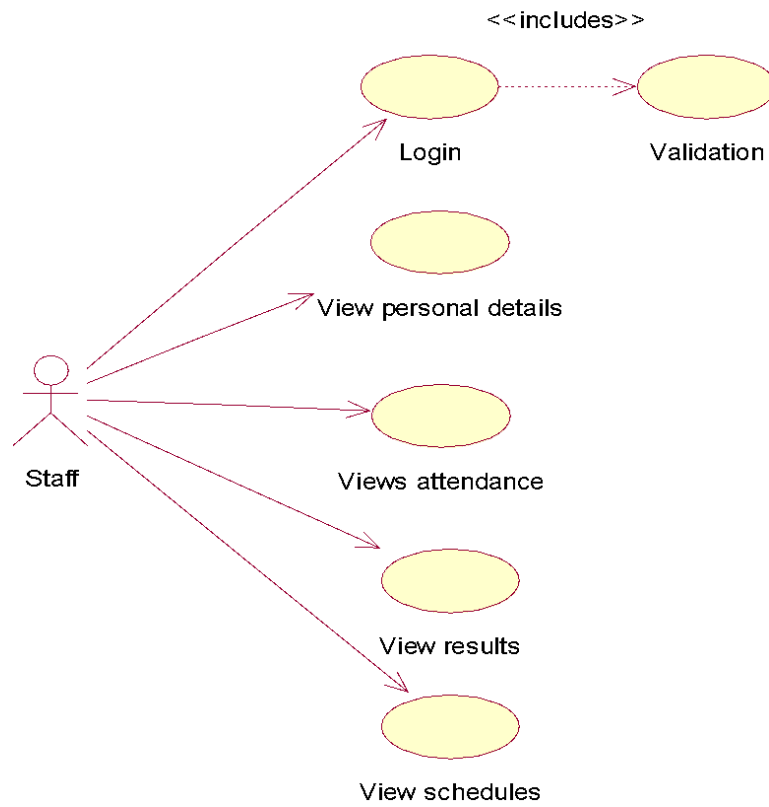


Fig. 5.9

Student Use Case

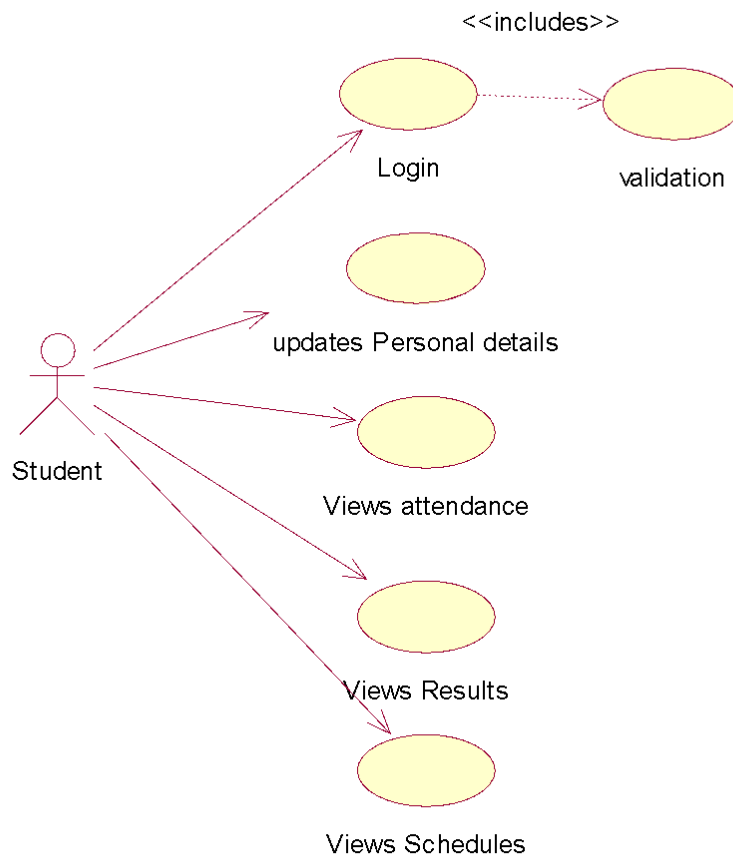


Fig. 5.10

5.2 CONTROL FLOW DIAGRAMS

5.2.1 Activity diagrams

- An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

Login Process

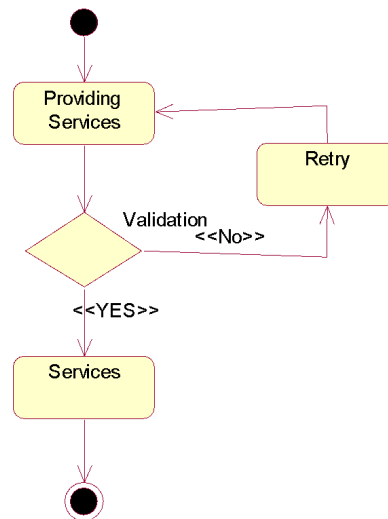


Fig. 5.11

Registration Process

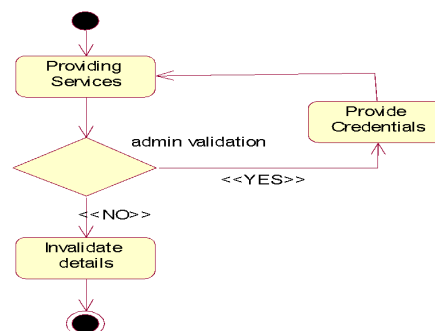


Fig. 5.12

Administrator Process

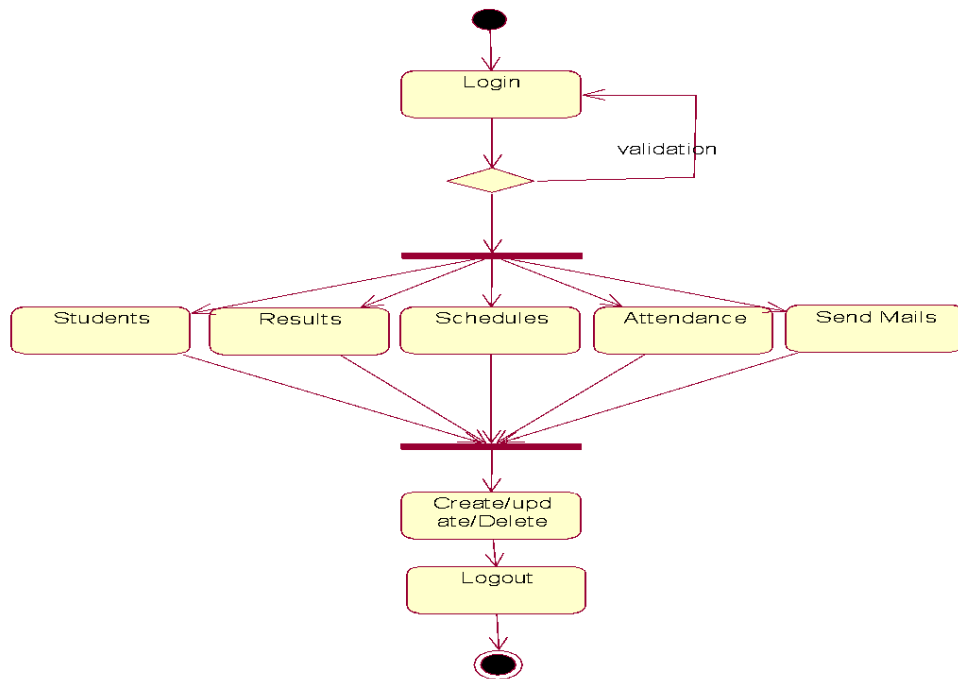


Fig. 5.13

Staff Process

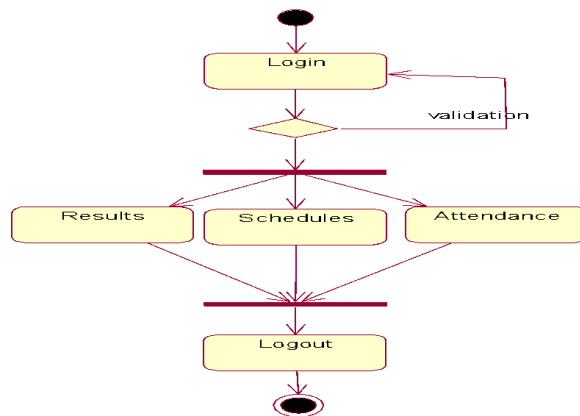


Fig. 5.14

Student Process

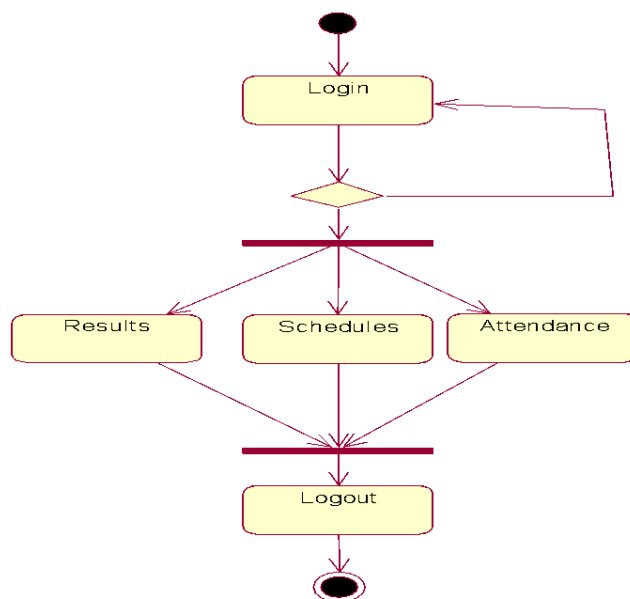


Fig. 5.15

5.3 component diagram

Component is a physical Part of a system that conforms to and provides realization of set of interfaces. A component is a self contained unit that encapsulates state and behavior of various set of classifiers. A component provides set of classes and interfaces with some functionality and GUI interfaces which may be required to in several services.

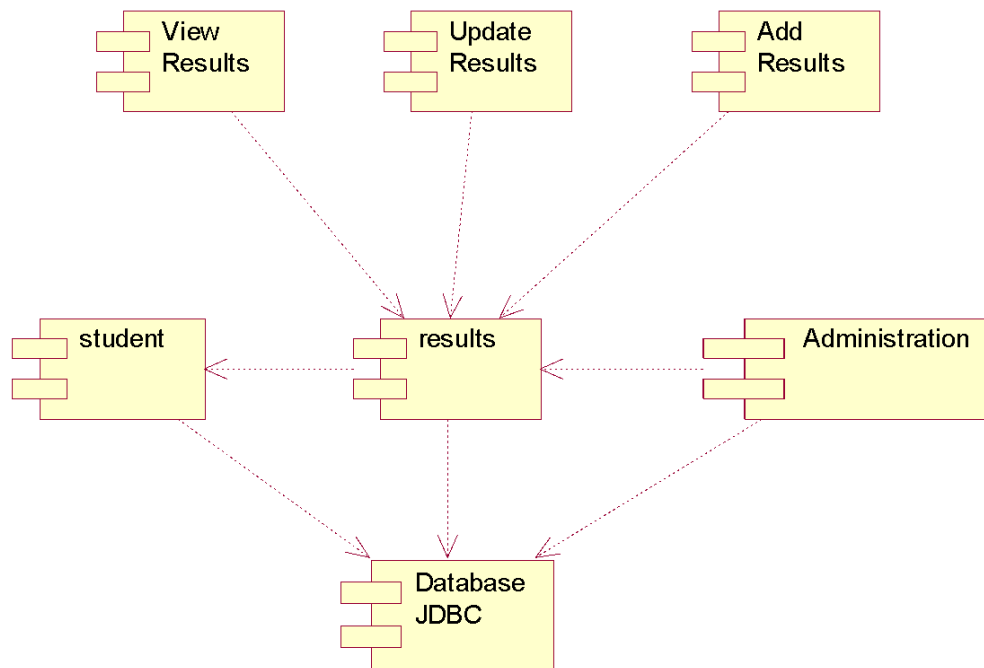


Fig. 5.16

5.4 Deployment diagram

- A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them.
- Graphically, a deployment diagram is collection of vertices and arcs.

