

DISTRIBUTED GRADING SYSTEM

A Project Report Submitted to the Acharya Nagarjuna University, Guntur



For the partial fulfillment of requirements for the Award of the Graduation

BACHELOR OF SCIENCES

By

A. S .L. Prasanna (Y193016044)

D. Pavanisahithya (Y193016052)

B. Renuka (Y193016045)

K. P. kumari (Y193016057)

CH. Rajarajeswari (Y193016050)

M. Kavya (Y193016059)

16-3-22 To 18-5-22

Under the Internal Guidance of

N. RAJYA LAKSHMI

Asst. Professor Dept. of Computer Science



Y.A. GOVT. DEGREE COLLEGE FOR WOMEN

CHIRALA - 523155

**Y.A GOVT. DEGREE COLLEGE FOR WOMEN
CHIRALA – 523155**



CERTIFICATE

This is to certify that it is a bonifide work entitles "**DISTRIBUTED GRADINGSYSTEM**" have one by "**DIVVELA. PAVANISAHITHYA (Y193016052)**". In the academic year 2021 – 2022 in partial fulfillment of the requirements for the award of degree of **Bachelor of Sciences** in the Department of **Computer Science**.

Project Guide

N. RAJYA LAKSHMI, M.Tech.,
Dept. of Computer Science
Y.A. Govt. Degree College For Women,
Chirala.

Head of the Department

N. RAJYA LAKSHMI, M.Tech.,
Dept of Computer Science,
Y.A. Govt. Degree college for Women,
Chirala.

Signature of the External

Signature of the Internal

Principal
Y.A. Govt. Degree College for Women
CHIRALA, Praksam Dist., (A.P.)

ACKNOWLEDGEMENT

I sincerely thank the following distinguished personalities who have given their advice and support for successful completion of this work.

I would like to thank **Smt. Dr. CH. RAMANAMMA**, Principal, **Y.A. Govt. Degree College for Women, Chirala**, for providing the Online Resources and allowed to carry out this Work.

I Extend my Sincere thanks to **Smt. N. RAJYA LAKSHMI**, Head of the Dept. for her kind Co-operation and providing the Required resources.

I am very thankful to my project guide **Smt. N. RAJYA LAKSHMI**, Asst. Professor, Dept. of computer Science, **Y.A. Govt. Degree College for Women, Chirala**, for his Co-operation and inspiring guidance and immense help during the course of Project.

I Extend our Sincere thank to all other teaching and non-teaching staff of the department and who helped directly or indirectly for their co-operation and encouragement.

This Project is a proud presentation of my work to the greatness of our professors and friends who have encouraged me all through the duration of project.

D. Pavanisahithya

DIVVELA. PAVANISAHITHYA

(Y193016052)

DECLARATION

This is to state that the dissertation titled “**DISTRIBUTED GRADING SYSTEM**” carried out at the Department of Computer Science, **Y.A. Govt. Degree College for Women, Chirala** is based on the original work carried by me under the guidance of **Sri. N. RAJYA LAKSHMI, Asst. Professor** towards the partial fulfillment of requirements for award of **B.Sc.** This Project work has not been submitted to any other University or Institution either in part or full towards any other degree up to my knowledge and belief.

Date: 22-06-2022
Place: Chirala

D. Pavanisahithya
Project Associates:
D. PAVANI SAHITHYA
(Y193016052)

ABSTRACT

The main objective of the project is to distribute the task across multiple computers in lieu of using a standalone computer for giving grades to the students.

Sri Harsha Degree College is a college located in Hyderabad. It has various branches in Hyderabad. Presently the college is following manual method for maintaining student details and processing the results of the students.

The college is interested in computerizing the details of the students. As there is more number of students located in various branches it is not efficient to maintain the database and process the logic in a single server. To avert this, the method we have implemented is the distributed environment to enhance the performance.

The company mainly deals with the following aspects under a computerized environment

- It allows the user to maintain the entire details of the students including his examination details. This is done using a simple method.
- It helps in processing the results of the students in a distributed environment.

Presently the above mentioned activities are being done manually; it has become a hazardous task for the management of the college. So the college entrusted this project work to AXEND LABS at Hyderabad. Some part of this work has been taken up by us for preparing a web – enabled system.

CONTENTS

Page No

Chapter-1: Introduction

1.1. Background.....	1
1.2. Existing System.....	2
1.3. Proposed System	2
1.4. Team Setup.....	2

Chapter-2: System Analysis

2.1. Conversion.....	5
2.2. Development Environment.....	6
2.3. Execution Environment.....	6
2.4. Analysis Diagrams	6
2.5. Tables/Entities in the Database.....	14
2.6. Description of Interfaces.....	15

Chapter-3: Design Aspects

3.1. Hierarchy Chart	16
3.2. Brief Description of Databases in various Modules.....	16
3.3. Table with Sample data.....	17
3.4. Table giving Reports.....	18
3.5. Interfaces (Blank Forms)	21

Chapter-4: Walkthrough

4.1. Screens	24
--------------------	----

Chapter-5: Conclusion:

5.1. Goals that are met.....	34
5.2. Goals that are not met	34
5.3. Concluding Remarks.....	34

Chapter -1

Introduction

1.1. Background:

Sri Harsha Degree College is a college located in Hyderabad. It has various branches in Hyderabad. Presently the college is following manual method for maintaining student details and processing the results of the students.

The college is interested in computerizing the details of the students. As there is more number of students located in various branches it is not efficient to maintain the database and process the logic in a single server. To avert this, the method we have implemented is the distributed environment to enhance the performance.

The company mainly deals with the following aspects under a computerized environment

- It allows the user to maintain the entire details of the students including his examination details. This is done using a simple method.
- It helps in processing the results of the students in a distributed environment.

Presently the above mentioned activities are being done manually; it has become a hazardous task for the management of the college. So the college entrusted this project work to AXEND LABS at Hyderabad. Some part of this work has been taken up by us for preparing a web – enabled system.

THE AXEND LABS, Hyderabad:

AXEND LABS was found in 1994, AXEND LABS is organized as three strategic Business units catering to Software Development, Software services and data managements and training areas. The company is steered by a highly motivated team of technologists with solid software development experience.

The company is steered by a high performing and young board of directors, all of who have an average software industry experience of 8 years. The management is result oriented, which is achieved by consensus. Key issued is brainstormed and an optimal solution ensured. All the business functions, the strategic business units and related responsibilities are shared among the directors thus ensuring smooth and efficient business process.

DISTRIBUTED GRADING SYSTEM

AXEND LABS is leading provider of Client/Server solutions for small to medium corporations. For existing client/server installations Inforise offer solutions that are web-centric in tune with the changing needs of corporation. So to say, at this very moment you needs of existing client server solution to support web integration, Inforise start with understanding your information needs to provide you a combination of tailor-made off-site and on-site solutions instantly leveraging the advantages of the Internet. Inforise client and server and web bridging solutions help you to get incrementally migrate to the new network centric computing paradigm.

AXEND LABS implemented mini client/server applications for a variety of business processors and developed software libraries for finance, inventory, sales orders, purchases, manufacturing, human resources and production process in an enterprise.

1.2. Existing System:

The following are the various problems in the existing system.

- Maintaining the operations in stand alone system has become a tedious task,
- In the existing system a user cannot send two or more requests at a time.
- It is difficult to maintain the more number of calculations in a single computer, as it shows a huge impact on the performance.

1.3. Proposed System:

The below listed are the objectives of the proposed system.

- Development of user friendly system, which can use in globally across multiple computers.
- The main advantage in this proposed system is communication of two or more systems simultaneously.
- Easy of updating the calculations.

The maintenance can be done more efficiently and effectively.

1.4. Team Setup:

The organization has formed a group of programmers with one project manager, one senior programmer and a trainee to solve the problem. I have taken part in some aspects of the project i.e. designing and coding few interfaces including documentation and preparing user manuals.

Chapter -2 System Analysis

The requirement has been investigated by a team of experts and the whole system is as described below:

Preliminary Study:

This section describes the architecture of the system. The diagram below shows the three hosts that were part of the demonstration, as well as the processes running on each host. The diagram also shows the communication between the various processes. The dotted lines indicate the communication a process has with the rmiregistry, while the solid lines show the actual remote method invocation.

There are also a number of UML diagrams attached that show the overall design of the system. The diagrams show the inheritance relationships between the classes. There is also an event diagram attached that shows the order of operations while the system is running.

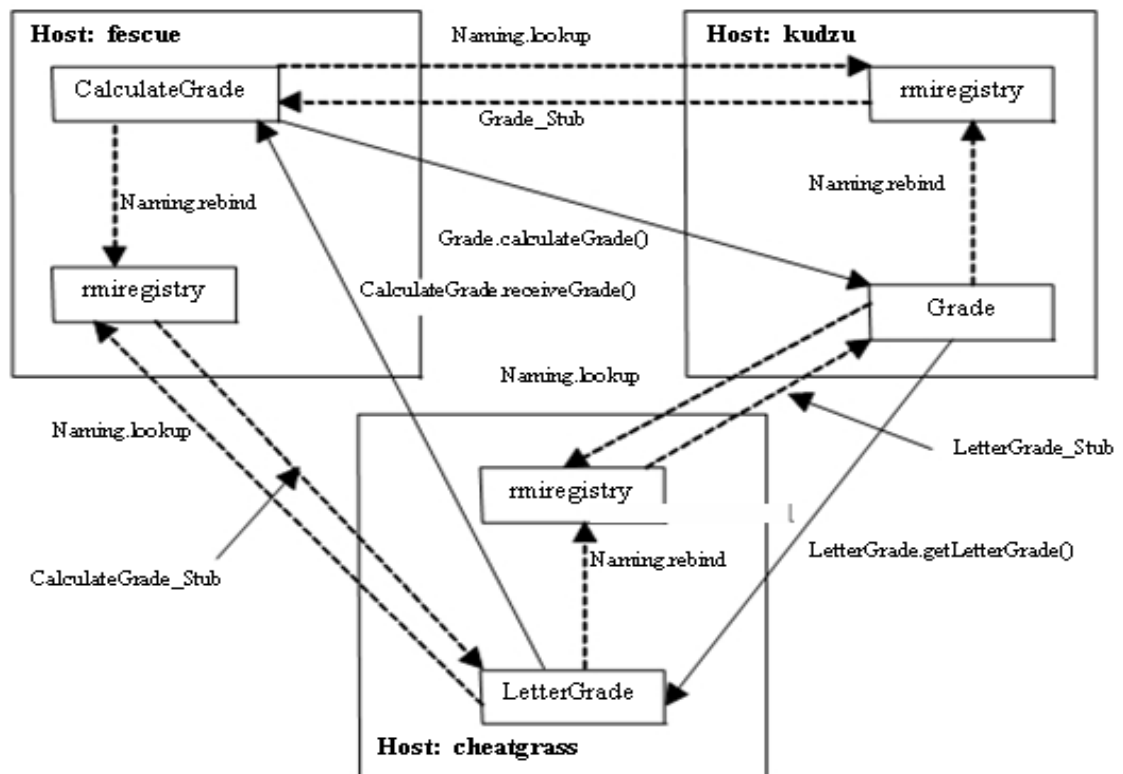


Figure 1. 1 Architecture of Distributed Grading System

DISTRIBUTED GRADING SYSTEM

The system consists of three processes running on different machines. The first process, called CalculateGrade, runs on fescue. The first thing this process does is it registers with the naming registry on the local host. It does this so one of its remote methods, receiveGrade, can be called remotely. The process then gets a reference to the Grade object by performing a lookup on the rmi registry running on kudzu. It then invokes the calculateGrade method on the remote object, passing as a parameter an instance of the RemoteFile class.

After calling this method, the CalculateGrade process waits to receive a final letter grade from the third process, which is called LetterGrade. Once it receives that grade, it prints the grade out to the screen.

The Remote File class is used to simulate a file. RMI requires that an object that is a parameter to a remote method must be serializable.

