SACRAMENTO STATE

COLLEGE OF ENGINEERING & COMPUTER SCIENCE

CSc 230

Software System Engineering

FINAL REPORT

Project Management System

Prof.: Doan Nguyen

Submitted By:

Parita Shah

Ajinkya Ladkhedkar

Spring 2015

Table of Content

List of Figures

List of Tables

1. Customer Statement of Requirements

a. Goals

The main goal of this software is to help the professor (customer) in the process of managing the projects of the students. Rather than manually storing and remembering documents and details about each project, professor can use this software to swiftly go through the process with the help of few mouse clicks only.

b. Problem Statement

In the existing set-up, professor has to note down all the students' details manually. He also has to sort these students into groups according to their projects. These tasks are usually done on paper or excel sheet or any such document. This can make the professor's task tedious since details can be lost/mishandled or multiple copies of same group details can be generated if any change has to be done.

Also, the professor has to keep a track of upcoming deadlines regarding a project. Moreover the professor may not be able to keep a check on a group's progress. We are developing project management system to reduce his memory load and help the professor simplify the process of managing all these details.

c. Proposed Solution

In order to solve the above mentioned problems, we are developing a software which will manage all the details of the project. The professor will be able to add students' details in the system. Moreover students can be grouped together for a group project and the details will then appear for the project as a whole group.

Once the software is set up with all the details, the whole process of project management gets automated. The project will provide a tracking system through which the professor can track the progress of each project. Professor will have to pre-define the deadlines for all projects. Once a group has finished a task for a given deadline, professor can check it as achieved in the system. We plan to show a progress bar for this functionality. We shall show the name of the task on that progress bar. The tracking of progress will be shown in a graphic display which will make the GUI appealing.

We will provide have a reminder service which shall post the upcoming deadline set up by the professor. For example, if the professor has set up a due date for UML diagrams submission, this deadline will be displayed on the home page as soon as the professor logs into the system.

Also at any point, if the professor wants to download the details of the projects, he can create report by exporting all details in an excel sheet. Once all the projects are finished and the semester is over, professor can reset the software using the report generation functionality. Finally we shall provide a functionality to reset the system by clearing all existing records. So the system will be ready to use for the next semester.

2. Software Development Lifecycle Model

Software model used: **Agile Model**

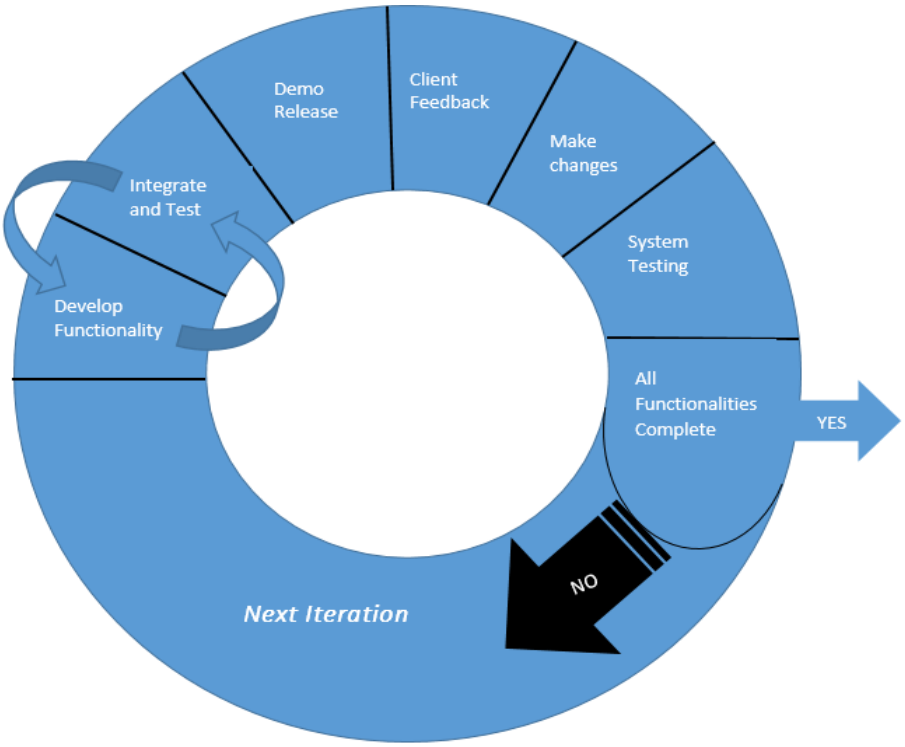


Figure 2.1 Agile Model

Features of Agile Model:

- Rapid and adaptive response to any change **at any point** in the project lifecycle.
- Active communication amongst all stakeholders.
- Faster delivery of the software.
- Advantageous for short-lived projects like ours.
- Driven by customer requirements.

3. Functional Requirements

FR Id.	Requirement	Priority	Ownership	Description
FR – 1	Add_new_user	2	Parita Shah	Professor will create login credentials for himself.
FR – 2	User_login	1	Parita Shah	System will authenticate user from above entered credentials.
FR – 3	Add_student_record	1	Ajinkya Ladkhedkar	Enter student details like Name, Student ID and e-mail.
FR – 4	Edit_student_record	2	Ajinkya Ladkhedkar	Professor can edit any student's details
FR – 5	Delete_student_record	4	Ajinkya Ladkhedkar	Professor can delete a selected student's record.
FR – 6	Search_student_details	2	Parita Shah	Professor can search a record pertaining to a specific student by name or student ID.
FR – 7	Create_new_group	1	Ajinkya Ladkhedkar	Professor will create a group for students working under single project.
FR – 8	Edit_group_details	2	Ajinkya Ladkhedkar	Professor can edit details of a particular group searching by the project name.
FR – 9	Delete_group_record	4	Ajinkya Ladkhedkar	Professor can delete a group record if needed.
FR – 10	Add_deadline	2	Parita Shah	For each milestone professor can add a deadline.
FR – 11	Edit_deadline	3	Parita Shah	Professor can modify the deadline.
FR – 12	Check_deadline_achieved	2	Parita Shah	If a deadline is met for a project, professor can mark it as achieved.
FR – 13	Display_reminder	3	Parita Shah	System will display a reminder for upcoming deadline like a ticker on the home page.
FR – 14	Generate_report	2	Parita Shah	Professor can download all project regarding details in an excel spreadsheet at any point in the semester.
FR – 15	Reset_system	4	Ajinkya Ladkhedkar	Once the professor is done using the software for a semester, he can reset the system. This will delete all records from the system and make the system ready for the next semester.

Table 3.1 Functional Requirements

4. Non-Functional Requirements

NF ID.	Requirement	Description
NF – 1	Usability	We are using concepts of Human computer interaction. All the webpages will have consistent GUI all over the system. We are providing navigation buttons on each page to move back and further.
NF – 2	Interoperability	System will lead you to homepage in maximum 3 mouse clicks.
NF – 3	Space	We shall normalize the database upto 2NF. This will reduce redundancy in the database and hence occupy less memory space.
NF – 4	Reliability	Software will provide feedback for any error. Also, the user can navigate back to previous page.
NF – 5	Security	Requires user authentication. Only Admin can use the application. Code cannot be changed by the user.
NF – 6	Portability	Since the system is deployed on web server, it can be accessed by any web browser on any computer.
NF – 7	Standards	We are using coding standards by giving meaningful names to functions and variables. This will help in easy maintenance and future modification.