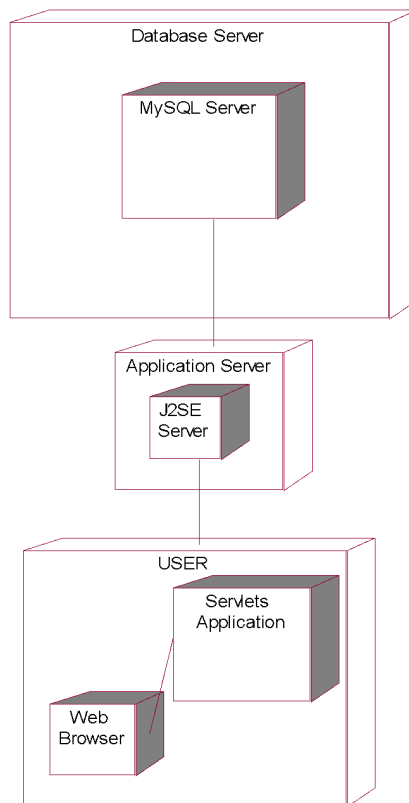


Fig. 5.17

5.5 ER Diagram

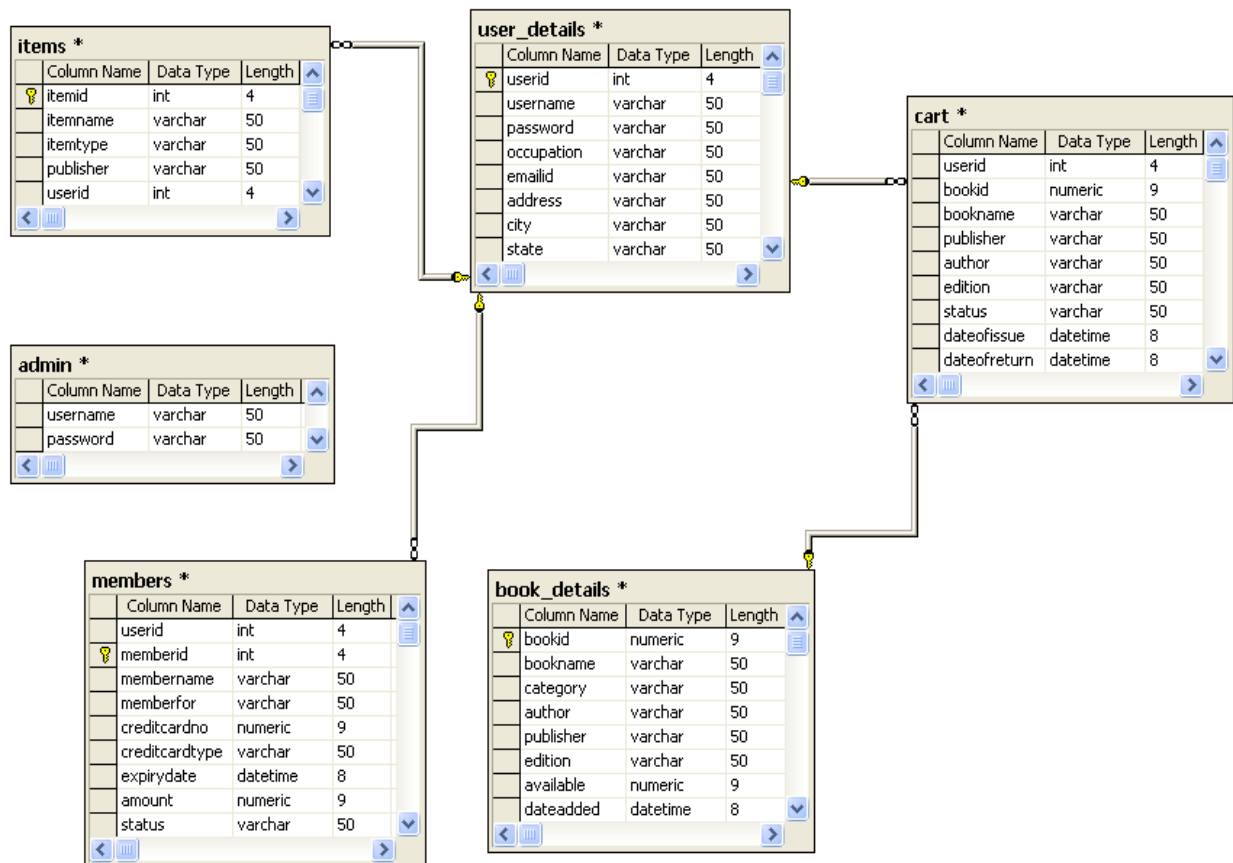


Fig. 5.18

3. TESTING

Testing Methodologies

- Black box Testing:
- White box Testing.
- Gray Box Testing.

Testing:

- The process of executing a system with the intent of finding an error.
- Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
- Quality is defined as justification of the requirements
- Defect is nothing but deviation from the requirements
- Defect is nothing but bug.
- Testing --- The presence of bugs
- Testing can demonstrate the presence of bugs, but not their absence
- Debugging and Testing are not the same thing!
- Testing is a systematic attempt to break a program or the AUT
- Debugging is the art or method of uncovering why the script /program did not execute properly.

Testing Methodologies:

- **Black box Testing:** is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.
Usually Test Engineers are involved in the black box testing.
- **White box Testing:** is the testing process in which tester can perform testing on an application with having internal structural knowledge.
Usually The Developers are involved in white box testing.
- **Gray Box Testing:** is the process in which the combination of black box and white box techniques are used.

Example for GUI Test cases:

T.C. No	Description	Expected value	Actual value	Result
1	Check for all the features in the screen	The screen must contain all the features		
2	Check for the alignment of the objects as per the validations	The alignment should be in proper way		

1. Positive Test Cases:

- The positive flow of the functionality must be considered
- Valid inputs must be used for testing
- Must have the positive perception to verify whether the requirements are justified.

Example for Positive Test cases:

T.C. No	Description	Expected value	Actual value	Result

1	Check for the date Time Auto Display	The date and time of the system must be displayed		
2	Enter the valid Roll no into the student roll no field	It should accept		

2. Negative Test Cases:

- Must have negative perception.
- Invalid inputs must be used for test.

Example for Negative Test cases:

T.C. No	Description	Expected value	Actual value	Result
1	Try to modify the information in date and time	Modification should not be allow		
2	Enter invalid data in to the student details form, click on Save	It should not accept invalid data, save should not allow		