This allows the object to be transferred from one virtual machine to another. Since input streams aren't serializable, a new object had to be created to simulate a file when it's passed to a remote method. The remote file class does just that. It allows the programmer to easily send the contents of a file to a remote method, so the file can be processed on a remote machine. The constructor of the class takes a string as a parameter. The string indicates the name of the file that will be used. The constructor reads the contents of the file into a byte array. Any method that operates on the remote file class can then call the getInputStream function. This function returns an input stream to the byte array that stores the contents of the original file. Once a process has the input stream, it can operate on it just as it would for a normal file. So the RemoteFile object makes it appear to the application that the store locally.

The second process, called Grade, registers with the naming registry on kudzu and then waits for a client to call one of its methods. The method that the CalculateGrade process calls takes a RemoteFile object as an input parameter. The remote file contains a number of integers, which represents grades that a student has accumulated in a course. The process reads in the contents of the file, and calculates the average of these grades. When that is complete, the process gets a reference to the LetterGrade object by performing a lookup on the rmi registry running on cheatgrass. The reference is then used to call the getLetterGrade method of the LetterGrade object. This method takes one parameter, which is the average that was just calculated.

DISTRIBUTED GRADING SYSTEM

The third process, called LetterGrade, registers with the naming service running on cheatgrass and then waits for a client to call one of its methods. The method that the grade process calls takes one parameter, which represents the final average of a student. This process then assigns a LetterGrade for that average. So if the average is an 85, this process will assign grade of 'B'. Once the LetterGrade is determined, the process calls the receive method of the CalculateGrade process, which is running on the first machine. This method takes one parameter which is the final lettergrade.

2.1. Conversion of the Statement of Problem:

In this project, the main aim is, each and every server must be communicating with all other servers. Here mainly one objective is available.

Main Module:

In this the administrator is given to a server to perform some request operations onto different servers which are distributed globally. Here the constraint is these distributed servers are by default connected to LAN or WAN and some default security levels.

In this project there are two modules. The main emphasis is on the second module and it is the main basic objective of the project. The brief description of both the modules is as follows.

Overview of Modules:

Module 1: This module is very simple i.e. it interacts with the database to store, retrieve and manipulate the data in the database. In this module we concentrate mainly on the maintaining the student details in the database.

Module 2: This module is given to a server to perform some request operations onto different servers which are distributed globally. Here the constraint is these distributed servers are by default connected to LAN or WAN and some default security levels.

DISTRIBUTED GRADING SYSTEM

2.2. Development Environment:

Tools Used/ Required

Platform	: Windows 2000/XP
Server	: Apache Tomcat 5.5
Database	: Oracle 10g
Database Connectivity	: JDBC
Front end	: JAVA Technologies (JDK 1.4) [i.e., Servlets/JSP, JavaBeans,RMI], HTML
Scripting Technology	: Java Script
Editor	: MyEclipse 5.0 IDE

2.3. Execution Environment:

Hardware

Pentium processor with 800 MHz. speed 256 MB RAM.
40 GB Hard Disk.
NIC card with Network accessibility.

Software

The system must also contain
Oracle Database to be installed, configured
Apache server 5.5 be installed, configured and
The Application need to be deployed onto the Server.

2.4. Diagrams

a) Dataflow Diagrams:

Data flow diagram is a structural analysis tool that is used for graphical representation of Data processes throughout any organization. The data flow approach emphasis is on the logic underlying the system, by using combination of only 4 symbols. It follows a top down approach. A full description of a system actually consists a set of DFD's, which comprises of various levels. And initial overview model is exploded to lower level diagrams that show additional features of the system. Further each process can be broken down into a more detailed DFD. This is a repetitive process until sufficient details are described.

DISTRIBUTED GRADING SYSTEM

CONTEXT Level DFD

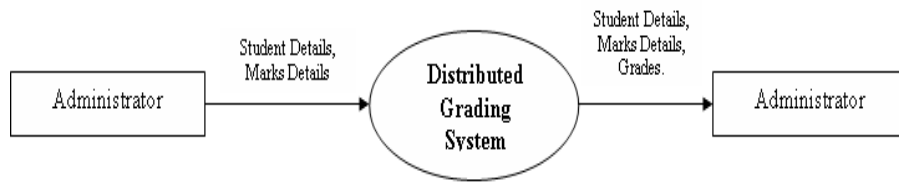


Figure 2. 1 Zero Level DFD of the Distributed Grading System

First Level DFD

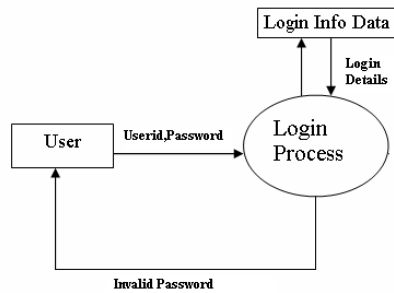


Figure 2. 2 First Level DFD for Login Process

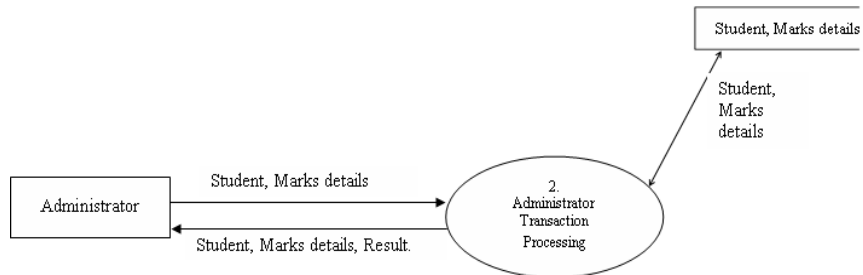


Figure 2. 3 First Level DFD for Administrator Activities

DISTRIBUTED GRADING SYSTEM

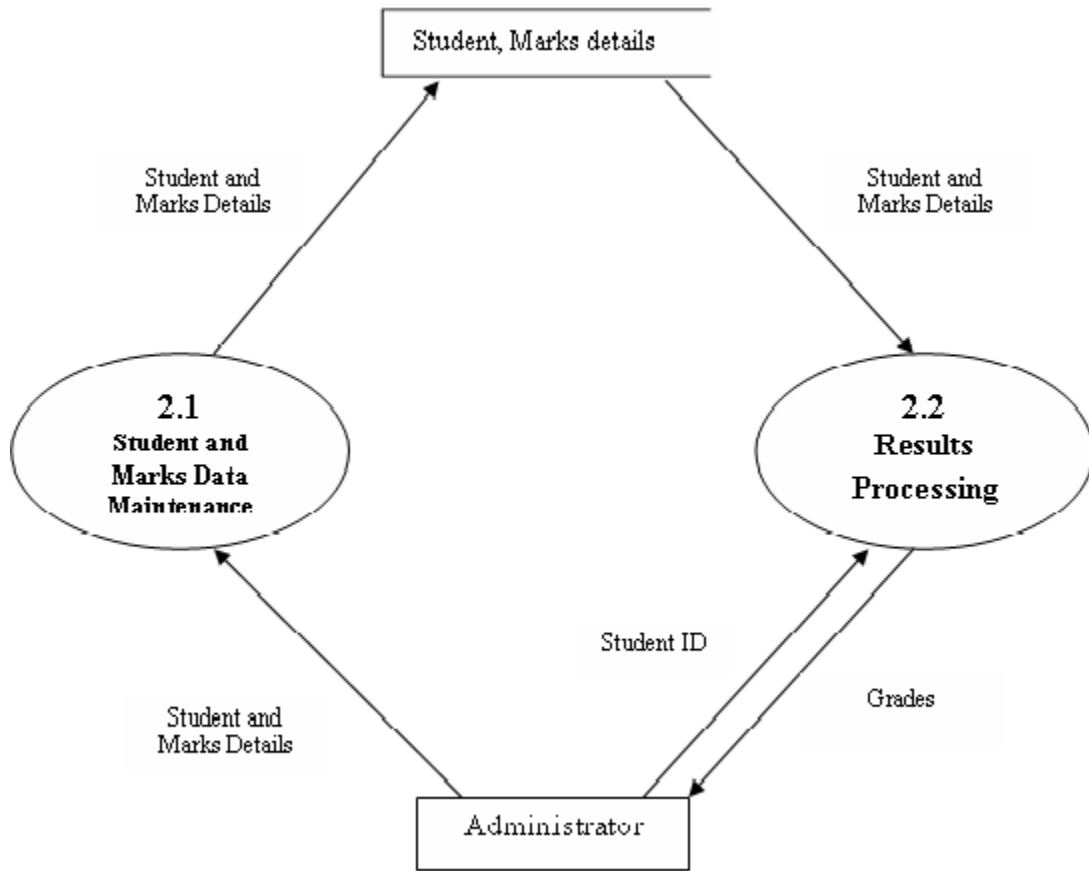


Figure 2.3. 1 Second Level DFD for Administrator

b).Entity-Relationship Diagrams

ER (Entity-Relationship) Diagram is used to represent the relationship between entities in the Database.

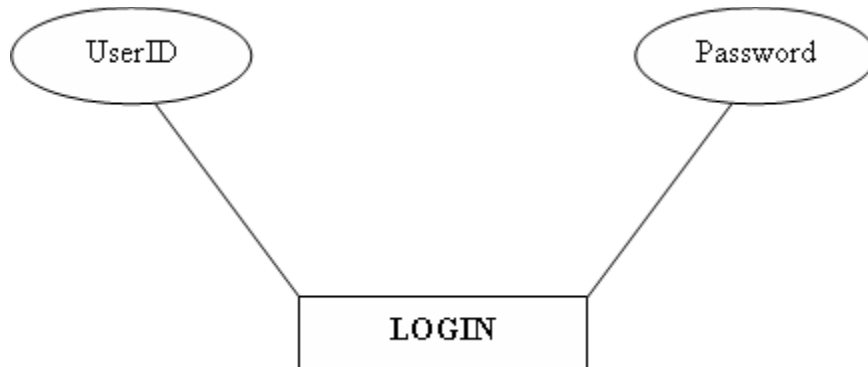


Figure 2. 4 ER- Diagram1

DISTRIBUTED GRADING SYSTEM

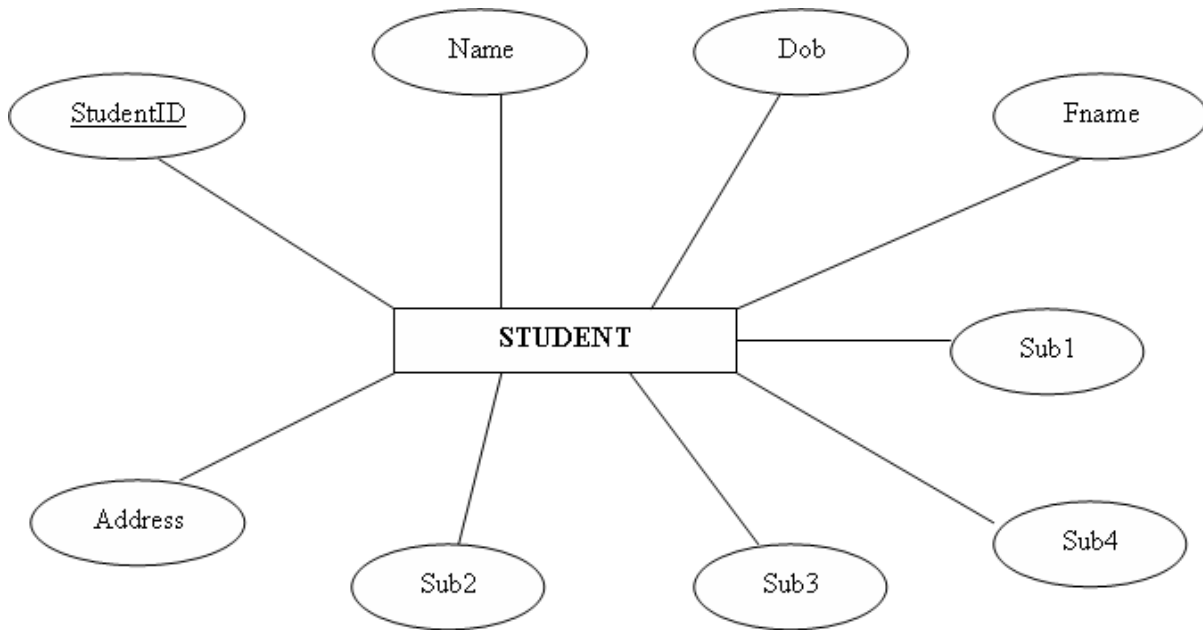


Figure 2. 5 ER – Diagram 2

C). UML Diagrams

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

UML is a notation that resulted from the unification of Object Modeling Technique and Object Oriented Software Technology .UML has been designed for broad range of application.