Table 4.1 Non Functional Requirements

5. Tools used:

Platform: Java 1.7

Web server: Apache Tomcat 7

Communication Framework: Eclipse Luna

Database Management System: Oracle 10g

6. Languages used:

Backend: SQL

Application Logic: JSP, JDBC

Frontend: HTML, CSS

7. System Diagram:

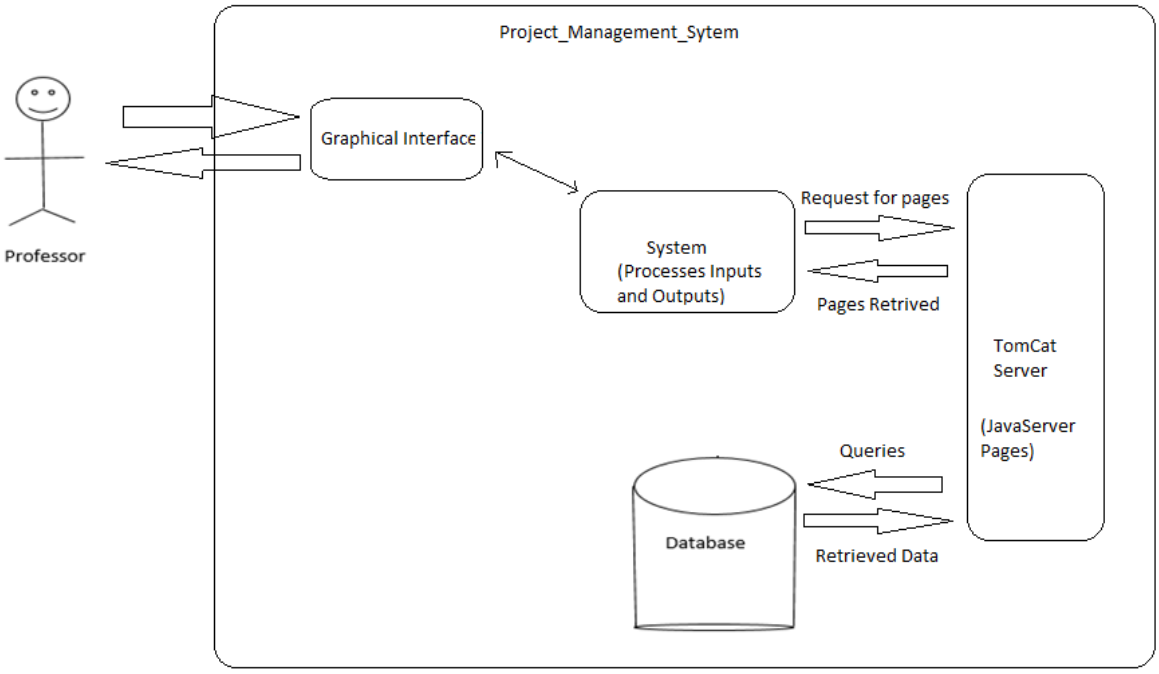


Figure: 7.1 System Diagram

8. Class Diagram

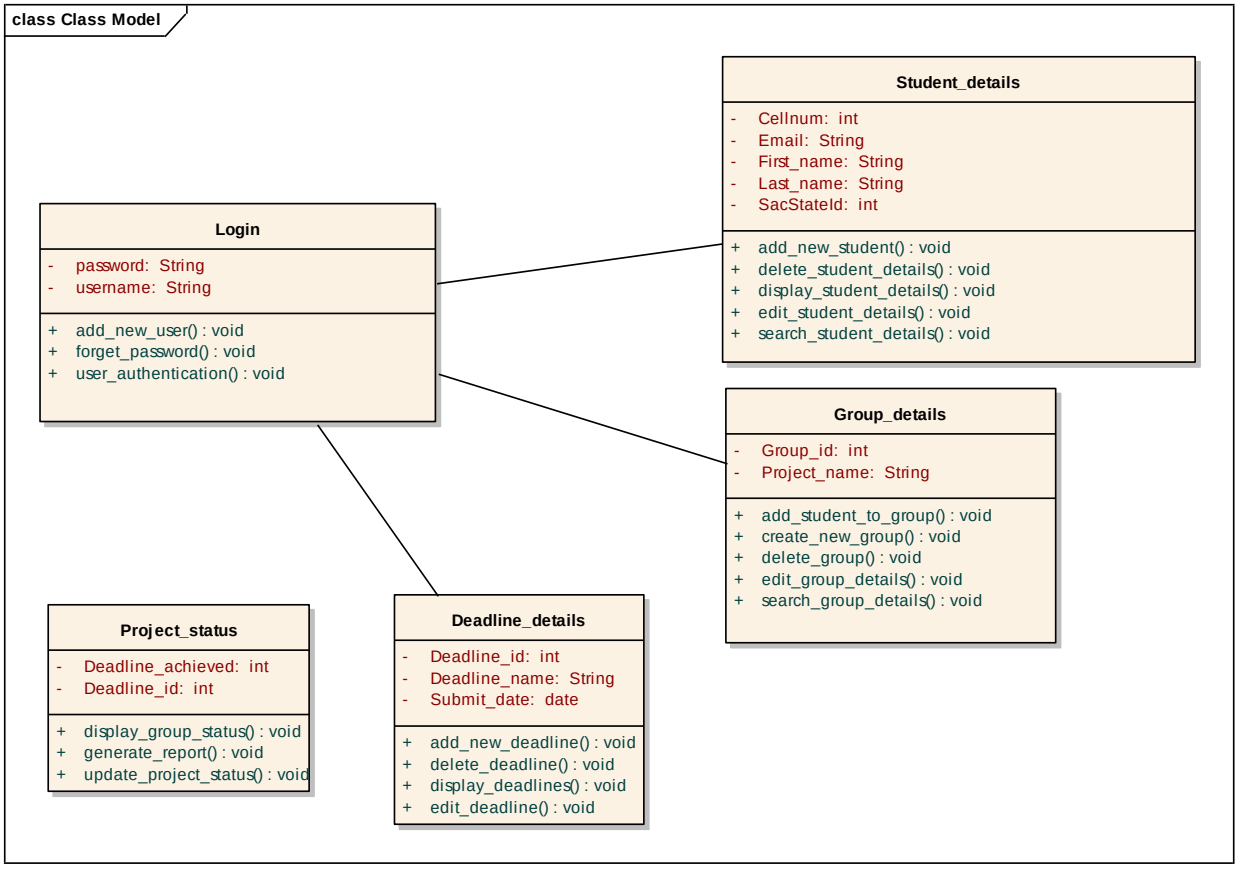


Figure 8.1 Class Diagram

9. Database Design

Table 1: Login

Description: Contains all details of the user of the system.

Constraint: Username is the primary key. No two users can have the same username.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOGIN	FIRSTNAME	Varchar2	20	-	-	-	✓	-	-
	LASTNAME	Varchar2	20	-	-	-	✓	-	-
	USERNAME	Varchar2	20	-	-	1	-	-	-
	PASSWORD	Varchar2	30	-	-	-	✓	-	-
	CELLPHONE	Varchar2	10	-	-	-	✓	-	-

Table 2: Student Details

Description: Contains all details of students as entered by the user of the system.

Constraint: Sac State ID is the primary key. This should be unique for all students in the database.

Group_ID is the foreign key. Students if assigned to any project (group) will have the group id in this field.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT_DETAILS	F_NAME	Varchar2	20	-	-	-	✓	-	-
	L_NAME	Varchar2	20	-	-	-	✓	-	-
	SID	Varchar2	9	-	-	1	-	-	-
	EMAIL	Varchar2	30	-	-	-	✓	-	-
	CELLNUM	Varchar2	10	-	-	-	✓	-	-
	GROUP_ID	Number	-	-	0	-	✓	-	-

Table 3: Group Details

Description: Contains Name and ID of a group/project.

Constraint: Group ID is the primary key. No two groups can have the same Group ID.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GROUP_DETAILS	GROUP_ID	Number	-	-	0	1	-	-	-
	PROJECT_NAME	Varchar2	100	-	-	-	✓	-	-

Table 4: Deadline Details

Description: Contains details about deadlines that have to be achieved by the groups.

Constraint: Deadline ID is the primary key. Each deadline must have a unique ID.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEADLINE_DETAILS	DEADID	Number	-	-	0	1	-	-	-
	DEADLINE_NAME	Varchar2	100	-	-	-	✓	-	-
	SUBMIT_DATE	Date	7	-	-	-	✓	-	-

Table 5: Deadlines Achieved

Description: Contains details about which group/project has achieved which deadline.

Constraints: Here foreign keys Group ID and Deadline Id together form a compound primary key. A group can achieve a deadline only once.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEADLINES_ACHIEVED	GROUP_ID	Number	-	-	0	1	-	-	-
	DEADID	Number	-	-	0	2	-	-	-

Functional Requirements Specification

1. Stakeholders

Looking at the nature of our software, there will be only 2 types of stakeholders, Professor (user) and Developers (Parita Shah and Ajinkya Ladkhedkar). The system is being developed mainly for the professor to manage all projects of a class. So, professor will be responsible for carrying out all the functions in the application. The functions will include adding a user to the system, adding and modifying student and group details working on a project, track progress of progress of a project, generating reports and resetting the system in the end. Developers will be responsible for generating the software and connect the system to database in which all the details entered by the user will be stored.

2. Actors and Goals

Actor	Goal(s)
User (Professor)	Add his details to obtain credentials for the system. Add student details and group them according to the projects. Track progress of each project. Generate report and reset system.
Application	To provide interface for automating the management process. To allow user to carry out functionalities that is allowed to him.
Database server	Process queries fired through application which may retrieve or modify data.
Internet server	Allow Application to run.

Table 2.1 Actors and Goals

3. Use Case Diagram

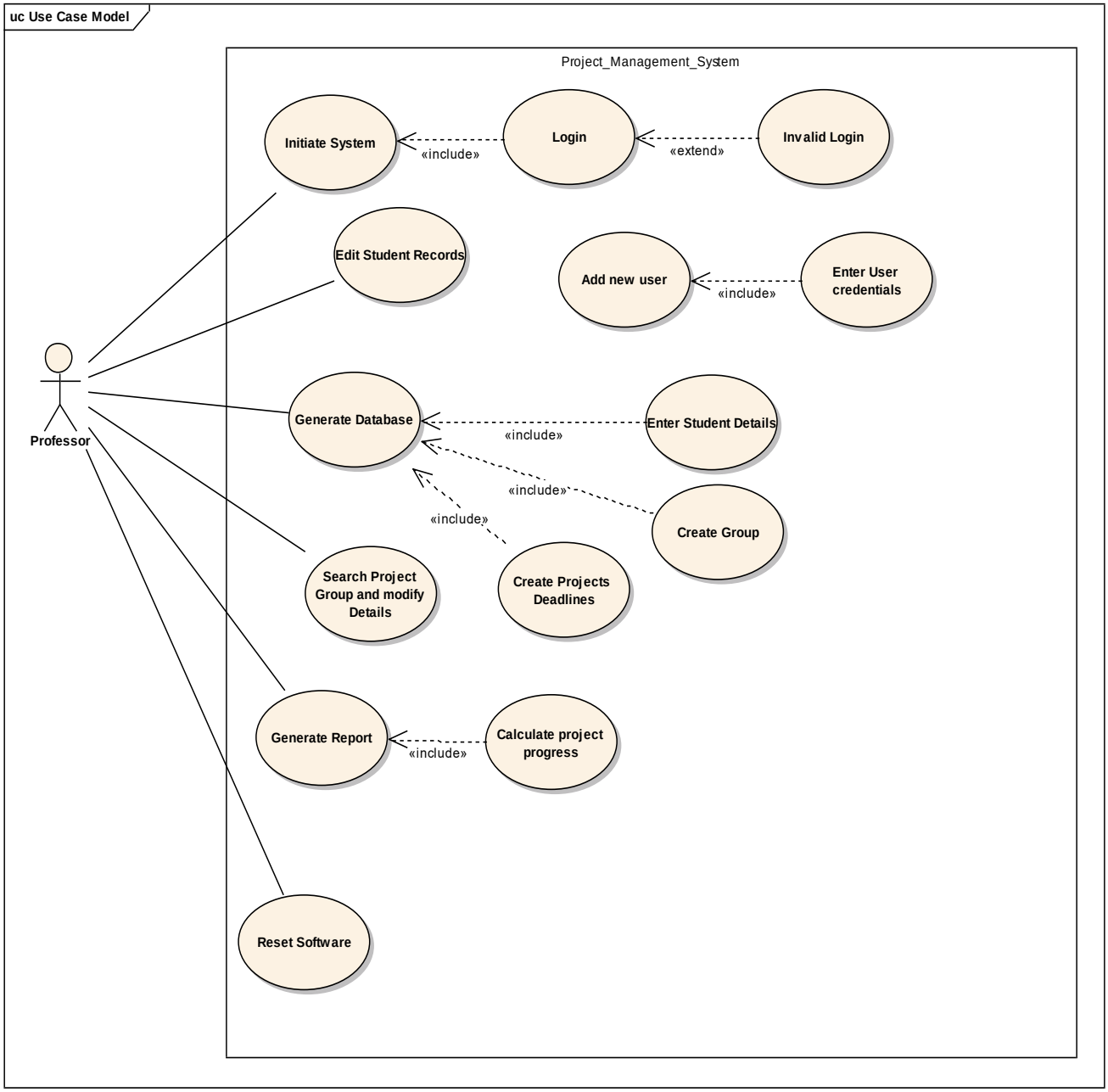


Figure: 3.1 Use Case Diagram

4. Use Case Description

Use Case	Name	Description	Pre-Conditions	Post Conditions
UC-1	Initiate System	User will initiate the system by accessing the web page	Web Server must be running and client should be connected to the internet.	User will be able to view the login page.
UC-2	Add new User	Professor will create a new user who will use the software.	Professor has to have a web browser to run the application.	User credentials will be available for user to log in the system.
UC-3	Login	User will login using the credentials.	The user should have login credentials for the system.	The user will be redirected to the home page of the application.
UC-4	Invalid Login	System will give error if user credentials are not valid.	User should enter login credentials.	System will generate feedback and user will have to login again with correct credentials.
UC-5	Generate Database	User will enter students, project and deadlines details.	User needs to be logged into the system. To create groups database, user needs to add student details.	New student record will be added into the database. After adding groups database, students will be allocated to their respective projects. New deadlines will be created.
UC-6	Edit Database	User will edit student and project details.	User needs to be logged into the system. There should be existing projects data in the database.	Database will be updated with the necessary changes.
UC-7	Search Project group and modify details	User can search for a project to change its details	User needs to be logged into the system. Projects database should be added into the system.	Project details will be modified.
UC-8	Create Project Deadlines	User can create deadlines and add the submit date.	User needs to be logged in into the system.	New deadlines will be added. Upcoming deadline will now be displayed on the homepage.
UC-9	Generate Report	User can generate report which will contain details of projects.	User needs to be logged in into the system. Details need to be added into the database.	Report will be generated in the form of excel sheet which will contain all details regarding any project.
UC-10	Calculate Project Progress	User can track the progress of a project.	User needs to be logged into the system.	User will be able to see how many deadlines a project has achieved.
UC-11	Reset software	Professor can reset the system for next semester use.	User needs to be logged into the system.	The system will be reset and all the data regarding user, students, projects and deadlines will be deleted from the database.