7. OUTPUT SCREENS

Login Page

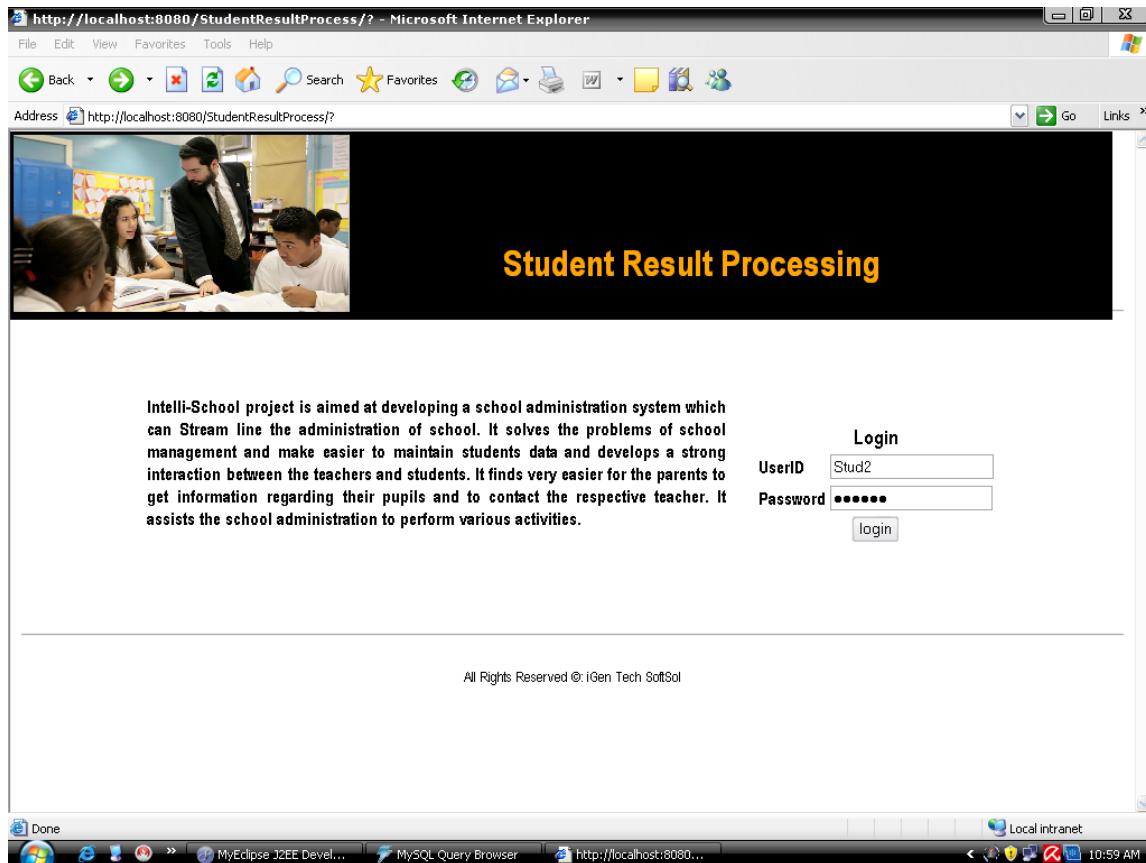


Fig. 7.1

Student Homepage

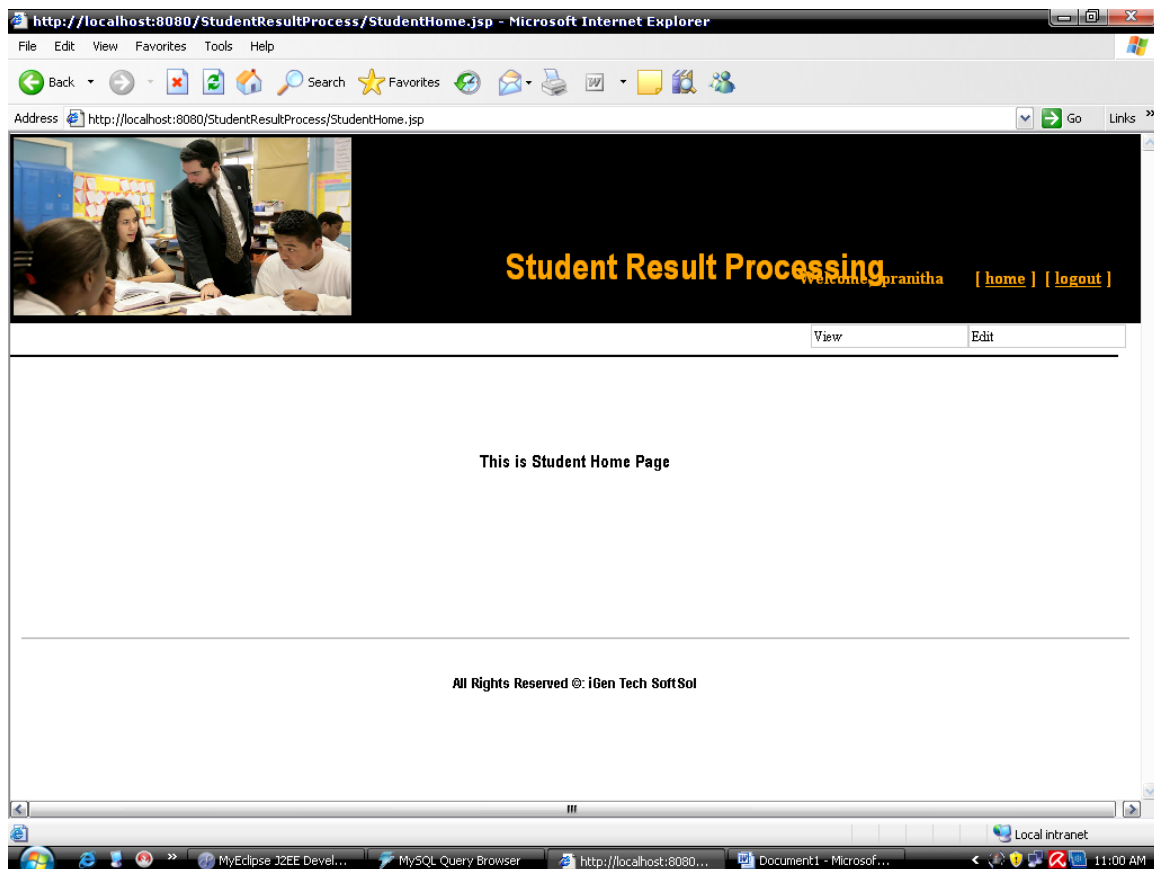


Fig. 7.2

Student View Details

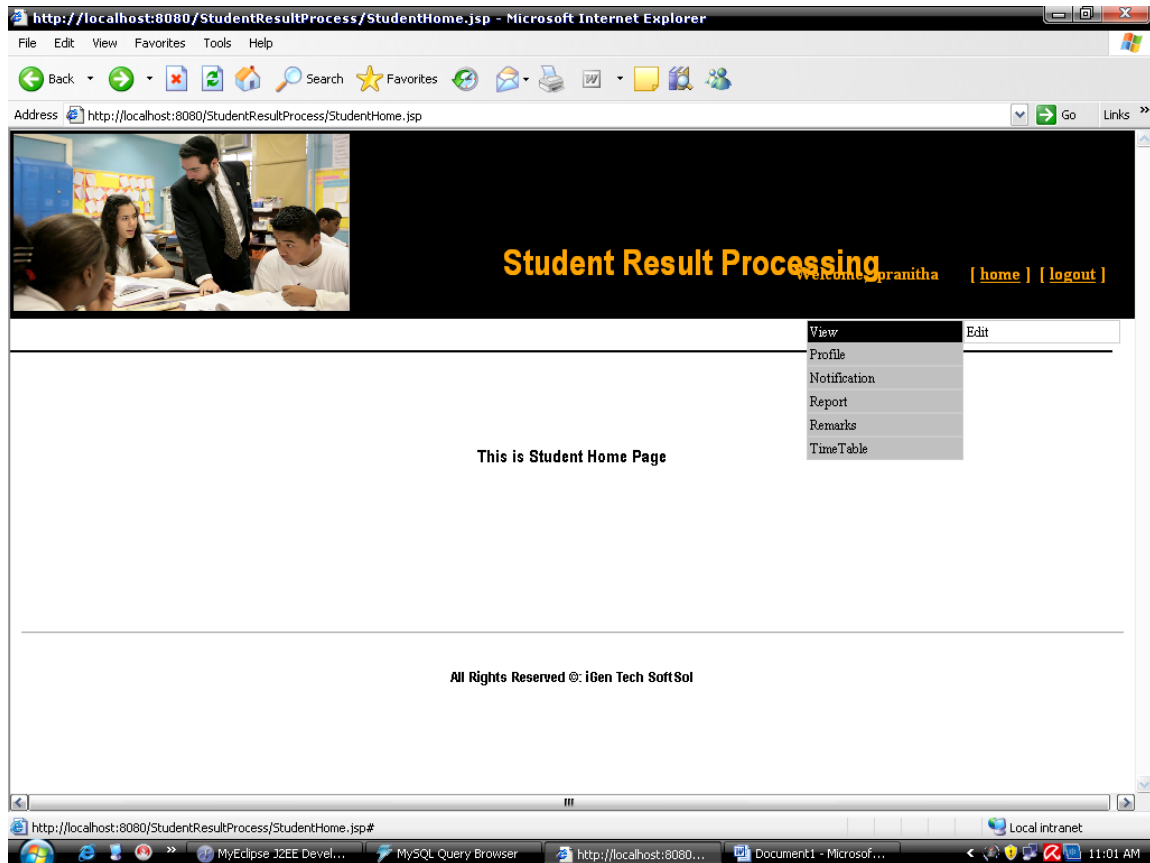


Fig. 7.3

