

DEGREE PLAN AUTOMATION

DELIVERABLE 4

Submitted by:
Sharanya Gottimukkula
Nanditha Bodanapu
Sriharshini Vallabhaneni
Aravind Thottempudi

CSCE 5430 - Software Engineering
Deliverable – 4

a. Requirements Destined for Development Phase – II

Functional Requirement	Development Phase	Start Date	End Date
FREQ-9	Phase 2	10/30/2018	11//03/18
FREQ-11	Phase 2	10/30/2018	11//03/18
FREQ-13	Phase 2	11/04/2018	11/06/2018

Implemented Requirements:

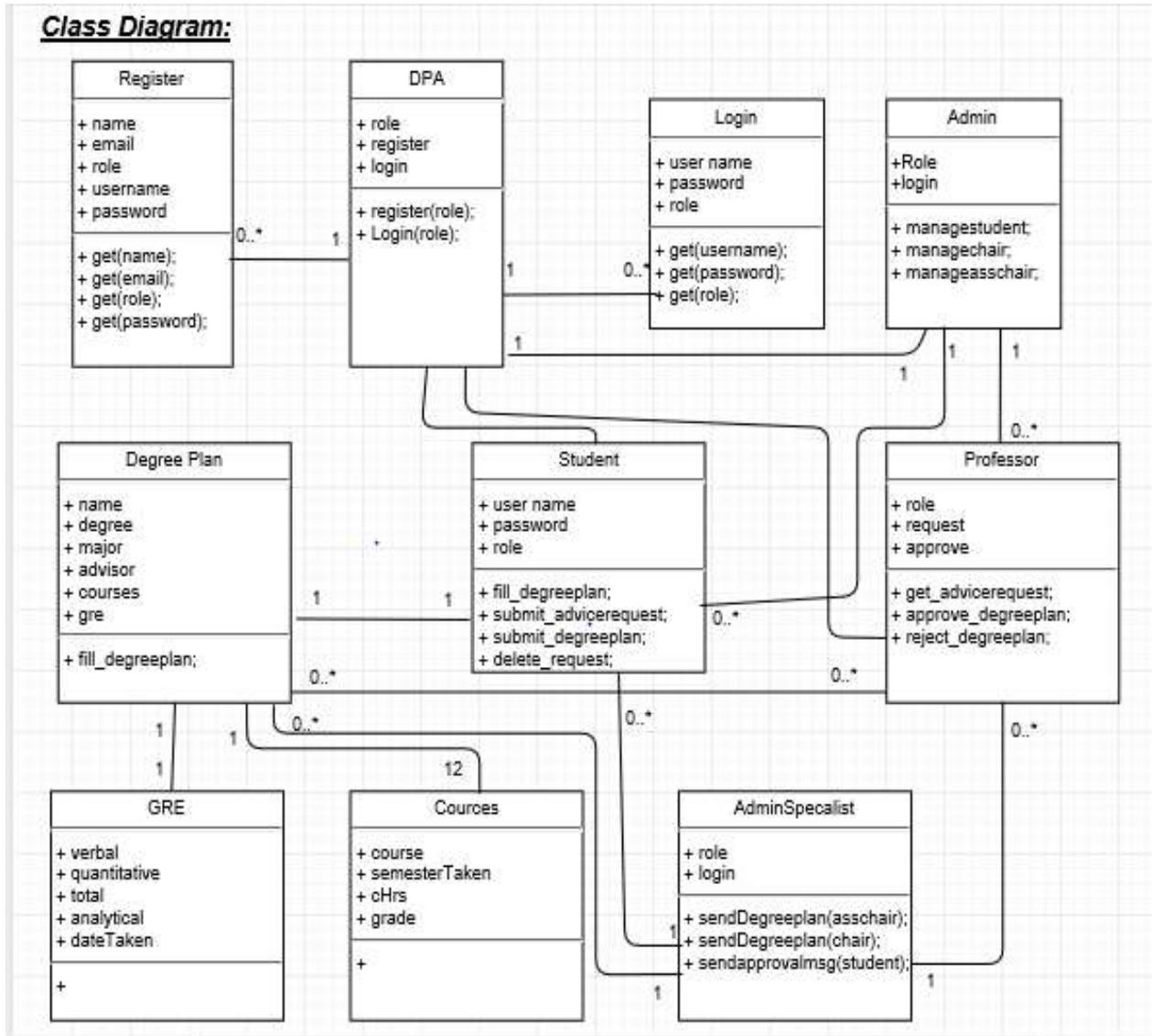
FREQ-9: Approve/Decline Degree Plan - Professor, FREQ -11: Manage Degree Plan – Administrative Specialist, FREQ-13: Account Management. We have successfully implemented all the requirements designated for phase-2 and started implementing the requirements designated for phase-III, as per the schedule listed in the deliverable-3.

Plan for three Development Phases

Functional Requirement	Development Phase	Start Date	End Date
FREQ-1	Phase 1	10/08/2018	10/11/2018
FREQ-2	Phase 1	10/12/2018	10/15/2018
FREQ-3	Phase 1	10/16/2018	10/19/2018
FREQ-5, FREQ-8	Phase 1	10/20/2018	10/25/2018
FREQ-6, FREQ-7	Phase 1	10/26/2018	10/29/2018
FREQ-9, FREQ-11	Phase 2	10/30/2018	11//03/18
FREQ-13	Phase 2	11/04/2018	11/06/2018
FREQ-10	Phase 3	11/06/2018	11/08/2018
FREQ-12	Phase 3	11/09/2018	11/12/2018
FREQ-4	Phase 3	11/13/2018	11/14/2018
FREQ-14	Phase 3	11/15/2018	11/17/2018
FREQ-15	Phase 3	11/18/2018	11/20/2018

b. 1. UML Diagrams

Class Diagram:



Use Case Text:

Main Success Scenario:

- 1: Student registers himself.
- 2: Student logs into the application.
- 3: Student sends advisory request form to professor.
- 4: Professor accepts the advisory request form from the student.
- 5: Student sends the degree plan form for approval to professor.
- 6: Professor approves the degree plan form.
- 7: Professor sends the approved degree plan form to Administrative specialist.
- 8: Administrative Specialist approves and forwards the approved degree plan form to Associate chair
- 9: The approved form from Associate Chair is sent to administrative specialist.

10: Administrative specialist sends the degree plan approved by associate Chair to Chair.

11: The Chair approves the degree plan.

12: The Chair forwards the degree plan approved by him administrative specialist.

Extensions:

4a: Professor rejects the advisory request.

1. If professor rejects the advisor return to step 1

6a: Professor rejects the degree plan.

1. If professor rejects the degree plan, return to step 5.

9a: Administrative Specialist rejects degree plan.

1. If administrative rejects the degree plan go to step 5.

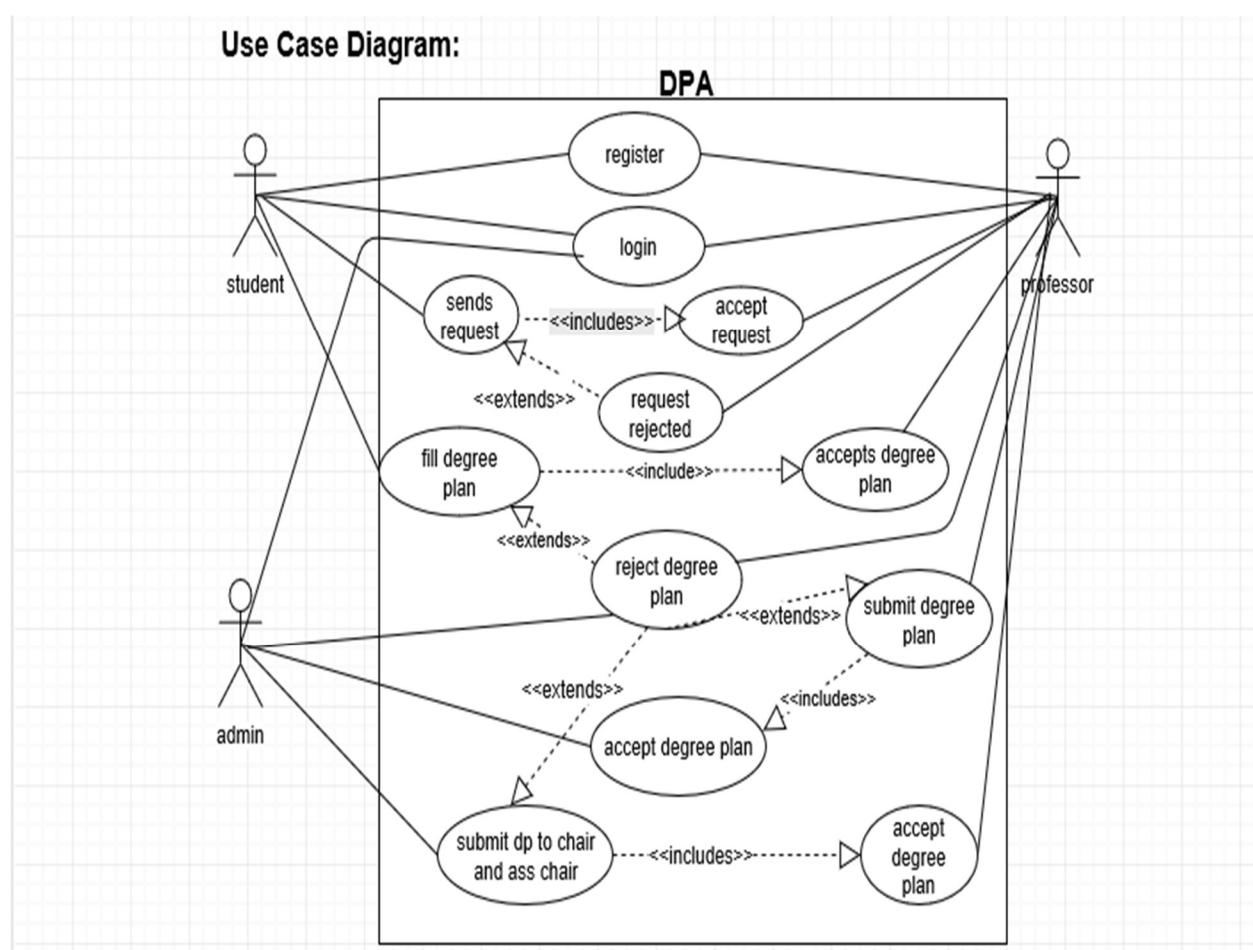
10a: Associate Chair has rejected the degree plan

1. If Associate Chair rejects the degree plan go to step 5.

11a: Chair has rejected the degree plan

1. If Chair rejects the degree plan go to step 5.

2a. User Case Diagram



Use Case Text: UpdateStudentInfo

1: Student registers himself.

2: Student logs into the application.

3: Student can view advisor.

4: Student can view degree plan status.

Extensions:

2a: Student updates account information.

Use Case Text: UpdateProfessorInfo

1: Professor registers himself.

2: Professor logs into the application.

3: Professor can view advisor request sent by students.

4: Professor can view degree plan form submitted by students.