DISTRIBUTED GRADING SYSTEM

i). UseCase Diagram:

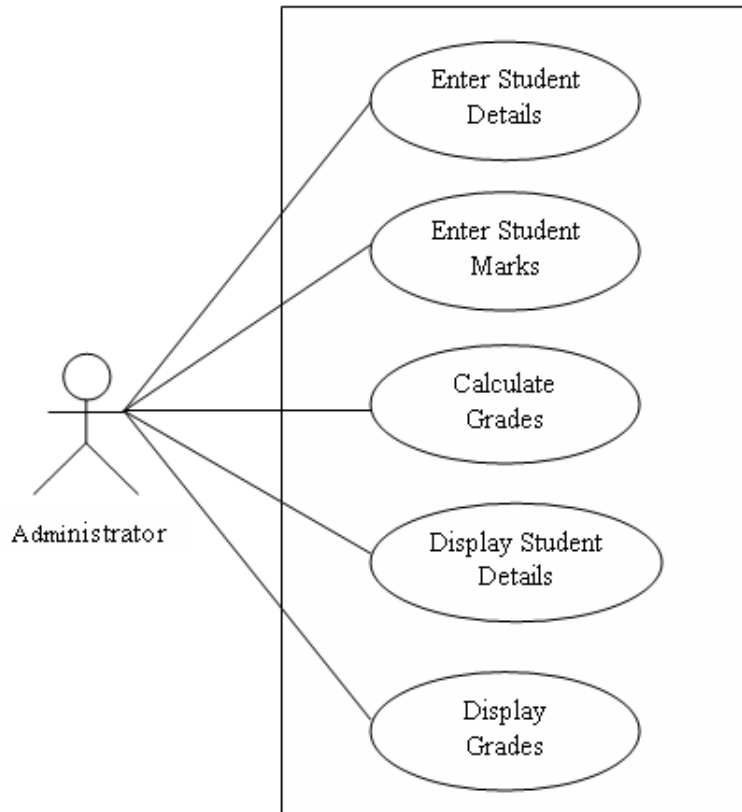


Figure 2. 6 UseCase Diagram for Administrator

DISTRIBUTED GRADING SYSTEM

ii). Sequence Diagram:

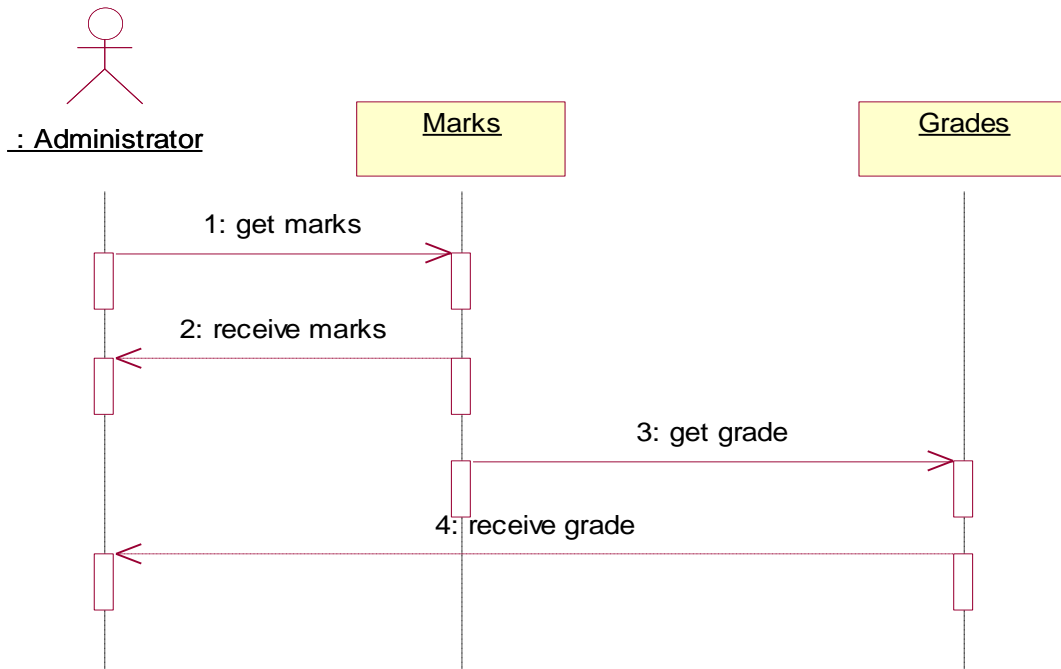


Figure 2. 7 Sequence Diagram for Administrator

DISTRIBUTED GRADING SYSTEM

iii). Collaboration Diagram:

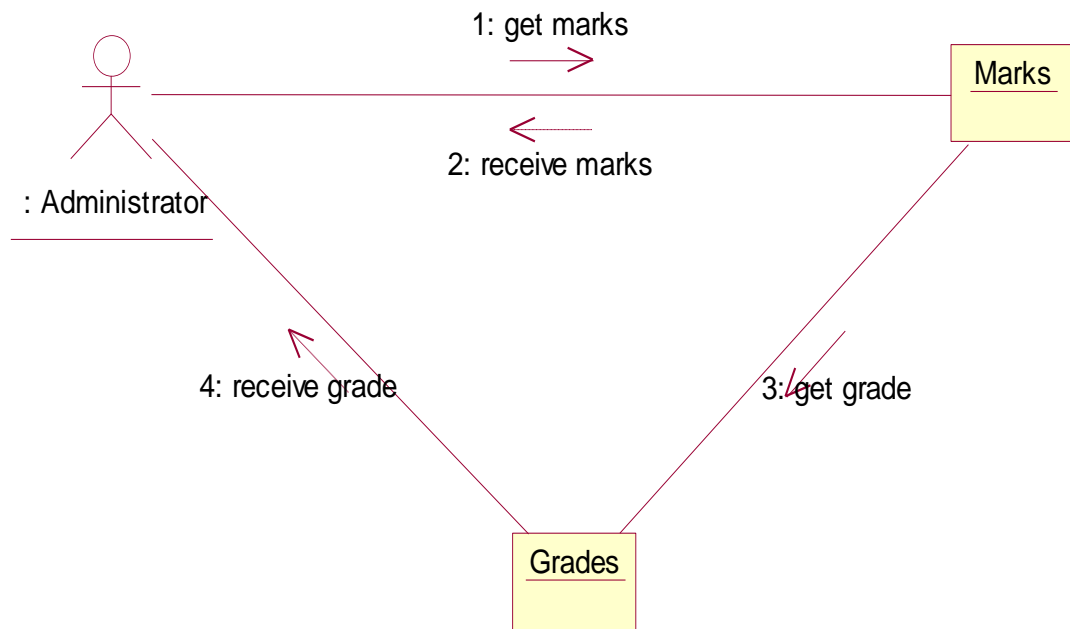


Figure 2. 8 Collaboration diagram for Administrator

DISTRIBUTED GRADING SYSTEM

iv). Class Diagram:

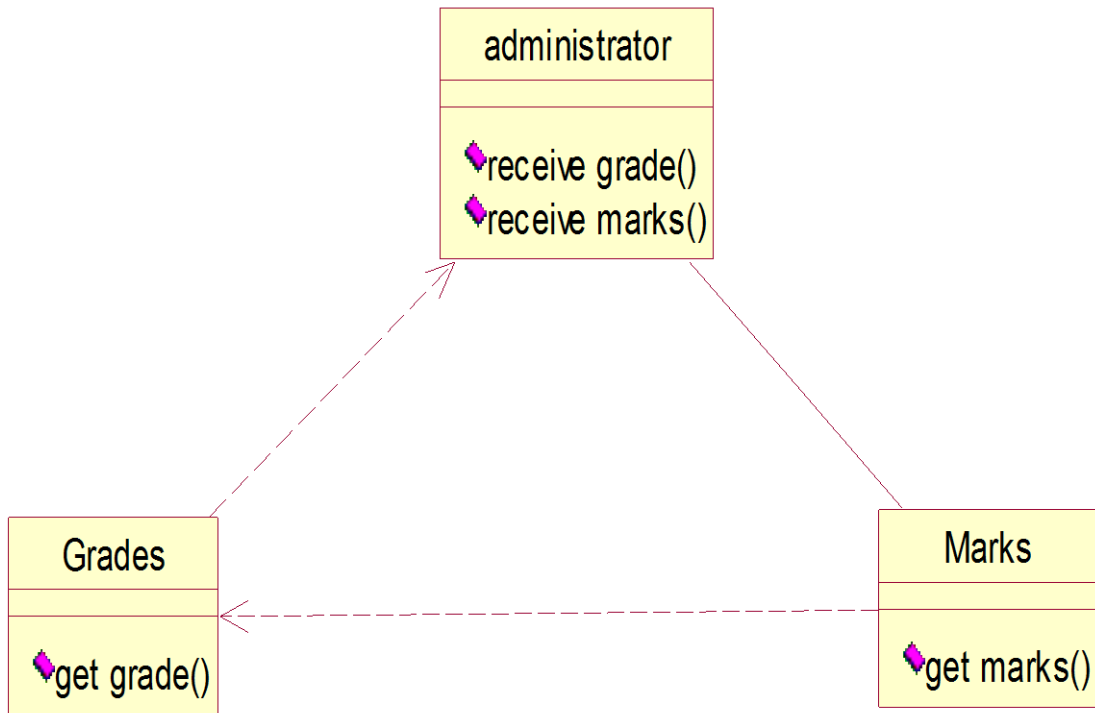


Figure 2. 9 Class Diagram for Administrator

DISTRIBUTED GRADING SYSTEM

2.5. Data Dictionary:

Relation Name	Attribute	Type	Size	Constraints
Login	userId	Number	4	NOT NULL
Login	Password	Text	10	
Student	StudentID	Number	4	NOT NULL
Student	Name	Text	15	NOT NULL
Student	DOB	Date		
Student	Fname	Text	15	
Student	Address	Text	40	
Marks	StudentID	Number	4	NOT NULL
Marks	Sub1	Number	3	NOT NULL
Marks	Sub2	Number	3	NOT NULL
Marks	Sub3	Number	3	NOT NULL
Marks	Sub4	Number	3	NOT NULL

2.6. Description of Interfaces:

- i. Front Tier:** This project is purely a web application. Each and every user expects a user friendly and attractive interface. In order to develop the interface we have uses HTML, Jsp. The HTML language is static and it contains lot of built-in controls which are very helpful in designing the interface, so this has been used to develop the attractive environment. Jsp is used for the dynamic activity that takes place at the time of execution.
- ii. Communication Tier:** For any application it is mandatory to have a communication path between Front -end and Back-end. The Concept of JDBC is used for this specific purpose and the concept of RMI (Remote method Invocation) is used from invoking the methods located on the remote computer as this application is developed for distributed environment. The brief description of these concepts has been discussed in the following sections.
- iii. Language Chosen:** As this being a web application there is possibility to use the same application in the variety of platforms. So the application should be platform independent. As we know that the JAVA is platform independent we have chosen JAVA language because of its platform independency nature. This language has enormous concepts which are helpful in developing the application in the distributed environment. So we have chosen JAVA language for developing this application.
- iv. Users of the System:** Presently the user of the system is only the administrator. The main intention here is to extend more number of users including the staff, students located at various numbers of branches.