Table 4.1 Use Case Description

5. Sequence Diagram

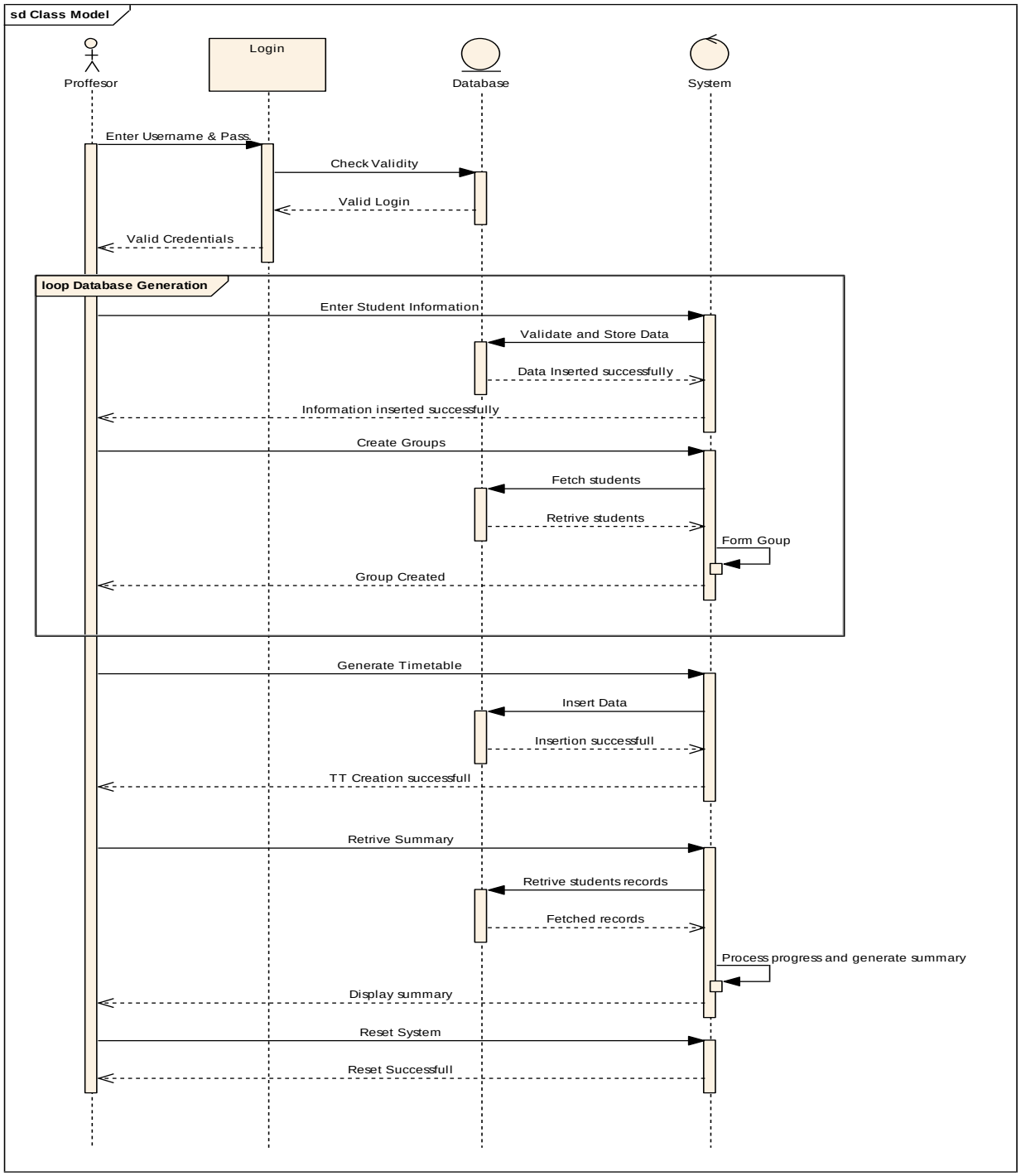


Figure 5.1 Sequence Diagram

6. Activity Diagram

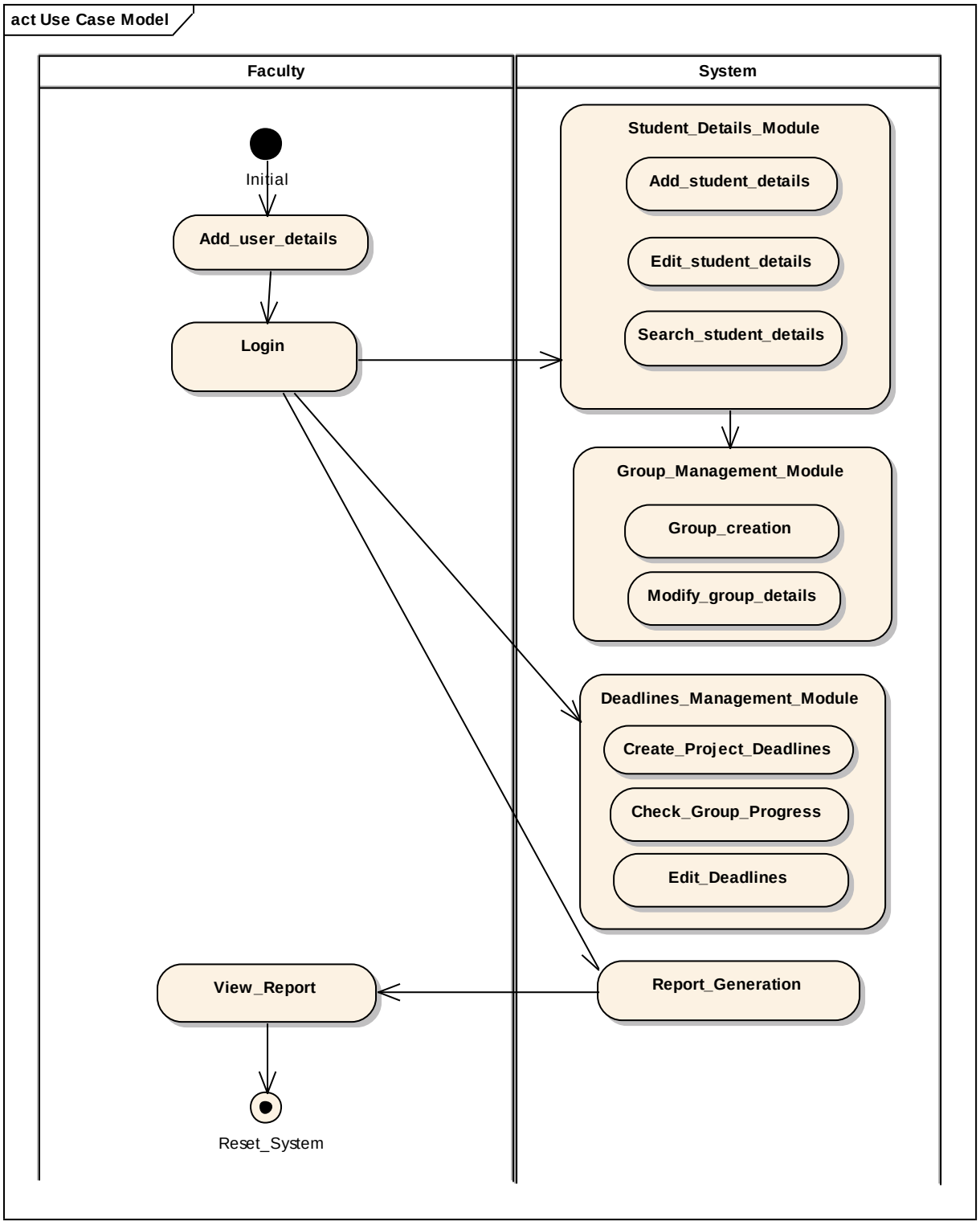


Figure 6.1 Activity Diagram

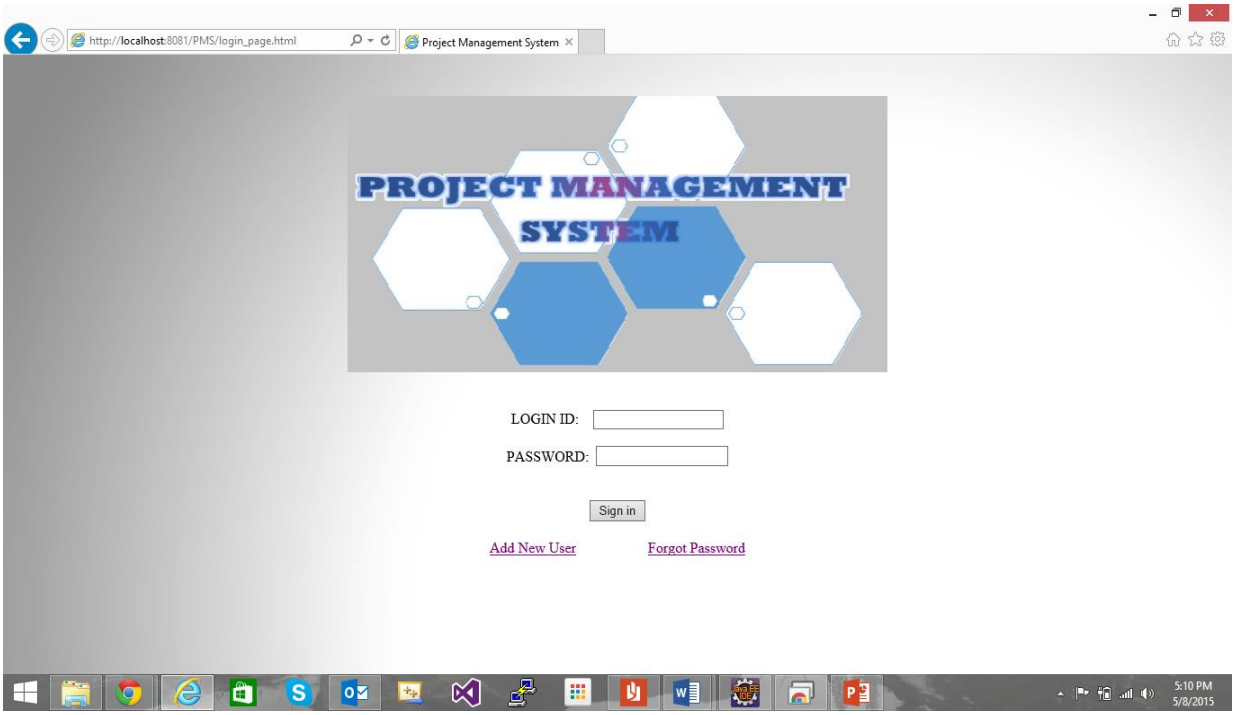
7. Traceability Matrix

Requirement	UC-1	*UC-2*	*UC-3*	*UC-4*	*UC-5*	*UC-6*	*UC-7*	*UC-8*
Add_new_user	X							
User_login		X						
Add_student_record			X					
Edit_student_record				X				
Delete_student_record				X				
Search_student_details				X				
Create_new_group				X				
Edit_group_details				X				
Delete_group_record				X				
Add_deadline						X		
Edit_deadline						X		
Check_deadline_achieved					X			
Display_reminder						X		
Generate_report							X	
Reset_system								X