Extensions:

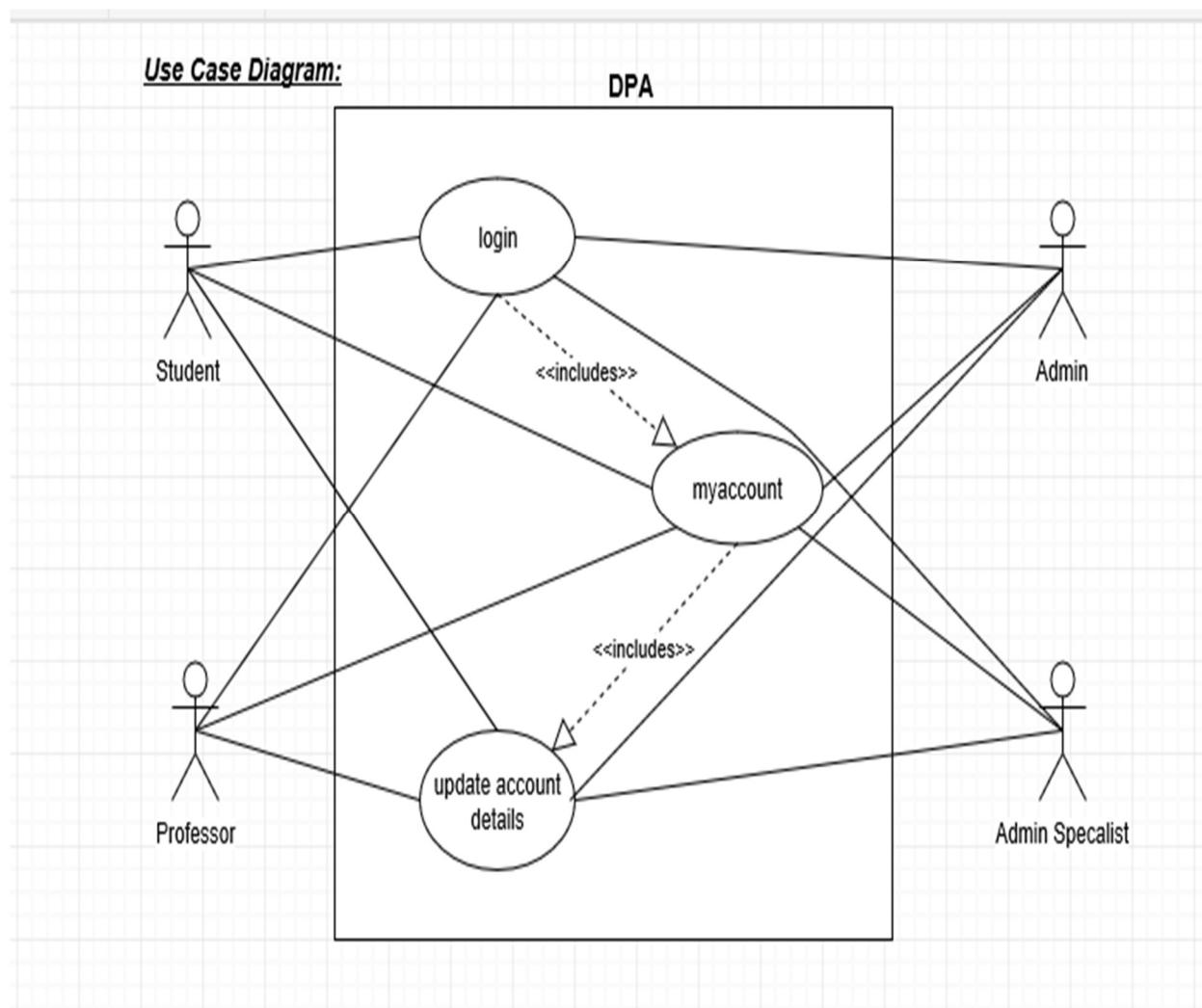
2a: Professor updates account information.

3a: Professor rejects advisory requests

5a: Professor rejects the degree plan.

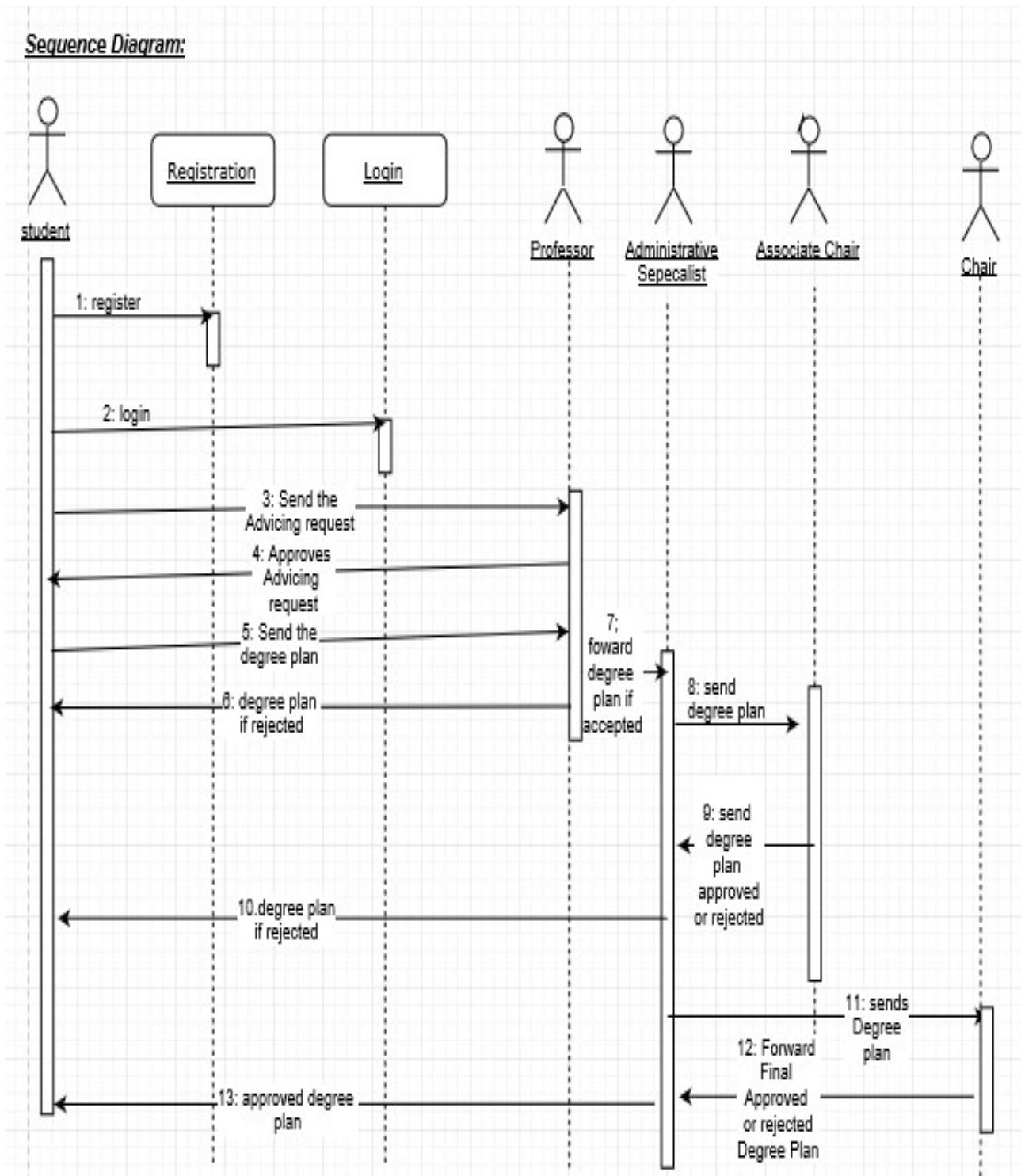
1. Professor writes comments which contains the reason for rejection.

2b. Use Case Diagram:



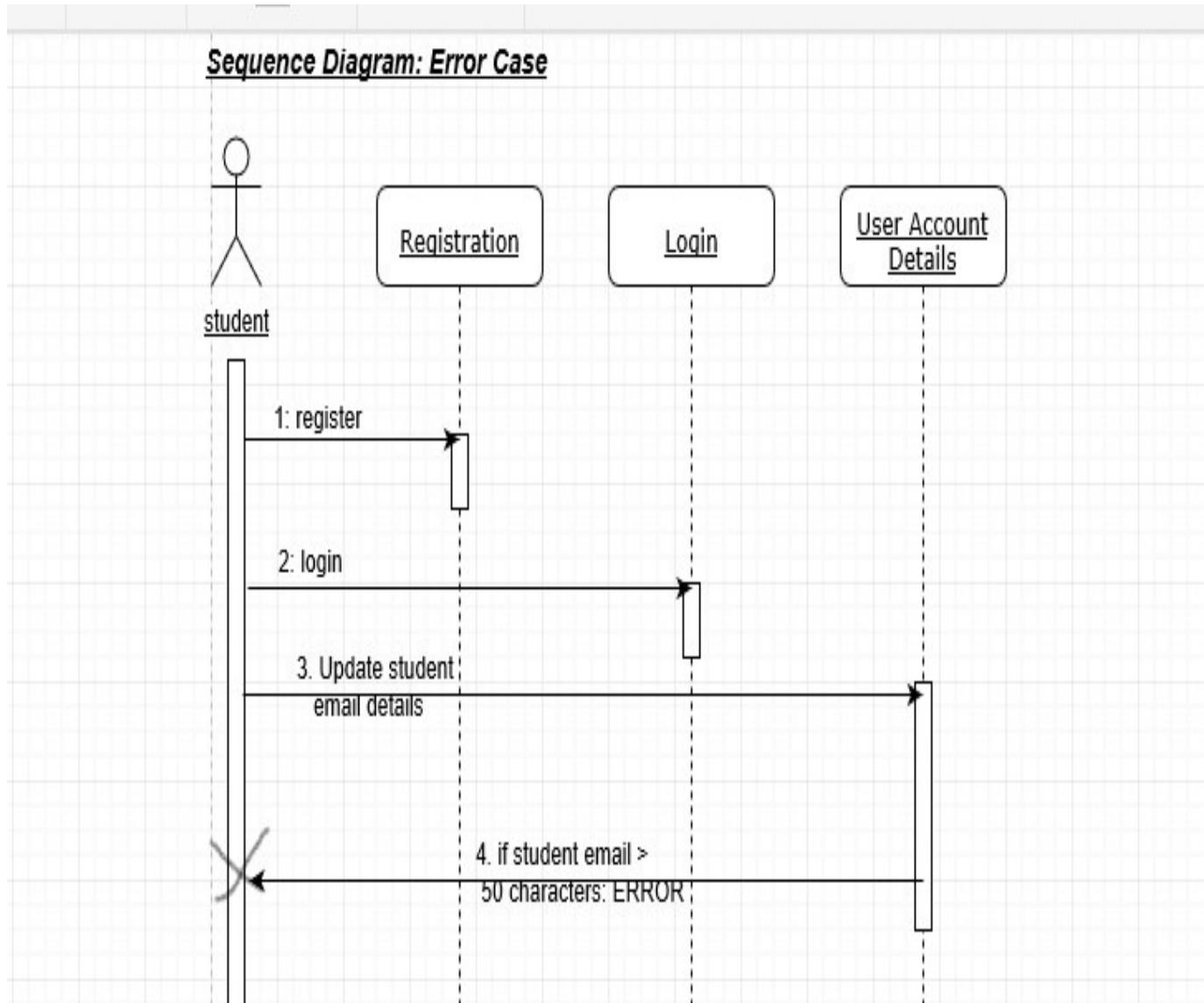
3a. Sequence Diagram: Success scenario

This sequence diagram displays the application flow starting from student registration till approval of degree plan and receiving the approval notice.



3b. Sequence Diagram: Error Case

Student registers himself to the application by entering the required details, now the student can log in to the system using the registered details. The student can update his/her account information using “MyAccount”. The allocated size for email in database is 50 varchar, however if the student enters an email id greater than 50 characters then it would throw an error their by terminating the user execution



c. Test Cases (Unit Tests)

This is test case tests getDegreePlan() method of degreePlanDao that is it checks if the getDegreePlan() method is getting the expected degreeplan i.e the correct degree plant from the database for the given student details. This test takes the student name and professor name as input and expects the output of the getDegreePlan() method to be the degree plan of the correct student. Assert compares expected degree plan with the degree plan returned by getDegreePlan() and if it is same, this test call will pass.^[2]

```
@Test
public void testViewDegreePlan() {
    DegreePlan degreePlan = new DegreePlan();
    when(degreePlanDao.getDegreePlan(anyString(),
        anyString())).thenReturn(degreePlan);
    DegreePlan result = degreePlanService.viewDegreePlan(anyString(),
        anyString());
    assertEquals(degreePlan, result);
}
```

This is test case tests getGREScores() method of DegreePlanDao class that is it checks if the getGREScores() method is getting correct gre scores for the given student name. This test takes the student name as input and compares it with the gre scores returned by getGREScores() and if it same as the expected, this test will pass.^[2]

```
@Test
public void testGetGREScores() {
    GRE gre = new GRE();
    when(degreePlanDao.getGREScores(anyString())).thenReturn(gre);
    GRE result = degreePlanService.getGREScores(anyString());
    assertEquals(gre, result);
}
```

This test case checks if the user getCourses() method of DegreePlanDao class. It takes the name of the student as input and checks if the returns course listed of the student is the expected course list.^[2]

```
@Test
public void testGetCourses() {
    List<Courses> mockCourseList = new ArrayList<Courses>();

    when(degreePlanDao.getCourses(anyString())).thenReturn(mockCourseList);
    List<Courses> result = new ArrayList<Courses>();
    result = degreePlanService.getCourses("sharanya");
    assertEquals(mockCourseList.size(), result.size());
}
```