Chapter -3
Design Aspects

3.1. Hierarchy Chart:

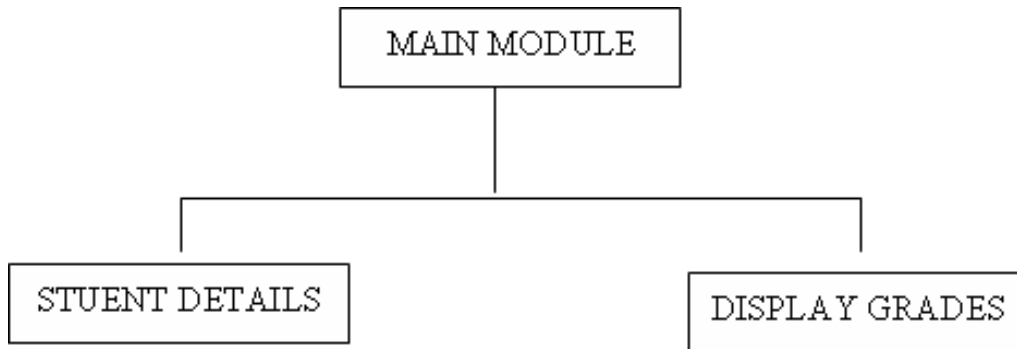


Figure 3. 1 Hierarchy Chart for MainModule

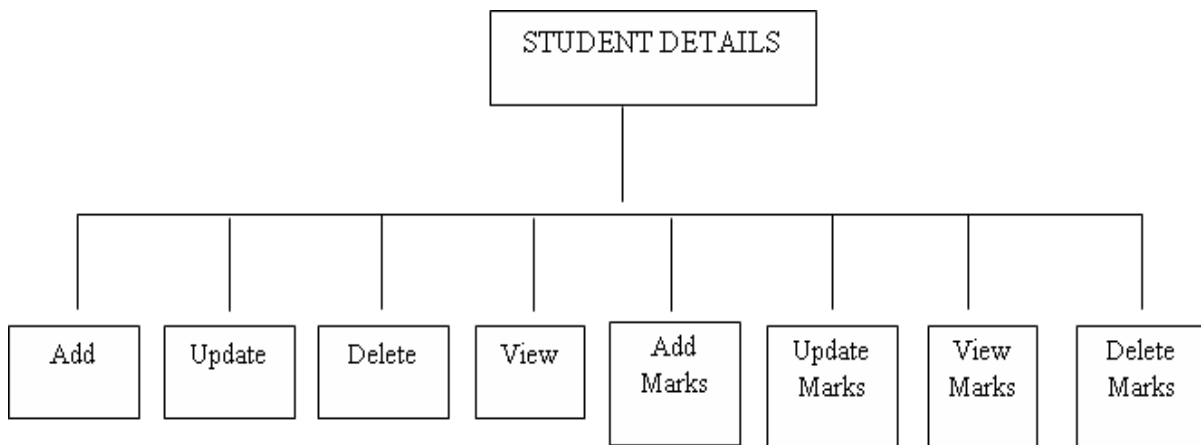


Figure 3. 2 Hierarchy Chart for StudentDetails Module

3.2. Brief Description of Database in Various Modules:

S.No.	Name of Table	Type of Table	Purpose
1.	Login	Independent	This table is used to maintain the list of authorized users.
2.	Student	Independent	This table stores the entire details of the student.
3	Marks	Dependent	The Marks details are maintained here and this references the Student table.

DISTRIBUTED GRADING SYSTEM

3.3. Tables giving Test Data:

i). Login

USERID	PASSWORD
-----	-----
101	AAAA
102	BBBB
103	CCCC
104	DDDD
105	EEEE

ii). Student

STUDENTID	NAME	DOB	FNAME	ADDRESS
-----	-----	-----	-----	-----
201	Ramesh	05-07-1985	Kumar	Newpet,Kuppam
202	Sanjeev	09-10-1984	Rajesh	MainRoad,Rayachoti
203	Babu	17-01-1985	Prasad	Newpet,Kuppam
204	Suresh	12-10-1984	Sreekanth	Newpet,Kuppam
205	Uday	19-06-1985	Sunil	Newpet,Kuppam

iii). Marks

STUDENTID	SUB1	SUB2	SUB3	SUB4
-----	-----	-----	-----	-----
201	15	45	54	43
202	54	85	87	71
203	75	45	39	43
204	56	57	68	75
205	92	75	54	81

3.4. Testing Aspects:

Testing is the process of executing the program with the intent of finding errors. It is a major quality measure employed during software development. During testing, the program is executed with a set of test cases and the output of the program is verified, the test cases is evaluated to determine if the program is performing as it is expected to perform. A set of test inputs execution conditions and expected results is said to be a test case.

A good test case is one that has a high probability of detecting an as yet undiscovered error. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

Psychology of Testing

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't works. Testing is the process of executing the program with the intent of finding errors. Testing is properly viewed as the destructive process of trying to find the errors (whose presence is assumed) in the program.

Testing Objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally we can say,

- Testing is the process of executing a program with the intent of finding errors.
- A successful test is one that uncovers as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

Levels of Testing

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development, they are

DISTRIBUTED GRADING SYSTEM

1. Unit Testing

Unit testing focuses on verification of the building blocks or the units i.e. the modules. Using procedural design as a guide, important control paths are tested. All units of RMS were successfully tested.

a) Black-Box Testing

Knowing the specific function that a product/ module has been designed to perform, test can be designed to demonstrate the operational validity of each function. At the same time searching for errors in each function it enables us to derive sets of input conditions that will fully exercise all functional requirements for a program. It attempts to find errors falling in the following categories.

- Incorrect and missing functions.
- Interface errors.
- Errors in data structures or database access.
- Errors in function logic.

Tests are designed to answer the following questions:

- How is the functional validity tested?
- What classes of input we make good test cases?
- Is the system particularly sensitive to certain input values?
- How are the boundaries of data classes isolated?
- What data rules data volume can the system tolerate?
- What effects is specify combination of data have on the system?

All the function in the system is tested in the order they appear in the process specification diagram and the hierarchy chart and the designed test cases are tested.

System testing strategies have the following genetic characteristics:

- Testing begins at module level and works outward toward integration the entire system.
- Different testing techniques appropriate at different points in the time.

DISTRIBUTED GRADING SYSTEM

- The developer and independent test group conduct tests.
- Testing and debugging are different activities, but debugging must be accompanied in a testing strategy.
- The different test strategies include.

b). White-Box Testing:

This is performed knowing the internal working of the system. Tests are conducted to ensure that “all gears mesh”, i.e. the internal operations perform according to the specification and all internal components have adequately been exercised. This can be done by close examination of procedural detail. Providing test cases that exercise specific set of conditions and or loops test logical paths through the software.

Using the white-box testing techniques, we derive test cases that are:

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical decisions on their true or false sides.
3. Execute all the loops with in their boundaries and within their operational bounds.
4. Exercise internal data structures to assure their validity.

By the test case each of the RMS modules have been executed at least once and tested for bugs. All the logical decisions on their true or false states are executed at least once to ensure the proper working if the system.

2. Integration Testing:

Integration testing is a construction time testing which deals with interfacing irregularities. The objective is to take unit tested modules and RMS follows top down integration technique.

3. Validation Testing:

Validation testing succeeds when the system function reasonably according to the users expectations. This is achieved by a series of black-box tests that demonstrate conformity with the requirements. There are two tests conducted for the system validation.

DISTRIBUTED GRADING SYSTEM

4. Alpha Testing:

A customer conducts the test at the developers place. The software is used in a natural setting with the developer looking over the shoulder and recording errors.

5. Beta Testing:

This test is conducted at one or more user sites by the end user. Here the developer is generally absent. Therefore beta testing is a “live application of software in an environment that can’t be controlled by the developer. The customer records all the errors that surface while testing and reports them to the developer for correction

3.5. Tables giving Reports:

View all student Details: This report displays the details of all the students and this report used the student table.

View all Marks Details: This report displays the marks particulars of all the students and this report used the marks table.

Displaying Grades Report: This report displays the grades of the students in a text file. This process uses the marks table

DISTRIBUTED GRADING SYSTEM

3.6 Blank Screens / Forms:

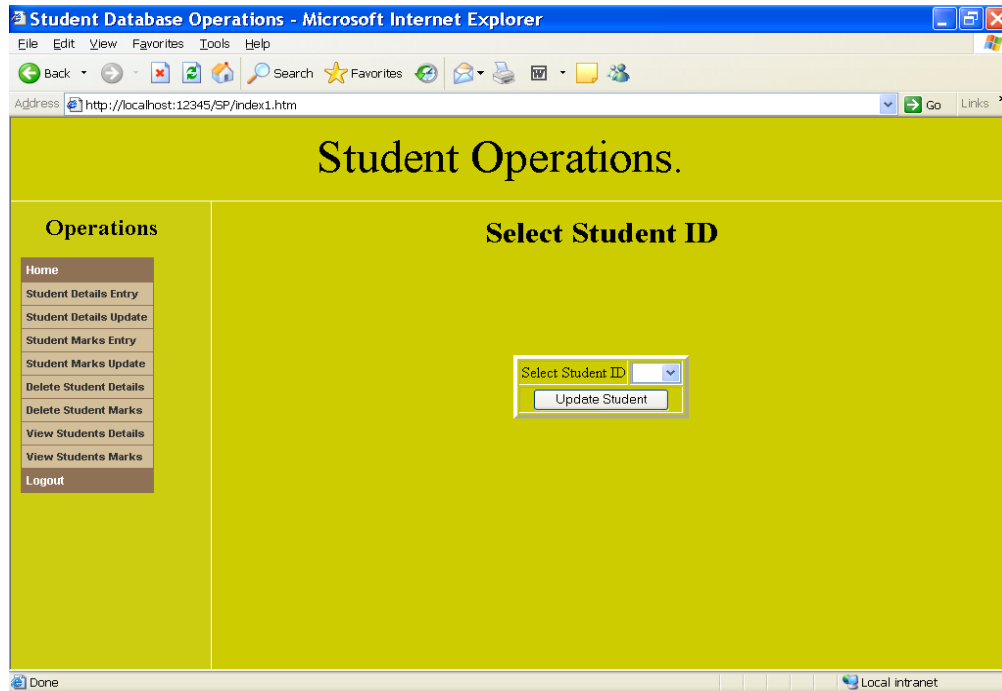
The screenshot shows a web browser window titled "Student Database Operations - Microsoft Internet Explorer". The address bar shows "http://localhost:12345/SP/index1.htm". The main content area has a yellow background and is titled "Student Operations.". On the left, there is a vertical menu under the heading "Operations" with the following items: Home, Student Details Entry, Student Details Update, Student Marks Entry, Student Marks Update, Delete Student Details, Delete Student Marks, View Students Details, View Students Marks, and Logout. The main area on the right is titled "Insert Data" and contains a form with the following fields: Student ID, Name, DOB, FName, and Address. Below these fields is a button labeled "Insert Student Data".