Student Edit Details

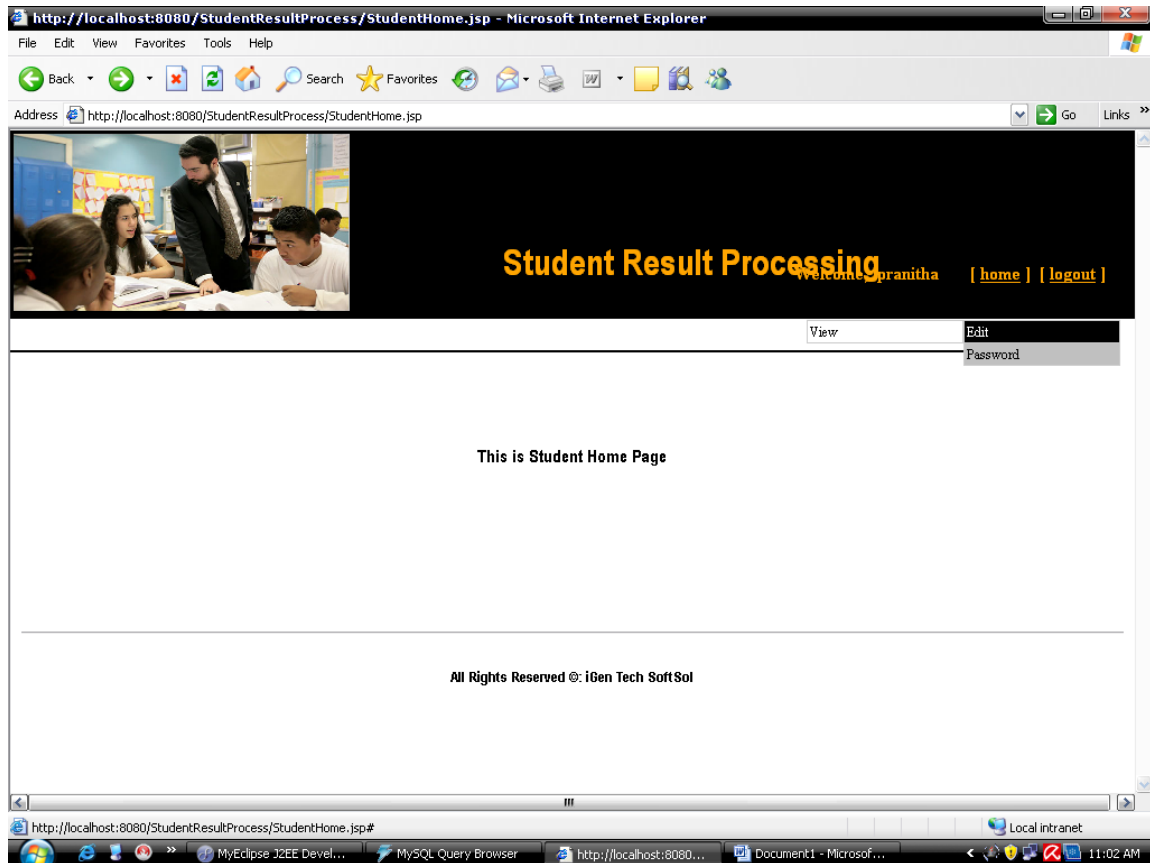


Fig. 7.4

View Student Profile

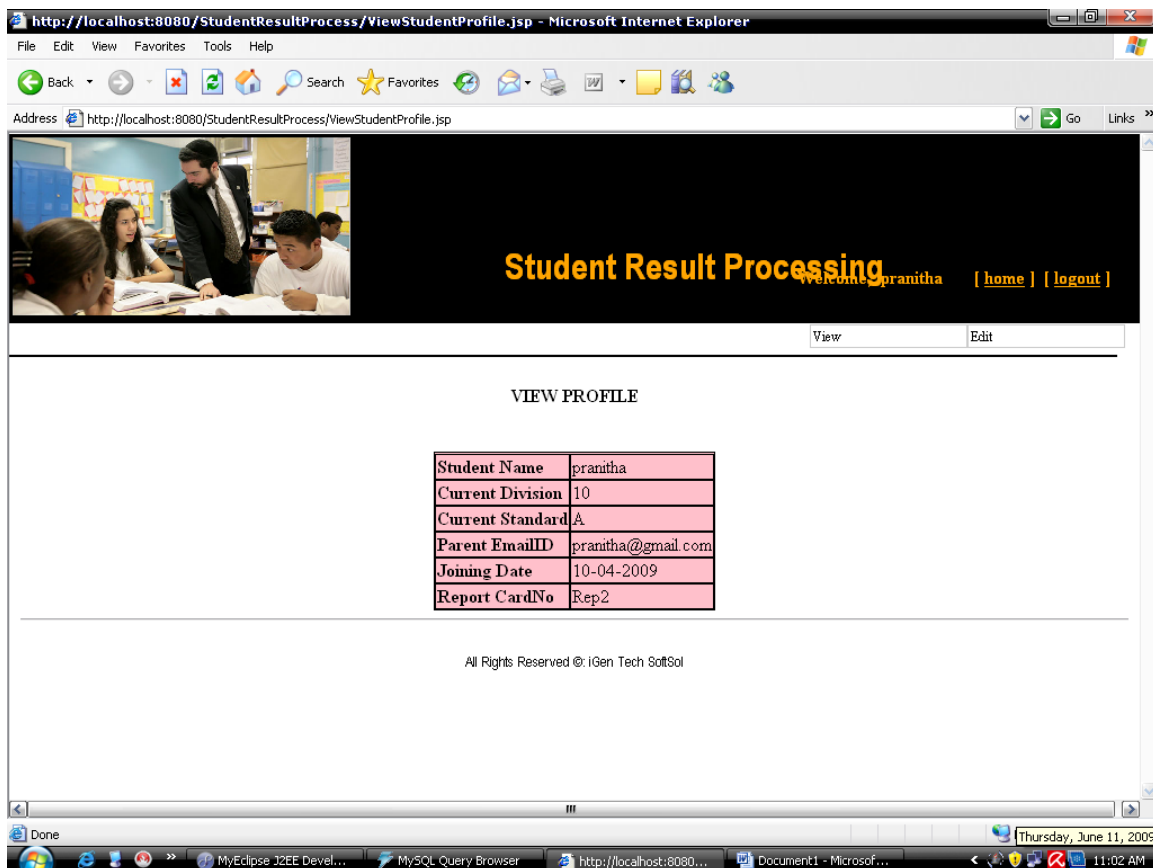


Fig. 7.5

Notification Details

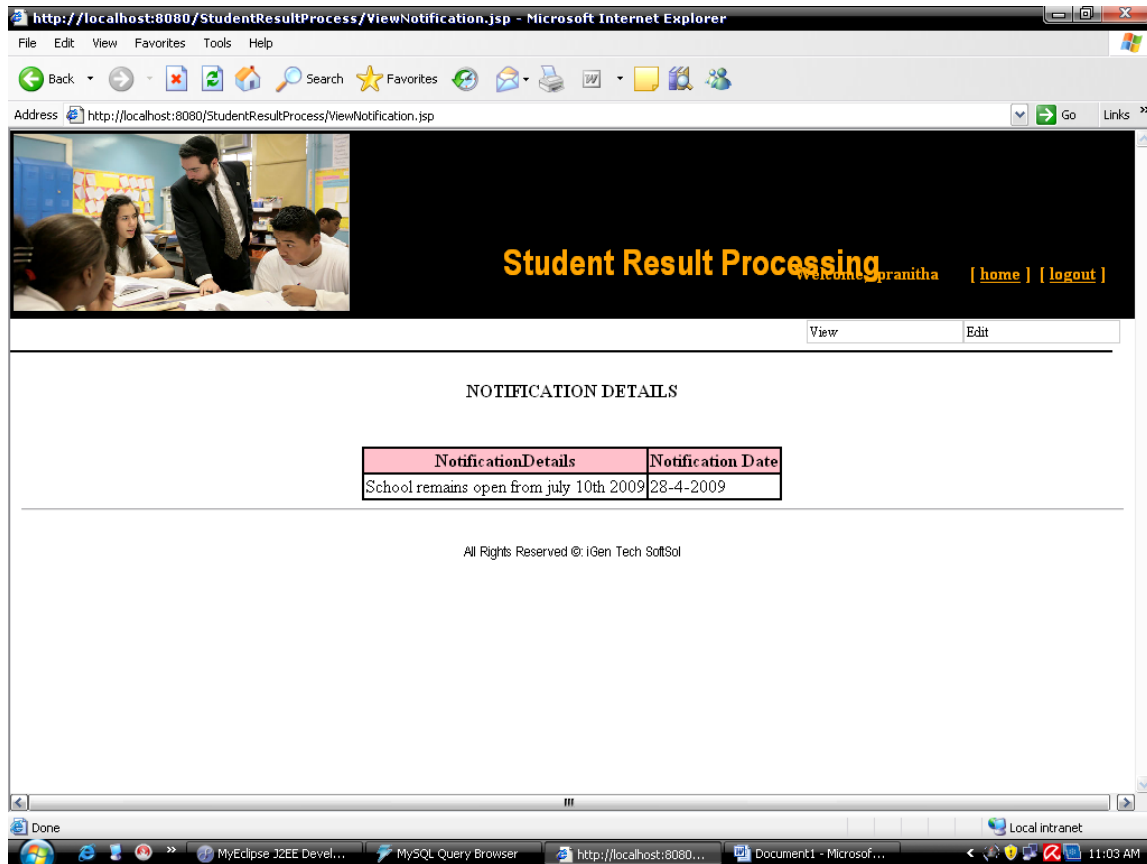


Fig. 7.6

View Marks Details:

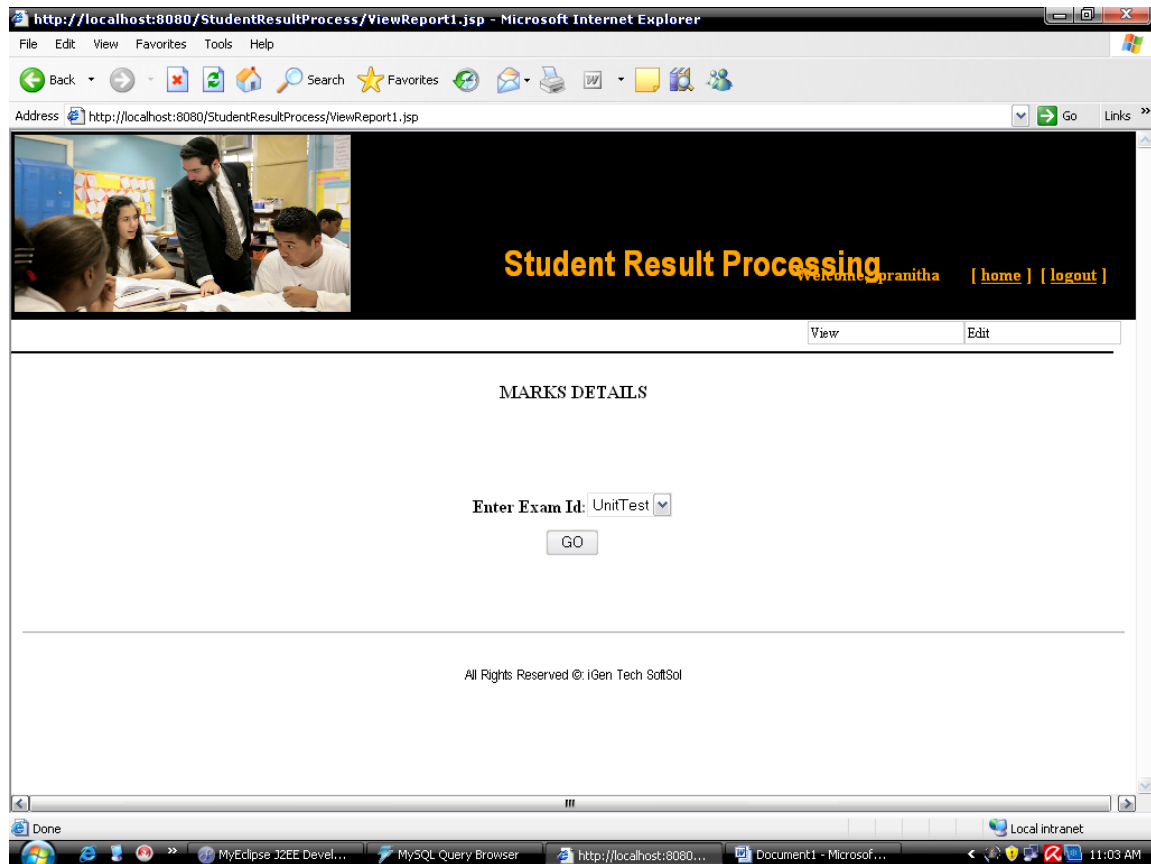


Fig. 7.7

View Remarks

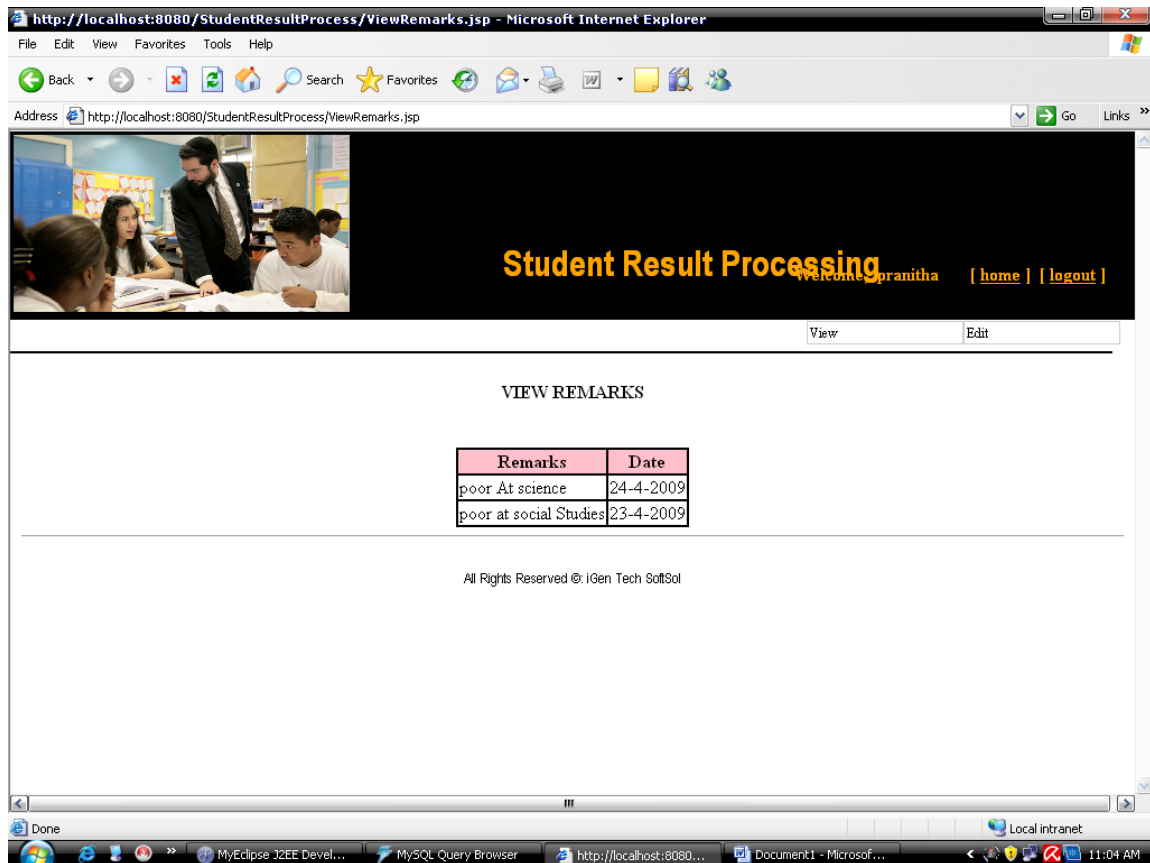


Fig. 7.8

View Schedule Details

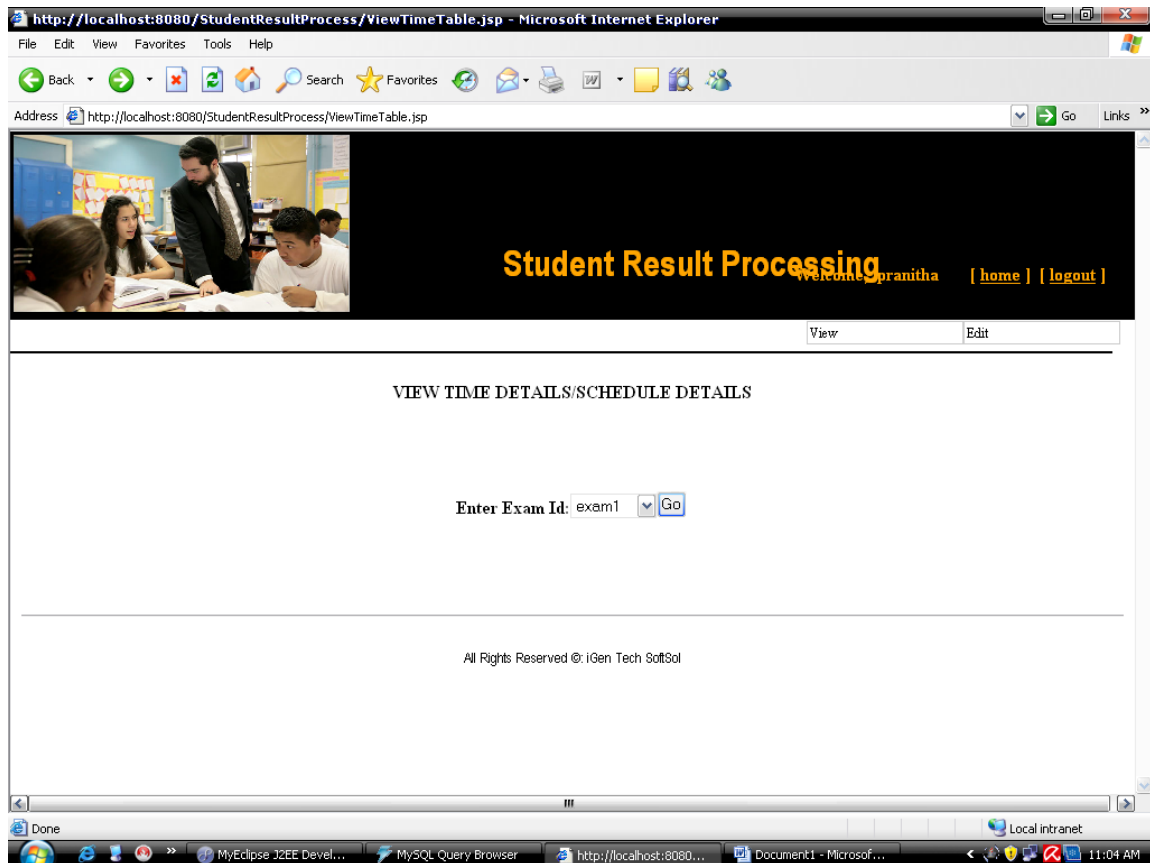


Fig. 7.9

View Details

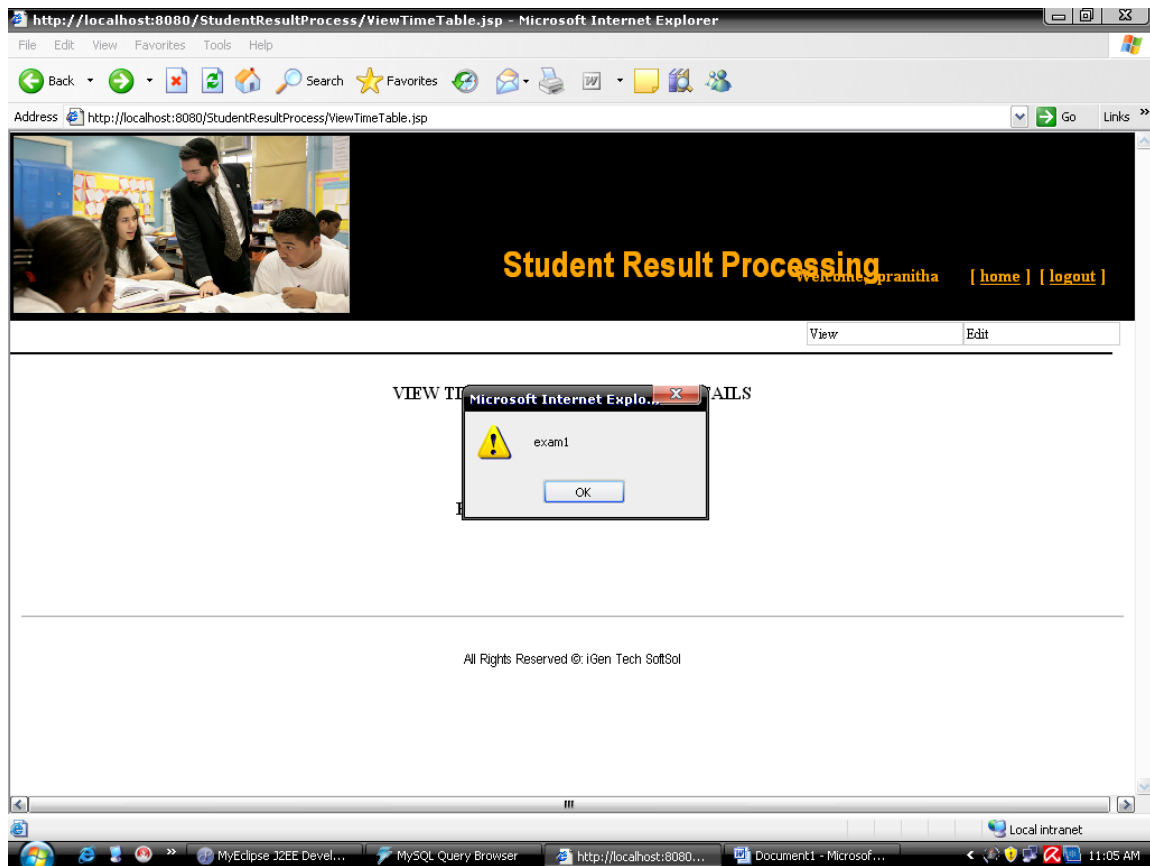


Fig. 7.10

Details Information

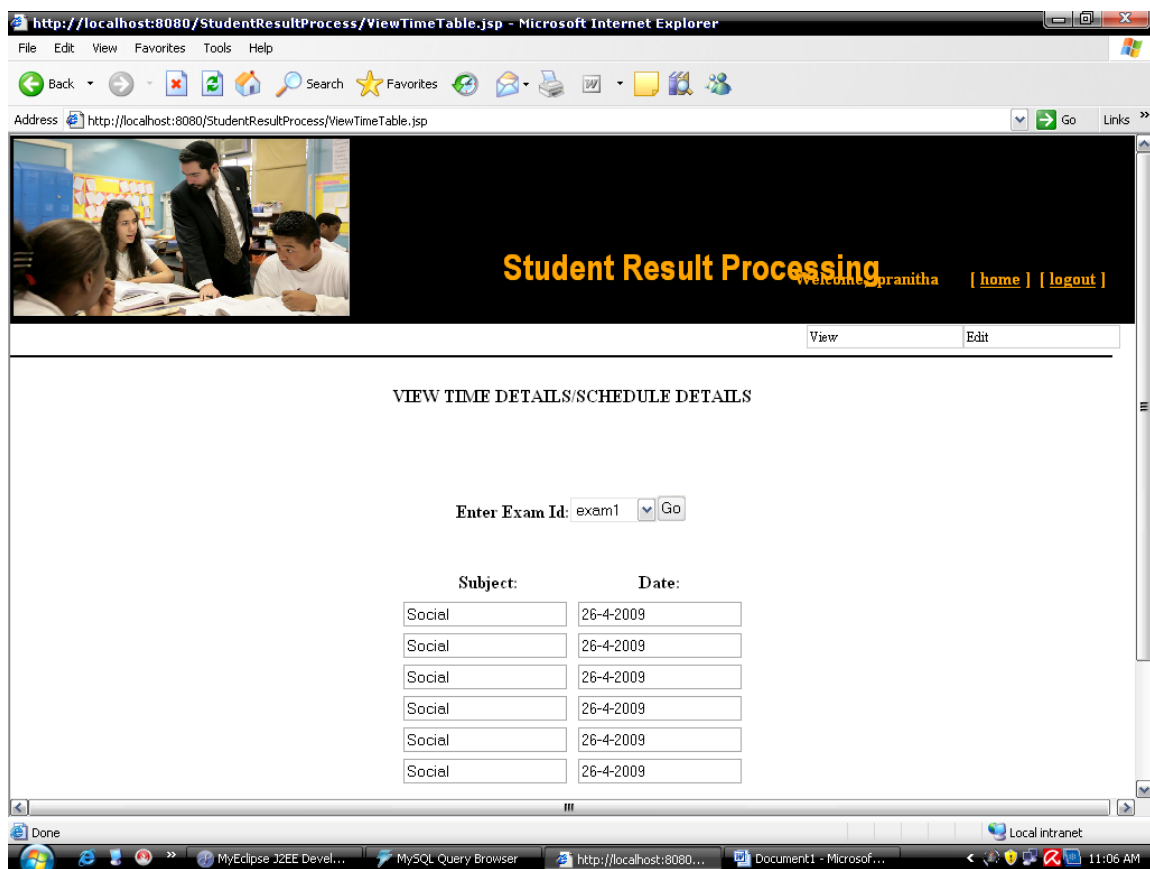


Fig. 7.11

Change Password

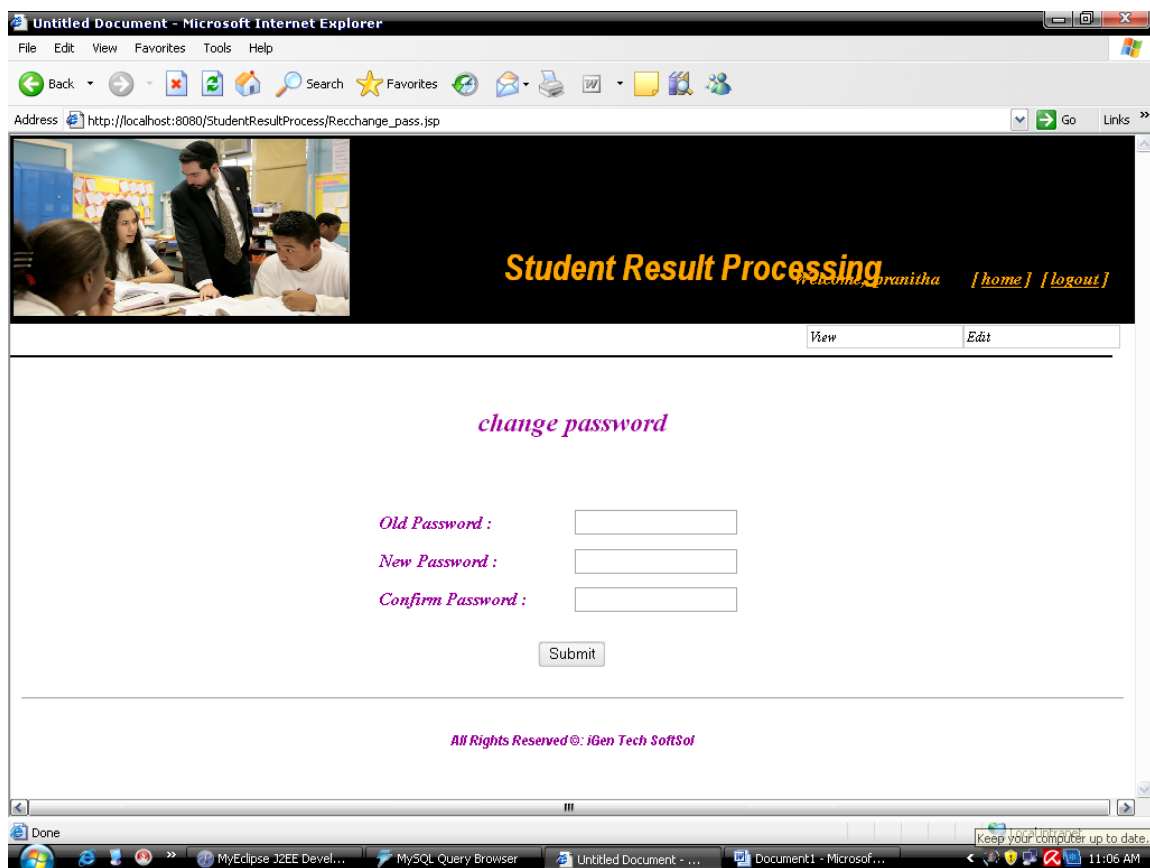


Fig. 7.12

Logout

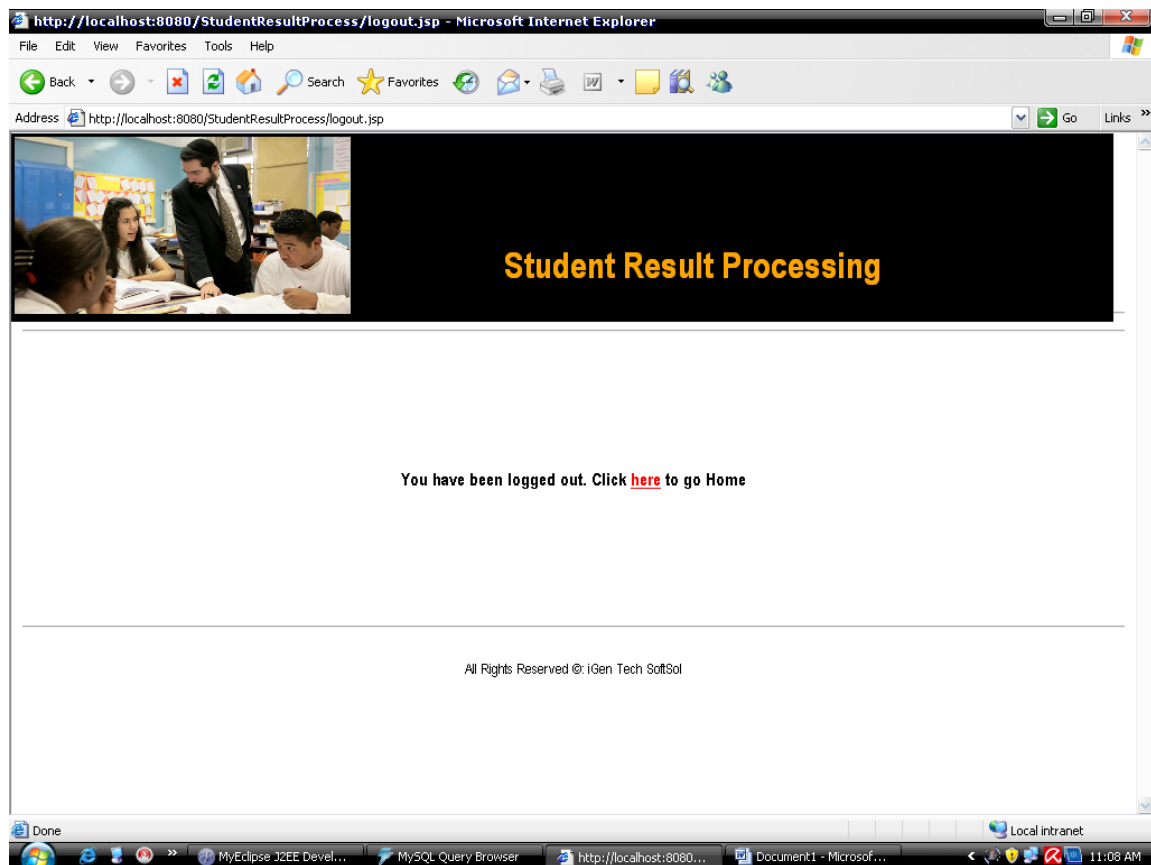


Fig.7.13

Faculty Login

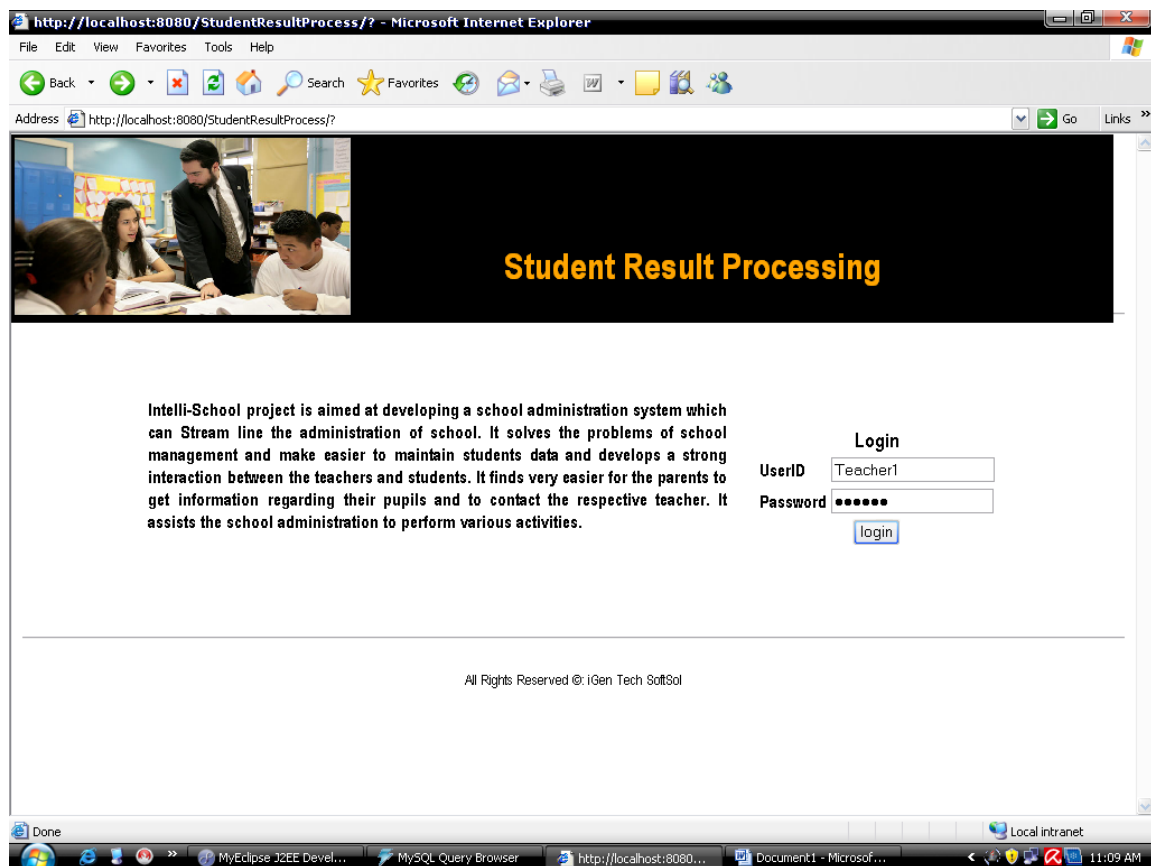


Fig. 7.14

Faculty Home Page

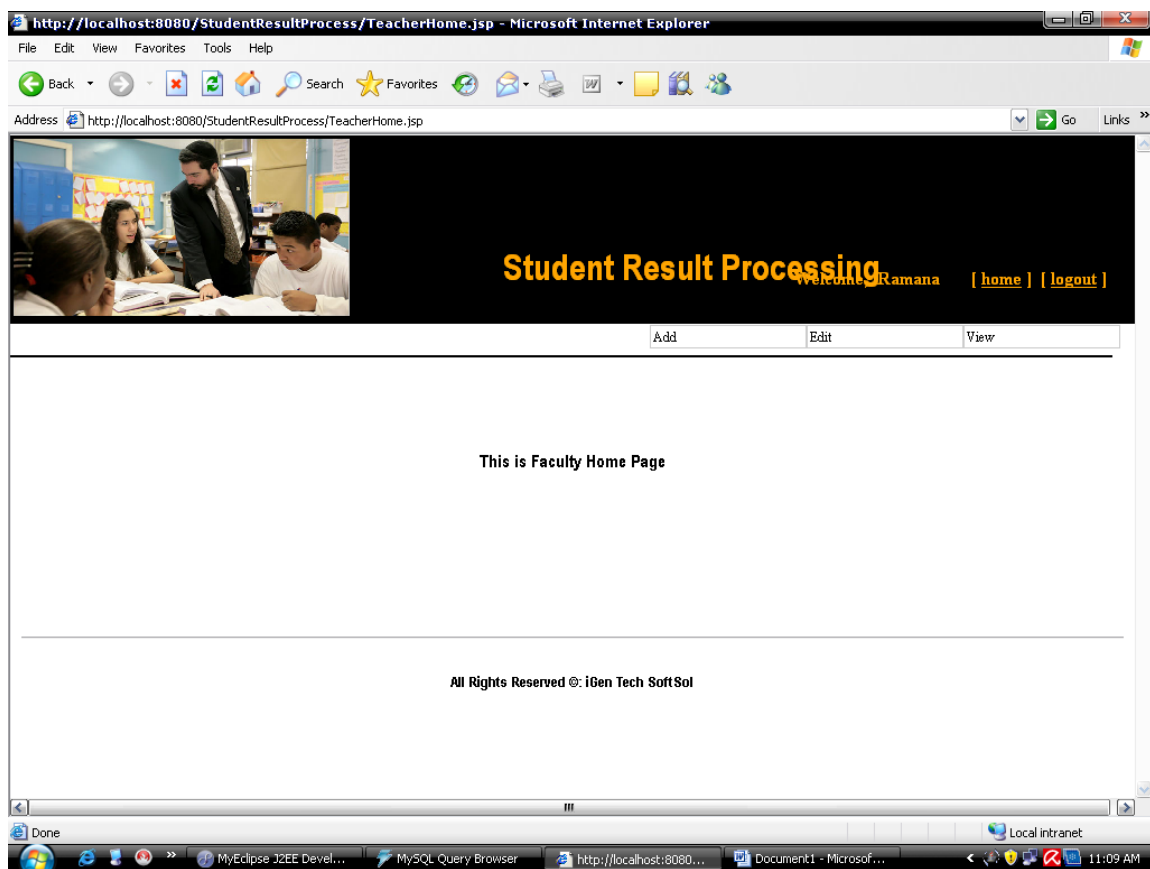


Fig. 7.15

Add Details

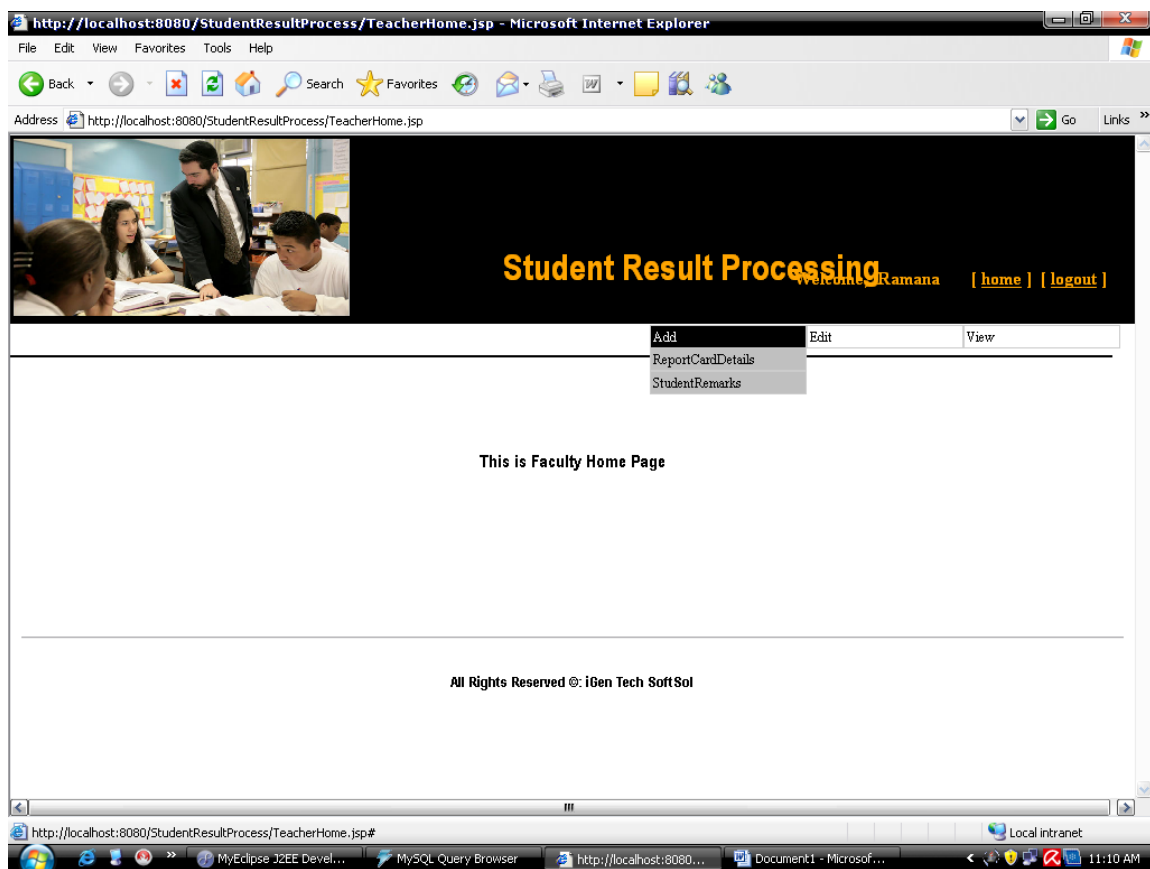


Fig. 7.16

Edit Information