This test case checks the submitDPToAdminSpecialist() method called in DegreePlanDao class. This test takes professor name, student name and professor signature as input and then checks if they are properly inserted by comparing the returned result to 1..^[2]

```
@Test
public void testSubmitToAdminSpecialist() {
    when(degreePlanDao.submitDPToAdminSpecialist(anyString(),
        anyString(), anyString())).thenReturn(1);
    int result =
    degreePlanService.submitToAdminSpecialist(anyString(), anyString(),
        anyString());
    assertEquals(1, result);
```

```
}
```

This test case checks if the length of the hashed password is equal to the expected hash length or not. It calls the hashString() method by duplicate password and hashing algorithm and checks if the length of the hashed password returned by the method is equal to the actual length 32. If it is same the test case will pass^[1]

```
@Test
public void testHashCodeLength() throws NoSuchAlgorithmException,
UnsupportedEncodingException {
    Hashing hashTest = new Hashing();
    String hashedValue = hashTest.hashString("Test@432", "MD5");
    int length = hashedValue.length();
    assertEquals(32, length);
}
```

This test checks if the dpProfessorReject() method is returning 1. This test case calls the dpProfessorReject() by passing in the professor name, student name and reject comments as input and checks if the returned result is equal to 1^[2]

```
@Test
public void testDPPprofessorreject() {
    when(degreePlanDao.dpProfessorReject(anyString(), anyString(),
anyString())).thenReturn(1);
    int result = degreePlanService.dpProfessorreject(anyString(),
anyString(), anyString());
    assertEquals(1, result);
}
```

This test tests the getDPStatus() method of DegreePlanDao class. It takes in student name and professor name as inputs and returns a string. This test checks if the returned status for the input values is equal to the expected values. ^[2]

```
@Test
public void testGetDPStatus() {
    String Status = "With Professor";
    when(degreePlanDao.getDPStatus(anyString(),
anyString())).thenReturn(Status);
    String actualResult = degreePlanService.getDPStatus(anyString(),
anyString());
    assertEquals(Status, actualResult);
}
```

This test case tests the getComments() method of the DegreePlanDao class. It takes in the student name and professor name as input and gets the reject comments of that particular student. This test checks if the returned comments are equal to the expected comments and if so the test case will succeed ^[2]

```
@Test
public void testGetComments() {
    when(degreePlanDao.getComments(anyString(),
anyString())).thenReturn("Please change core course 3");
    String result = degreePlanService.getComments(anyString(),
anyString());
    assertEquals("Please change core course 3", result);
}
```

This test case tests the getDegreePlan() method of the DegreePlanDao class. It sends username as input to the method getDegreePlan() and expects the output to be the degreePlan if the result returned by getDegreePlan () matches the degreePlan the test case will pass^[2]

```

@Test
public void testGetDegreePlan() {
    DegreePlan degreePlan = new DegreePlan();

    when(degreePlanDao.getDegreePlanData(anyString())).thenReturn(degreePlan);
    DegreePlan result = degreePlanService.getDegreePlan(anyString());
    assertEquals(degreePlan, result);
}

```

This test tests the updateDegreePlan() method of the DegreePlanDao. This test takes in the student id and student name as inputs and expects the result to be 1, if the result is 1 the test case succeeds.^[2]

```

@Test
public void testUpdateDegreePlan() {
    when(degreePlanDao.updateDegreePlan(any(),
        anyString())).thenReturn(1);
    int result = degreePlanService.updateDegreePlan(any(),
        anyString());
    assertEquals(1, result);
}

```

This test tests the getDegreePlans() method of the DegreePlanDao. This test takes checks the status and gets the degree plans matching that particular degree plan status. If the returned degree plan list if equal to the expected degree plans list the test succeeds.^[2]

```

@Test
public void testGetDegreePlans() {
    List<DegreePlan> degreePlans = new ArrayList<DegreePlan>();
    when(degreePlanDao.getDegreePlans()).thenReturn(degreePlans);
    List<DegreePlan> result = degreePlanService.getDegreePlans();
    assertEquals(degreePlans, result);
}

```

This case tests the getReceivedDegreePlan() method of the DegreePlanDao class. It takes in the student name and student id as the inputs and returns the degree plan associated with that particular student name and id. If the expected degree plan and actual returned degree plan are same the test case succeeds.

```

@Test
public void testGetReceivedDP() {
    DegreePlan dp = new DegreePlan();
    when(degreePlanDao.getReceivedDegreePlan(anyString(),
        anyString())).thenReturn(dp);
    DegreePlan result =
    degreePlanService.getReceivedDP(anyString(), anyString());
    assertEquals(dp, result);
}

```

This test case tests the submitDPToAssociateChair() method of the DegreePlanDao class. It takes in the student Id and student name as inputs and checks if the degree plan is properly submitted to the associate chair by checking the returned result against 1, if equal the test succeeds.^[2]

```

@Test
public void testSubmitToAssociateChair() {
    when(degreePlanDao.submitDPToAssociateChair(anyInt(),
        anyString())).thenReturn(1);
}

```

```

        int result = degreePlanService.submitToAssociateChair(anyInt(),
anyString());
assertEquals(1, result);
}

```

This test case tests the receivedDegreePlansLevel2() method of the DegreePlanDao class. It takes in the degree plan status as the input and returns the degree plans with the status equal to the input status. If the returned degree plans list is equal to the actual list then the test succeeds.

```

@Test
public void testGetReceivedDegreePlansLevel2() {
    List<DegreePlan> mockDegreePlans = new ArrayList<DegreePlan>();
    when(degreePlanDao.receivedDegreePlansLevel2(anyString())).thenReturn(mockDegreePlans);
    List<DegreePlan> result =
degreePlanService.getReceivedDegreePlansLevel2(anyString());
    assertEquals(mockDegreePlans, result);
}

```

This test case tests the submitDPToASL2() method of the DegreePlanDao class. It takes in the professor name, student name and associate chair signature as inputs and checks if the degree plan is correctly submitted by comparing the result against 1, if so the test succeeds.

```

@Test
public void testSubmitToASL2() {
    when(degreePlanDao.submitDPToASL2(anyString(), anyString(),
anyString())).thenReturn(1);
    int result = degreePlanService.submitToASL2(anyString(),
anyString(), anyString());
    assertEquals(1, result);
}

```

This test case tests the submitDPToChair() method of the DegreePlanDao class. It takes in student Id and student name as inputs and checks if the degree plan is submitted to the chair successfully. It compares the returned result against 1 and if equal the test case succeeds.

```

@Test
public void testSubmitToChair() {
    when(degreePlanDao.submitDPToChair(anyInt(),
anyString())).thenReturn(1);
    int result = degreePlanService.submitToChair(anyInt(),
anyString());
    assertEquals(1, result);
}

```

This test tests the receivedDegreePlansLevel3() method of the DegreePlanDao class. It takes in the degree plan status as input and checks if the returned degree plans list is same as the expected. If so the test case succeeds.

```

@Test
public void testGetReceivedDegreePlansLevel3() {
    List<DegreePlan> degreePlans = new ArrayList<DegreePlan>();

    when(degreePlanDao.receivedDegreePlansLevel3(anyString())).thenReturn(degreePlans);
    List<DegreePlan> result =
degreePlanService.getReceivedDegreePlansLevel3(anyString());
}

```

```

        assertEquals(degreePlans, result);
    }
}

```

This test tests the submitToASL3() method of the DegreePlanDao class. It takes in Professor name, student name, signature as inputs and checks if the degree plan is correctly submitted to the admin specialist or not. It compares the returned result with 1 and if equal the test case succeeds.

```

@Test
public void testSubmitToASL3() {
    when(degreePlanDao.submitDPToASL3(anyString(), anyString(),
anyString())).thenReturn(1);
    int result = degreePlanService.submitToASL3(anyString(),
anyString(), anyString());
    assertEquals(1, result);
}

```

This test tests the testSendApprovaltoStudent() method of the DegreePlanDao class. It takes in the student id and student name as input and expects the result to be 1. If equal the test case succeeds.

```

@Test
public void testSendApprovaltoStudent() {
    when(degreePlanDao.sendApprovaltoStudent(anyInt(),
anyString())).thenReturn(1);
    int result = degreePlanService.sendApprovaltoStudent(anyInt(),
anyString());
    assertEquals(1, result);
}

```

This testGetCoreCourses() tests getCoreCourses() method of DegreePlanDaoImpl.java . The getCoreCourses() method returns List of core courses from the mandatorycourses table. This test case mocks the Jdbc template to return a list object of type course. If the actualCourseList and the mockCourseList values are equal then the assert statement returns true, if not the test case will fail.

```

@Test
public void testGetCoreCourses() {
    List<Course> mockCourseList =new ArrayList<Course>();
    mockCourseList.add(new Course());

    when(mockJDBCTemplate.query(Matchers.anyString(),
Matchers.any(CourseRowMapper.class)))
    .thenReturn( mockCourseList);

    DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
    List<Course> actualCourseList=mockObj.getCoreCourses();
    Assert.assertEquals(actualCourseList, mockCourseList);
}

```

This testGetOptionalCourses() tests getOptionalCourses() method of DegreePlanDaoImpl .java. The getOptionalCourses() method returns list of optional courses from the courses table. This test case mocks the Jdbc template to return a list object of type String. If the actualStringList and the mockStringList values are equal then the assert statement returns true, if not the test case will fail.

```

    @Test
    public void testGetOptionalCourses() {

        List <String> mockStringList=new ArrayList<String>() ;
        mockStringList.add(new String());
        when(mockJDBCTemplate.query(Matchers.anyString(),
        Matchers.any(OptionalCourseRowMapper.class)))
        .thenReturn( mockStringList);

        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        List actualStringList=mockObj.getOptionalCourses();
        Assert.assertEquals(actualStringList,mockStringList);

    }

```

This testGetDegreePlan() tests getDegreePlan(String sName, String userName) method of DegreePlanDaoImpl.java . The getDegreePlan (String sName, String userName) method accepts two input parameters of type string , one for student name and the other for username of student and returns an object of the DegreePlan.java (which is a model). This test case mocks the Jdbc template to return a DegreePlan object. If the actualDegreePlan and the mockDegreePlan values are equal then the assert statement returns true, if not the test case will fail.

```

    @Test
    public void testGetDegreePlan() {

        DegreePlan mockDegreePlan =new DegreePlan();

        when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
        Matchers.any(BeanPropertyRowMapper.class)))
        .thenReturn( mockDegreePlan);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        DegreePlan actualDegreePlan =mockObj.getDegreePlan("sName",
        "userName");
        Assert.assertEquals(mockDegreePlan, actualDegreePlan);

    }

```

This testGetGREScores() tests getGREScores(String sName)method of DegreePlanDaoImpl.java. The getGREScores(String sName)method accepts an input parameters of type string , for student name and returns an object of the GRE.java (which is a model). This test case mocks the Jdbc template to return a GRE object. If the actualgre and the mockgre values are equal then the assert statement returns true, if not the test case will fail.

```

    @Test
    public void testGetGREScores() {

        GRE mockgre = new GRE();

        when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
        Matchers.any(BeanPropertyRowMapper.class)))
        .thenReturn( mockgre);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        GRE actualgre =mockObj.getGREScores("sName");
        Assert.assertEquals(mockgre, actualgre);

```

```
}
```

This testGetCourses() tests getcourses(String sName) method of DegreePlanDaoImpl.java. The getcourses(String sName)method accepts an input parameters of type string , for student name and returns an List object to type Courses. This test case mocks the Jdbc template to return a List object to type Courses. If the actualCourselist and the mockCourselist values are equal then the assert statement returns true, if not the test case will fail.

```
@Test
public void testGetCourses() {
    List<Courses> mockCourselist =new ArrayList<Courses>();
    mockCourselist.add(new Courses());
    when(mockJDBCTemplate.query(Matchers.anyString(),
Matchers.any(CoursesRowMapper.class)))
        .thenReturn( mockCourselist);
    DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
    List<Courses> actualCourselist = mockObj.getcourses("sName");
    Assert.assertEquals(actualCourselist, mockCourselist);
}
```

This testSubmitDPToAdminSpecialist() tests submitDPToAdminSpecialist(String userName, String sName, String sign)method of DegreePlanDaoImpl.java . The submitDPToAdminSpecialist(String userName, String sName, String sign)method accepts three input parameters of type string, for username, student name and signature, and returns an Integer value which denotes the success/failure of the update statement (1-denotes success, 0 -denotes failure). This test case mocks the Jdbc template to return a integer. If the mockval and the actualresult values are equal then the assert statement returns true, if not the test case will fail.

```
@Test
public void testSubmitDPToAdminSpecialist() {
int mockval=1;
when(mockJDBCTemplate.update(Matchers.anyString(),objCap.capture()))
    .thenReturn(mockval);
DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
int actualresult= mockObj.submitDPToAdminSpecialist("userName", "sName",
"sign");
Assert.assertEquals(actualresult, mockval);
}
```

This testGetDPStatus() tests getDPStatus(String userName, String majorProfessor) method of DegreePlanDaoImpl.java. The getDPStatus(String userName, String majorProfessor)method accepts two input parameters of type string , for username, majorProfessor name, and returns an string value which denotes the status of the degree plan. This test case mocks the Jdbc template to return an String. If the mockval and the Actualval values are equal then the assert statement returns true, if not the test case will fail.

```
@Test
public void testGetDPStatus() {
    String Mockval="str";

```

```

        when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
 Matchers.any(Class.class))).thenReturn(Mockval);
 DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();

 String Actualval=mockObj.getDPStatus("userName", "majorProfessor");
 Assert.assertEquals(Actualval, Mockval);

}

```

This testGetComments() tests getComments(String userName, String majorProfessor) method of DegreePlanDaoImpl.java. The getComments(String userName, String majorProfessor) method accepts two input parameters of type string, for username, majorProfessor name, and returns an string value which contains comments issued by the major professor. This test case mocks the Jdbc template to return a String. If the Mockval and the Actualval values are equal then the assert statement returns true, if not the test case will fail.

```

@Test
public void testGetComments() {

    String Mockval="str";
    when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
 Matchers.any(Class.class))).thenReturn(Mockval);
    DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();

    String Actualval=mockObj.getComments("userName", "majorProfessor");
    Assert.assertEquals(Actualval, Mockval);

}

```

This testGetDegreePlanData() tests getDegreePlanData(String userName) method of DegreePlanDaoImpl.java. The getDegreePlanData(String userName) method accepts an input parameters of type string, for username and returns an object of the DegreePlan.java (which is a model). This test case mocks the Jdbc template to return a DegreePlan object. If the actualDegreePlan and the mockDegreePlan values are equal then the assert statement returns true, if not the test case will fail.

```

@Test
public void testGetDegreePlanData() {

DegreePlan mockDegreePlan =new DegreePlan();

when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
 Matchers.any(BeanPropertyRowMapper.class)))
 .thenReturn( mockDegreePlan);
DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
DegreePlan actualDegreePlan =mockObj.getDegreePlanData( "userName");
Assert.assertEquals(mockDegreePlan, actualDegreePlan);

}

```

d. Team Member Contribution

Name of the Member	Components Developed	Overall Contribution (%)
Sharanya Gottimukkula	<p>FREQ-11</p> <p>Designed HTML pages that allows admin specialist to verify the students degree plan and send it to chair and associate chair.</p> <p>Involved in writing viewdegreeplan.jsp, DegreePlanServiceImpl and DegreePlanDaoImpl classes which handles the degree plan related request and responses and data flow</p>	25 %
Nanditha Bodanapu	<p>FREQ-13</p> <p>Designed HTML pages that allow users to view their account information and also the page that allows the users to update information.</p> <p>Involved in writing updateAccountInformation() method in all the user controllers (StudentController, ProfessorController, AdminController and AdminSpecialistController()).</p>	25 %
Aravind Swamy Thottempudi	<p>FREQ-11</p> <p>Involved in creating the database tables required for the degree plan management by admin specialist and also in writing few methods in DegreePlanServiceImpl class as a part of degree plan management</p>	25 %
Sriharshini Vallabhaneni	<p>FREQ-9</p> <p>Involved in writing the approveDegreePlan() and rejectDegreePlan() methods in the degreePlanController which allows the professor, admin specialist , chair and associate chair to approve or decline the degree plan.</p>	25 %

e. User Manual

Degree Plan Automation System:

This is web application. In order to use this system, user needs to install the below software:

Eclipse

MYSQL workbench and server

Google chrome

Type the URL “localhost:8080/dpa/login” in the browser and hit enter to access the main page of the DPA system.

This page has login fields and a link to new user registration page and forgot password page.



The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:8080/dpa/login (highlighted with a red oval)
- Toolbar:** Back, Forward, Stop, Refresh, Home, Bookmarks, etc.
- Bookmark Bar:** Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
- Header:** DPA CSCE UNT
- Navigation Bar:** Home, DPI, Help
- Login Form:**
 - UserName: Enter UserName
 - Password: Enter password
 - role: Admin (dropdown menu)
 - login button
- Links:** NewUser? Register, Forgot Password
- Page Footer:** © 2018 SHAN CSCE UNT, Contact Us: +1 xxx-xxx-xxxx

The login fields allow admin, student, professor, chair, associate chair and admin specialist to login by typing in their username, password and choosing appropriate role from the dropdown field, which has options as in screenshot below. All the three fields are required. The username and password fields are validated and will not allow special characters.

DPA CSCE UNT

Registered Successfully, Please Login

Login

UserName	sharanya
Password
Role	Admin Admin Student Professor Chair Associate Chair Admin Specialist



By clicking on the “NewUser?Register” link on the main page of the website user will be redirected to registration page

DPA CSCE UNT

Register

Name	Enter Name
Email	Enter Email
Role	Select
UserName	Enter UserName
Password	Enter password
RetypePassword	Re-Enter password
<input type="button" value="register"/>	

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

This registration page allows only students and professors to register. Role dropdown has only two roles student and professor. All the fields are mandatory and have validations, which are notified to the user on focus.

DPA CSCE UNT

Home DPI Help

Register

Name	<input type="text" value="Enter Name"/>	Name should be 5 to 20 characters length
Email	<input type="text" value="Enter Email"/>	
Role	<input type="text" value="Select"/>	
UserName	<input type="text" value="Select"/> <input type="text" value="student"/> <input type="text" value="professor"/>	UserName should be 5 to 20 characters length
Password	<input type="text" value="Enter password"/>	
RetypePassword	<input type="text" value="Re-Enter password"/>	Password length should be between 5 to 15 characters. It should not contain special characters

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

Upon successful registration the user will be redirected to the login page with the success message

DPA CSCE UNT

Home DPI Help

Registered Sucessfully. Please Login

Login

UserName	<input type="text" value="Enter UserName"/>
Password	<input type="text" value="Enter password"/>
role	<input type="text" value="Admin"/>
<input type="button" value="login"/>	

NewUser? Register
Forgot Password

Activate Windows
Go to PC settings to activate Windows.

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

User login as Admin: If you login with role as admin, you will be directed to admin home page where you has options to manage chair, associate chair, students/professors and admin specialist.

DPA CSCE UNT

Home DPI StaffDirectory StudentList Help CSCEUNT_DPA

Manage Users

Select
Manage Chair
Manage Associate Chair
Manage Students/Professors
Manage Admin Specialist

Verify and Approve Students/Professors

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

9:34 AM 10/29/18

By clicking on manage chair you get options to add, delete or update chair. By clicking on manage associate chair you get options to add, delete or update associate chair. By clicking on manage professors/students you get options to delete or update professors/students. By clicking on manage admin specialist you get options to add, delete or update admin specialist.

DPA CSCE UNT

Home DPI StaffDirectory StudentList Help CSCEUNT_DPA

Manage Users

Manage Chair

Verify and Approve Students/Professors

DELETE Chair
UPDATE Chair
ADD Chair

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

By clicking on ADD chair you will be directed to a add chair page where you can fill in the chair details and add him/her.

The screenshot shows the DPA CSCE UNT website with a green header bar. Below the header, there is a navigation menu with links: Home, DPI, StaffDirectory, StudentList, Help, and a dropdown menu labeled CSCEUNT_DPA. The main content area is titled "ADD Chair" and contains five input fields: Name, Email, Role, UserName, and Password. A "adduser" button is located below the password field.

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

Admin can also view the student list and staff directory by clicking on the links “StaffDirectory” and “StudentList” options on the menu bar.

The screenshot shows the DPA CSCE UNT website with a green header bar. The navigation menu includes a link labeled "StudentList" which is circled in red. Below the header, the main content area is titled "Student List" and displays a table with six rows, each containing a student's name, department, and email address, followed by a "Send Message" button. At the bottom of the page, there is a copyright notice and a contact information line.

Name	Department	
Aravind Thotempudi	aravindthotempudi@my.unt.edu	<button>Send Message</button>
George Joseph	georgejoseph@my.unt.edu	<button>Send Message</button>
Nanditha Bodanapu	nandithabodenapu@my.unt.edu	<button>Send Message</button>
Sharanya Gottimukkula	sharanyagottimukkula@my.unt.edu	<button>Send Message</button>
Sri Harshini	sriharshinivallabhaneni@my.unt.edu	<button>Send Message</button>

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

DPA CSCE UNT

Home DPI StaffDirectory Sent Requests Help

CSCE Staff Directory		
Name	Email	
Dr. Armin Mikler	arminmikler@unt.edu	<button>Send Request</button>
Bryant Barett	BryantBarett@unt.edu	<button>Send Request</button>
Robert Akl	roberakl@unt.edu	<button>Send Request</button>

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

You can logout by clicking the button logout present in the dropdown with your username display.

DPA CSCE UNT

Home DPI StaffDirectory StudentList Help

CSCEUNT_DPA ▾
CSCEUNT_DPA
MyAccount
Logout

Manage Users

Select

Verify and Approve Students/Professors

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

User login as Professor: If you login, choosing role as a professor, chair or associate chair you will be directed to a professor home page. Professor home displays list of students, whose request has been accepted by you and also it has a button to view degree plan which is enabled, if your student already submitted the degree plan for approval.

Professor Home Page

DPA CSCE UNT

Home DPI Student List Received Requests Help mikler

My Students

Name	Department	Degree Plan
dolly	CE	Not Yet Submitted
sharanya	CS	View Degree Plan

© 2018 SHAN CSCE UNT Activate Windows
Contact Us: +1 xxx-xxx-xxxx Go to PC settings to activate Windows.

Professor as an associate chair or chair (home page):

If the professor is an Associate Chair or Chair of the department, the home page will also display a list of degree plans received from the admin specialist.

DPA CSCE UNT

Home DPI Student List Received Requests Help robert

My Students

Name	Department	Degree Plan
sharmila	CE	Not Yet Submitted

Degree Plans Received from Admin Specialist

Name	Student ID	View Degree Plan
sharanya	11256784	View Degree Plan

© 2018 SHAN CSCE UNT Activate Windows
Contact Us: +1 xxx-xxx-xxxx Go to PC settings to activate Windows.

You have options to view Student list and received requests by clicking on the options “StudentList” and “ReceivedRequests” on the navigation bar at the top.

The screenshot shows the DPA CSCE UNT application interface. At the top, there is a green header bar with the text "DPA CSCE UNT". Below it is a black navigation bar containing links for "Home", "DPI", "Student List", "Received Requests", and "Help". A user profile dropdown menu is open, showing the name "bryant". The main content area is titled "Student List" and displays a table with five rows of student information. Each row includes the student's name, department, email address, and a "Send Message" button. The table has a light gray background with alternating row colors.

Name	Department	
Aravind Thotempudi	aravindthotempudi@my.unt.edu	Send Message
George Joseph	georgejoseph@my.unt.edu	Send Message
Nanditha Bodanapu	nandithabodanapu@my.unt.edu	Send Message
Sharanya Gottimukkula	sharanyagottimukkula@my.unt.edu	Send Message
Sri Harshini	sriharshinivallabhaneni@my.unt.edu	Send Message

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

You can accept the request by clicking on the Accept Request button and reject the request by clicking on the Reject Request button

The screenshot shows the DPA CSCE UNT application interface. At the top, there is a green header bar with the text "DPA CSCE UNT". Below it is a black navigation bar containing links for "Home", "DPI", "Student List", "Received Requests" (which is highlighted with a red oval), and "Help". A user profile dropdown menu is open, showing the name "bryant". The main content area is titled "Received Requests" and displays a table with two rows of received requests. Each row includes the requester's name, department, admission semester, year, message, and "Accept Request" and "Reject Request" buttons. The table has a light gray background with alternating row colors.

Name	Department	Admission Semester	Admission Year	Message		
george	CS	fall	2017	please accept	Accept Request	Reject Request
sharanya	CS	fall	2017	mmmmm	Accept Request	Reject Request

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

You can approve the degree plan received from students as well as degree plans received from administrative specialist by clicking “Accept Degree Plan” and can reject by clicking “Reject Degree Plan” button.

CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
Admission to candidacy is recommended:		Total Semester Hours Requires: 36	

[Accept DegreePlan](#) [Reject DegreePlan](#)

© 2018 SHAN CSCE UNT

Activate Windows
Go to PC settings to activate Windows.

Contact Us: +1 xxx-xxx-xxxx

The professor needs to sign after clicking “Accept Degree Plan” to approve and send it to the administrative specialist.

Supporting Courses

Course Prefix, Number and Title	Semester Taken	C. Hrs	Grade
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A

Admission to candidacy is recommended: Total Semester Hours Requires: 36

Signature

Please type Full Name

[Submit to AdminSpecialist](#)

© 2018 SHAN CSCE UNT

Activate Windows
Go to PC settings to activate Windows.

Contact Us: +1 xxx-xxx-xxxx

The professor needs to add comments, which allows the student to update the degree plan as per the comments and resubmit it. All this needs to be done when you click “Reject Degree Plan” button.

DPA CSCE UNT

Home DPI Help

Student Name sharanya

Comments Enter comments of rejection

submit



You can logout by clicking the button logout present in the dropdown with your username display.

DPA CSCE UNT

Home DPI Student List Received Requests Help

Logout (highlighted with a red circle)

Student List

Name	Department	
Aravind Thottempudi	aravindthottempudi@my.unt.edu	Send Message
George Joseph	georgejoseph@my.unt.edu	Send Message
Nanditha Bodanapu	nandithabodanapu@my.unt.edu	Send Message
Sharanya Gottimukkula	sharanyagottimukkula@my.unt.edu	Send Message
Sri Harshini	sriharshinivallabhaneni@my.unt.edu	Send Message

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

User login as Student: If you login by choosing the role as student, you will be directed to the student home. Student home displays the major professor name and email and also has an option to start new degree plan and also to check the degree plan status.

The screenshot shows the DPA CSCE UNT student home page. At the top, there is a green header bar with the text "DPA CSCE UNT". Below it is a black navigation bar with links: Home, DPI, StaffDirectory, Sent Requests, and Help. A user profile dropdown menu is open, showing the name "sharanya". The main content area starts with a section titled "My Advisor" which lists a professor's name and email, with a "Send Message" button. Below this is a section titled "My Degree Plan" containing two buttons: "Start New Degree Plan" and "Degree Plan Status".

You can send requests to the professors by clicking on the “staffdirectory” link in the menu and then clicking “send request” button beside the professor.

The screenshot shows the DPA CSCE UNT staff directory page. At the top, there is a green header bar with the text "DPA CSCE UNT". Below it is a black navigation bar with links: Home, DPI, StaffDirectory, Sent Requests, and Help. A user profile dropdown menu is open, showing the name "sharanya". The main content area is titled "CSCE Staff Directory" and lists three professors with their names and emails, each accompanied by a "Send Request" button.

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

Activate Windows
Go to PC settings to activate Windows.

You can view the sent requests by clicking on “sent requests” link in the menu and then click delete if you want to delete the sent request.

DPA CSCE UNT

Home DPI StaffDirectory Sent Requests Help

Sent Requests

Name	Status
Bryant Barrett	Request Sent

Delete Request

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

After clicking the start degree plan button you can choose the major for which you want to file the degree plan. Depending on the major option you choose you will be getting the dropdown courses related to that major.

DPA CSCE UNT

Home DPI StaffDirectory Sent Requests Help sharanya

My Advisor

Name	Email
Dr. Armin Mikler	arminmikler@unt.edu

Send Message

My Degree Plan

Start New Degree Plan

Select

Selected

Computer Science

Computer Engineering

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx

After choosing the major the degree plan form as in below screenshots will be displayed. The major will be auto populated as per your choice of major.

DPA CSCE UNT

[Home](#) [DPI](#) [StaffDirectory](#) [Sent Requests](#) [Help](#)
sharanya ▾

**Master's Degree Plan
Department of Computer Science And Engineering**

Name	<input type="text" value="Enter Name"/>	Student ID (EMP ID)	<input type="text" value="Enter ID"/>		
Local Address	<input type="text" value="Enter Address"/>				
UNT Email ID	<input type="text" value="Email ID"/>				
Degree To Be Earned	<input type="text" value="M.S. (Master of Science)"/>	Major	<input type="text" value="computerScience"/>		
Minor	<input type="text" value="Please enter minor"/>	Interest Area	<input type="text" value="Enter specialization"/>		
Major Professor	<input type="text" value="Enter Major Professor Name"/>	Co-Major Professor	<input type="text" value="Enter Co-Major Professor Name"/>		
Most Recent GRE Scores:	<input type="text" value="Verbal"/>	<input type="text" value="Quantitative"/>	<input type="text" value="Analytical"/>	Date Taken:	<input type="text" value="mm/dd/yyyy"/>

Please select Core Courses

Core Course A	<input type="text" value="CSCE 5450 Programming Languages"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Core Course B	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Core Course C	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Core Course D	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>

Please select Optional Courses

Optional Course 1	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Optional Course 2	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Optional Course 3	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>
Optional Course 4	<input type="text" value="select"/>	semester taken	<input type="text" value="Select"/>	C. Hrs	<input type="text" value="C. Hrs"/>	Grade	<input type="text" value="grade"/>

You have options to save the filled degree plan or to submit the degree plan. Submit also saves the degree plan for you and sends it to your major professor

Optional Course 5	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Optional Course 6	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Optional Course 7	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Optional Course 8	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Total Semester Hours	Enter total semester hours						
				Save	submit		

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

The core course options you get when you choose Computer Science are different from what you get when you choose Computer Engineering.

Core course list when Computer Science is chosen

Degree To Be Earned	M.S.(Master of Science)	Major	computerScience				
Minor	Please enter minor	Interest Area	Enter specialization				
Major Professor	Enter Major Professor Name	Co-Major Professor	Enter Co-Major Professor Name				
Most Recent GRE Scores:	Verbal	Quantitative	Analytical	Date Taken:	mm/dd/yyyy		
Please select Core Courses							
Core Course A	CSCE 5450 Programming Languages select CSCE 5430 Software Engineering	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course B	CSCE 5450 Programming Languages CSCE 5650 Compiler Design	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course C	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course D	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade

Core course list when Computer Engineering is chosen

UNI Email ID	Email ID						
Degree To Be Earned	M.S.(Master of Science)						
Major	computerEngineering						
Minor	Please enter minor						
Interest Area	Enter specialization						
Major Professor	Enter Major Professor Name						
Co-Major Professor	Enter Co-Major Professor Name						
Most Recent GRE Scores:	Verbal	Quantitative	Analytical				
	Date Taken:	mm/dd/yyyy					
Please select Core Courses							
Core Course A	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course B	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course C	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course D	select	semester taken	Select	C. Hrs	C. Hrs	Grade	grade

If the degree plan is rejected the student can view the reject comments, update the degree plan and resubmit the degree plan but clicking on the link “update and resubmit the degree plan”

DPA CSCE UNT

[Home](#) [DPI](#) [StaffDirectory](#) [Sent Requests](#) [Help](#)

My Advisor

Name	Email
Dr. Armin Mikler	arminmikler@unt.edu

[Send Message](#)

My Degree Plan

[Start New Degree Plan](#)

[Degree Plan Status](#)

Chair Rejected

Please change optional course 3

Update and resubmit the Degree Plan

You can make changes in your previous degree plan and click “update” button to update and resubmit the degree plan.

Optional Course 6	CSCE 5350 Fundamentals of Databases	semester taken	spring	C. Hrs	3	Grade	A
Optional Course 7	CSCE 5350 Fundamentals of Databases	semester taken	spring	C. Hrs	3	Grade	A
Optional Course 8	CSCE 5350 Fundamentals of Databases	semester taken	spring	C. Hrs	3	Grade	A
Total Semester Hours	36						

[Save](#) [Update](#)

© 2018 SHAN CSCE UNT

Activate Windows
Go to PC settings to activate Windows.

Contact Us: +1 xxx-xxx-xxxx

You can click on the download the approved degree plan link to download the degree plan approved in the CSCE department

DPA CSCE UNT

Home DPI StaffDirectory Sent Requests Help

sharanya ▾

My Advisor

Name	Email	
Dr. Armin Mikler	arminmikler@unt.edu	Send Message

My Degree Plan

[Start New Degree Plan](#)

[Degree Plan Status](#)

Degree Plan approved in
the CSCE Department

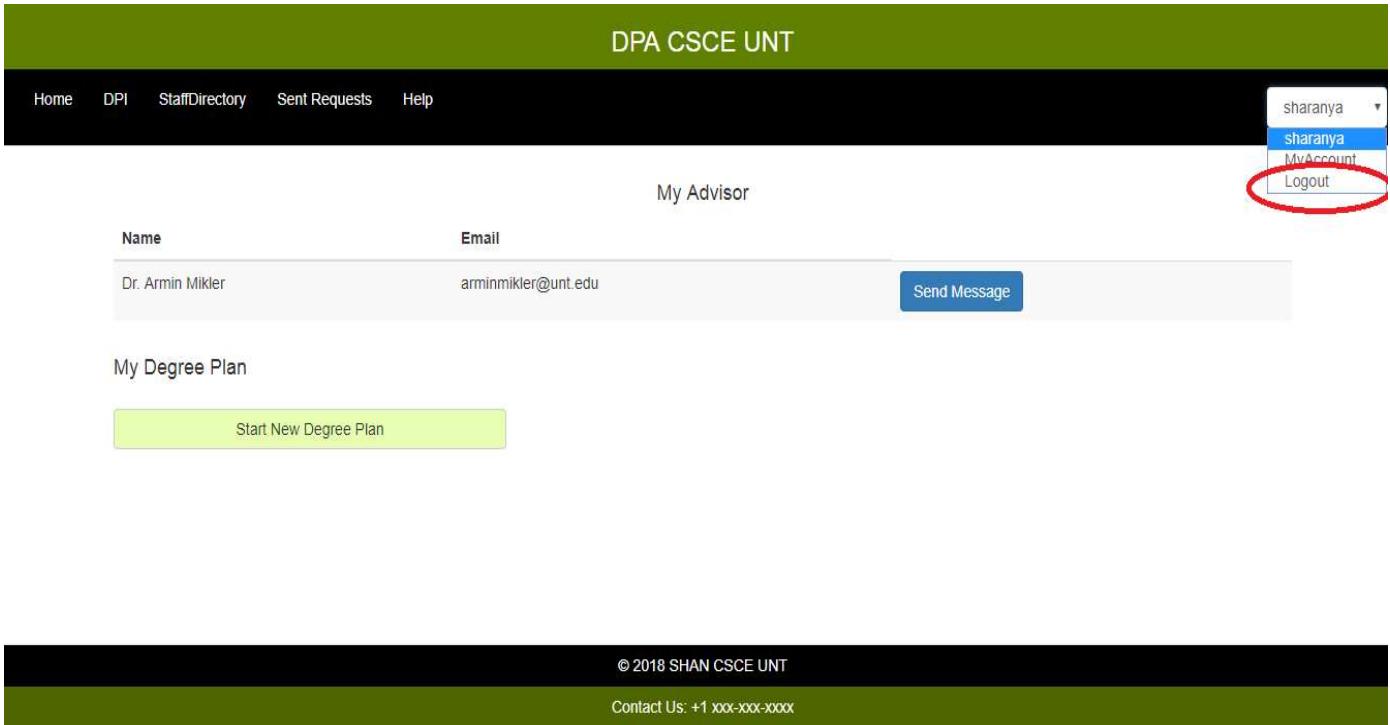
[Download the Approved Degree Plan](#)

© 2018 SHAN CSCE UNT

Activate Windows
Go to PC settings to activate Windows.

Contact Us: +1 xxx-xxx-xxxx

You can logout by clicking the button logout present in the dropdown with your username display.



DPA CSCE UNT

Home DPI StaffDirectory Sent Requests Help

sharanya

sharanya

My Account

Logout

My Advisor

Name	Email
Dr. Armin Mikler	arminmikler@unt.edu

Send Message

My Degree Plan

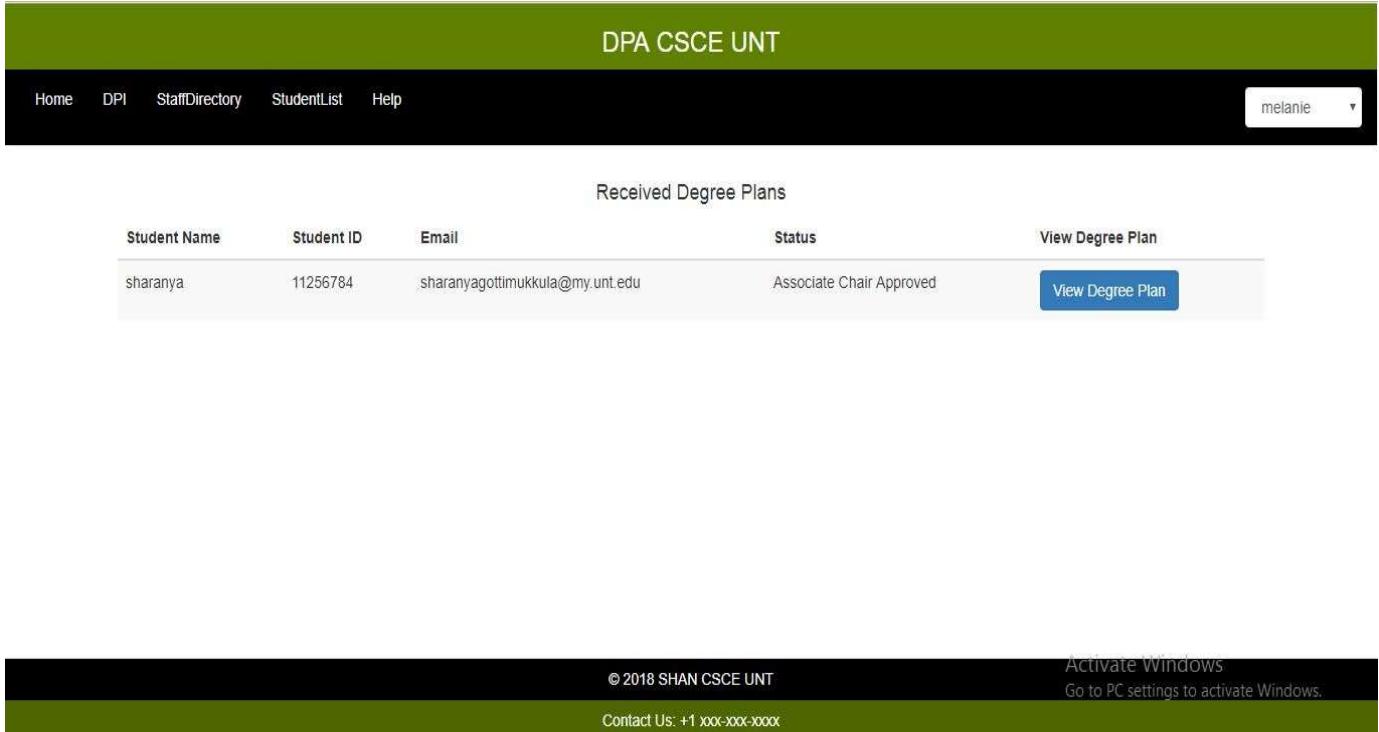
Start New Degree Plan

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

User as Admin Specialist:

If the user logs in with role as an admin specialist the user will be redirected to the admin specialist home. Admin specialist home displays the degree plans received from the professors.



DPA CSCE UNT

Home DPI StaffDirectory StudentList Help

melanie

Received Degree Plans

Student Name	Student ID	Email	Status	Action
sharanya	11256784	sharanyagottimukkula@my.unt.edu	Associate Chair Approved	View Degree Plan

Activate Windows
Go to PC settings to activate Windows.

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

Admin specialist can send the received degree plan to the associate chair and chair for approval by clicking on the button “Submit to Associate Chair” or “Submit to Chair” buttons.

The screenshot shows a software interface for managing degree plans. At the top, there is a table listing courses:

CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A

Below the table, there are two input fields:

- “Admission to candidacy is recommended: _____”
- “Total Semester Hours Requires: 36”

Underneath these fields is another input field: “Advisor or Major Professor: Dr. Armin Mikler”. At the bottom of the screen, there are two buttons: “Submit to Associate Chair” (blue background) and “Reject DegreePlan” (white background). The footer contains copyright information, contact details, and activation links:

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx
Activate Windows
Go to PC settings to activate Windows.

The second part of the screenshot shows a similar interface, but with different input fields:

- “Associate Chair for Graduate Studies: Dr. Robert Akl”
- “Submit to Chair” (blue background)
- “Reject DegreePlan” (white background)

The footer is identical to the first screenshot.

Admin specialist can send the approval notice to the student when the status of degree plan is chair approved that is after the chair approving the degree plan.

The screenshot shows a software interface for managing degree plans. At the top, there are four input fields:

- “Admission to candidacy is recommended: _____”
- “Total Semester Hours Requires: 36”
- “Advisor or Major Professor: Dr. Armin Mikler”
- “Associate Chair for Graduate Studies: Dr. Robert Akl”

Below these fields is another input field: “Department Chair: Dr. Bryant Barrett”. At the bottom of the screen, there is a single button: “Send the Approved Degree Plan to Student” (blue background). The footer contains copyright information, contact details, and activation links:

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx
Activate Windows
Go to PC settings to activate Windows.

View account information and update (All users):

The user can click on the MyAccount option shown in the below screenshot and then view the account details.

The screenshot shows the DPA CSCE UNT website. At the top, there is a green header bar with the text "DPA CSCE UNT". Below it is a black navigation bar with links: Home, DPI, StaffDirectory, Sent Requests, and Help. On the right side of the navigation bar is a "MyAccount" dropdown menu. The menu items are: sharmila, MyAccount (which is highlighted with a red oval), and Logout. The main content area is titled "Account Details" and contains four input fields: Name (sharmila), Email (sharmila@my.unt.edu), Role (student), and UserName (sharmila). Below these fields is a blue "Update Account Details" button. At the bottom of the page, there is a dark footer bar with the text "© 2018 SHAN CSCE UNT", "Contact Us: +1 xxx-xxx-xxxx", "Activate Windows", and "Go to PC settings to activate Windows."

The user can change the editable details and click update account details button to update the details.

This screenshot is identical to the one above, showing the "Account Details" page. However, the "Name" field now contains "sharmila1" instead of "sharmila". All other fields (Email, Role, UserName) remain the same. The "Update Account Details" button is present at the bottom. The footer information at the bottom of the page is also identical to the previous screenshot.

f. Instructions to compile and run both program and Test Cases

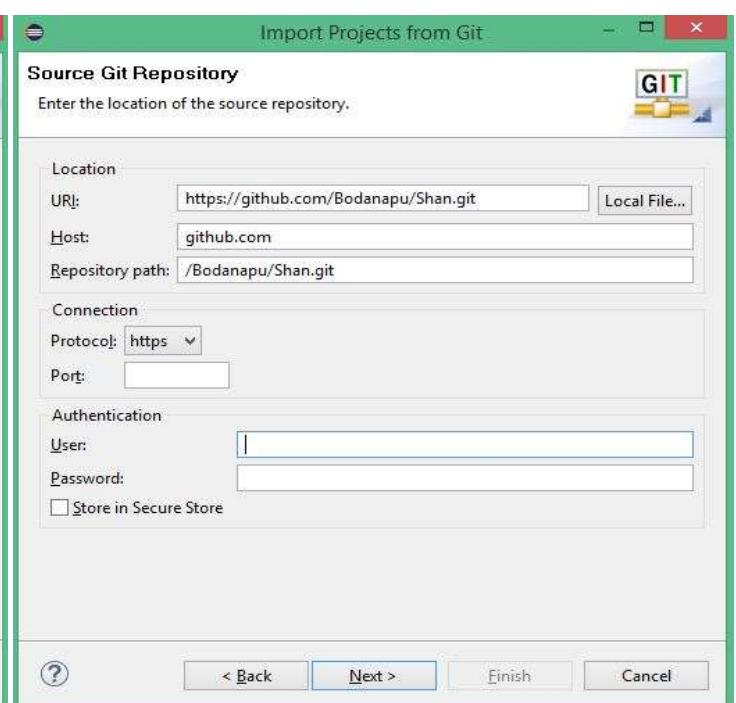
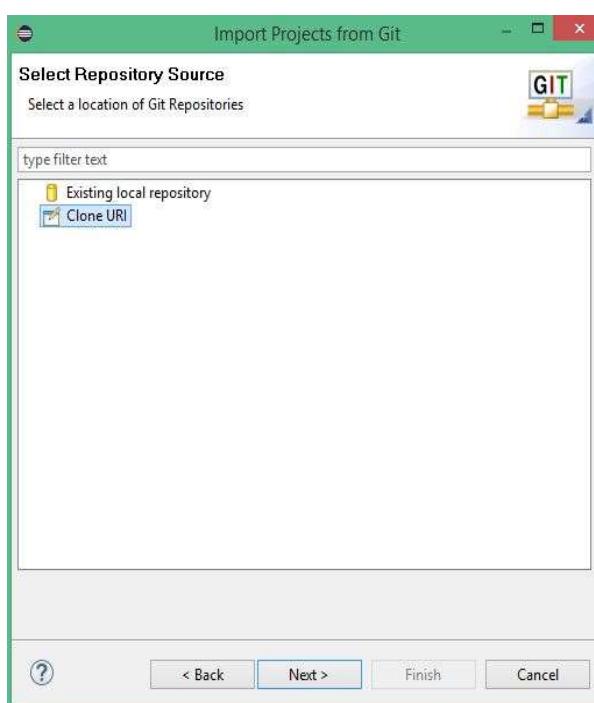
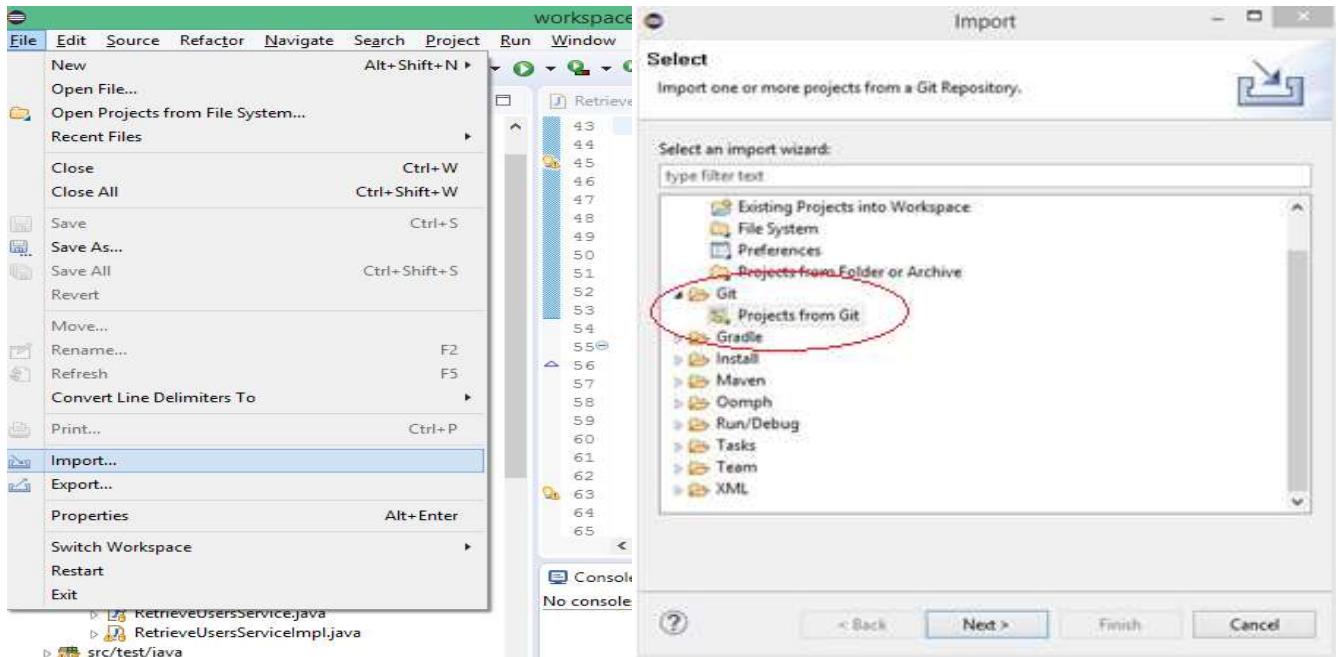
Software Required:

Eclipse

MYSQL workbench and server

Import the code in to eclipse^[3]:

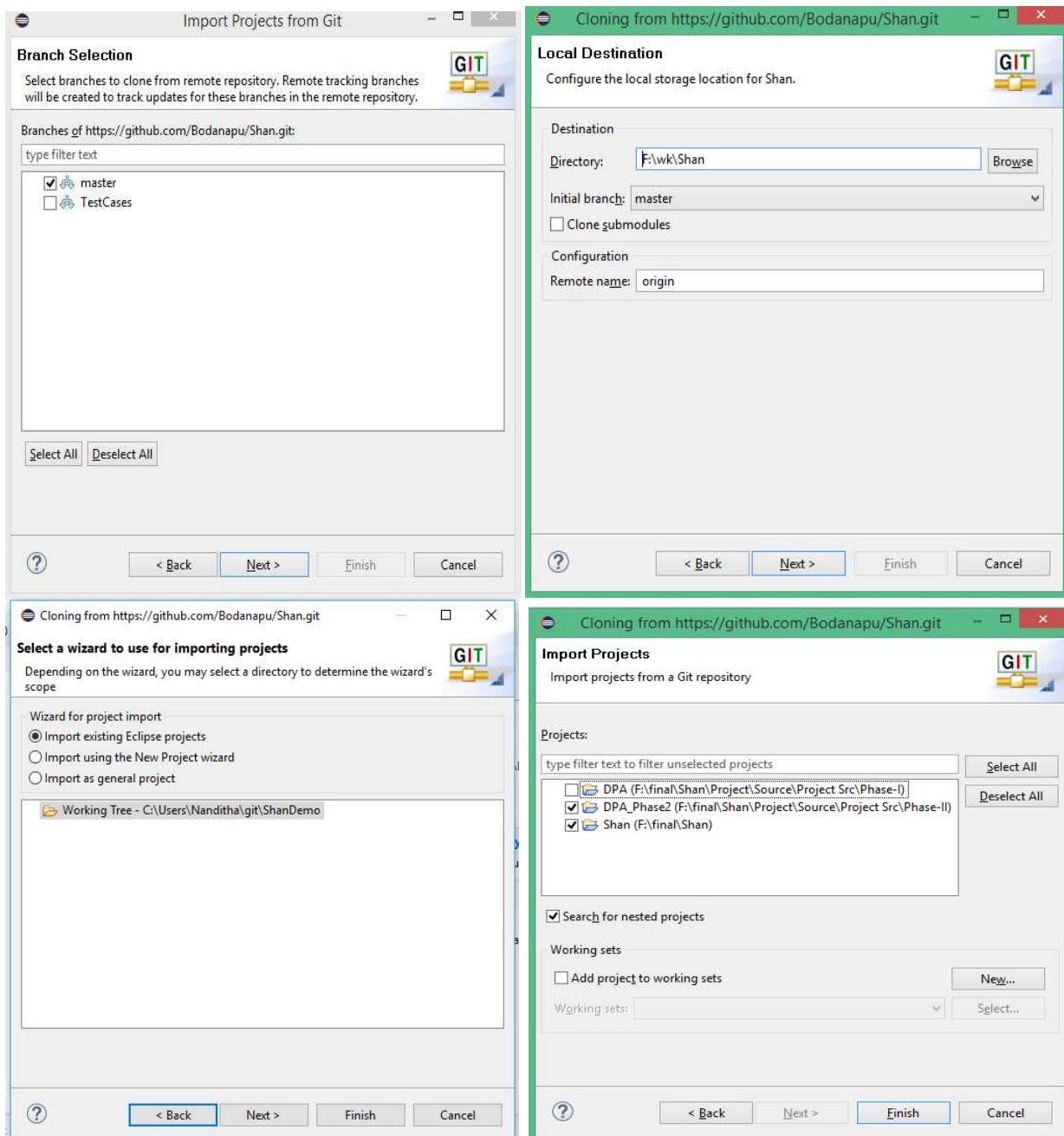
Open the eclipse workspace and import the code from GIT repository by clicking:
File -> import->Git->Projects from Git->clone URI



Type in the URI: <https://github.com/Bodanapu/Shan.git> and enter your github account details, username and password and click next. You can then select the branch “master” and click next. You will have chosen a folder to store the project and import it as import existing eclipse project and then click next, you will see three choices to clone 1.DPA, 2. DPA_Phase2, 3. SHAN choose both option 2, 3 and click fetch. You will now find the DPA_Phase2 maven project with the entire SHAN repository structure in your eclipse project explorer.

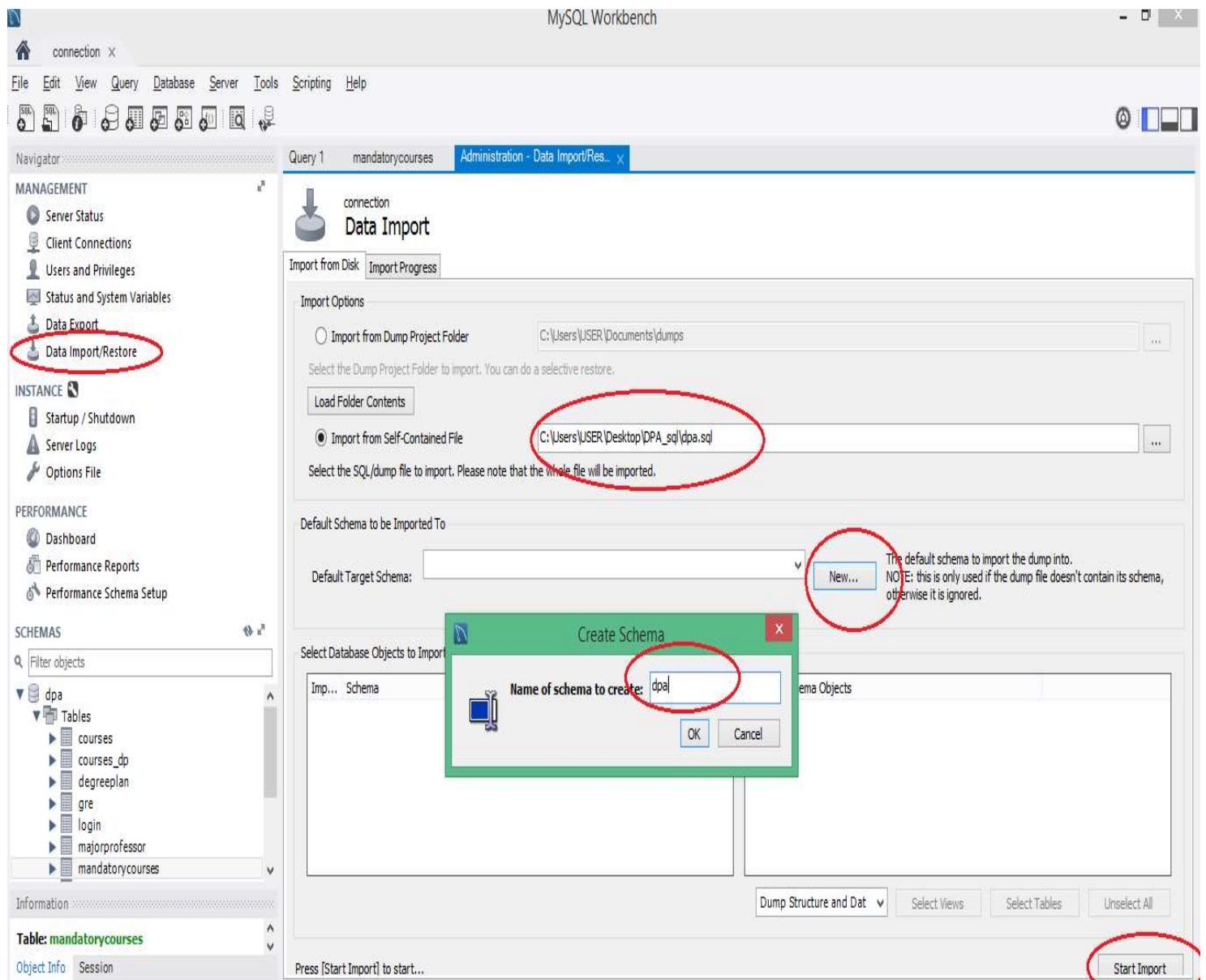
Build path to jdk 1.8:

Right click on the DPA project -> Bulid Path -> Configure Build path -> Add Library -> JRE System Library -> Alternate JRE -> browse to jdk1.8.0_152 location on your system and click finish -> apply and close.



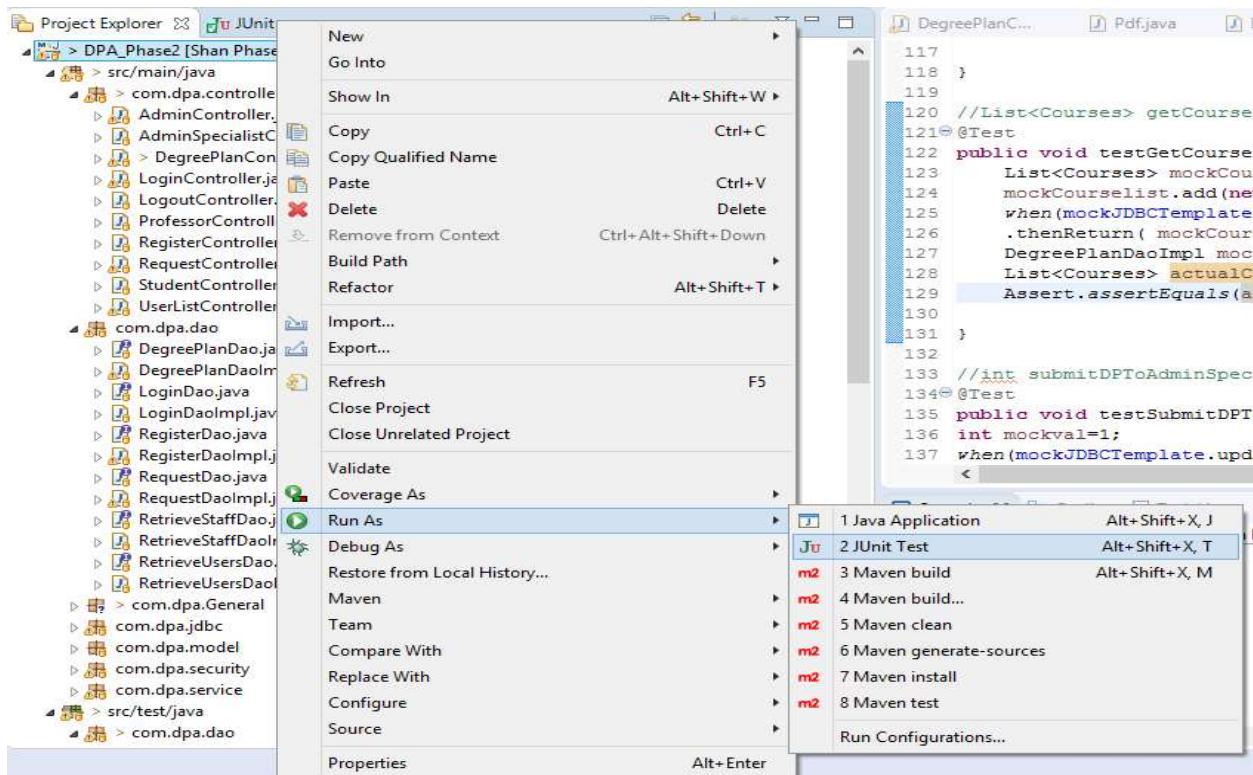
Import dpa.sql file into workbench (Database Import) [4]:

You will find a dpa.sql file in the repository path “Shan/Project/Source/ProjectSrc/Phase-I/Src/Main/Db/”. Start the SQL server and open the MYSQL workbench and create a connection using username and password as root then click import and choose the self contained dpa.sql, create a new schema “dpa” and start import. After import you will have the dpa database created will all the tables in the workbench.



Procedure to Run the Test Cases:

Once you find the maven project DPA in your eclipse project explorer right click on the DPA and select the option Run As -> JunitTest, the test cases will run and show the test cases passed and failed both.



Finished after 12.315 seconds

Runs: 54/54 Errors: 0 Failures: 0

- | Test Case | Runner | Duration |
|---|---------|-----------|
| com.dpa.dao.RegisterDaoImplTest | JUnit 4 | (0.892 s) |
| com.dpa.service.RequestServiceImplTest | JUnit 4 | (0.0) |
| com.dpa.dao.RetrieveStaffDaoImplTest | JUnit 4 | (0.5) |
| com.dpa.dao.RetrieveUsersDaoImplTest | JUnit 4 | (0.4) |
| com.dpa.service.LoginServiceImplTest | JUnit 4 | (0.00) |
| com.dpa.dao.LoginDaoImplTest | JUnit 4 | (0.391 s) |
| com.dpa.dao.RequestDaoImplTest | JUnit 4 | (0.453 s) |
| com.dpa.dao.DegreePlanDaoImplTest | JUnit 4 | (9.06) |
| com.dpa.security.HashingTest | JUnit 4 | (0.312 s) |
| com.dpa.service.DegreePlanServiceImplTest | JUnit 4 | |



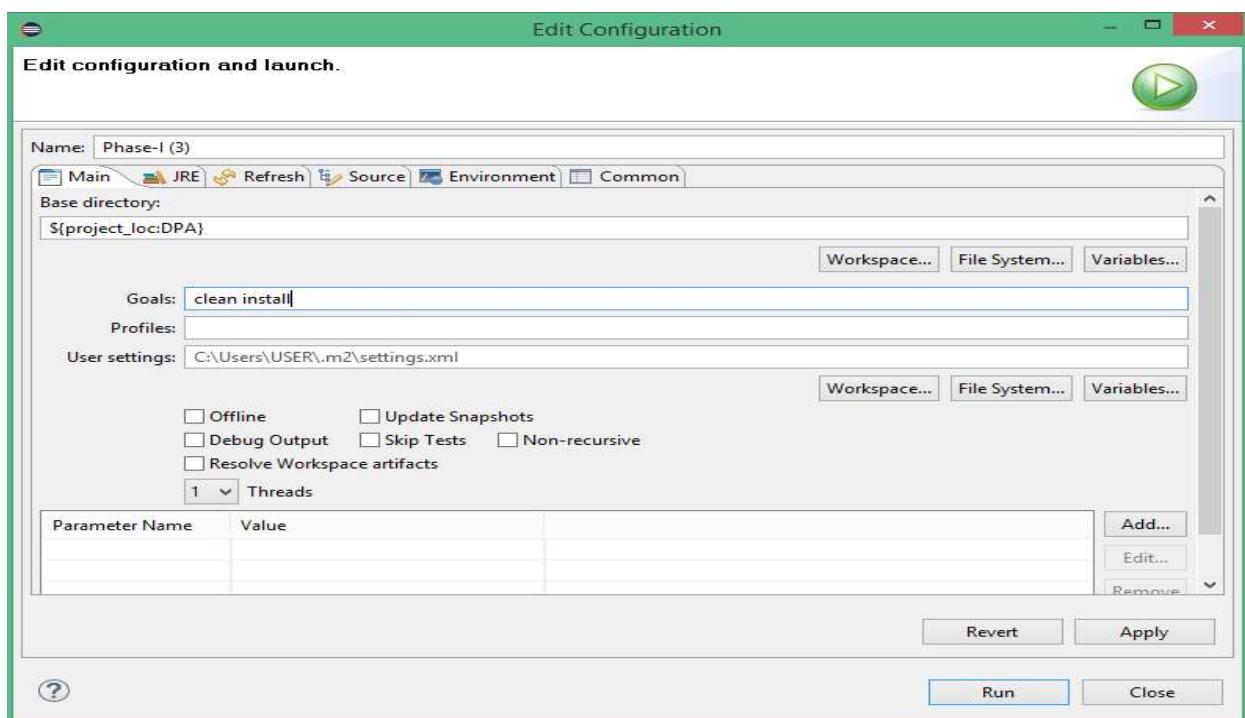
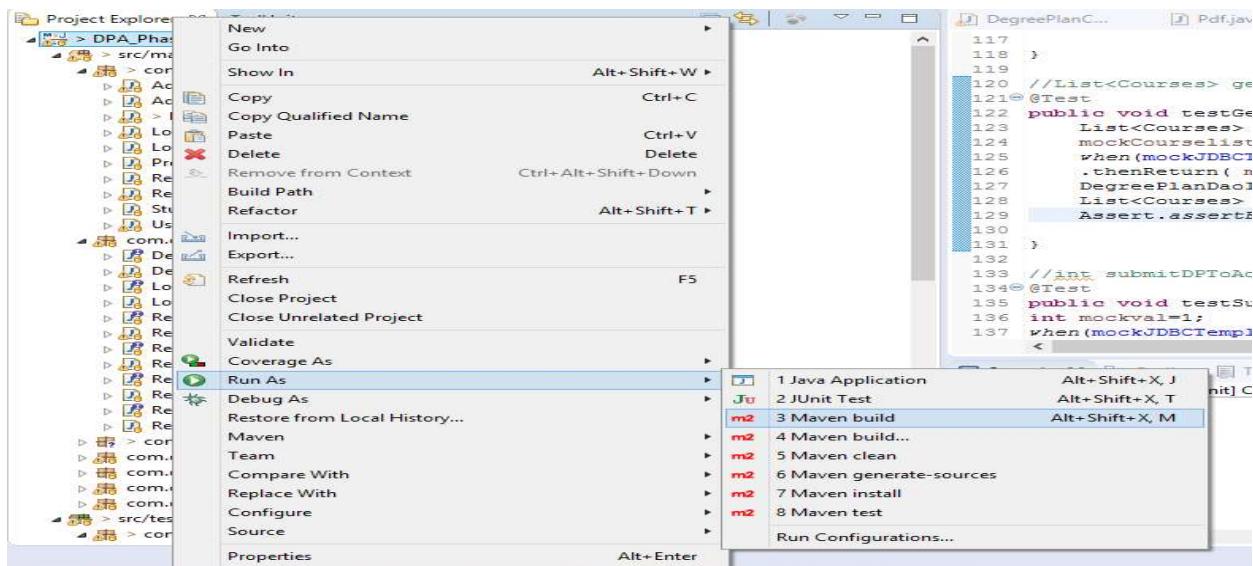
Procedure to Run the Program and check using web browser:

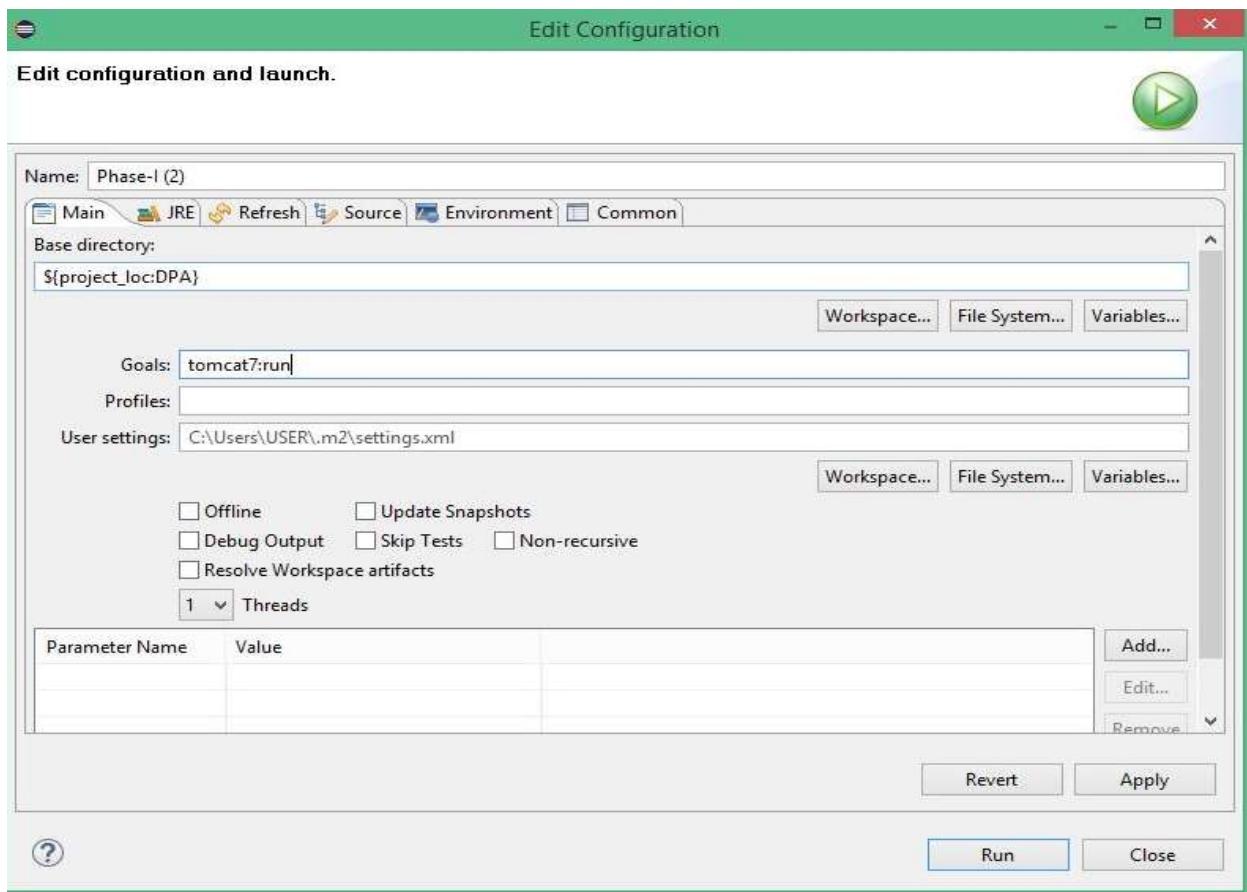
Right click on the maven project DPA_Phase2 in the eclipse project explorer and choose Run As -> Maven build. When you are first time running the project the Maven build asks you to set the goal. Type “clean install” in the goals field and click run.

The maven will download and build all the dependencies of the project present in the “pom.xml” file.

Then click Run As -> Maven build and set the goal as “tomcat7:run” this runs the code using tomcat 7. Tomcat 7 is auto downloaded using the plugin present in the “pom.xml” file.

The project will now compile and run the code and gives a URL “localhost:8080/dpa/login” . Type URL this in the chrome browser (Google Chrome) and you can access the website.





Results :

```
Tests run: 26, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ DPA ---
[INFO] Packaging webapp
[INFO] Assembling webapp [DPA] in [F:\workspace_dpa\Shan\Project\Source\Project Src\
[INFO] Processing war project
[INFO] Copying webapp resources [F:\workspace_dpa\Shan\Project\Source\Project Src\Pr
[INFO] Webapp assembled in [2348 msecs]
[INFO] Building war: F:\workspace_dpa\Shan\Project\Source\Project Src\Phase-I\target
[INFO] WEB-INF\web.xml already added, skipping
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ DPA ---
[INFO] Installing F:\workspace_dpa\Shan\Project\Source\Project Src\Phase-I\target\DE
[INFO] Installing F:\workspace_dpa\Shan\Project\Source\Project Src\Phase-I\pom.xml t
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 31.012 s
[INFO] Finished at: 2018-10-29T14:14:53-05:00
[INFO] -----
```

```
Console Outline Task List
Phase-I (1) [Maven Build] C:\Program Files\Java\jdk1.8.0_181\bin\javaw.exe (Oct 29, 2018, 2:09:06 PM)
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.shan.dpa:DPA >-----
[INFO] Building DPA Maven Webapp 0.0.1-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] >>> tomcat7-maven-plugin:2.2:run (default-cli) > process-classes @ DPA >>>
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ DPA ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory F:\workspace_dpa\Shan\Project\Source\Project Src\Phase-I\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.6.1:compile (default-compile) @ DPA ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< tomcat7-maven-plugin:2.2:run (default-cli) < process-classes @ DPA <<<
[INFO]
[INFO]
[INFO] --- tomcat7-maven-plugin:2.2:run (default-cli) @ DPA ---
[INFO] Running war on http://localhost:8080/
[INFO] Using existing Tomcat server configuration at F:\workspace_dpa\Shan\Project\Source\Project Src\Phase-I\target\tomcat
[INFO] create webapp with contextPath:
Oct 29, 2018 2:09:24 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
Oct 29, 2018 2:09:24 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Tomcat

```

Activate Windows

Go to PC settings to activate Windows

Reference:

- [1]. <https://www.youtube.com/watch?v=o5k9NOR9lrI>
- [2]. <https://www.youtube.com/watch?v=d2KwvXQgQx4>
- [3]. <https://stackoverflow.com/questions/6760115/importing-a-github-project-into-eclipse>
- [4]. <https://www.linode.com/docs/databases/mysql/deploy-mysql-workbench-for-database-administration/>