Screen 3.1An Interface for Entering Student Details

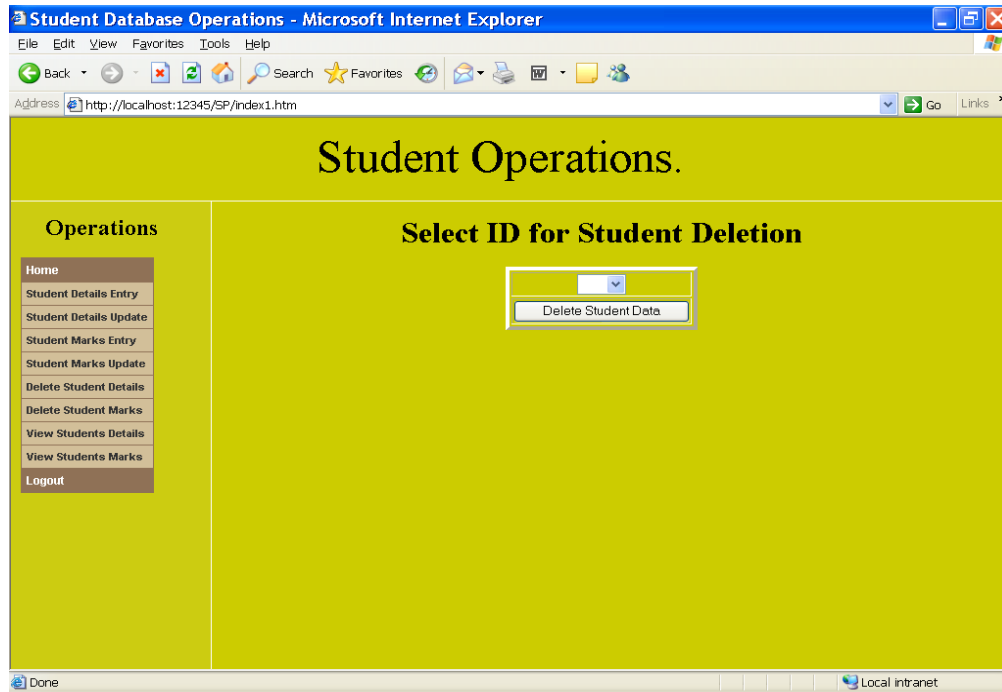
The screenshot shows the same web browser window as above. The main content area is still titled "Student Operations.". The left menu is identical. The main area on the right is now titled "Insert Student Marks" and contains a form with the following fields: Select Student ID (a dropdown menu), Sub 1, Sub 2, Sub 3, and Sub 4. Below these fields is a button labeled "Insert Student Marks".

Screen 3.2An Interface for Entering Student Marks Details

DISTRIBUTED GRADING SYSTEM



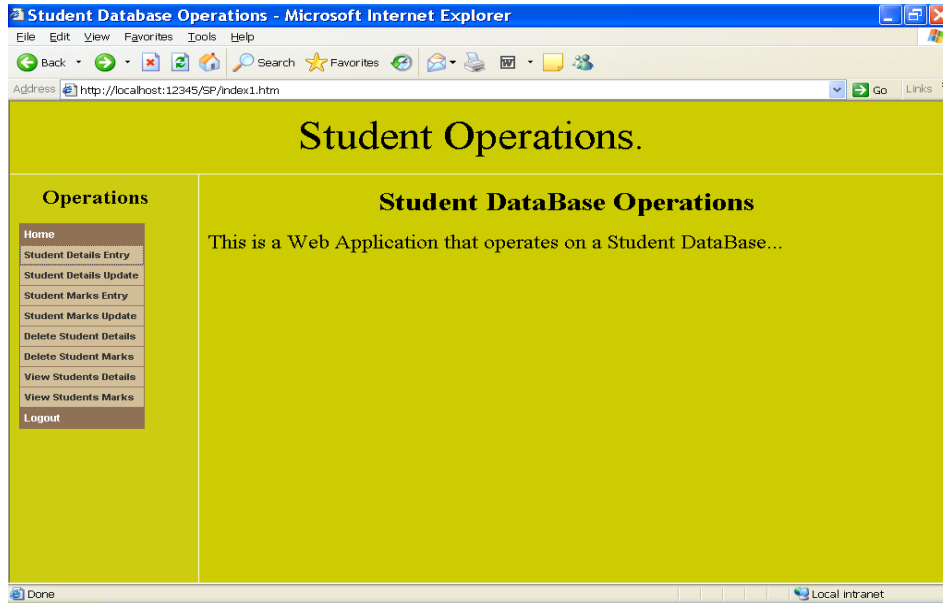
Screen 3. 3 An Interface for Updating Student Details



Screen 3. 4 An Interface for Deleting Student Details

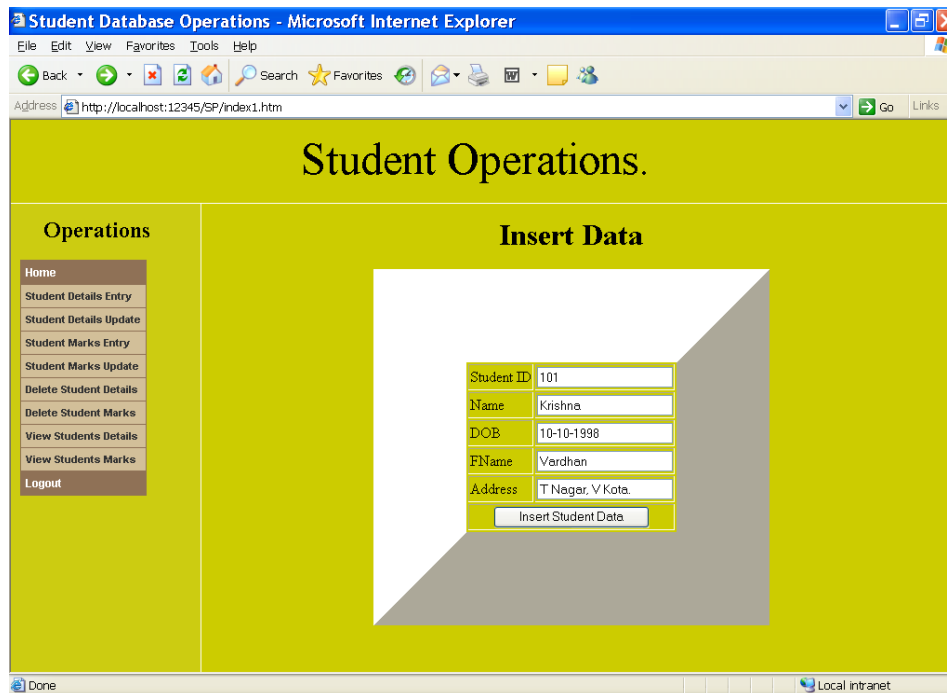
Chapter - 4 Walk Through

The following section gives an overview of interfaces in this package.



Screen 4. 1.1 Home Page of Student Operations

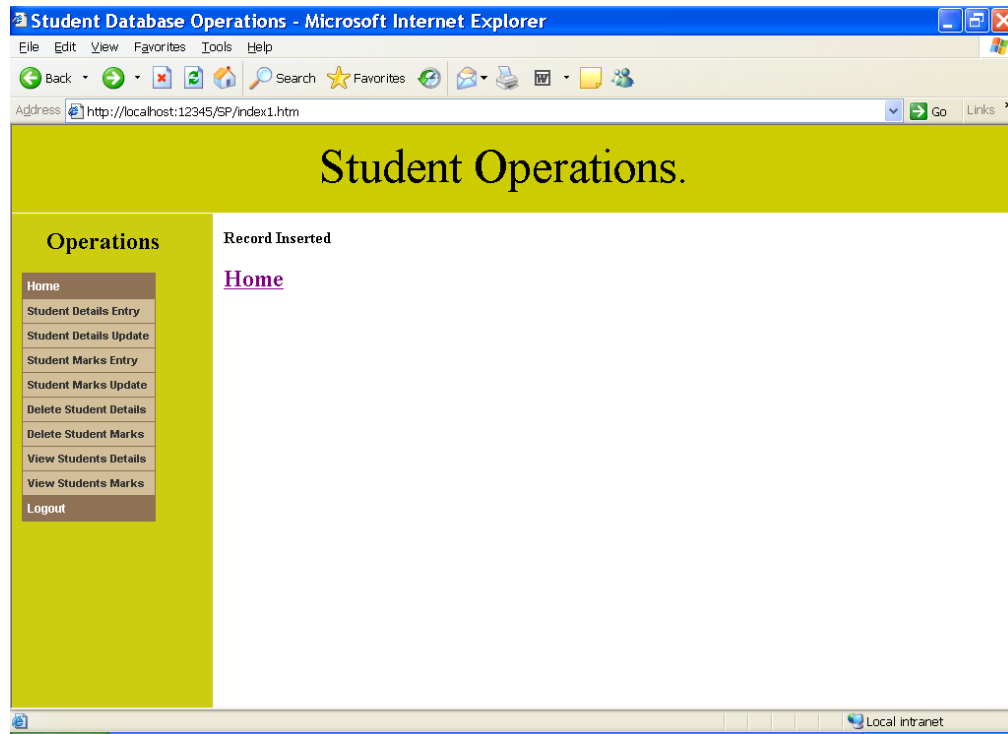
Description: This is the home page of the application. This page is displayed as soon as the project is executed.



Screen 4. 1.2 Entering Student Details

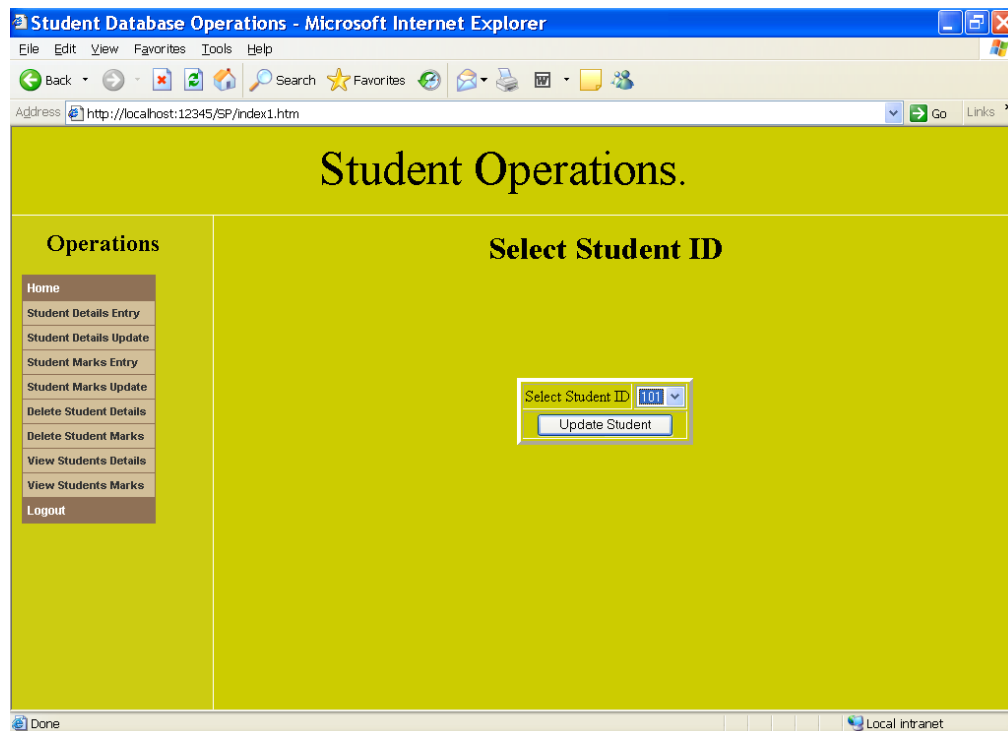
DISTRIBUTED GRADING SYSTEM

Description: This screen is used to enter the details of the student.