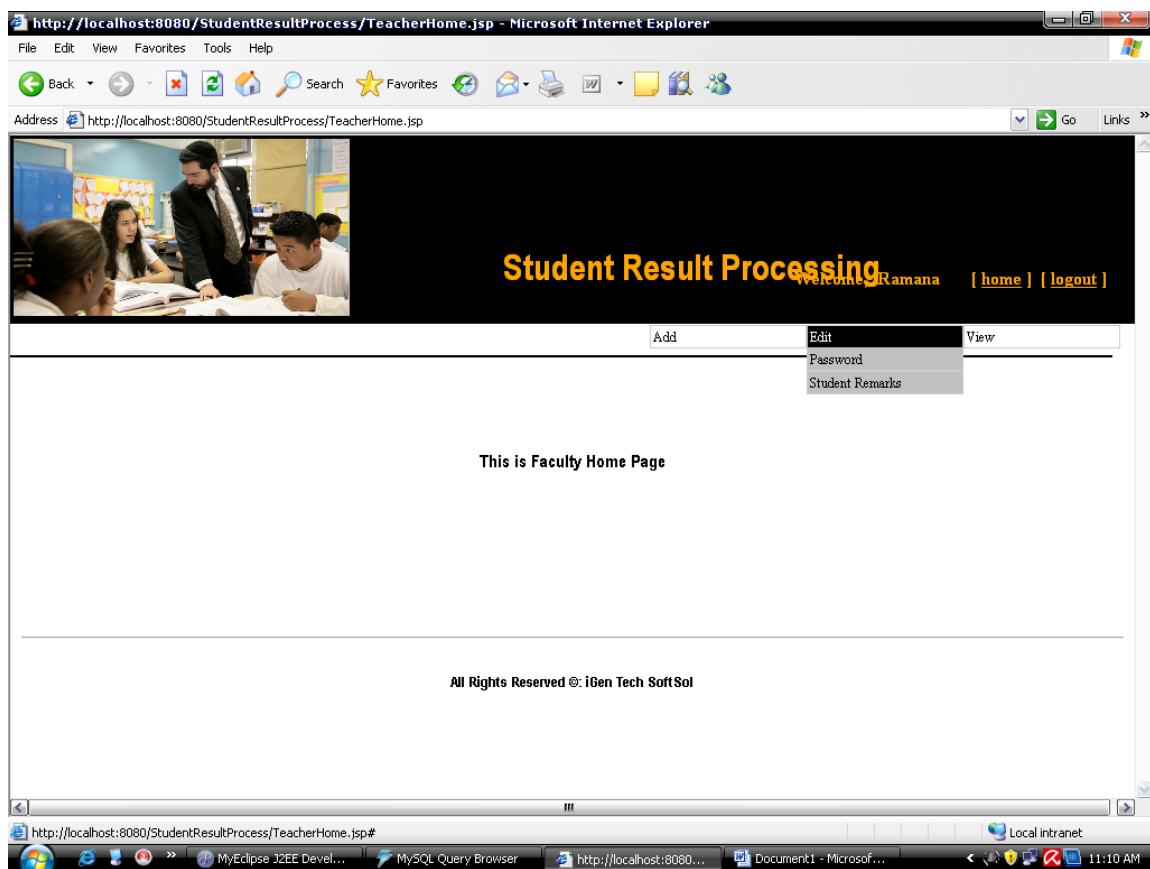


Fig. 7.17

View Details

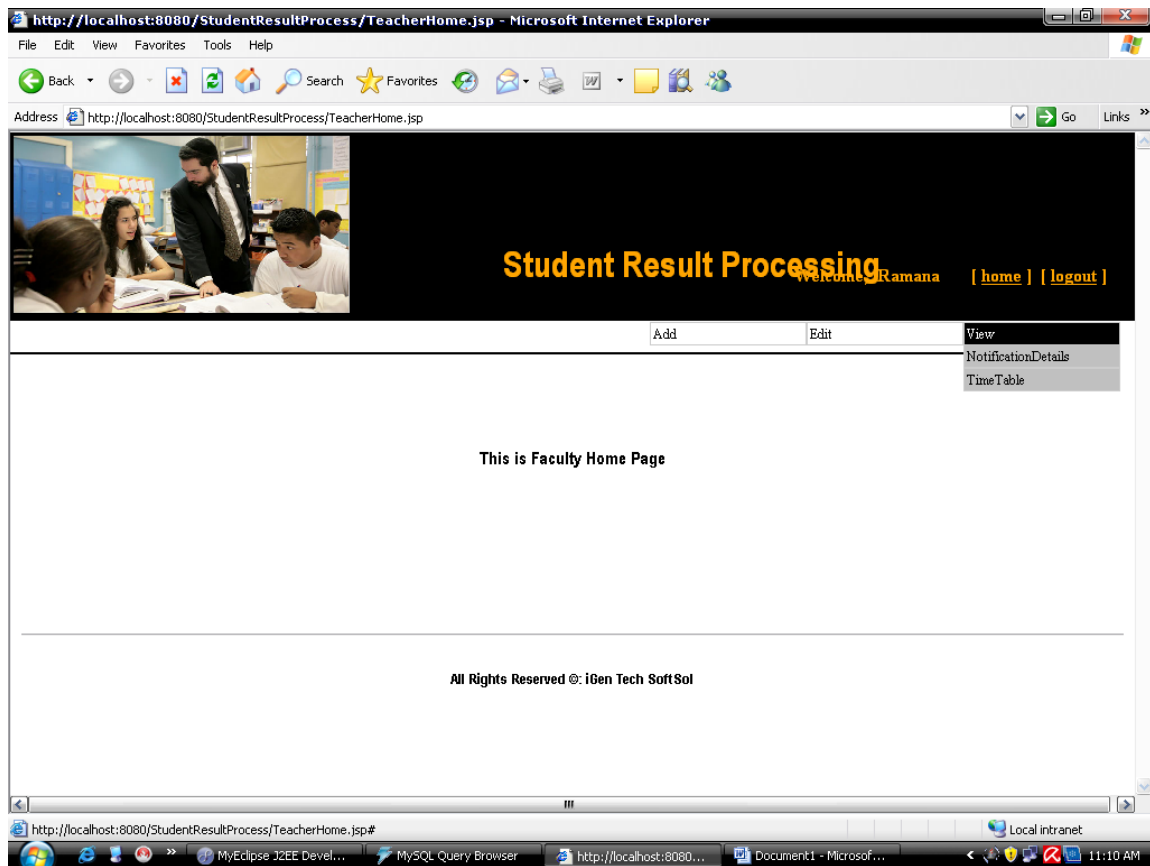


Fig. 7.18

Add Report card Details:

The screenshot shows a web browser window with the address `http://localhost:8080/StudentResultProcess/AddReportCardDetails.jsp`. The page has a header with a logo of students in a classroom and the title "Student Result Processing" in orange. Below the title, it says "Welcome, Ramana" and has links for "[home]" and "[logout]". There are three tabs: "Add", "Edit", and "View", with "Add" being the active tab. The main content area is titled "PUBLISH REEPORT" and contains a form with the following fields:

- Student ID:
- Exam Name:
- Actual Marks:
- Subject Name:
- Total Marks:

Below the form is an "ADD" button. At the bottom of the page, it says "All Rights Reserved © iGen Tech SoftSol". The Windows taskbar at the bottom shows the time as 11:11 AM and includes icons for MyEclipse J2EE Development, MySQL Query Browser, and a Microsoft Word document.

Fig. 7.19

Student Remarks

http://localhost:8080/StudentResultProcess/AddStudentRemarks.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Home Print Mail News RSS Feeds

Address http://localhost:8080/StudentResultProcess/AddStudentRemarks.jsp Go Links

Student Result Processing Welcome Ramana [home] [logout]

Add Edit View

ENTER STUDENT REMARKS

Student ID: Stud1

Enter Remarks: etfhhfhjhj

Date:

ADD

Calendar - Microsoft Int...

June 2009

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Enter Year: 1980 Get

All Rights Reserved © iGen Tech SoftSol

Done Local intranet 11:12 AM

Fig. 7.20

Change Password

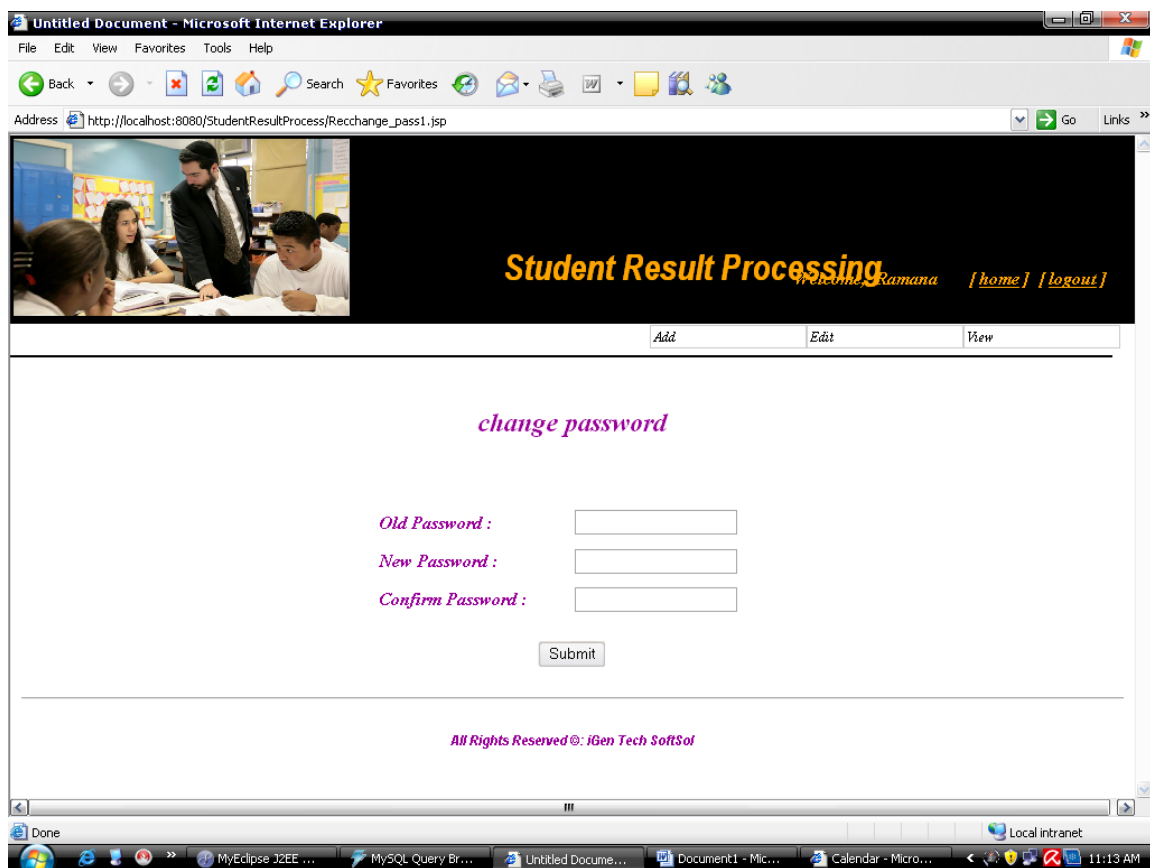


Fig. 7.21

Student Remarks

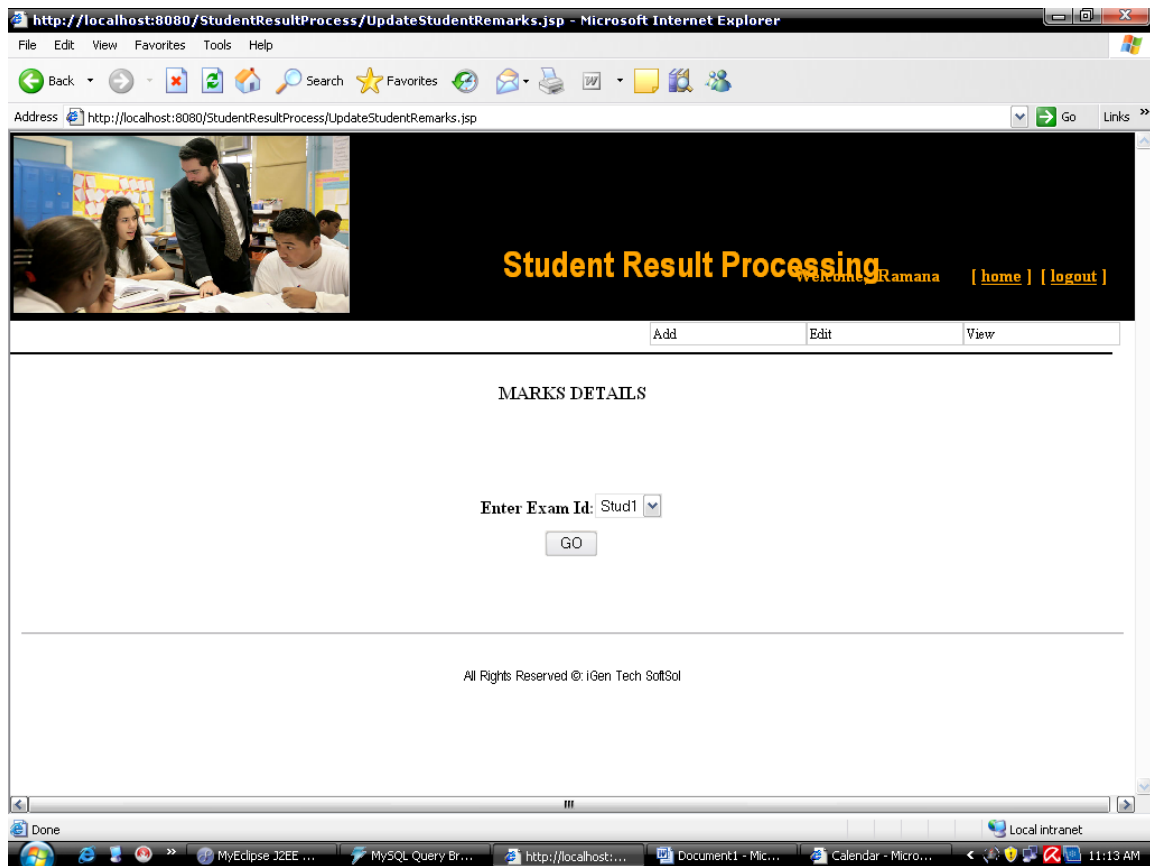


Fig. 7.22

Notification Details

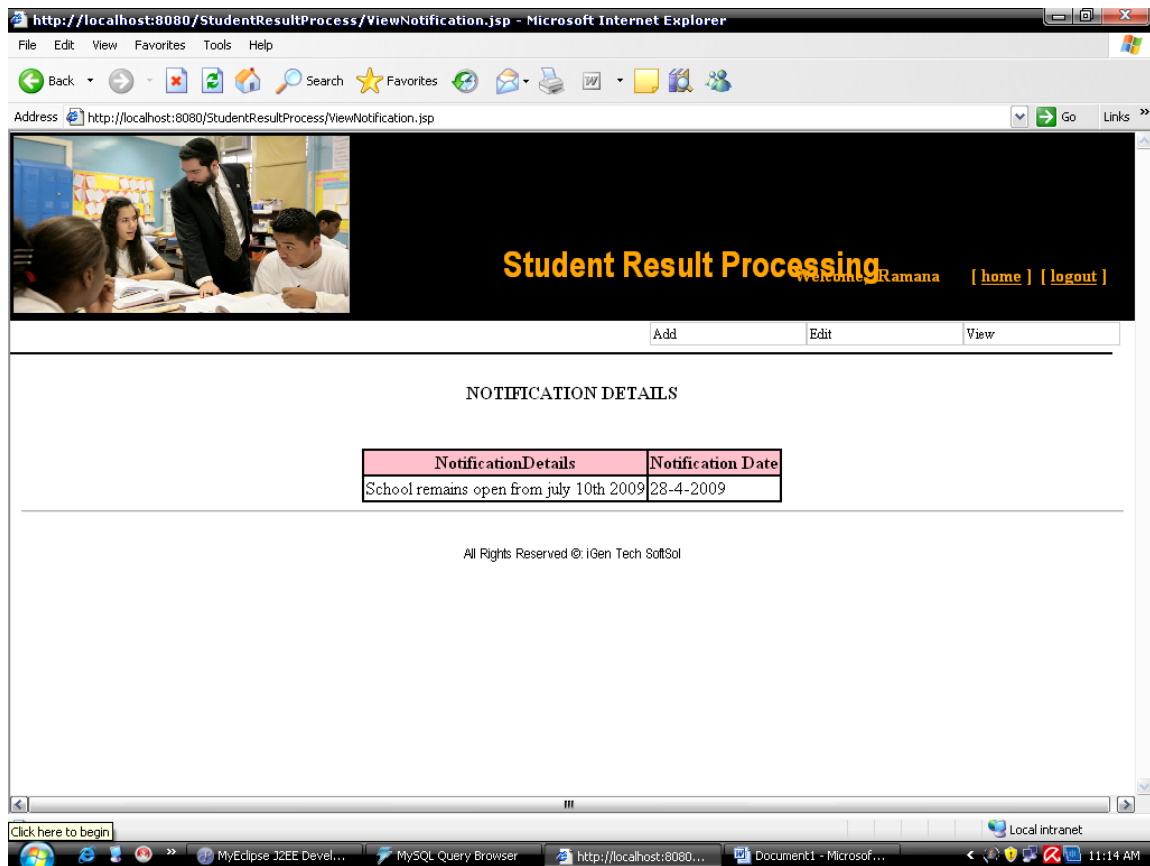


Fig. 7.23

View time Table Details

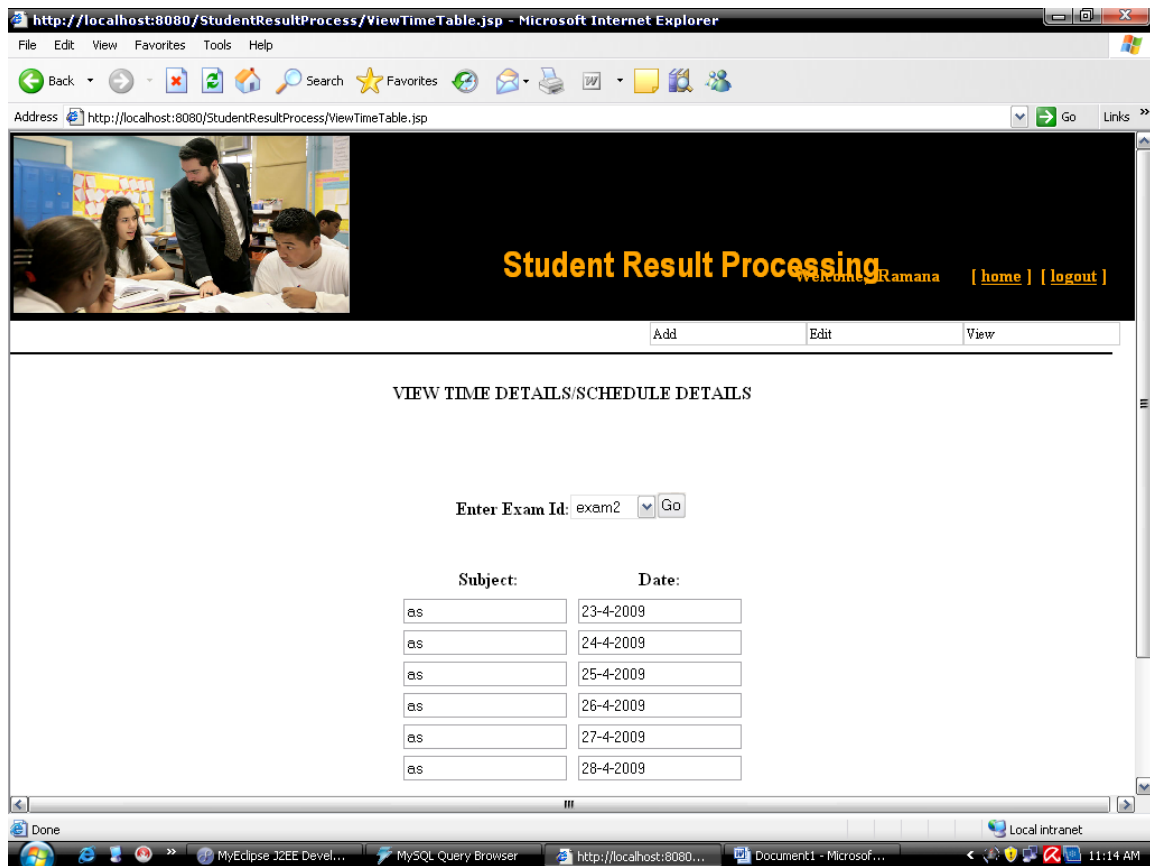


Fig. 7.24

Log Out

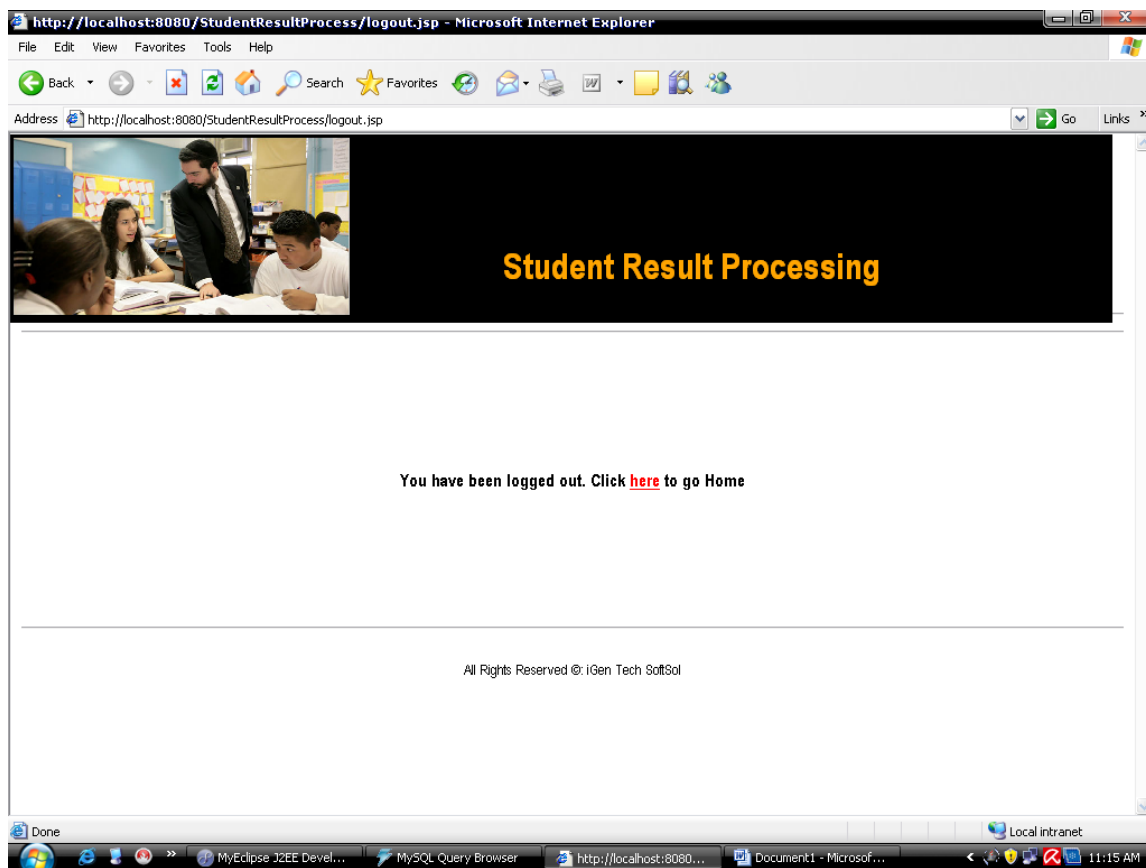


Fig. 7.25

8.

CONCLUSION

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.

9. FUTURE ENHANCEMENTS

Now the developed System is a web based system, it gives all the student details. In the future the results can be directly printed, and this functionality can be made available to the user. This can also be enhanced by giving the user more services such as aggregate calculation etc.,

9.REFERENCES

- Herbert Schildt, Java Complete Reference, 5th edition
- Dietel and Dietel, Java How To program
- Pressman, Software Engineering, 4th edition
- Raghurama Krishnan, Database Management Systems

- <http://www.java.sun.com>
- <http://www.oracle.com>