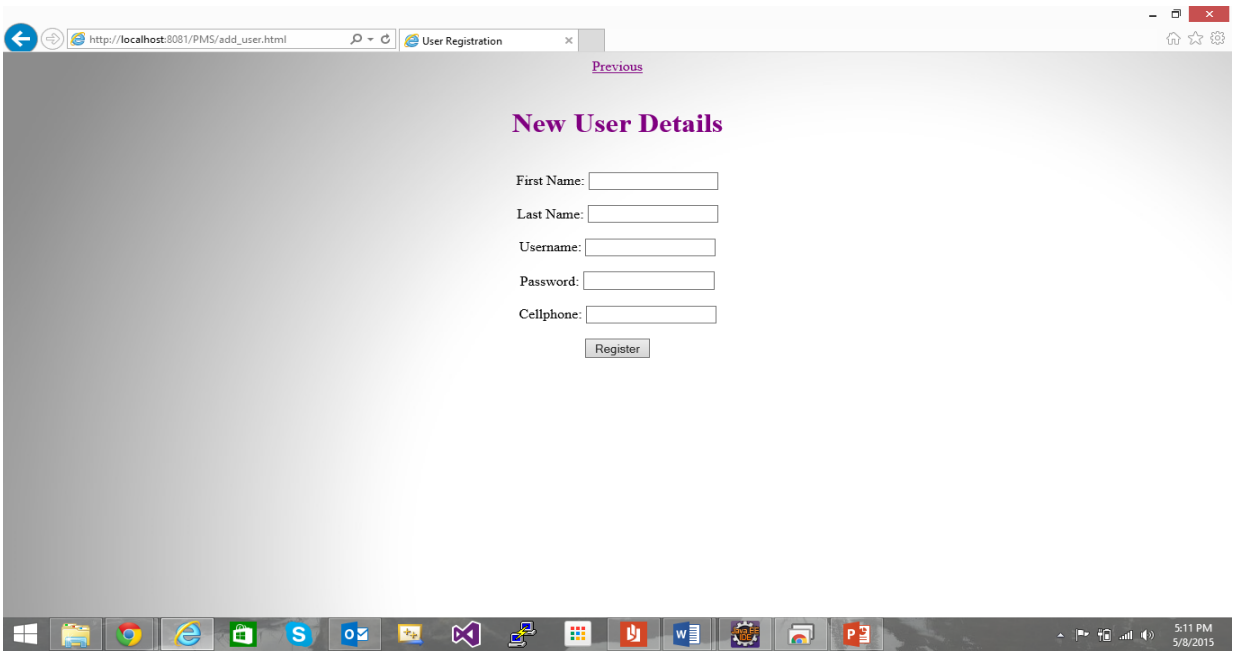
Table 7.1 Traceability Matrix

8. Application Screenshots

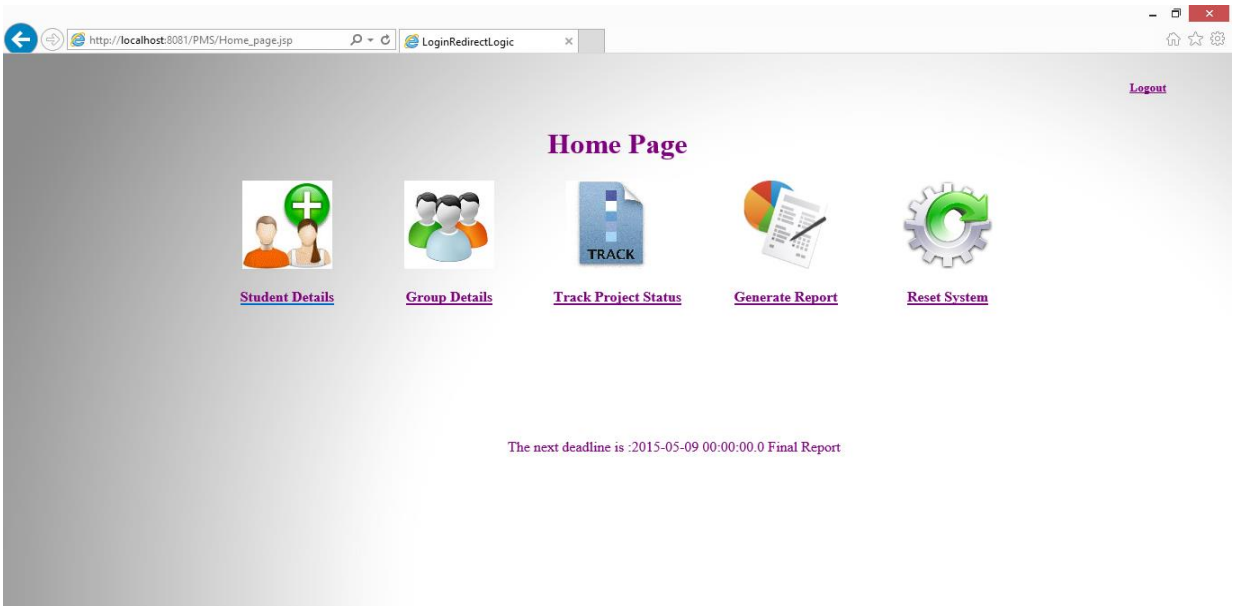
a. Login Screen: User can enter his login credentials to enter into the system. A new user can also create his login credentials using Add new user feature. Also if a user has forgotten his password, he can fetch that using the Forgot Password link.



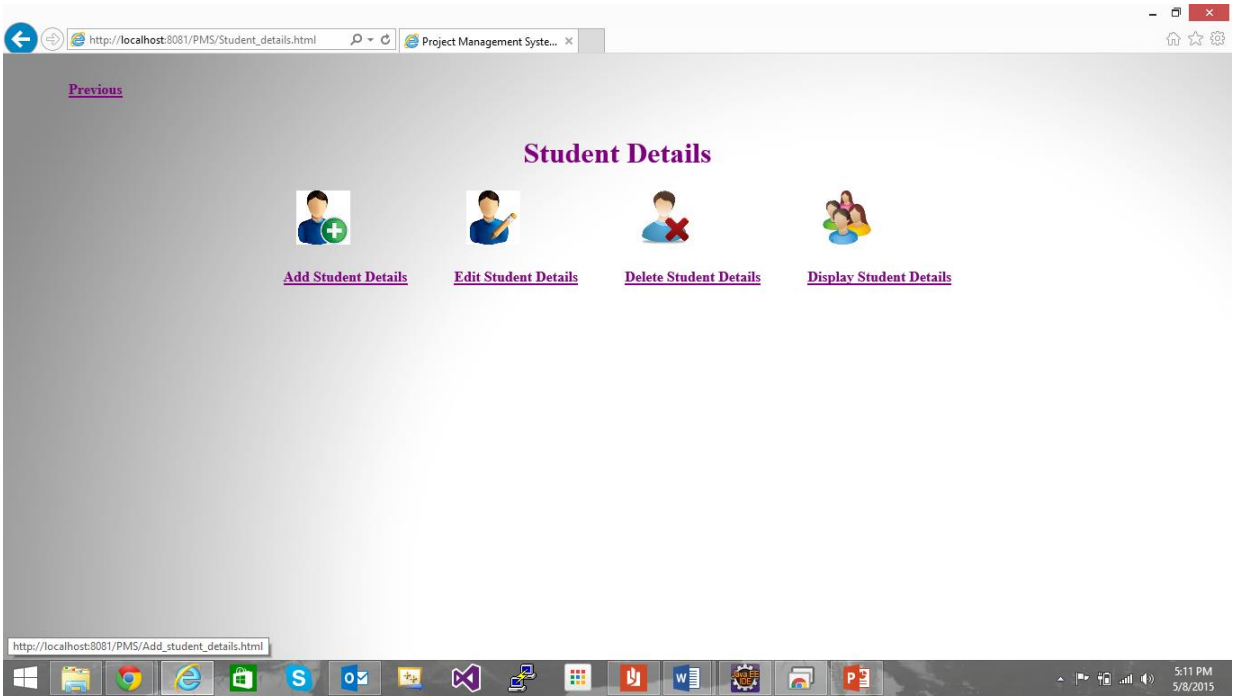
b. Add New User Screen: User will add his details into the system to create login credentials for himself.



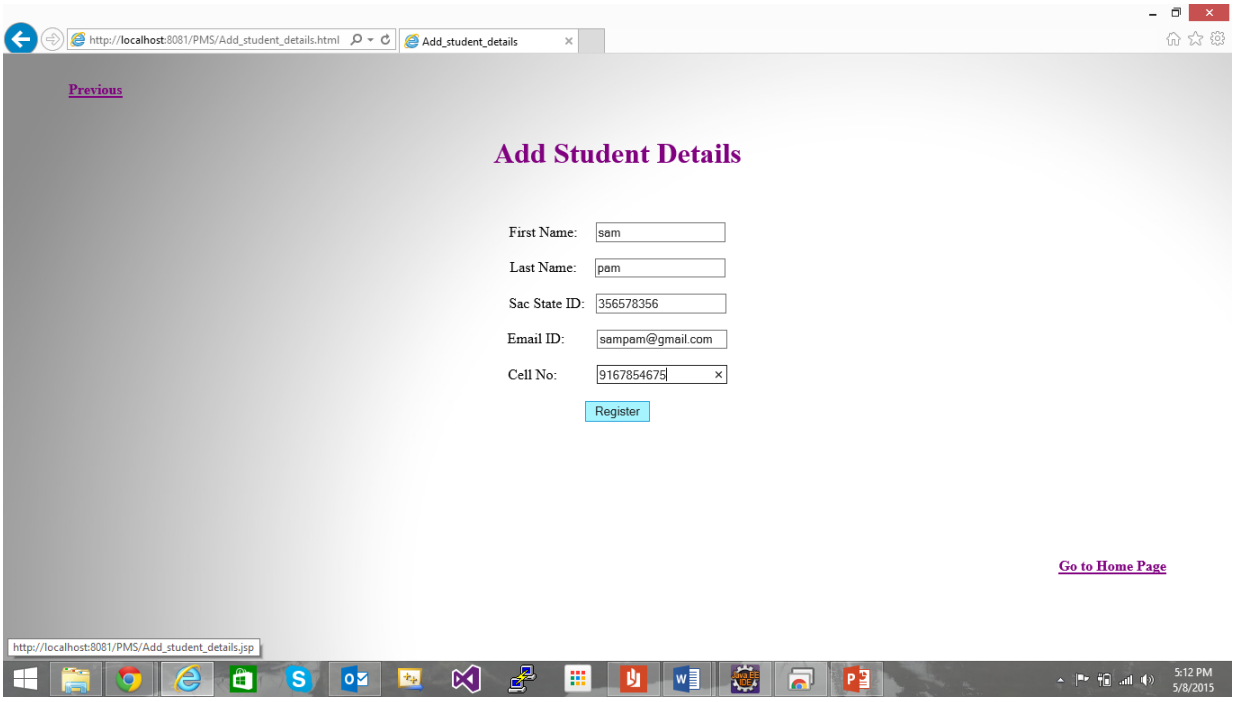
c. Home Page: User can see the links for all functionalities here. User can logout from the application from this screen. Also the upcoming deadline will be displayed on this screen.



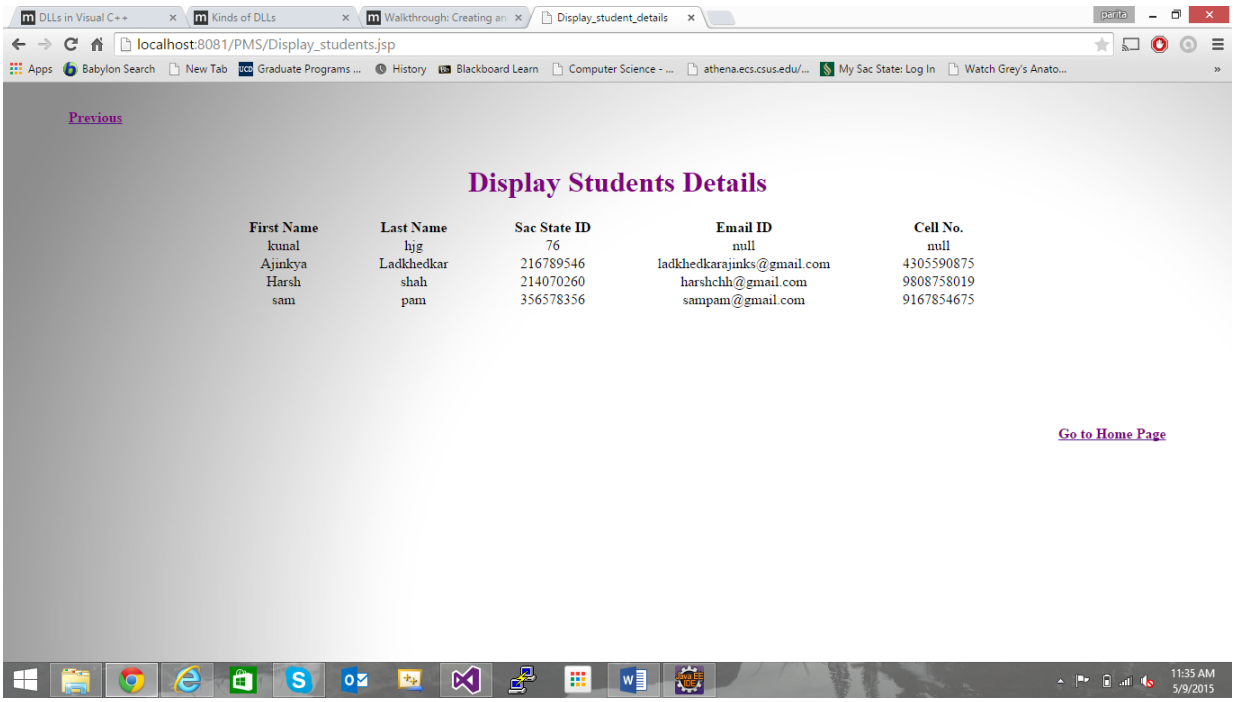
d. Student Details Screen: Links for Adding, Editing, Deleting and Displaying students details can be found on this screen. User can also navigate to homescreen by clicking on the previous button on the top left corner of the screen.



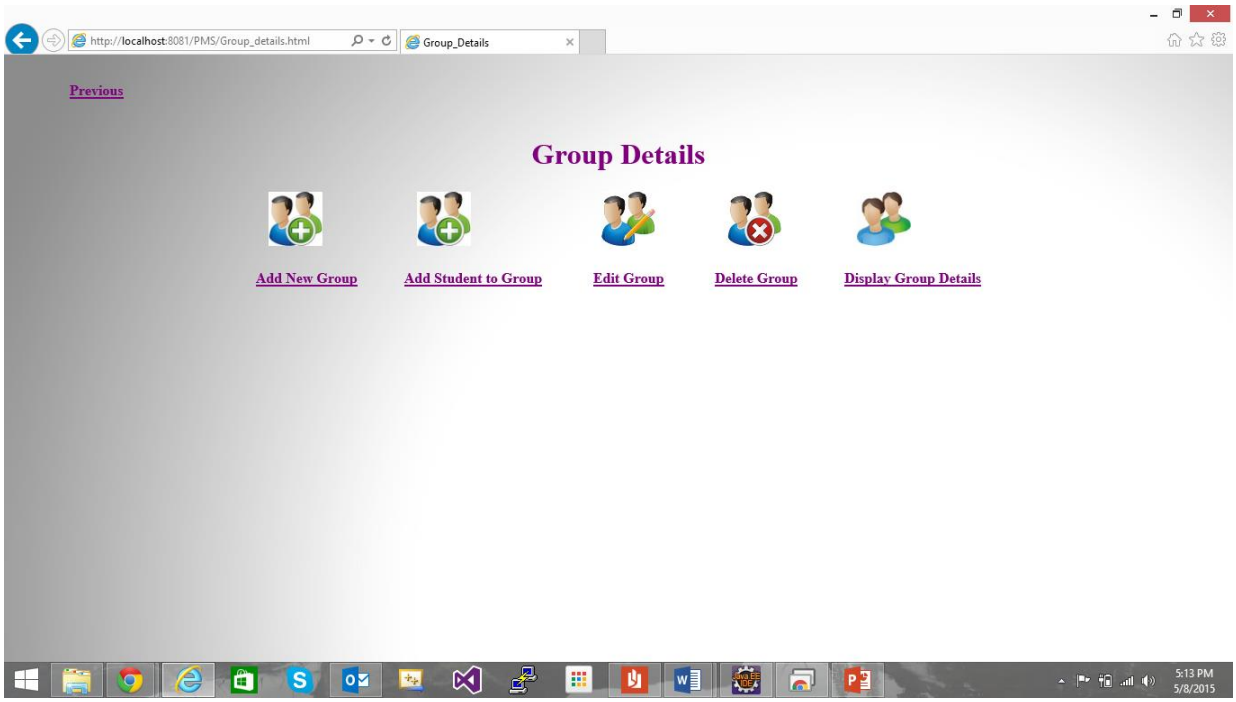
e. Add Student Details Screen: User can add all students details on this screen. As discussed previously, Sac State ID should be unique for all students.



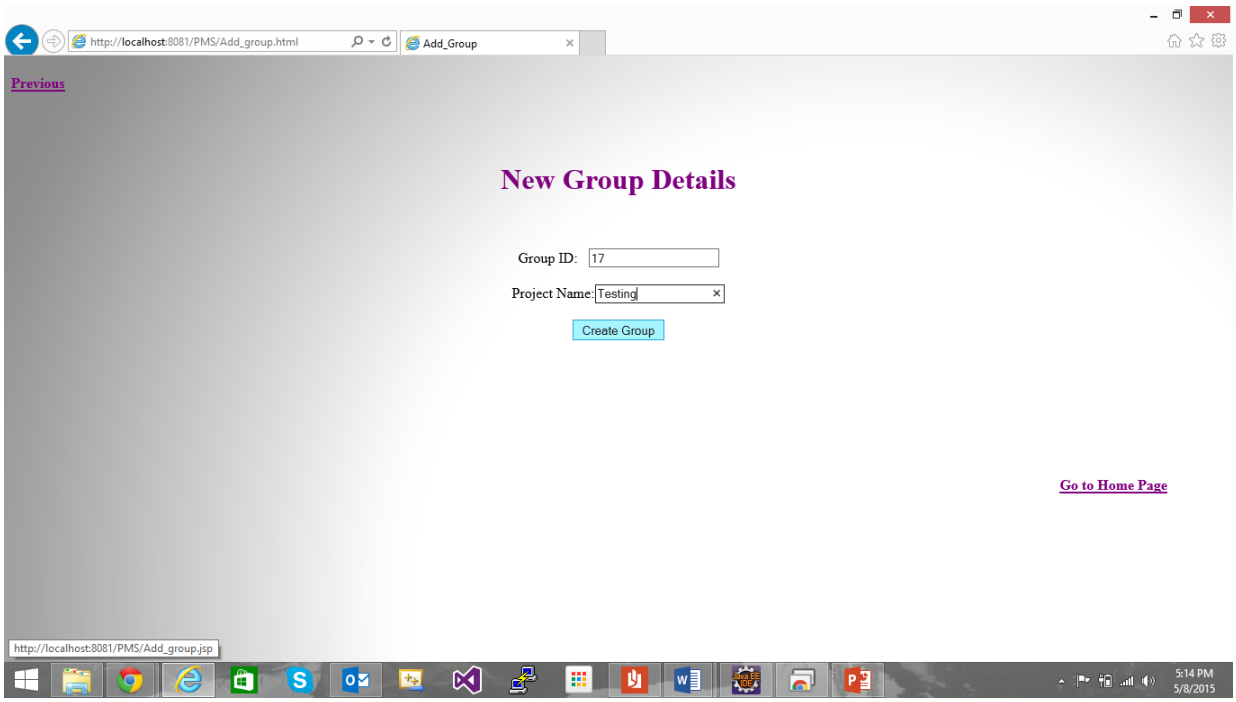
f. Display Students Details: User can see the details of all the students that exist in the database.



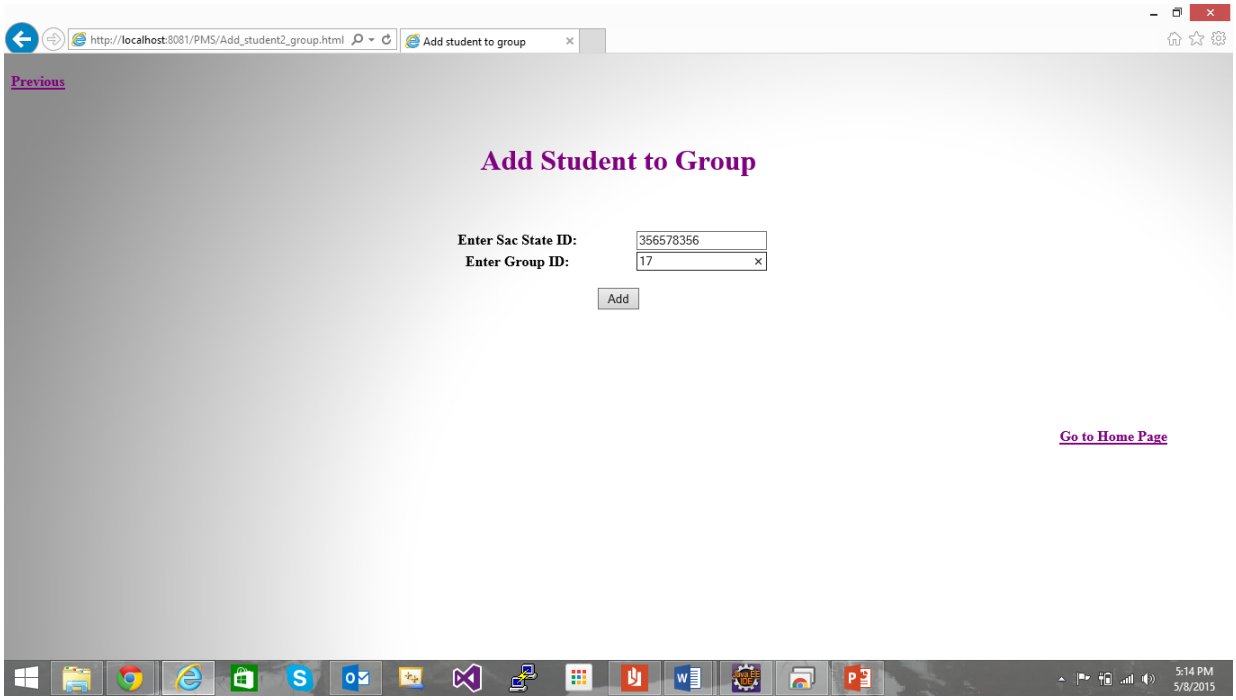
g. Group Details Page: User can access the links to Add new group, Add any student to any group, Edit, Delete and Display group details.



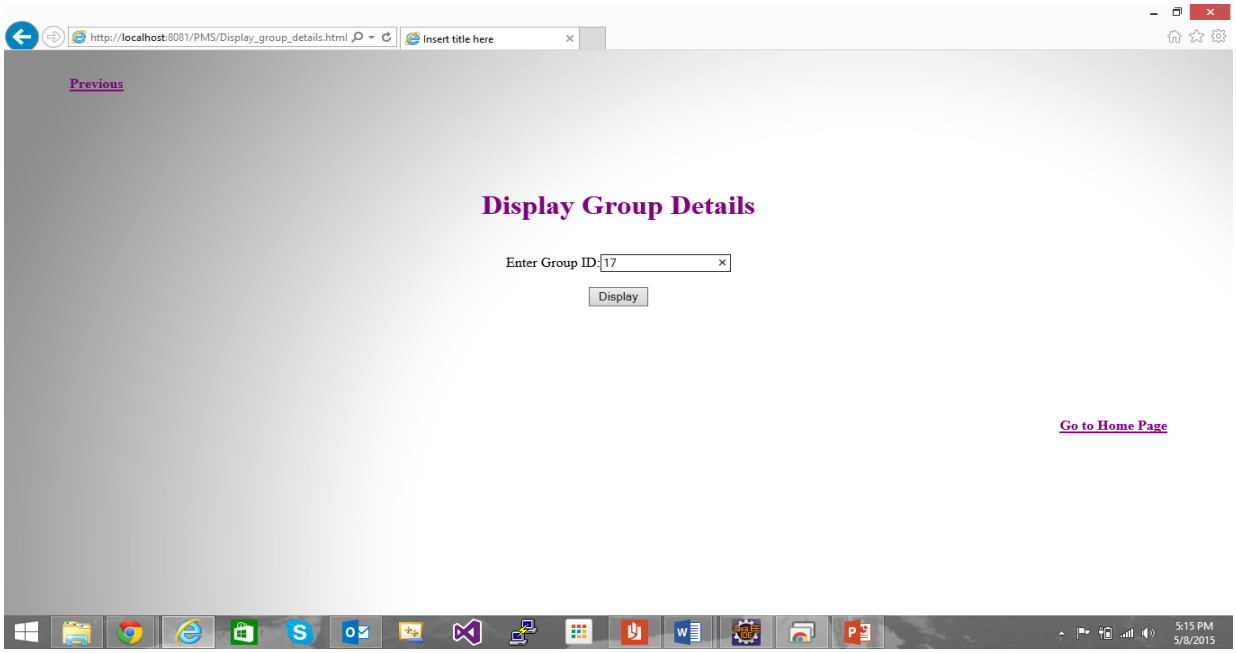
h. Add New Group Page: User can add a new group into the database. As discussed previously, the Group ID should be a unique number.



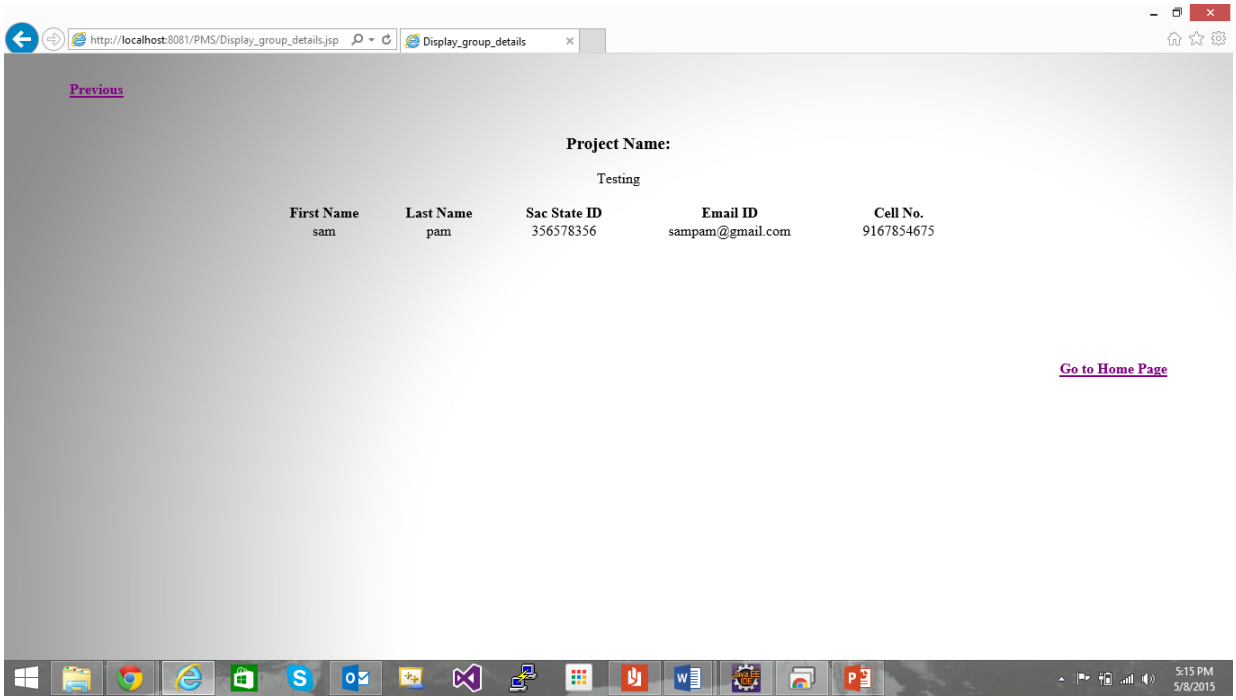
i. Add Student to Group: User can Add a student to a group if they are working as a group.



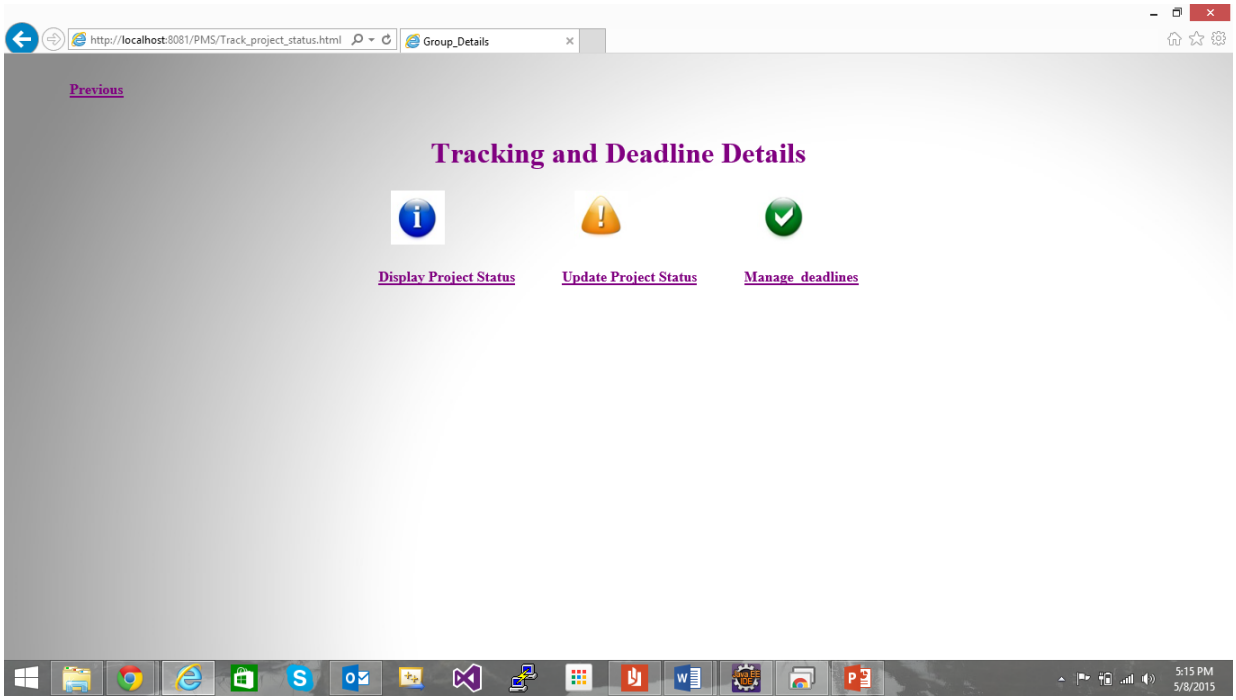
j. Display Group Details: Search by Group ID page: In order to view details of a group, the user needs to enter the unique Group ID first.



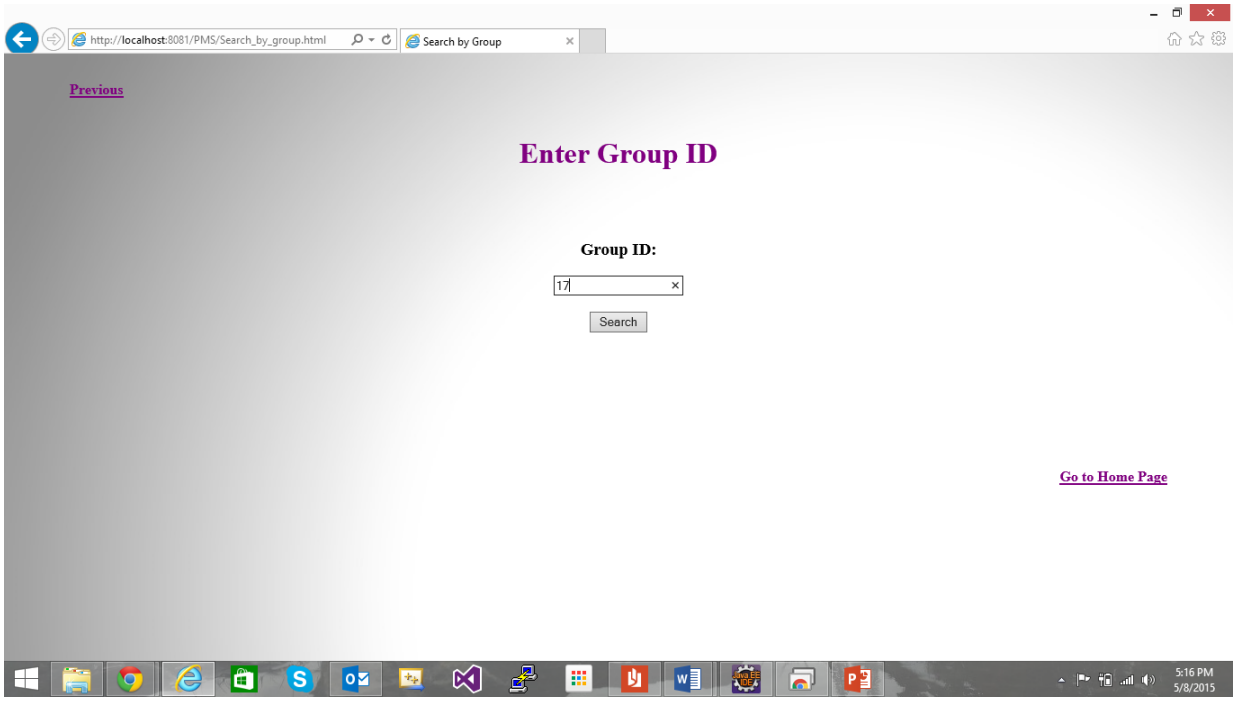
k. Display Group Details: User can all details of a particular group.



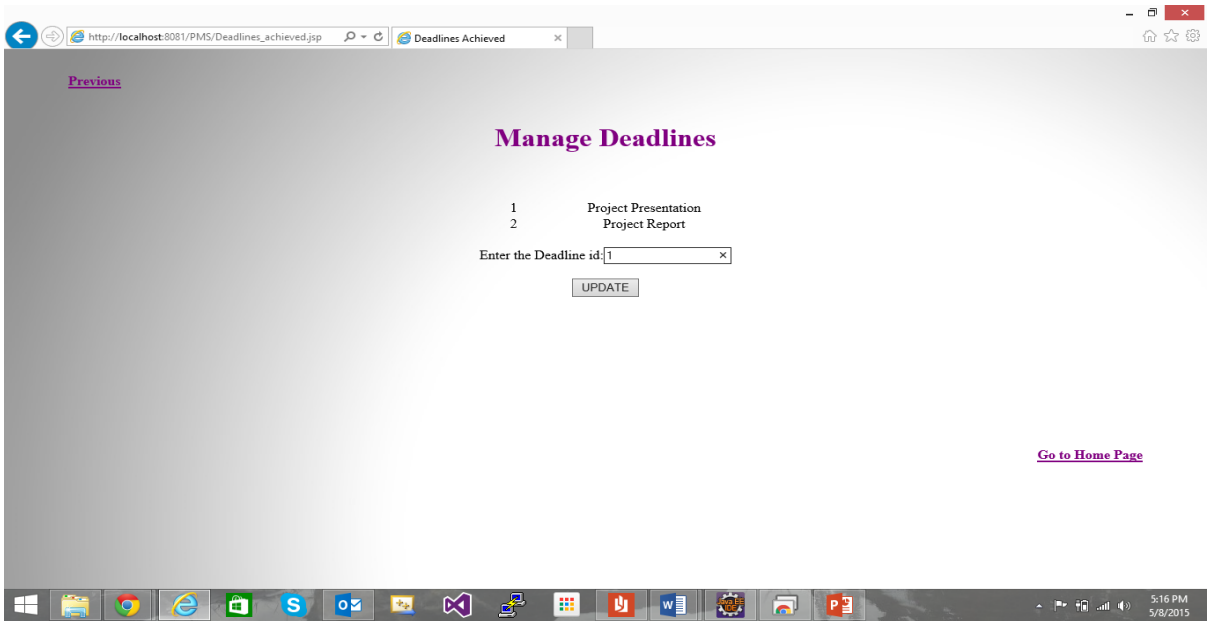
l. Progress Tracking and Deadline Details: User can access links to Display and Update Project status and Manage Deadlines options.



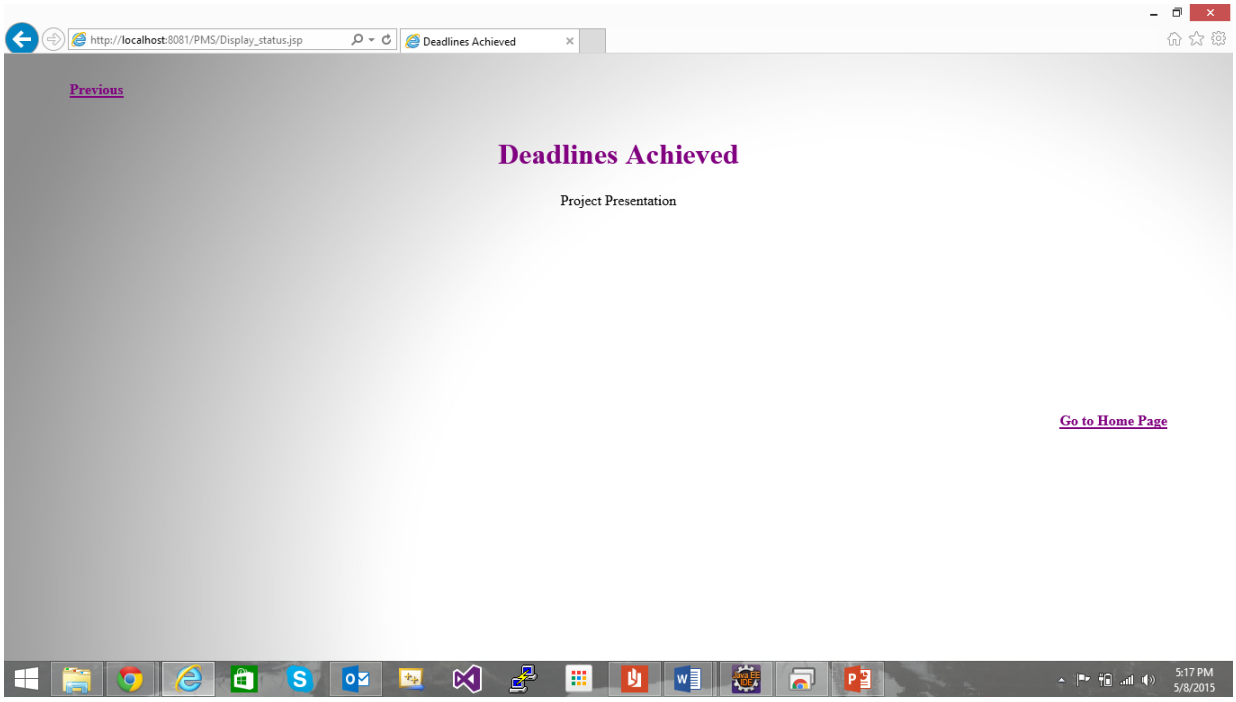
m. Add Deadline Achieved: Search by Group ID: If a group has achieved any deadline, the user will search the group using Group ID and then add that deadline to the list of deadlines achieved by the group.



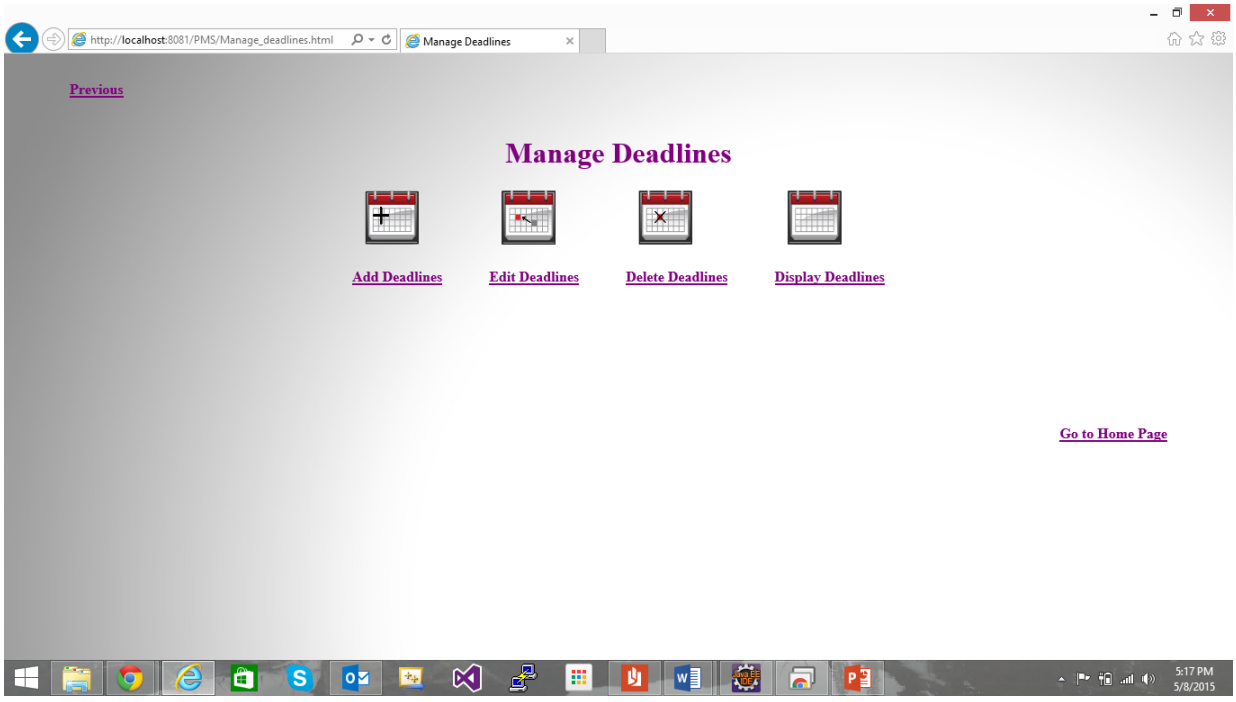
n. Add Deadlines Achieved: Enter Deadline ID that is Achieved: Here the user will add the deadlines achieved by the group by entering the Deadline ID.



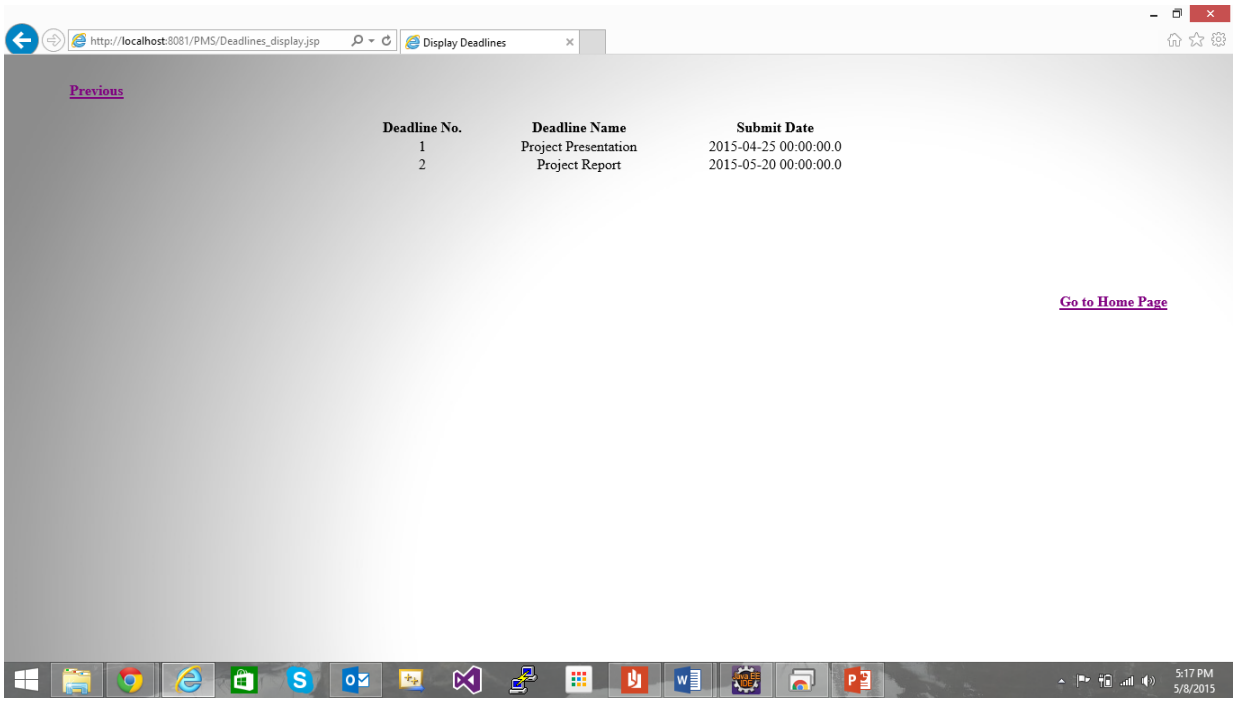
o. Display Deadlines Achieved by a Group: This screen will display the deadlines achieved by a particular group.



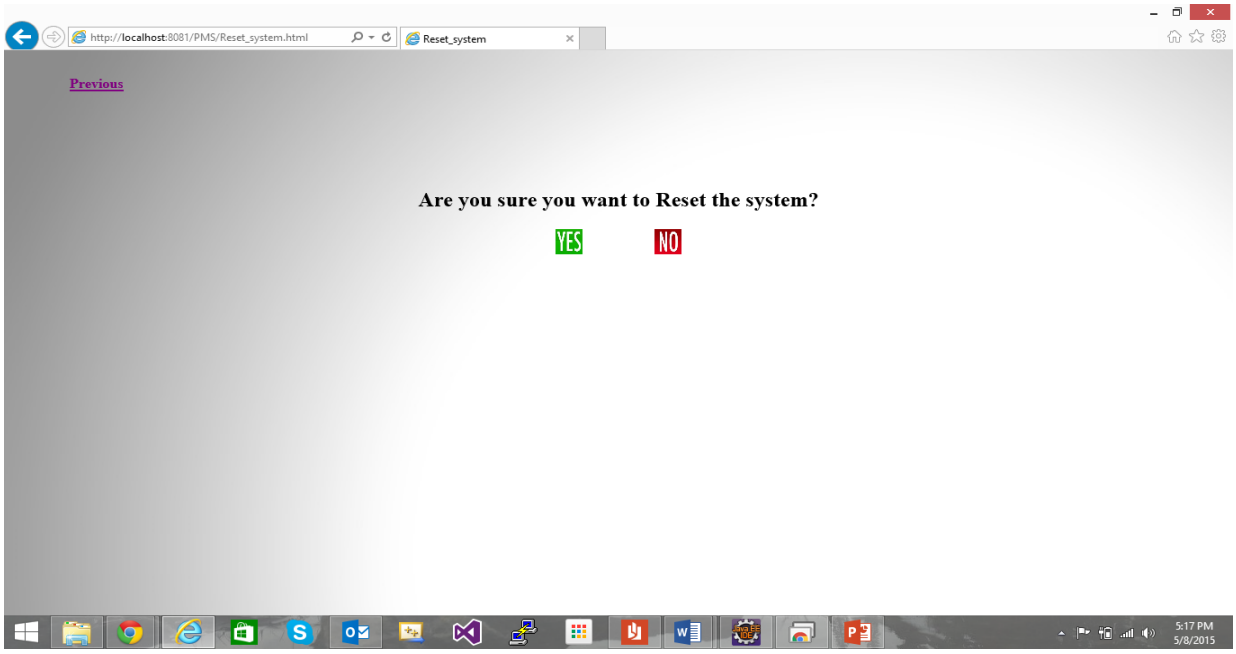
p. Manage Deadlines: The user can access the links to Add, Edit, Delete and Display deadlines on this screen.



q. Display Existing Deadlines: The user can view all the deadlines added into the database.



r. Reset System: The user can reset the system by click on the Reset system link on the home screen. This will delete all the information entered into the database.



9. Sample Code for Displaying Student Details

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<%@ page import="java.sql.*"%>
<%@ page import="oracle.jdbc.pool.OracleDataSource"%>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Deadlines Achieved</title>
</head>
<link rel="stylesheet" type="text/css" href="Home_page.css">
<body class="center">
<table><tr><td><a href="Track_project_status.html"><h4 align="left">Previous</h4></a></td></tr></table>
        <%
int group_id=Integer.parseInt(request.getParameter("group_id"));
int deadline_catched[] = new int[50];
int i=0,k=0,l=0;
String dname[]=new String[200];
Connection con1 = null;
Connection con2 = null;


        try {
                        //step1 load the driver class
                        Class.forName("oracle.jdbc.pool.OracleDataSource");
                        //step2 create  the connection object
                        con1 = DriverManager.getConnection(
                                        "jdbc:oracle:thin:@//127.0.0.1:1521/XE", "system",
                                        "parita");

                        con2 = DriverManager.getConnection(
                                        "jdbc:oracle:thin:@//127.0.0.1:1521/XE", "system",
                                        "parita");


                        //step3 create the statement object
                        Statement stmt1 = con1.createStatement();
                        Statement stmt2 = con2.createStatement();


                        String query1 = "select deadid from deadlines_achieved where group_id =
"+group_id+"";

                        ResultSet rs1=stmt1.executeQuery(query1);

                        while(rs1.next())
                        {
                                deadline_catched[i]=rs1.getInt("deadid");
                                i++;
                        }
```

```
%>
                    <table align="center">
                    <tr>
                    <th><h1>Deadlines Achieved</h1></th>
                    </tr>
<%
                    //----------------------------------------
                    for(int j=0;j<i;j++)
                    {
                              String query2="select deadline_name from deadline_details where
                              deadid="'+deadline_catched[j]+"'";
                              ResultSet rs2=stmt2.executeQuery(query2);
                              rs2.next();
%>
                              <tr>
                              <td><%=rs2.getString("deadline_name") %></td>
                              </tr>
<%
                    }
%>
                    </table>
<%
                    stmt2.close();
                    stmt1.close();


                    //step5 close the connection object

                    con2.close();
                    con1.close();
          }

               catch (Exception e)
               {
                    System.out.println(e);
               }
          %>
<br><br><br><br><br><br><br><br><br><br><br><table align="right"><tr><td><a href="Home_page.jsp"><h4
align="left">Go to Home Page</h4></a></td></tr></table>
</body>
     </html>
```

10. Sample Test Cases

Objective	Pre-condition	Steps	Description	Test Data	Expected Result	Actual Result	Status
Verify the Login functionality with valid credentials.	Login Page should be available	Step-1	Navigate to Login Page		Login Page should be displayed.	Login Page displayed.	Pass
		Step-2	Enter valid user name in user name text box	User Name	Entered user name should be display.	Entered user name displayed.	Pass
		Step-3	Enter valid password in password text box	Password	Entered Password should be display in Password text box in encrypted format.	Entered Password displayed in Password text box in encrypted	Pass
		Step-4	Click on Submit button		User should be login successfully.	User successfully	Pass
To Register new students	Navigational Button should be available	Step-1	Navigate to Add Student Page				Pass
		Step-2	Insert Data into fields	Fields		Valid data inserted into	Pass
		Step-3		Condition	Student Registered	Student Registered	Pass
		Step-4	Click on OK button		Main Menu Displayed	Main Menu Displayed	Pass
To display all deadlines	Navigational Button should be available	Step-1	Navigate to Main Menu				Pass
		Step-2	Select Track Project status and display	Deadlines	Displayed all deadlines	Deadlines displayed	Pass
To create group	Students should be registered	Step-1	Navigate to Main Menu				Pass
		Step-2	Select Create group		Enter details page displayed	Page displayed for entering data	Pass

Table 10.1 Sample Unit Test Cases

11. Current Status

- User can add, delete and retrieve user details and login into the system using the login credentials.
- He can add, edit and delete student details and project details.
- User can add, edit, delete and display deadlines for all projects.
- User can track the status of each project and mark if a project has fulfilled a deadline.
- User will be able to see recent deadlines on the home screen.

- User can reset the system.

12. Next steps

- Generate Report containing all details.
- Work on GUI
- Validate some fields.
- Final Report

13. Future

- Email functionality for professor.
- Create App for Android/iOS

14. Lessons Learnt from 230

- We learnt various Software development lifecycle models. We learnt models like Waterfall model, Extreme Programming, Agile Model and many more. This helped us to choose which model was best for us to follow for development of our project.
- We learnt to document various stages of our project development. Starting from Requirement Gathering to Analysis, Development and Testing and so on.
- We learnt how to prepare UML diagrams according to our project specifications.
- We learnt how to perform unit, functional, stress and system testing.
- Real world examples given by professor helped us a lot in understanding the situations outside of the university and train ourselves for them.
- We learnt time management and team work.

15. Conclusion

- This software will help the professor in managing the projects of a class.
- He will be able to store all the details regarding students, projects and various intermediate deadlines.
- Mainly, he will be able to track the progress of each project.
- He will also be able to generate report containing all the above details.

- He will be able to reset the system and wipe off all data from the database and it will be ready to use for the next class.
- We have tried our best to make the process of managing projects easy for the professor using dedicated software.

16. Acknowledgments

- We would like to thank Professor Doan Nguyen for this guidance and encouragement throughout the project. He always helped in all possible ways.