Screen 4. 1.2 Message stating that the Record is Inserted

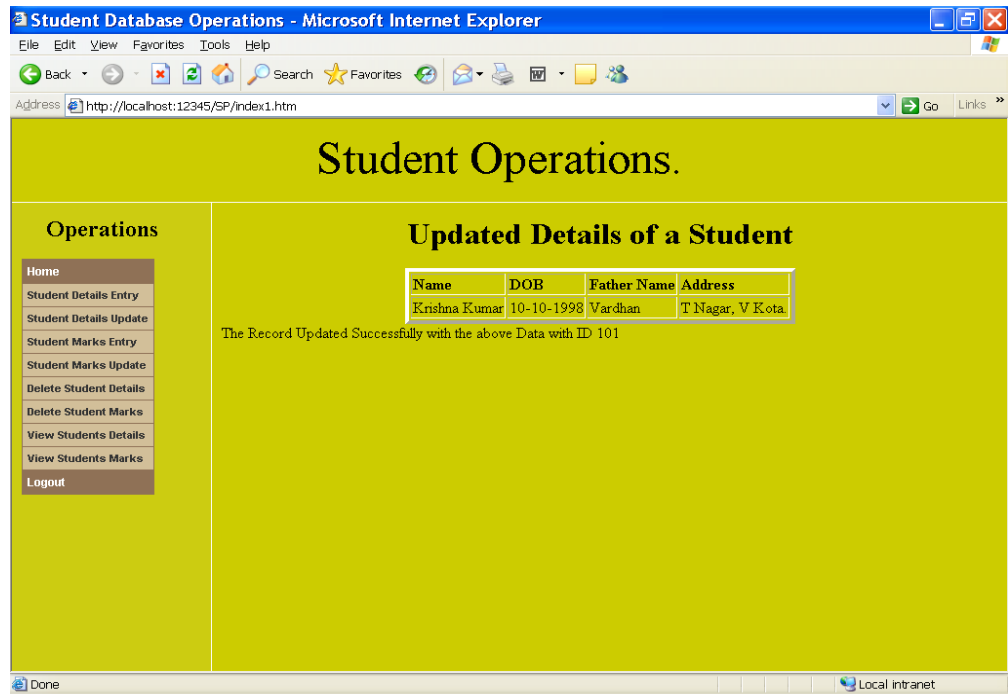
Description: This screen displays the message saying that the record has been inserted.



Screen 4.1. 3 Updating Student Details based on their StudentID

DISTRIBUTED GRADING SYSTEM

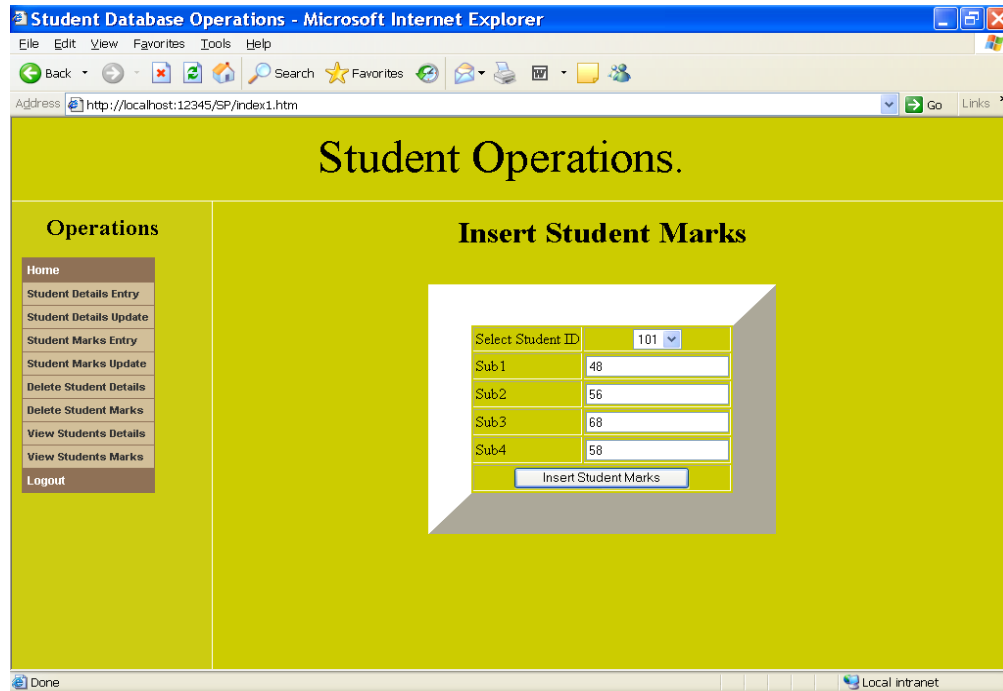
Description: This screen is for updating the details of the student. Here the user has to select the studentID to perform this operation.



Screen 4.1. 4 Updated Details of the Student

Description: This screen displays the updated information of the student as soon as the update operation is performed.

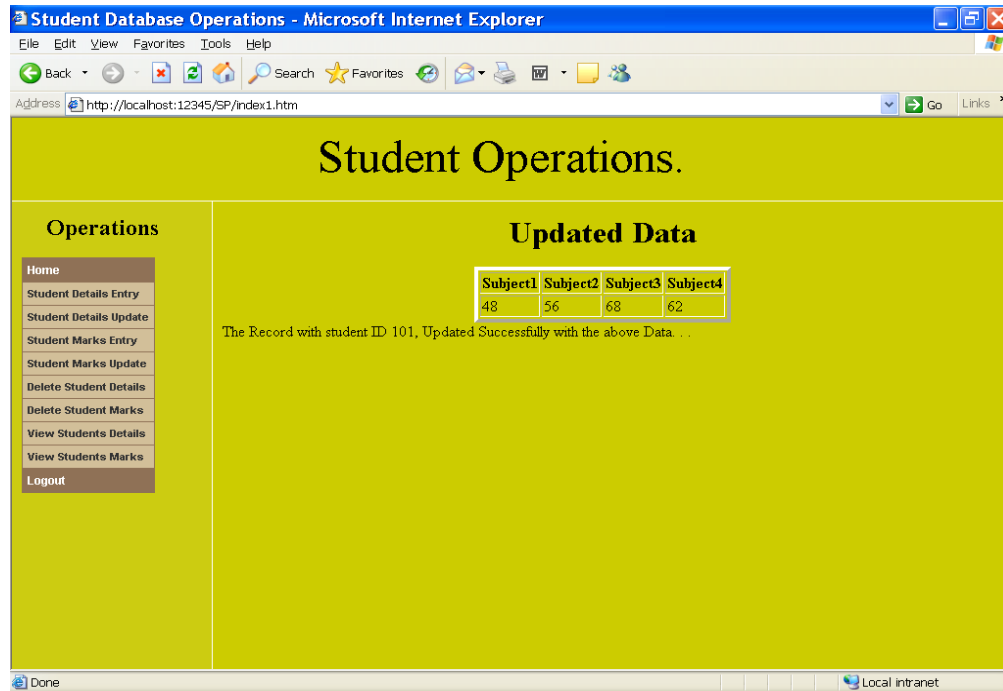
DISTRIBUTED GRADING SYSTEM



Screen 4.1. 5 Entering Student Marks Details

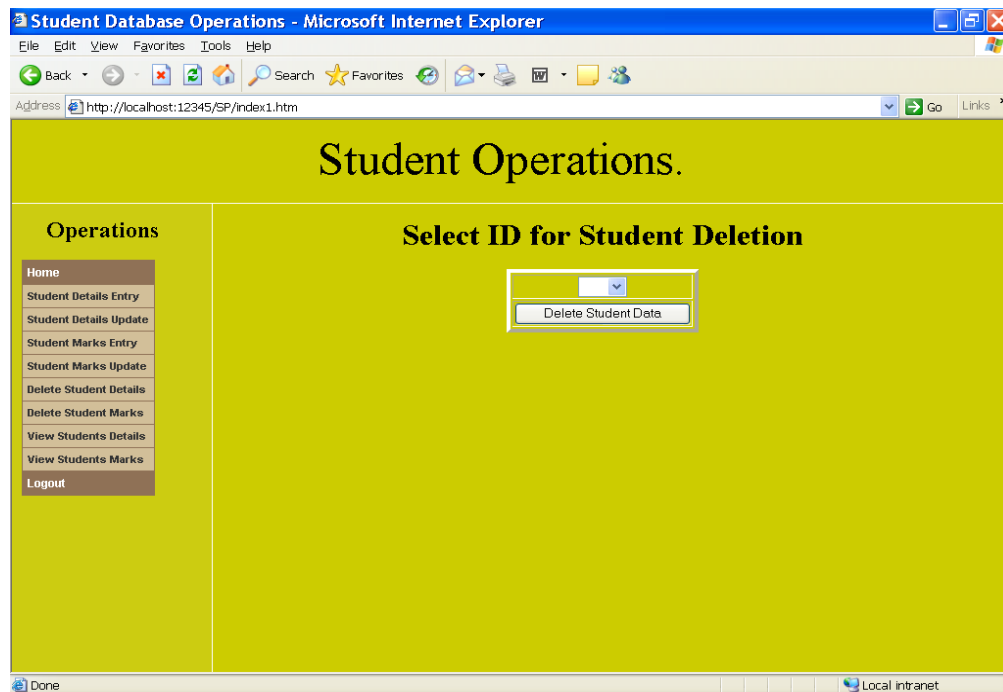
Description: This screen is for entering the marks particulars of the student. Here the user has to select the StudentID to enter the marks particulars.

DISTRIBUTED GRADING SYSTEM



Screen 4.1. 6 Updated Marks details of the Student

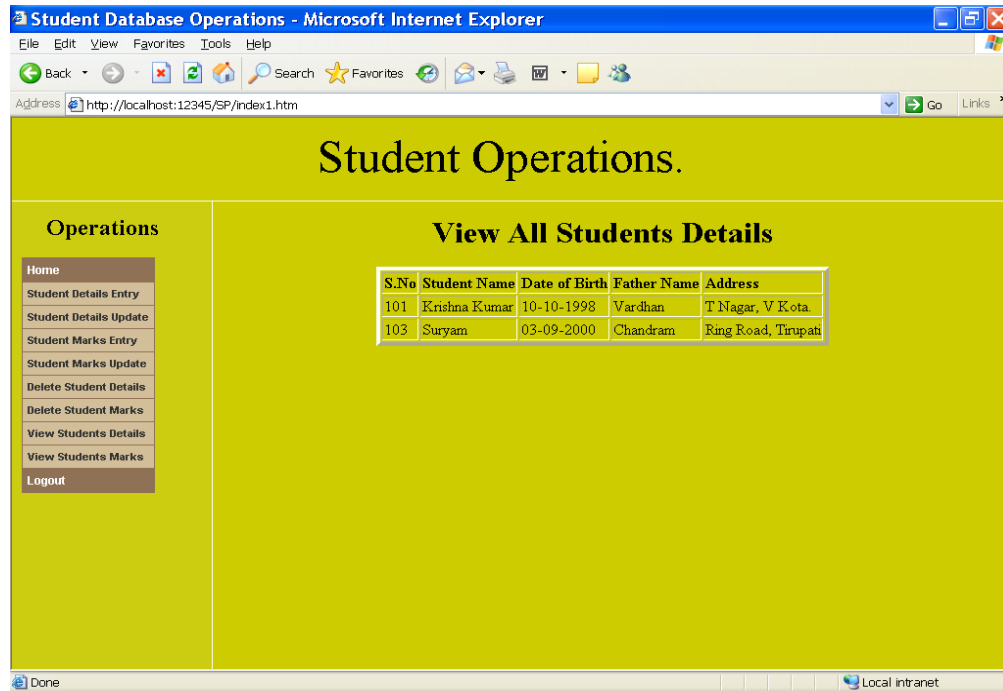
Description: This Screen displays the update Marks particulars of the student after performing the update operation.



Screen 4.1. 7 Deleting Student Details based on StudentID

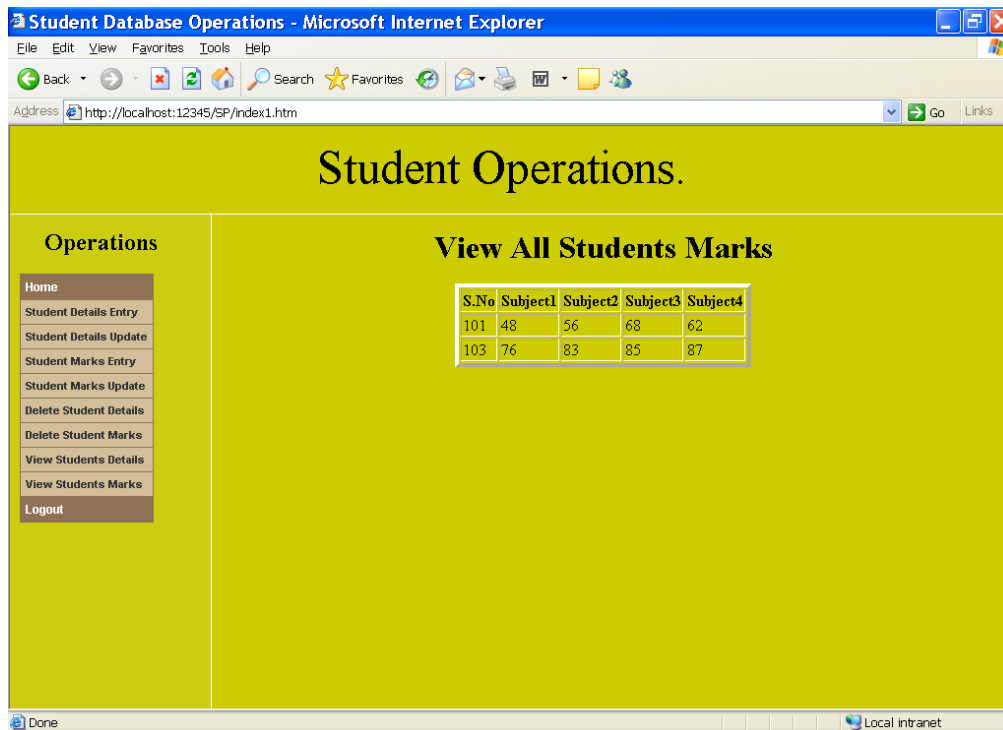
Description: This Screen is for deleting the details of the student. Here the User has to select the studentID to perform the operation.

DISTRIBUTED GRADING SYSTEM



Screen 4.1. 8 Displaying Student Details

Description: This Screen displays the details of all the students available in the database.

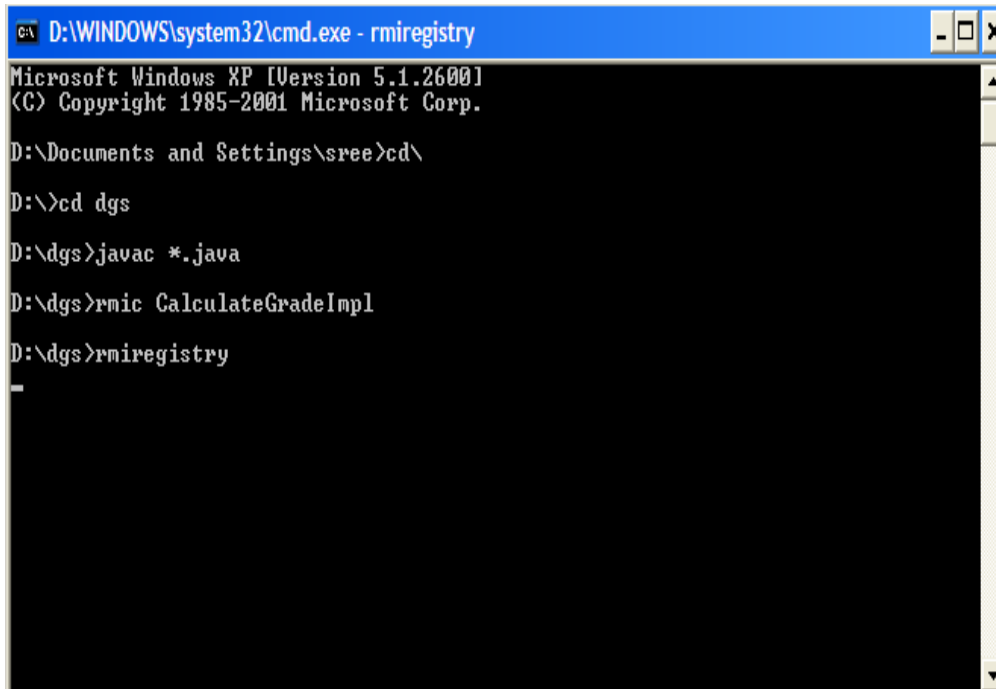


Screen 4. 9 Displaying Marks particulars

Description: This screen displays the marks details of all the students.

DISTRIBUTED GRADING SYSTEM

a). System -1: Screen 1

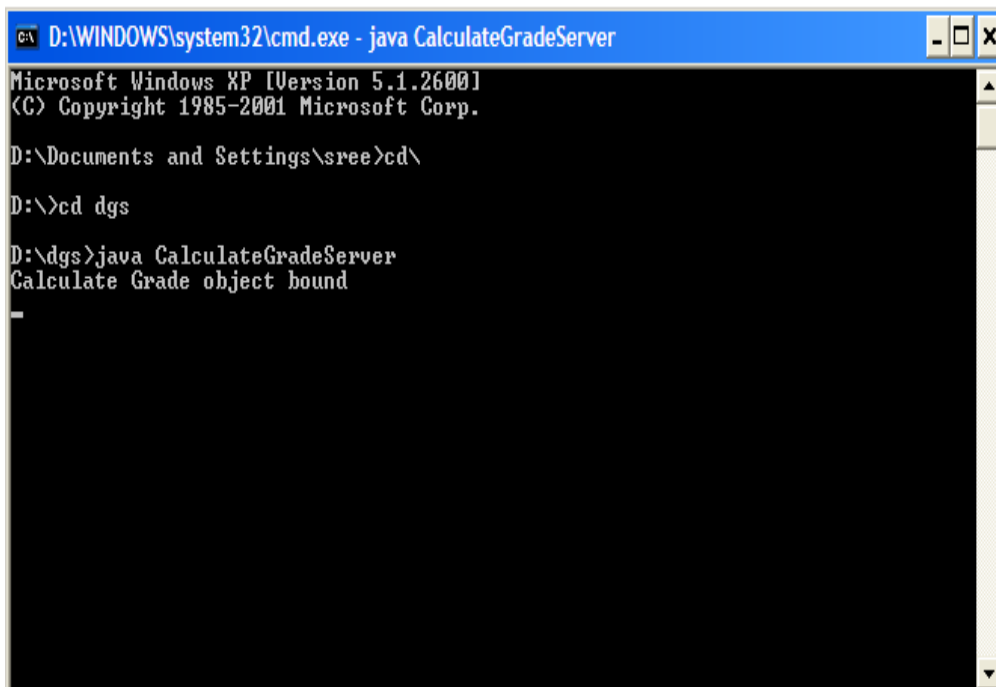


```
C:\ D:\WINDOWS\system32\cmd.exe - rmiregistry
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>javac *.java
D:\dgs>rmic CalculateGradeImpl
D:\dgs>rmiregistry
-
```

Screen 4. 10 Creating rmic for CalculateGradeServer

b). System -1: Screen2



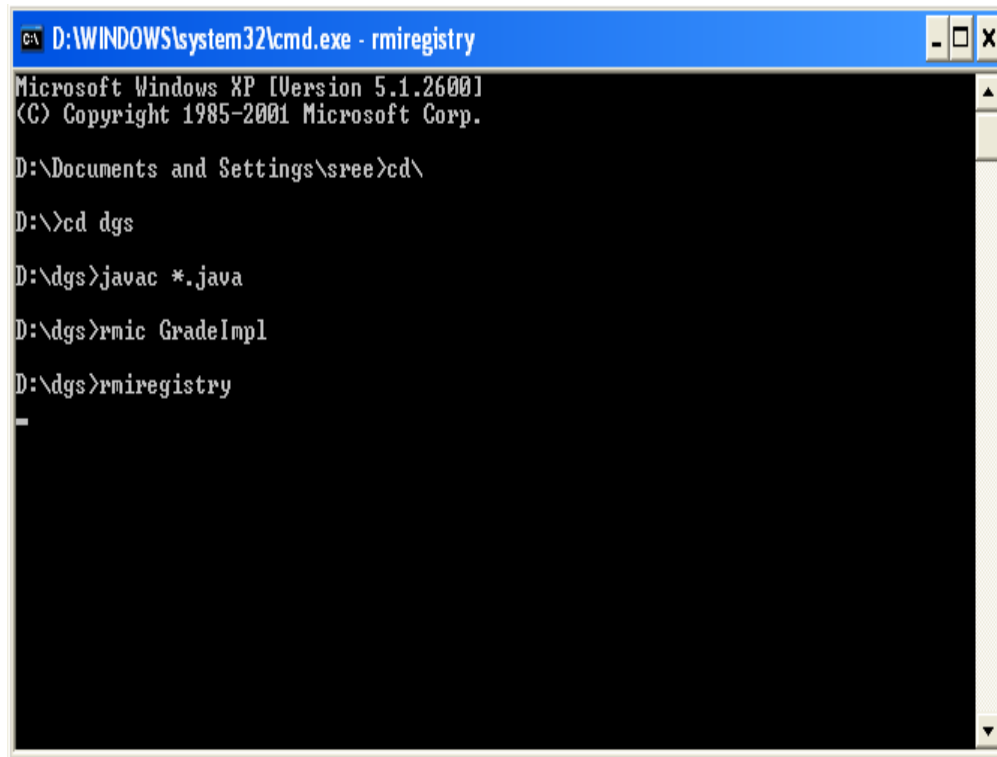
```
C:\ D:\WINDOWS\system32\cmd.exe - java CalculateGradeServer
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>java CalculateGradeServer
Calculate Grade object bound
-
```

Screen 4. 11 Generating Object For Calculategradeserver

DISTRIBUTED GRADING SYSTEM

c). System -2: Screen 1

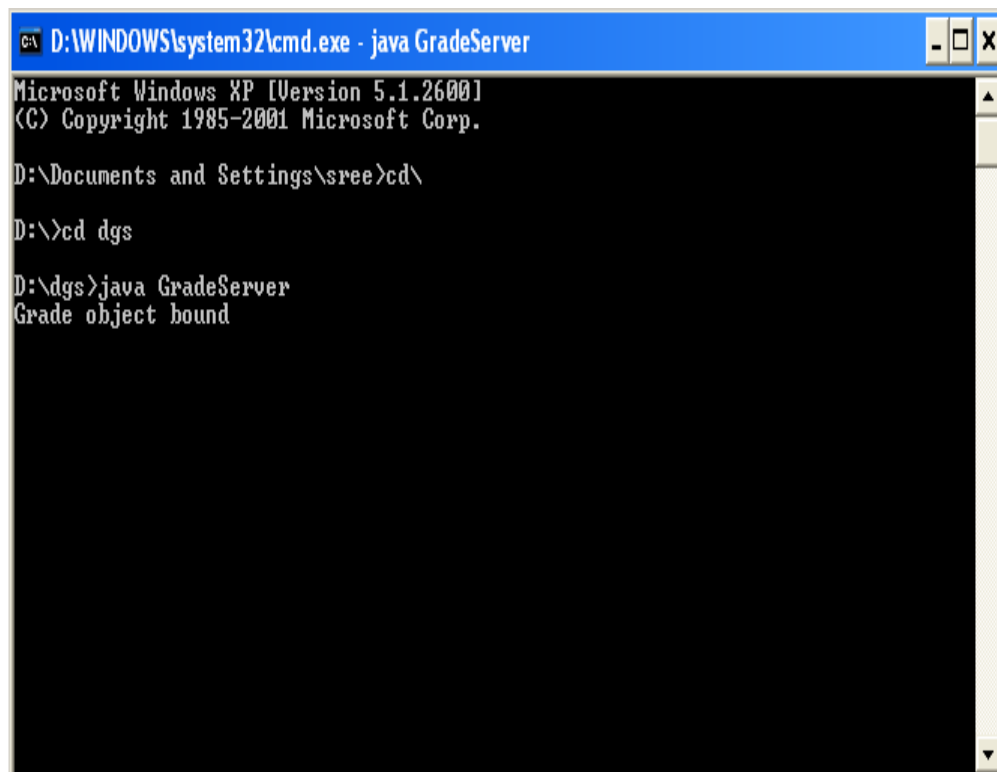


```
CA D:\WINDOWS\system32\cmd.exe - rmiregistry
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>javac *.java
D:\dgs>rmic GradeImpl
D:\dgs>rmiregistry
-
```

Screen 4. 12 Creating rmic for GradeServer

d). System -2: Screen 2



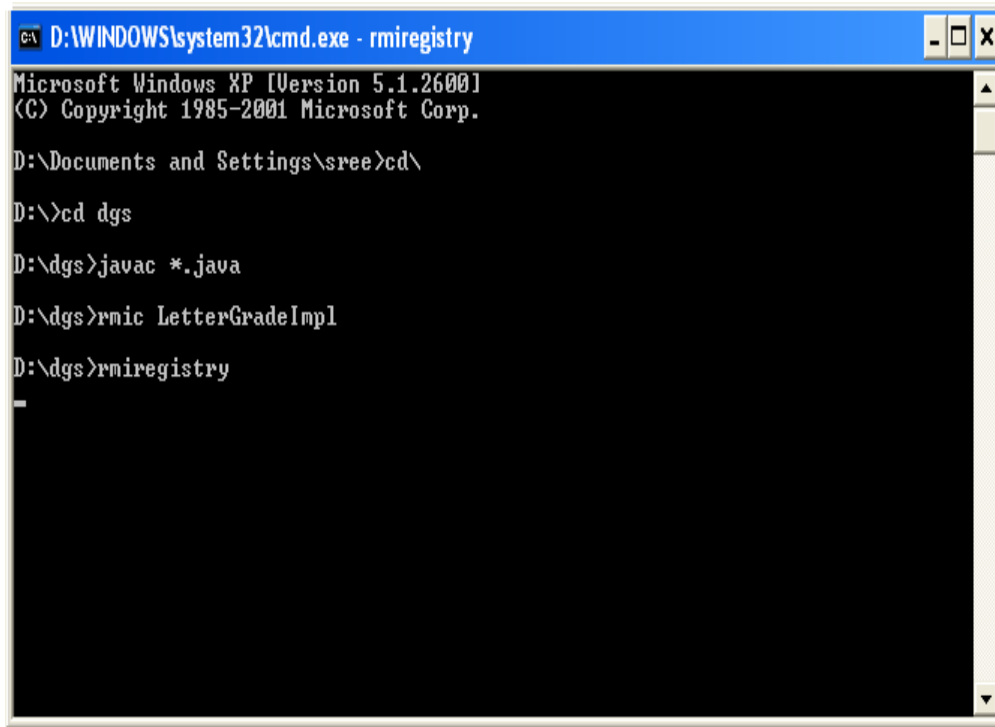
```
CA D:\WINDOWS\system32\cmd.exe - java GradeServer
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>java GradeServer
Grade object bound
```

DISTRIBUTED GRADING SYSTEM

Screen 4. 13 Generating object for GradeServer

e). System -3: Screen 1

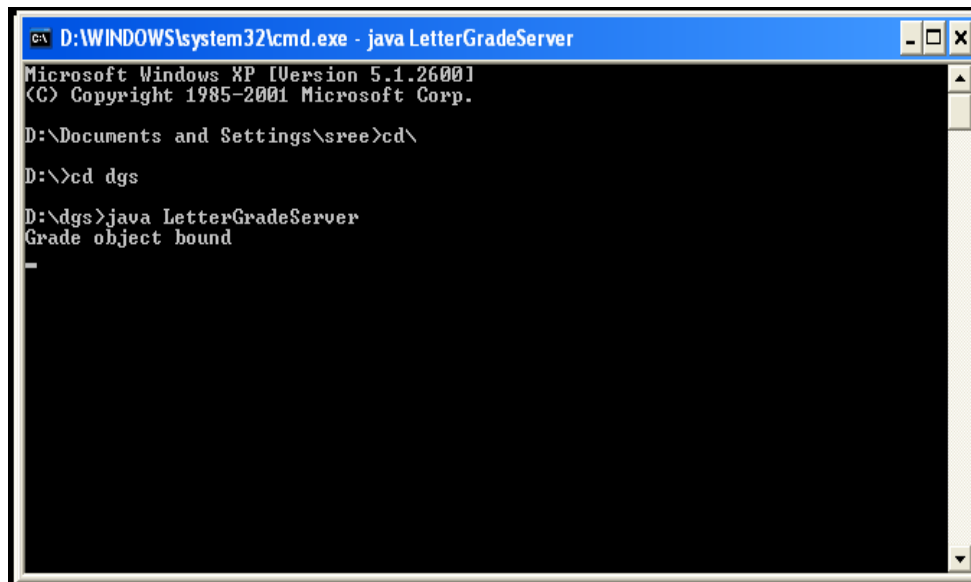


```
cmd D:\WINDOWS\system32\cmd.exe - rmiregistry
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>javac *.java
D:\dgs>rmic LetterGradeImpl
D:\dgs>rmiregistry
-
```

Screen 4. 14 Creating rmic for LetterGradeServer

f). System -3: Screen 2

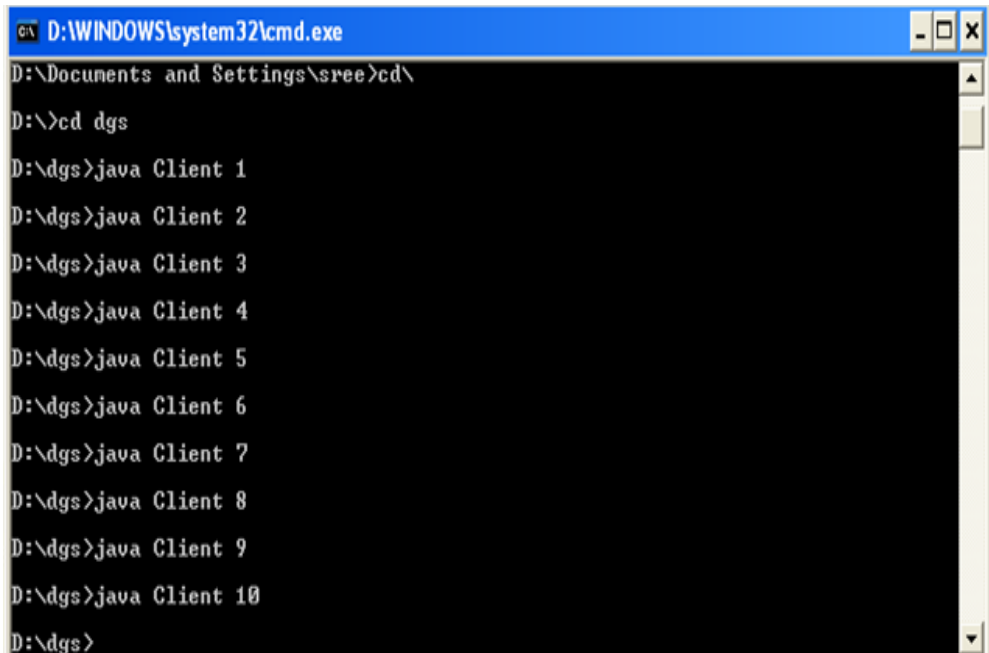


```
cmd D:\WINDOWS\system32\cmd.exe - java LetterGradeServer
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>java LetterGradeServer
Grade object bound
-
```

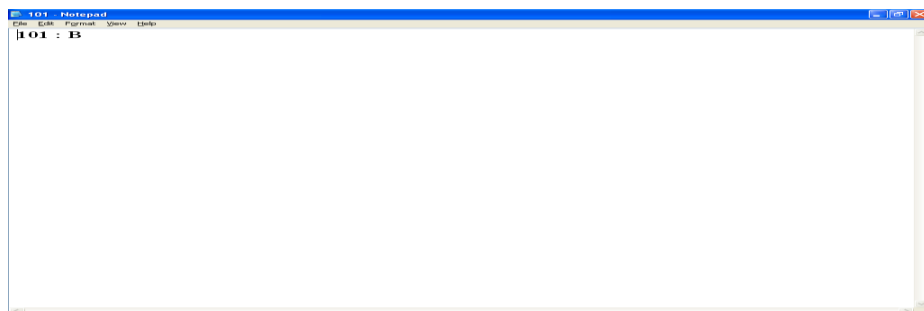
Screen 4. 15 Generating Object for LetterGradeServer

DISTRIBUTED GRADING SYSTEM



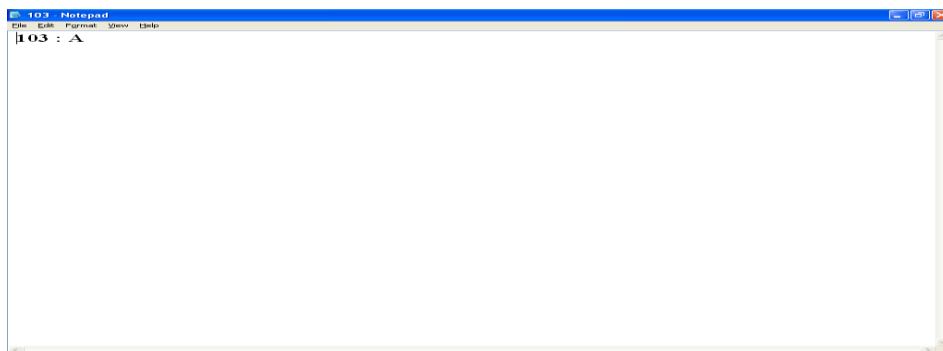
```
D:\WINDOWS\system32\cmd.exe
D:\Documents and Settings\sree>cd\
D:\>cd dgs
D:\dgs>java Client 1
D:\dgs>java Client 2
D:\dgs>java Client 3
D:\dgs>java Client 4
D:\dgs>java Client 5
D:\dgs>java Client 6
D:\dgs>java Client 7
D:\dgs>java Client 8
D:\dgs>java Client 9
D:\dgs>java Client 10
D:\dgs>
```

Screen 4. 16 Running a Client program



```
101 : B
```

Screen 4. 17 Displaying Grade of a Student in a File



```
103 : A
```

Screen 4. 18 Displaying Grade of a Student in a File

Chapter -5

Conclusion

5.1. Goals that are met:

Presently the application is in the starting stage. It was developed for just testing purpose. Here we have only one user i.e. the administrator. The activities under the administrator like inserting student details, modifying the existing details, displaying the student details including examination marks details and displaying the results are successful in the distributed environment. By using this basic step we can enhance the project to maintain the entire organization in a distributed environment.

5.2. Goals that are not met:

The actual goal of this project is to maintain the entire organization details including the entire details of the staff and etc. As this being the first step, it was not possible to develop the entire details and put into testing. So only a minute part has been took and tested whether the concept of distributed environment is success or not. The main goal of the project will be achieved while enhancing the project.

5.3. Conclusion Remarks:

The result of the Distributed Grading System is that, it communicates remotely two or more systems and gets the result as final grades of students by using the RMI (Remote Method Invocation) technology.

Finally we developed a System that would be able to operate across multiple platforms (i.e. UNIX and Windows).For ease of demonstration, we had the system running on three UNIX machines, but the system was also run successfully using two UNIX machines and one Windows machine.

BIOGRAPHY

References for the Project Development were taken from the following Books and Web Sites.

Oracle

PL/SQL Programming by Scott Urman

SQL complete reference by Livion

JAVA Technologies

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

JAVA2 Networking by Pistoria

JAVA Security by Scotl oaks

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

HTML

HTML Black Book by Holzner

JDBC

Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman