DEGREE PLAN AUTOMATION DELIVERABLE 5

Submitted by:

Sharanya Gottimukkula

Nanditha Bodanapu

Sriharshini Vallabhaneni

Aravind Thottempudi

CSCE 5430 - Software Engineering
Deliverable – 5

a. Requirements Destined for Development Phase – III

Functional Requirement	Development Phase	Start Date	End Date
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

Implemented Requirements:

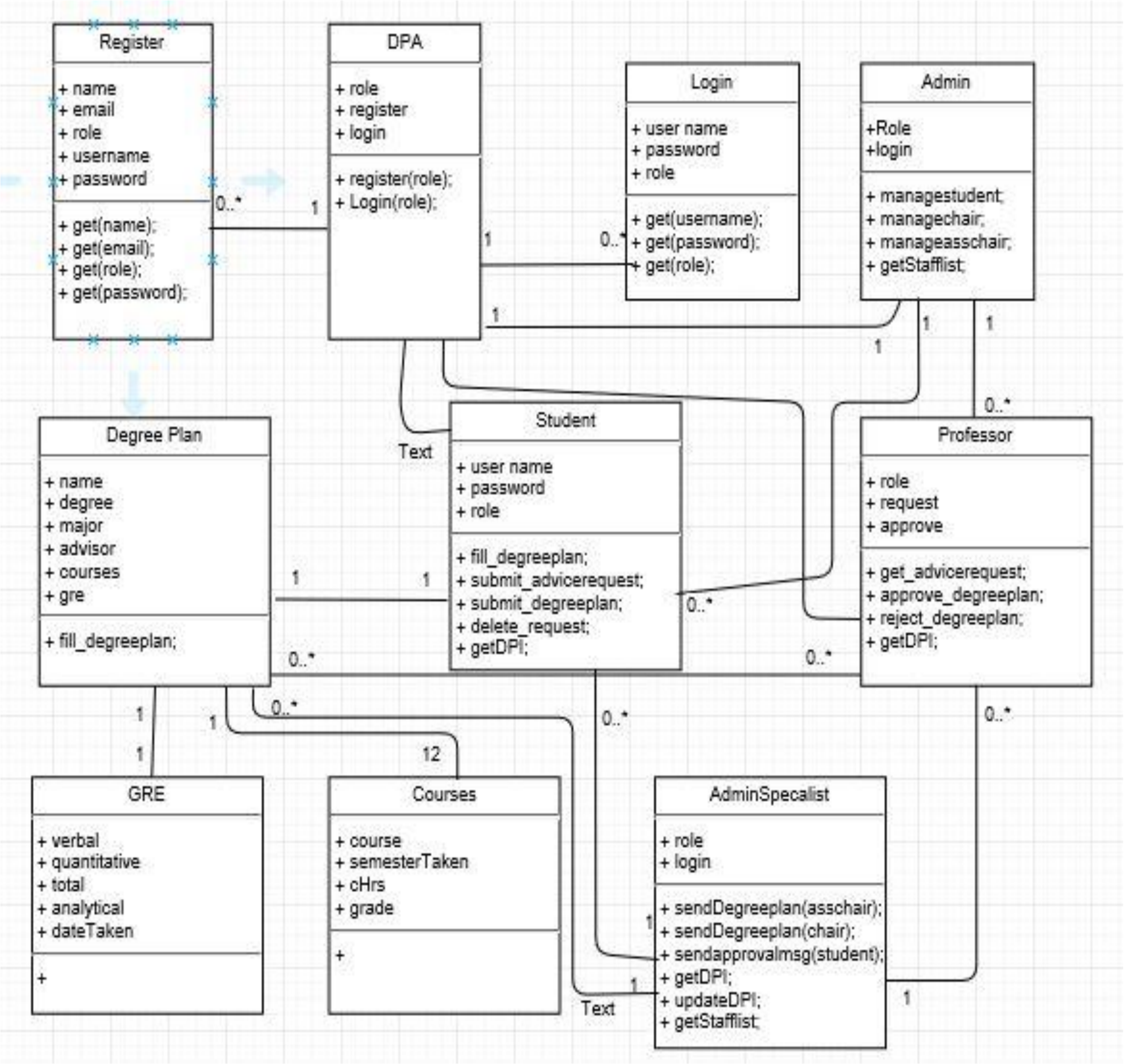
FREQ-10: Manage CSCE Department degree plan information - Administrative Specialist, FREQ - 12: Send approval notice to the student – Administrative Specialist, FREQ-4: Verify and approve students/professors – admin, FREQ-14: Contact Us – All users, FREQ-15: Notifications – All users were destined for phase 3. We have successfully implemented all the requirements designated for phase-3 as per the schedule listed in the deliverable-4.

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. UML Diagrams:

Class Diagram:



Use Case 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

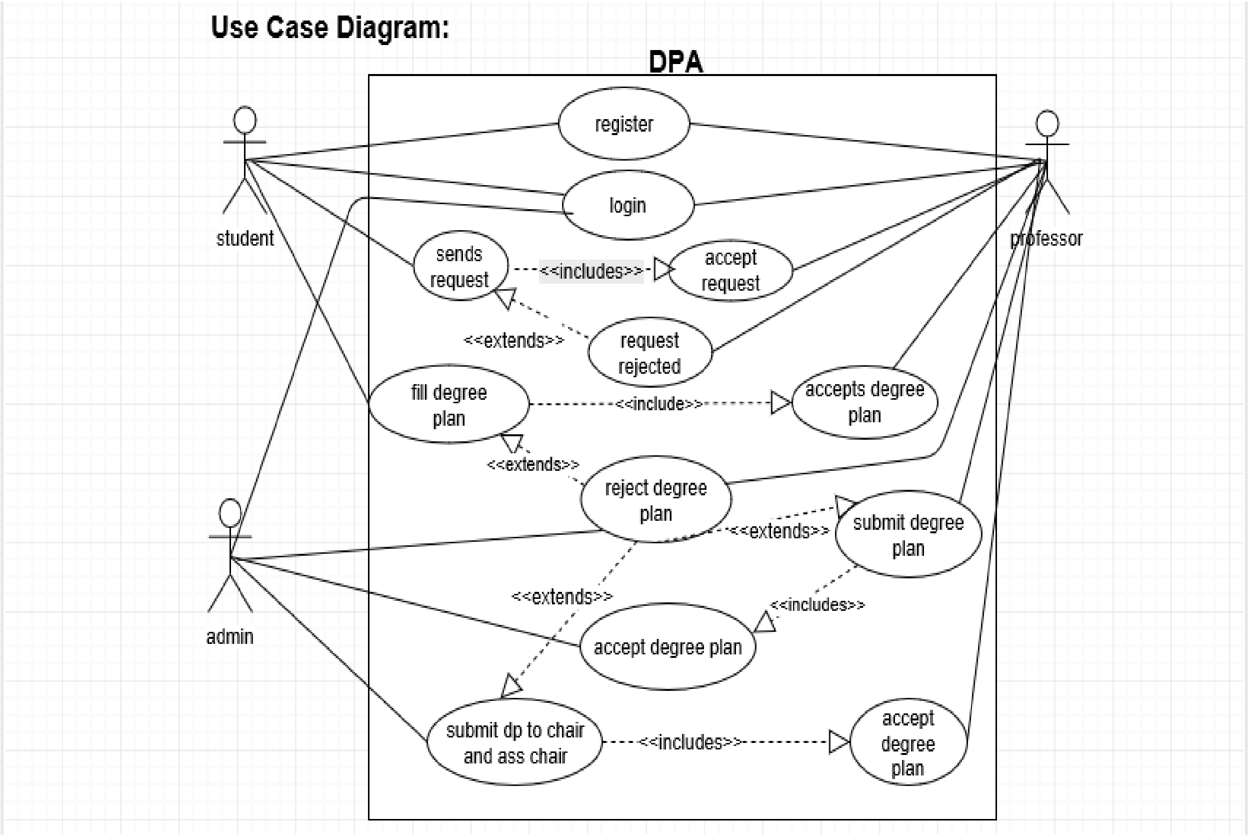
6a: Professor rejects the degree plan.
1. If professor rejects the degree plan, return to step 5.

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.

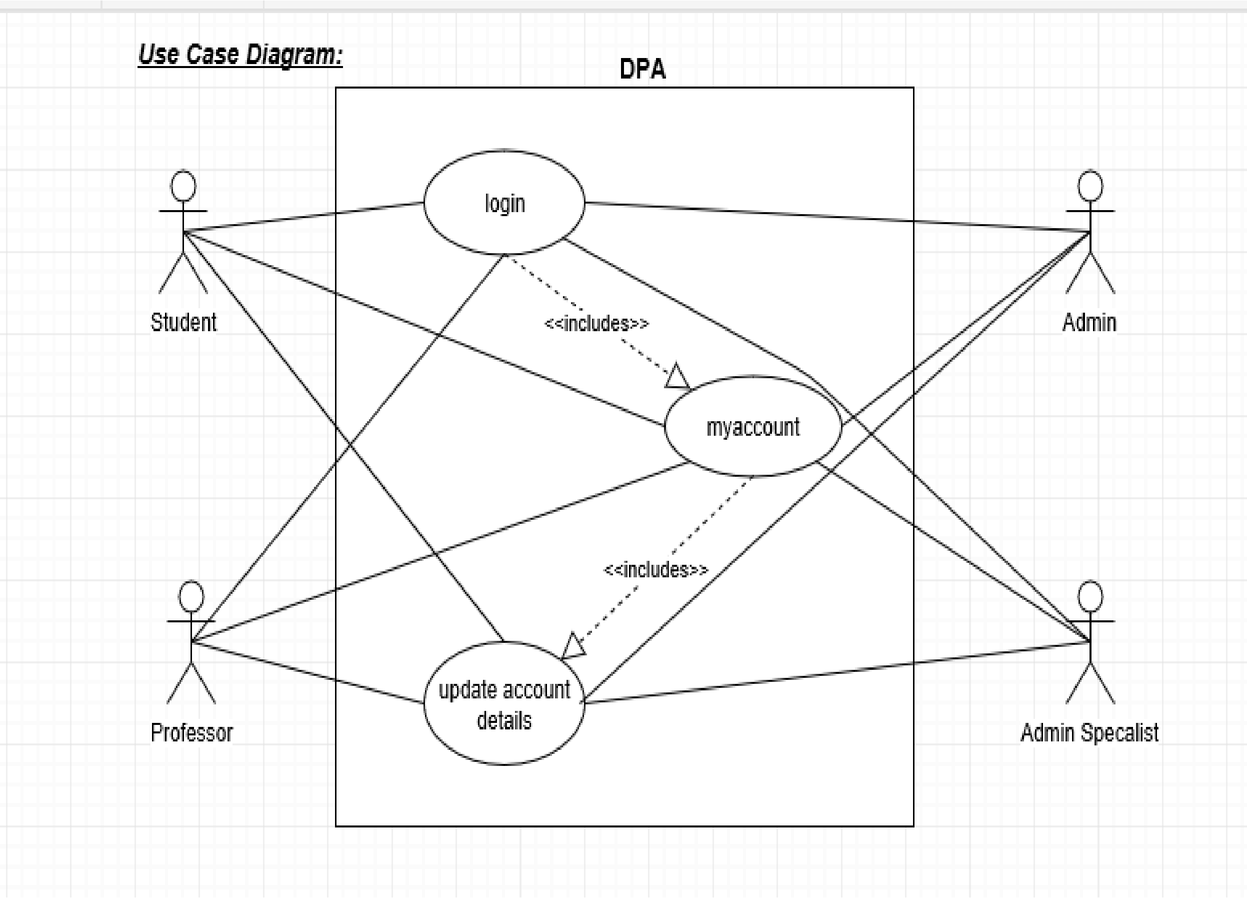
2a. User Case Diagram



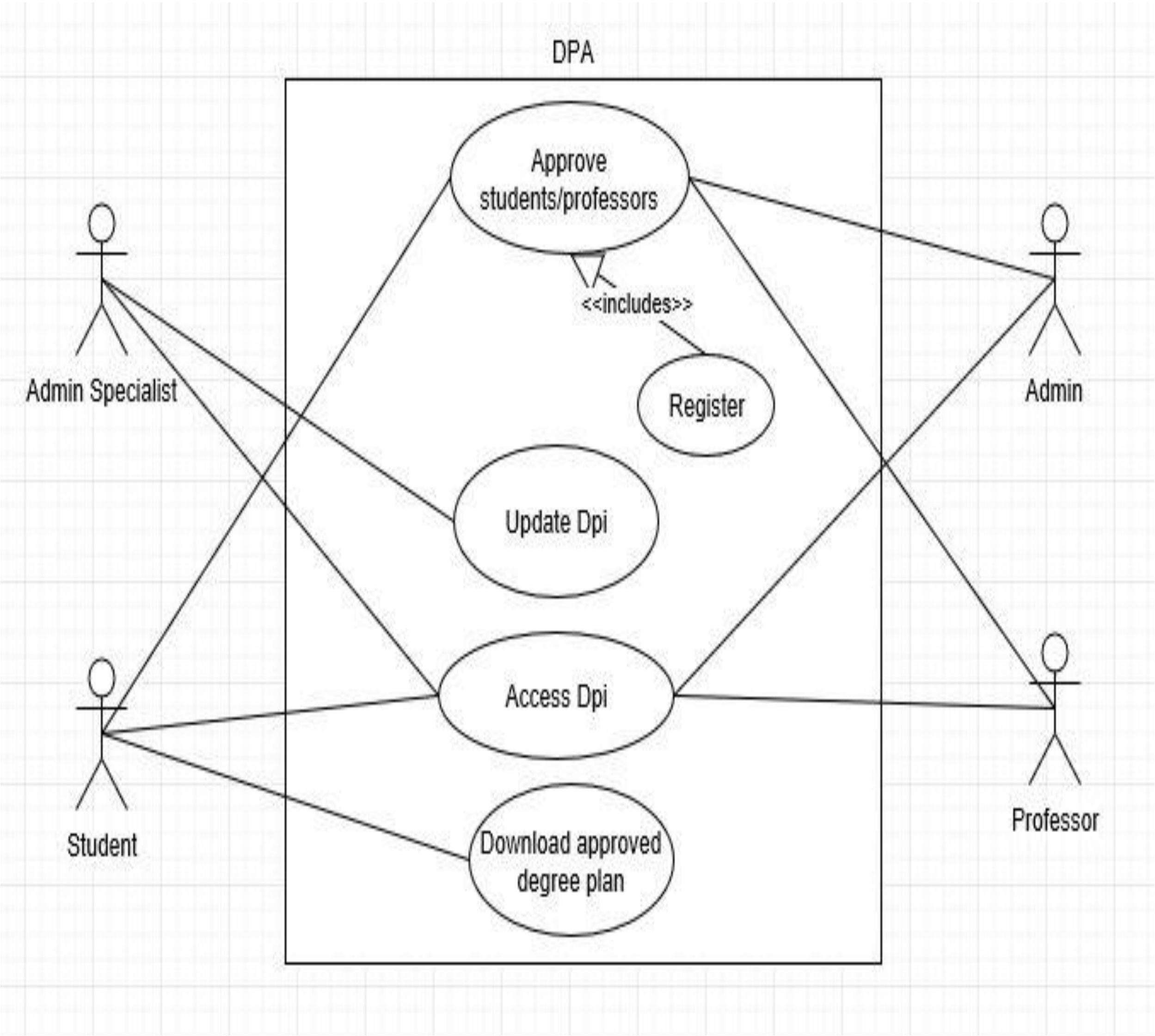
Use Case Text: UpdateStudentInfo
1: Student registers himself.
2: Student logs into the application.
3: Student can view advisor.
4: Student can view degree plan status.
Extensions:
2a: Student updates account information.

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:



2c Use Case Diagram:



Use case Text:

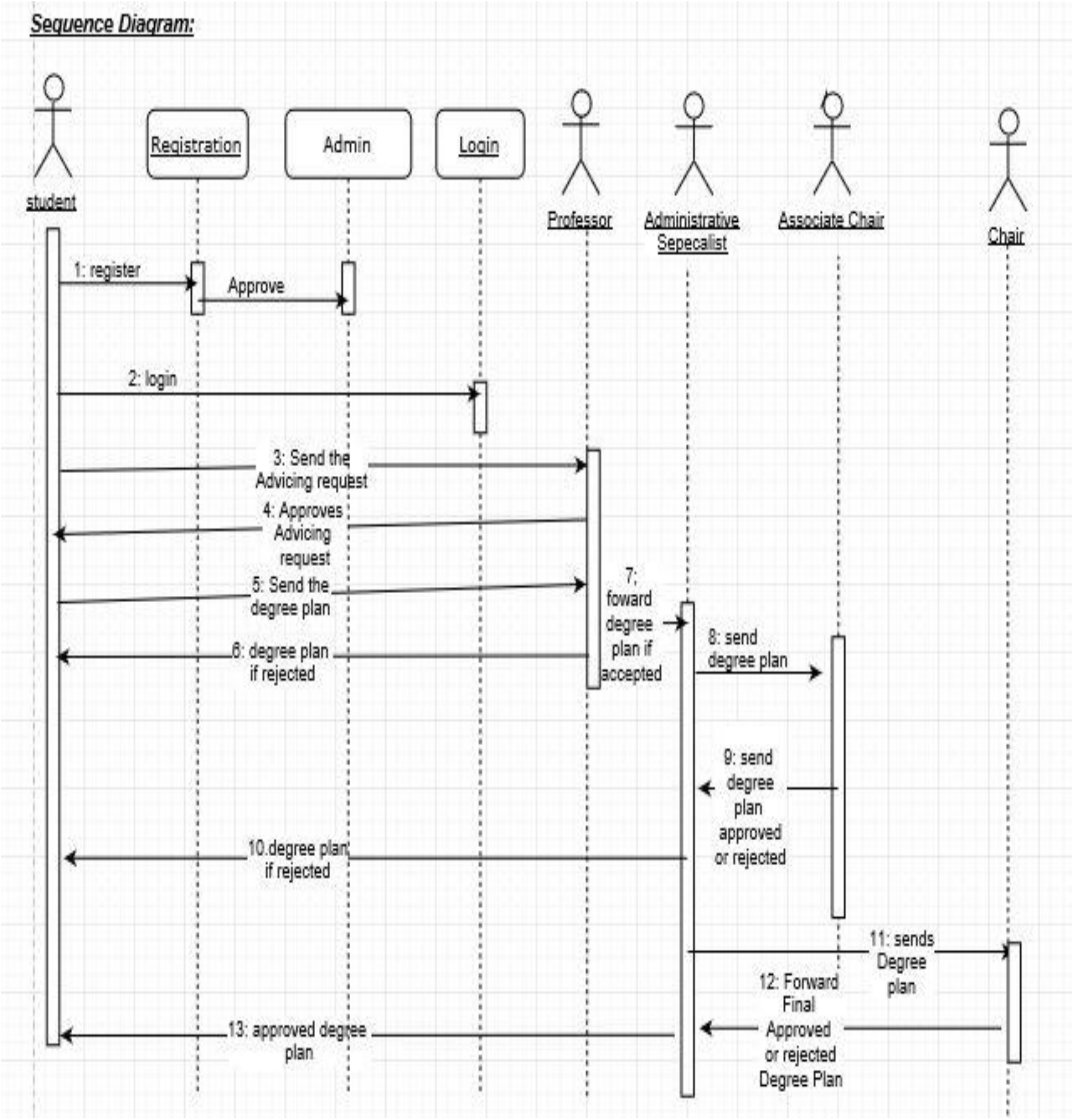
1. Students and professors register in DPA.
2. Admin will approve the registered user.
3. Admin can update the DPI.
4. Students, professors and admin can access the DPI.
5. Students can download the final approved degree plan

Extensions:

1. If the admin reject the student or professor registration then they are not able to use the application.

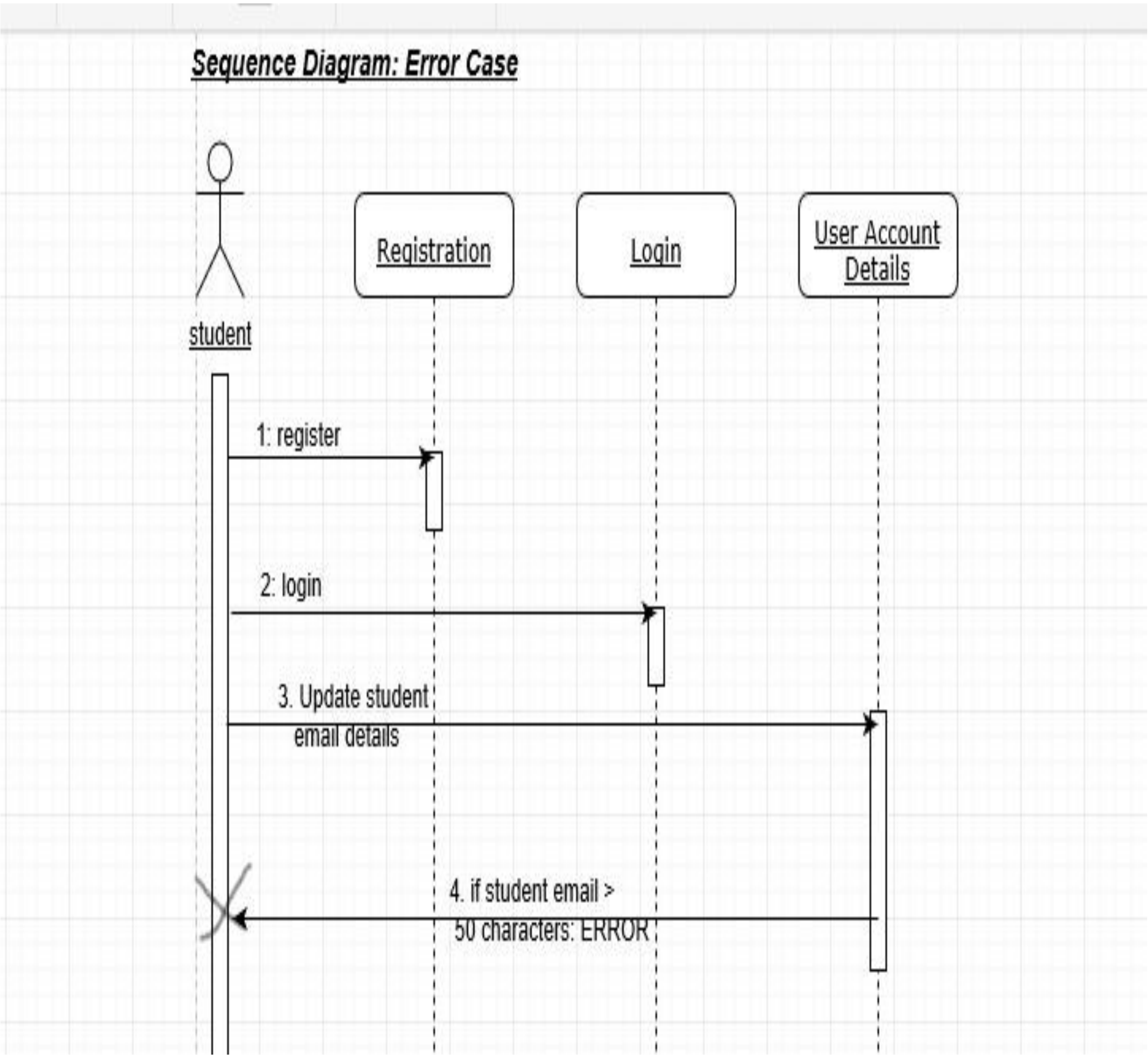
3a. Sequence Diagram: Success scenario

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


Sequence Diagram:
student
Registration
Admin
Login
Professor
Administrative Sepecalist
Associate Chair
Chair
1: register
Approve
2: login
3: Send the Advicing request
4: Approves Advicing request
5: Send the degree plan
6: degree plan if rejected
7: foward degree plan if accepted
8: send degree plan
9: send degree plan approved or rejected
10.degree plan if rejected
11: sends Degree plan
12: Forward Final Approved or rejected Degree Plan
13: approved degree plan

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


Sequence Diagram: Error Case
student
Registration
Login
User Account Details
1: register
2: login
3. Update student email details
4. if student email > 50 characters: ERROR

c. Test Cases (Unit Tests)

This is test case tests getDPI() method of degreePlanDao that is it checks if the getDPI() method is getting the expected degree plan information i.e the correct degree plan data from the. This test takes no input parameters and returns the string that is information, if the expected information and the returned information is same the test case succeeds.[2]

```
    @Test
    public void testgetDPI() {
        String real="sample text";
        when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
        Matchers.any(Class.class)))
        .thenReturn(real);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        String result = mockObj.getDPI();
        assertEquals(real, result);
    }
```

This is test case tests getInterestAreas() method of degreePlanDao that is it checks if the getInterestAreas () method is getting the expected that is correct interest area list from the database. This test takes no input parameters and returns the list of type InterestArea, if the expected list and the returned list is same the test case succeeds.[2]

```
    @Test
    public void testgetInterestAreas() {
        List <InterestArea> mockDPList=new ArrayList<InterestArea>() ;
        mockDPList.add(new InterestArea());
        when(mockJDBCTemplate.query(Matchers.anyString(),
        Matchers.any(InterestAreaRowMapper.class)))
        .thenReturn(mockDPList);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        List <InterestArea> actualDPList=mockObj.getInterestAreas();
        Assert.assertEquals(actualDPList,mockDPList);
    }
```

This is test case tests getAccountDetails() method of AccountDetailsDao that is it checks if the getAccountDetails () method is getting the exact details of the user. This test takes username as the input parameter and returns the account details of the particular user, if the expected details and the returned details are same the test case succeeds.[2]

```
    @Test
    public void testGetAccountDetails() {
        Register mockRegister =new Register();

        when(mockJDBCTemplate.queryForObject(Matchers.anyString(),
        Matchers.any(BeanPropertyRowMapper.class)))
        .thenReturn( mockRegister);
        AccountDetailsDaoImpl mockObj= new AccountDetailsDaoImpl();
        Register actualRegister =mockObj.getAccountDetails( "userName");
        Assert.assertEquals(mockRegister, actualRegister);
    }
```

This is test case tests updateAccountDetails() method of AccountDetailsDao that is it checks if the updateAccountDetails() method is updating the details of correct user. This test takes register as the input parameter and updates the details of a the particular username in the register, if the returned result is equal to the mockval the test succeeds.[2]

```
    @Test
    public void testUpdateAccountDetails() {
        int mockval=1;

    when(mockJDBCTemplate.update(Matchers.anyString(),objCap.capture()))
        .thenReturn(mockval);
        AccountDetailsDaoImpl mockObj= new AccountDetailsDaoImpl();
        Register register = new Register();
        int actualresult= mockObj.updateAccountDetails(register);
        Assert.assertEquals(actualresult, mockval);
    }
```

This is test case tests approveUser() method of AdminDao that is it checks if the approveUser() method is approving the correct user, This test takes username as input and if the returned result is equal to the mockval the test succeeds.[2]

```
    @Test
    public void testApproveUser() {
        int mockval=1;

    when(mockJDBCTemplate.update(Matchers.anyString(),objCap.capture()))
        .thenReturn(mockval);
        AdminDaoImpl mockObj= new AdminDaoImpl();
        int actualresult= mockObj.approveUser("userName");
        Assert.assertEquals(actualresult, mockval);
    }
```

This is test case tests rejectUser() method of AdminDao that is it checks if the rejectUser () method is rejecting the correct user, This test takes username as input and if the returned result is equal to the mockval the test succeeds.[2]

```
    @Test
    public void testRejectUser() {
        int mockval=0;

    when(mockJDBCTemplate.update(Matchers.anyString(),objCap.capture()))
        .thenReturn(mockval);
        AdminDaoImpl mockObj= new AdminDaoImpl();
        int actualresult= mockObj.rejectUser("userName");
        Assert.assertEquals(actualresult, mockval);
    }
```

This test case tests getAccoutDetails() method of AccountInformation ServiceImpl class that is it checks if the getAccoutDetails() method is getting the details of the correct user or not. It takes username as input and returns a register type data. If the returned result is equal to mock value then the test case succeeds.[1]

```
      @Test
      public void testGetAccountDetails() {
            Register mock = new Register();

      when(accountDetailsDao.getAccountDetails(anyString())).thenReturn(mock)
;
            Register result =
            accountInformationServiceImpl.getAccountDetails(anyString());
            assertEquals(mock, result);
      }
```

This test case tests updteAccoutDetails() method of AccountInformation ServiceImpl class that is it checks if the updteAccoutDetails () method is updating the details of the correct user or not. It takes username as input and returns 1 if the details are updated successfully. If the returned result is equal to 1 then the test case succeeds.[2]

```
      @Test
      public void testUpdateAccountDetails() {
            Register register = new Register();

      when(accountDetailsDao.updateAccountDetails(register)).thenReturn(1);
            int result =
accountInformationServiceImpl.updateAccountDetails(register);
            assertEquals(1, result);
      }
```

This test case tests getDPI() method of DegreePlanServiceImpl class that is it checks if the getDPI() method is getting the correct degree plan information from the database. If the returned degree plan information is equal to the expected degree plan information then the test succeeds[2]

```
@Test
      public void testgetDPI() {
            String real="sample text";
            when(degreePlanDao.getDPI()).thenReturn(real);
            String result = degreePlanService.getDPI();
            assertEquals(real, result);
      }
```

This test case tests getInterestAreas() method of DegreePlanServiceImpl class that is it checks if the getInterestAreas() method is getting the correct interest areas list from the database. If the returned interest areas is equal to the expected one then the test succeeds [2]

```
      @Test
      public void testgetInterestAreas() {
      List<InterestArea> mockInterestAreas = new ArrayList<InterestArea>();

      when(degreePlanDao.getInterestAreas()).thenReturn(mockInterestAreas);
            List<InterestArea> result = degreePlanService.getInterestAreas();
            assertEquals(mockInterestAreas.size(), result.size());
      }
```

This test case tests approveUser() method of AdminServiceImpl class that is it checks if the approveUser () method is approving the correct user to use the application. It takes username as input and returns 1 on correct approval. If the returned result is equal to 1 then the test succeeds [2]

```
      @Test
      public void testApproveUser() {
            when(adminDao.approveUser(anyString())).thenReturn(1);
            int result = adminServiceImpl.ApproveUser(anyString());
            assertEquals(1, result);
      }
```

This test case tests rejectUser() method of AdminServiceImpl class that is it checks if the rejectUser() method is rejecting the correct user from using the application. It takes username as input and returns 1 on correct rejection. If the returned result is equal to 1 then the test succeeds [2]

```
      @Test
      public void testRejectUser() {
            when(adminDao.rejectUser(anyString())).thenReturn(1);
            int result = adminServiceImpl.rejectUser(anyString());
            assertEquals(1, result);
      }
```

This test case tests rejectUser() method of AdminServiceImpl class that is it checks if the rejectUser() method is rejecting the correct user from using the application. It takes username as input and returns 1 on correct rejection. If the returned result is equal to 1 then the test succeeds [2]

```
      @Test
      public void testGetNeedApprovalList() {
            List<Register> mockList = new ArrayList<Register>();
            when(retrieveUsersDao.getToApprovalList()).thenReturn(mockList);
                        List<Register> result =
            retrieveUsersServiceImpl.getNeedApprovalList();
            assertEquals(mockList, result);
      }
```

This test case tests getStaffList() method of RetrieveUsersServiceImpl class that is it checks if the getStaffList () method is getting the staff list correctly. If the returned register list is equal to mocklist then the test succeeds [2]

```
      @Test
      public void testGetstaffList() {
            List<Register> mockList = new ArrayList<Register>();
            when(retrieveUsersDao.getstaffList()).thenReturn(mockList);
            List<Register> result = retrieveUsersServiceImpl.getstaffList();
            assertEquals(mockList, result);
      }
```

This test case tests addOpCourse() method of DegreePlanDao. This test checks if the optional course is properly being added to the database or not. This takes course name as input and returns 1 on successful addition. If the returned result by is equal to 1 then the test succeeds [2]

```
@Test
public void testAddOpCourse() {
        when(mockJDBCTemplate.update(Matchers.anyString(),
objCap.capture()))
        .thenReturn(1);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        int result = mockObj.addOpCourse(anyString());
        Assert.assertEquals(1,result);
}
```

This test case tests deleteOpCourse() method of DegreePlanDao. This test checks if the optional course is properly getting deleted the database or not. This takes course name as input and returns 1 on successful deletion. If the returned result by is equal to 1 then the test succeeds [2]

```
@Test
public void testDeleteOpCourse() {
        when(mockJDBCTemplate.update(Matchers.anyString()))
        .thenReturn(1);
        DegreePlanDaoImpl mockObj= new DegreePlanDaoImpl();
        int result = mockObj.deleteOpCourse(anyString());
        Assert.assertEquals(1,result);
}
```

This test case tests addOpCourse() method of DegreePlanService. This test checks if the optional course is properly being added to the database or not. This takes course name as input and returns 1 on successful addition. If the returned result by is equal to 1 then the test succeeds [2]

```
@Test
public void testAddOpCourse() {
        when(degreePlanDao.addOpCourse(anyString()))
        .thenReturn(1);
        int result = degreePlanService.addOpCourse(anyString());
        Assert.assertEquals(1,result);
}
```

This test case tests deleteOpCourse() method of DegreePlanService. This test checks if the optional course is properly being deleted from the database or not. This takes course name as input and returns 1 on successful addition. If the returned result by is equal to 1 then the test succeeds [2]

```
@Test
public void testDeleteOpCourse() {
        when(degreePlanDao.deleteOpCourse(anyString()))
        .thenReturn(1);
        int result = degreePlanService.deleteOpCourse(anyString());
        Assert.assertEquals(1,result);
}
```

c. Test Cases (System Tests)

The systems tests include various tests to check all the functionality as per the requirements that is the tests check all the functionality by giving an input and checking the output according to the requirements specified and also performance of system under pressure and for destructive scenarios.

Test Case No	Trace to Requirement	Preconditions	Test Input/ Test Steps	Test output	Pass/ fail
1	FREQ-1	Valid URL: Localhost:8080/dpa/login	1. Click “NewUser? Register” link	The user should be redirected to registeration page Localhost:8080/dpa/register	pass
2	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the Name field and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for Name with a message required	pass
3	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the Email field and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for Email with a message required	pass
4	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the Role field unselected and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for Role with a message required	pass
5	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the UserName field and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for UserName with a message required	pass
6	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the Password field and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for Password with a message required	pass
7	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Leave the RetypePassword field and fill all the remaining fields 3.click register	The user should stay on the same register page. The screen should prompt for RetypePassword with a message required	pass
8	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Fill the Name with less than 5 characters or more than 20 characters and focus out to fill other fields	The user should stay on the same register page. The screen should display an error message in red saying “Name should be 5 to 20 characters length”	pass

9	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2.Enter the email without the "@", "." symbol each at once and fill in all the other fields. 3.click register	The user should stay on the same register page. The screen should prompt with a message incorrect email format	pass
10	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2. Fill the UserName with less than 5 characters or more than 20 characters and focus out to fill other fields	The user should stay on the same register page. The screen should display an error message in red saying "UserName should be 5 to 20 characters length"	pass
11	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2. Fill the password and ReTypePassword field with less than 5 characters or more than 15 characters or with special characters and focus out to fill other fields	The user should stay on the same register page. The screen should display an error message in red saying "Password length should be between 5 to 15 characters. It should not contain special characters"	pass
12	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2. Fill the password and ReTypePassword field with different values. 3.Click Register	The user should stay on the same register page. The screen should display an error message in red saying "Password Mismatch"	pass
13	FREQ-1	Valid URL: Localhost:8080/dpa/register	1.open registration page 2. Fill all the fields as per the requirement with no errors. 3.Click Register	The user should be directed to the login page, with a message "Registered Successfully, please login" displayed on the top of the login page in green	pass
14	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the URL localhost:8080/dpa/login in the web browser	The user should be directed to the login page	Pass
15	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the UserName 2.Enter the password 3. Select role values not present int the data base 4.Click login	The user should stay on the login page with an error message "invalid credentials" displayed in red	Pass
16	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the UserName 2.Enter the password present in the database 3. Select role not matching the username and password 4.Click login	The user should stay on the login page with an error message "invalid credentials" displayed in red	Pass
17	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the UserName present in the database 2.Enter the password wrong 3. Select correct role for the	The user should stay on the login page with an error message "invalid credentials" displayed in red	Pass

			username entered 4.Click login		
18	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the UserName wrong 2.Enter the password present in the database 3. Select correct role of for the password entered 4. Click login	The user should stay on the login page with an error message “invalid credentials” displayed in red	Pass
19	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the UserName with special characters and focus out	The user should stay on the login page with an error message “UserName should not contain special characters” displayed in red	Pass
20	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the username with no errors 2.Enter the password with special characters and focus out	The user should stay on the login page with an error message “password should not contain special characters” displayed in red	Pass
20	FREQ-2	Valid URL: Localhost:8080/dpa/login	1.Enter the username with no errors 2.Enter the password with no errors 3.select the correct role for entered username and password. 4. click login	The user should be redirected to the home page of the role chosen. If role = admin should be redirected to admin page If role = admin specialist should be redirected to admin specialist home If role = student should be redirected to student home If role = professor should be redirected to professor home	Pass
21	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Admin Specialist	The screen should display: Delete Admin Specialist Update Admin Specialist Add Admin Specialist Buttons to the right of the select box	pass
22	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Associate Chair	The screen should display: Delete Associate Chair Update Associate Chair Add Associate Chair Buttons to the right of the select box	pass
23	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Chair	The screen should display: Delete Chair Update Chair Add Chair	pass

				Buttons to the right of the select box	
24	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Students/Professors	The screen should display: Delete Students/ Professors Update Students /Professors Buttons to the right of the select box	pass
25	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Admin Specialist 3.Click Add Admin Specialist	The user should be redirected to the Add Admin Specialist page where admin has fields to add admin specialist	pass
26	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Associate Chair 4.Click Add Associate Chair	The user should be redirected to the Add Associate Chair page where admin has fields to add associate chair	pass
27	FREQ-3	Valid admin login	1.Login with admin role 2.click the manage users select box on the admin home 3.choose Manage Chair 4.Click Add Chair	The user should be redirected to the Add Chair page where admin has fields to add chair	pass
28	FREQ-5	Valid student login	1. Login with student role 2.Click on StaffDirectory link in the menu bar	The user should stay on same page and be presented with a professor list having SendRequest button to the right of professor details	pass
29	FREQ-5	Valid student login	1. Login with student role 2.Click on StaffDirectory link in the menu bar 2.Click on SendRequest button 3. Click on StaffDirectory link in the menu bar	The request should be sent to the appropriate professor and user should be redirected to the home page. The student should be given the professor list with the professor to whom request is already sent deleted.	pass
30	FREQ-5	Valid student login	1. Login with student role 2.Click on SentRequests link in the menu bar	The user should stay on same page and be presented with the requests sent by him to the professors	pass
31	FREQ-5	Valid student login	1. Login with student role 2.Click on SentRequests link in the menu bar 3.Click Delete Request 4. Click on SentRequests link in the menu bar	The request sent to the professor should be deleted and user should be redirected to the home page The student should find the sent request list with the	pass

				deleted request deleted from the sent request list	
32	FREQ-8	Valid Professor Login	1. Login with professor, chair or associate chair role 2.Click on the received requests link present in the menu bar	The professor should be presented with the list of student details who sent the requests and with “Accept Request” and “Reject Request” buttons beside each request.	pass
33	FREQ-8	Valid Professor Login	1. Login with professor, chair or associate chair role 2.Click on the received requests link present in the menu bar 3.Click Accept Request button	The user should be present redirected to the home page where the accepted request student list is displayed.	pass
34	FREQ-8	Valid Professor Login	1. Login with professor, chair or associate chair role 1.Click on the received requests link present in the menu bar 2.Click Reject Request button	The user should be redirected to the home page with that particular request rejected and removed from the list.	pass
36	FREQ-6	Valid Student Login	1.Login using student role 2.Click “Start New Degree Plan” button	The student should be redirected to the degree plan form where students have all the fields required for the degree plan	pass
37	FREQ-6	Valid Student Login	1.Login using student role 2.Click “Start New Degree Plan” button 3.Fill all the fields like, Name, student Id, Local address and rest leaving one field each time	The student should be present in the same page with the message required for the missed-out field	pass
38	FREQ-6	Valid Student Login	1.Login using student role 2.Click “Start New Degree Plan” button 3.Fill all the fields like, Name, student Id, Local address and rest leaving any of the grade field for the courses 4.Click save	The same degree plan page should be reloaded with a message degree plan successfully saved message displayed on the top in green color	pass
39	FREQ-6	Valid Student Login	1.Login using student role 2.Click “Start New Degree Plan” button 3.Fill all the required fields and non-mandatory fields as per the requirement 4. click save	The same degree plan page should be reloaded with a message degree plan successfully saved message displayed on the top in green color	pass

40	FREQ-7	Valid Student Login	1.Login using student role 2.Click “Start New Degree Plan” button 3.Fill all the required fields and non-mandatory fields as per the requirement 4. click submit	The same degree plan page should be reloaded with a message degree plan successfully submitted message displayed on the top in green color	pass
41	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details	The professor should be presented with the student’s degree plan, where professor have “accept degree plan” and “reject degree” plan buttons	pass
42	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details 3.click on accept degree plan button	The professor should be given an input field to sign and then get a submit to admin specialist button.	pass
43	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details 3.click on accept degree plan button 4.click on submit to admin specialist button	The professor should be present in the same page with a message “degree plan submitted to admin specialist” in green displayed on the top of the page.	pass
44	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details 3.click on reject degree plan button	The professor should be redirected to the reject comments page where the professor has a text area where the professor can fill in the suggestions for student to change degree plan.	pass
45	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details 3.click on reject degree plan button	The professor should be redirected to the reject comments page where the professor has a text area where the professor can fill in the suggestions for student to change degree plan.	pass
46	FREQ-9	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details	The reject comments page should be reloaded with the message “reject comment successfully submitted” message at the top of the	pass

			3.click on reject degree plan button 4.click on submit button on reject comments page	page.	
47	FREQ-11	Valid professor login	1.login using professor or chair or associate chair role 2. click “View Degree Plan” button to the right of student details 3.click on reject degree plan button	The professor should be redirected to the reject comments page where the professor has a text area where the professor can fill in the suggestions for student to change degree plan.	pass
48	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role	The home page of admin specialist displays the degree plans submitted by professor to the admin specialist along with the status and “view degree plan” button	pass
49	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan (status - Status with Admin Specialist)	The admin specialist will be redirected to a page where student’s degree plan is displayed and admin specialist should have “submit to associate chair” and reject buttons	pass
50	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan (status - Status with Admin Specialist) 3.Click submit to associate chair button	The admin specialist will be redirected to the home page that has the status of degree plan submitted to associate chair as “with associate chair”	pass
51	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan (status - Associate Chair Approved) 3.Click submit to chair button	The admin specialist will be redirected to the home page that has the status of degree plan submitted to chair as “with chair”	pass
52	FREQ-12	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan (status - Chair approved) 3.Click send approval notice to student	The same degree plan page will be reloaded with the message “sent approval notice to student” in green on the top of the page	pass
53	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan 3.click reject degree plan	The admin specialist should be redirected to reject comments page	pass

54	FREQ-11	Valid Admin Specialist login	1.login using Adminspecialist role 2.click view degree plan 3.click reject degree plan 4.click submit	The reject comments page will be reloaded with the message "reject comments submitted successfully" displayed on the top of the page	pass
55	FREQ-13	Valid login as any user	1.Login with any user credentials 2.click on the name at the right corner in the menu bar 3.select MyAccount	The user should be redirected to the page where he/she can find all the details like username, name, role and email along with the update button	pass
56	FREQ-13	Valid login as any user	1.Login with any user credentials 2.click on the name at the right corner in the menu bar 3.select MyAccount 4. change any field 5. click submit	The same account details page will be reloaded with the message "account details updated successfully" in green at the top of the page	pass
57	FREQ-13	Valid login as any user	1.Login with any user credentials 2.click on the name at the right corner in the menu bar 3.select MyAccount 4. change any field 5. click submit	The same account details page will be reloaded with the message "account details updated successfully" in green at the top of the page	pass
58	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI" link present on the menu bar	The admin specialist should be redirected to the page with all the degree plan information and update buttons after each field	pass
59	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI" link present on the menu bar 3.change degree plan information and click on update	The same updatedpi page should be reloaded with message "Degree plan information updated successfully" in green on the top of the page	pass
60	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI" link present on the menu bar 3.change core courses of any of the groups and click on update	The same updatedpi page should be reloaded with message "core course updated successfully" in green on the top of the page	pass
61	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI"	The same updatedpi page should be reloaded with message "deleted course	pass

			link present on the menu bar 3.click delete present on the right of any optional course	successfully" in green on the top of the page	
62	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI" link present on the menu bar 3.click "add optional course" present at the bottom of the page	The same updatedpi page should be reloaded with a text field to enter the course name and the add button at the right of the text field	pass
63	FREQ-10	Valid Admin Specialist login	1.login with admin specialist role 2.Click on the "UpdateDPI" link present on the menu bar 3.click "add optional course" present at the bottom of the page 3.click add	The same updatedpi page should be reloaded with a message "optional course successfully updated" in green at the top of the page	pass
64	FREQ-4	Valid admin login	1.Login with admin role 2. Click Approve/Reject students/professors	The admin should be presented with a list of students and professors registered to access website and waiting for approval.	pass
65	FREQ-4	Valid admin login	1.Login with admin role 2. Click Approve/Reject students/professors 3.Click Approve User	The same approval list page will be reloaded with a message "user approved successfully" in green on the top of the page	pass
66	FREQ-4	Valid admin login	1.Login with admin role 2. Click Approve/Reject students/professors 3.Click Reject User	The same approval list page will be reloaded with a message "User Rejected successfully" in green on the top of the page	pass
67	FREQ-14	Valid URL or any user login	1.click on contact us link present in the menu bar	The user should be redirected to the contact us page with a set of URL'S	Pass
68	FREQ-15	Valid user login	1.login with student role 2.click view degree plan status	The student should be notified with the status of the degree plan	pass
69	FREQ-15	Valid user login	1.login with student role 2.click view degree plan status 3.click on the download approved degree plan link	The approved degree plan of the student should be downloaded to the default downloads folder	pass
70	Logout	Valid user login	1.click on the name present at the right corner on the menu bar 2.click on the logout option.	The user session should be cleared and user should be redirected to the login page	pass

d. Team Member Contribution

Name of the Member	Components Developed	Overall Contribution (%)
Sharanya Gottimukkula	FREQ-10 Designed HTML pages that allows admin specialist to Manage CSCE Department degree plan information. Involved in writing informationdpi.jsp, dpi_courses.jsp, updateddpi.jsp, AdminSpecialistController and few methods of DegreePlanServiceImpl and DegreePlanDaoImpl classes like getDPI(), updateDPIInfo() and others which handles the degree plan information related data and functionalities.	25 %
Nanditha Bodanapu	FREQ-12 Designed HTML pages that admin specialist to send approval notice and students to download the approved degree plan. Involved in writing sendApprovaltoStudent(), downloadApprovedDP() methods of DegreePlanController and Pdf.java class.	25 %
Aravind Swamy Thottempudi	FREQ-4 Involved in writing approveUser(), approveUsers() and rejectUser() methods of the AdminController. Also involved in modifying the login and register tables to accommodate the changes for the admin approval before login	25 %
Sri Harshini Vallabhaneni	FREQ-14, FREQ-15 Involved in writing ContactUs(), staffDirectory( ) methods of admin controller. Also involved in creating the database tables required to hold contact information	25 %

e. User Manual

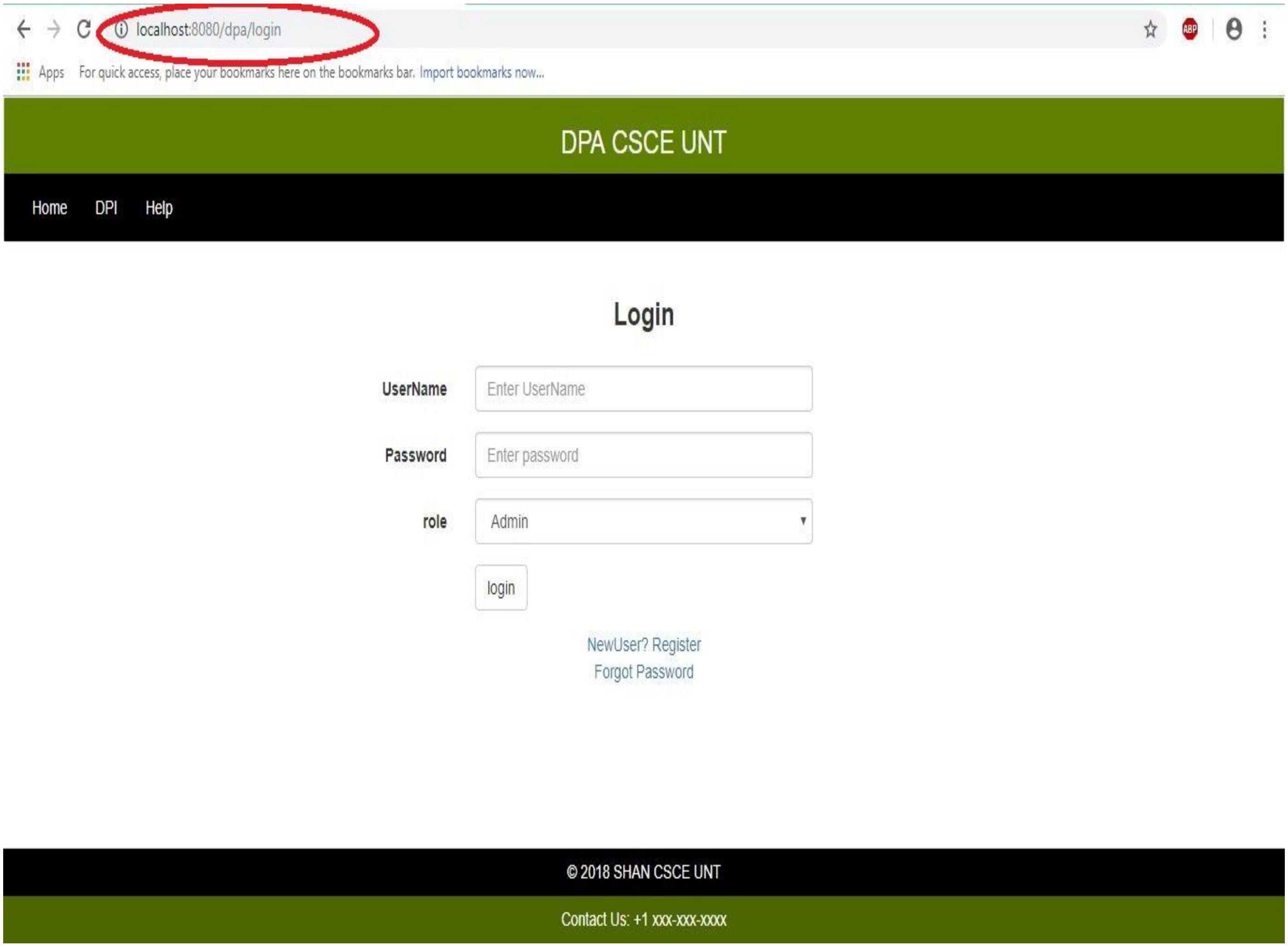
Degree Plan Automation System:

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

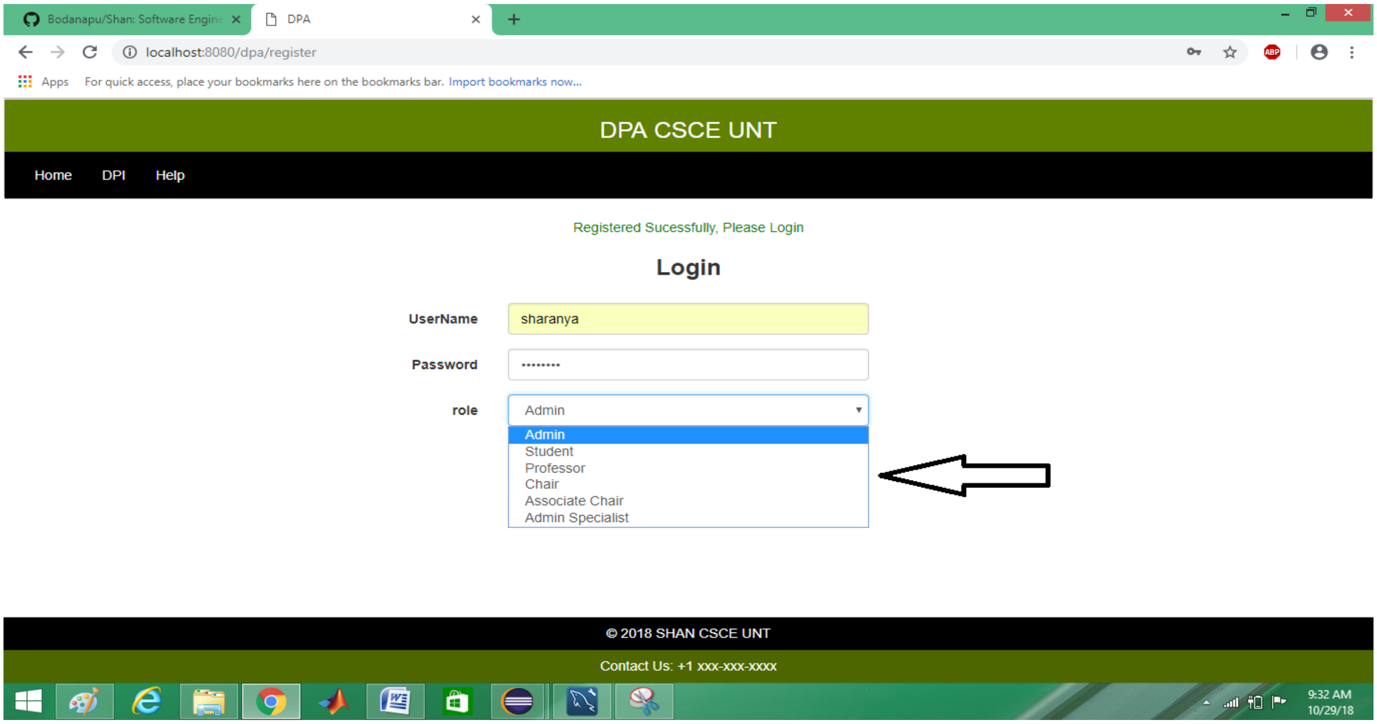
Eclipse
MYSQL workbench and server
Google chrome

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 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.


Bodanapu/Shan: Software Engine
DPA
localhost:8080/dpa/register
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
DPA CSCE UNT
Home
DPI
Help

Registered Sucessfully, Please Login

Login

UserName: sharanya

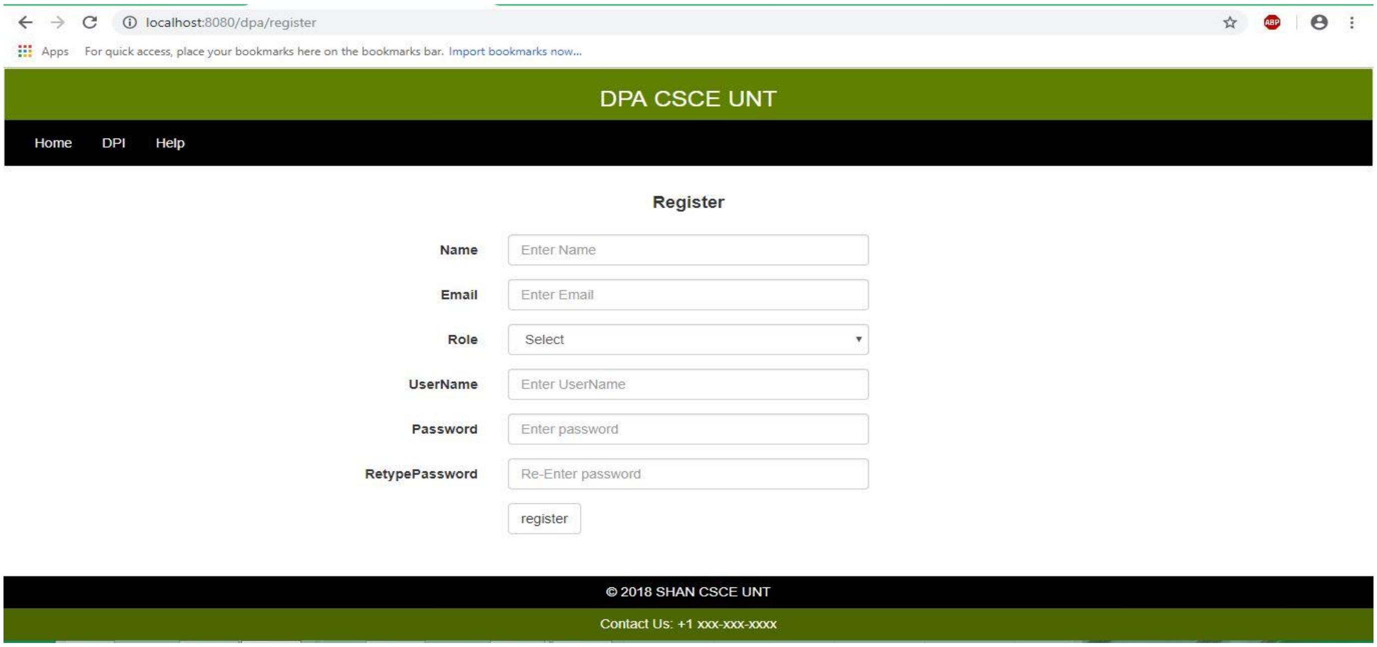
Password: ••••••••

role: Admin

- Admin
- Student
- Professor
- Chair
- Associate Chair
- Admin Specialist

© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx
9:32 AM
10/29/18

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


localhost:8080/dpa/register
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
DPA CSCE UNT
Home
DPI
Help

Register

Name: Enter Name

Email: Enter Email

Role: Select

UserName: Enter UserName

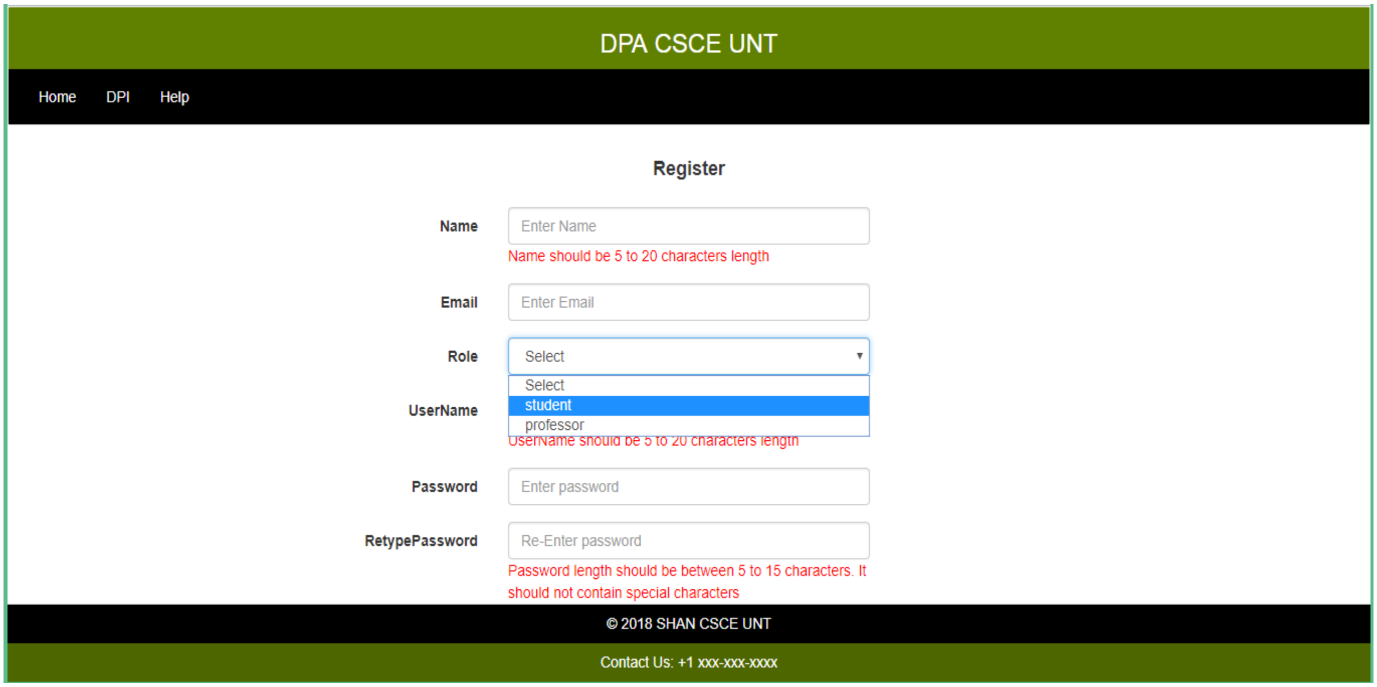
Password: Enter password

RetypePassword: Re-Enter password

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.

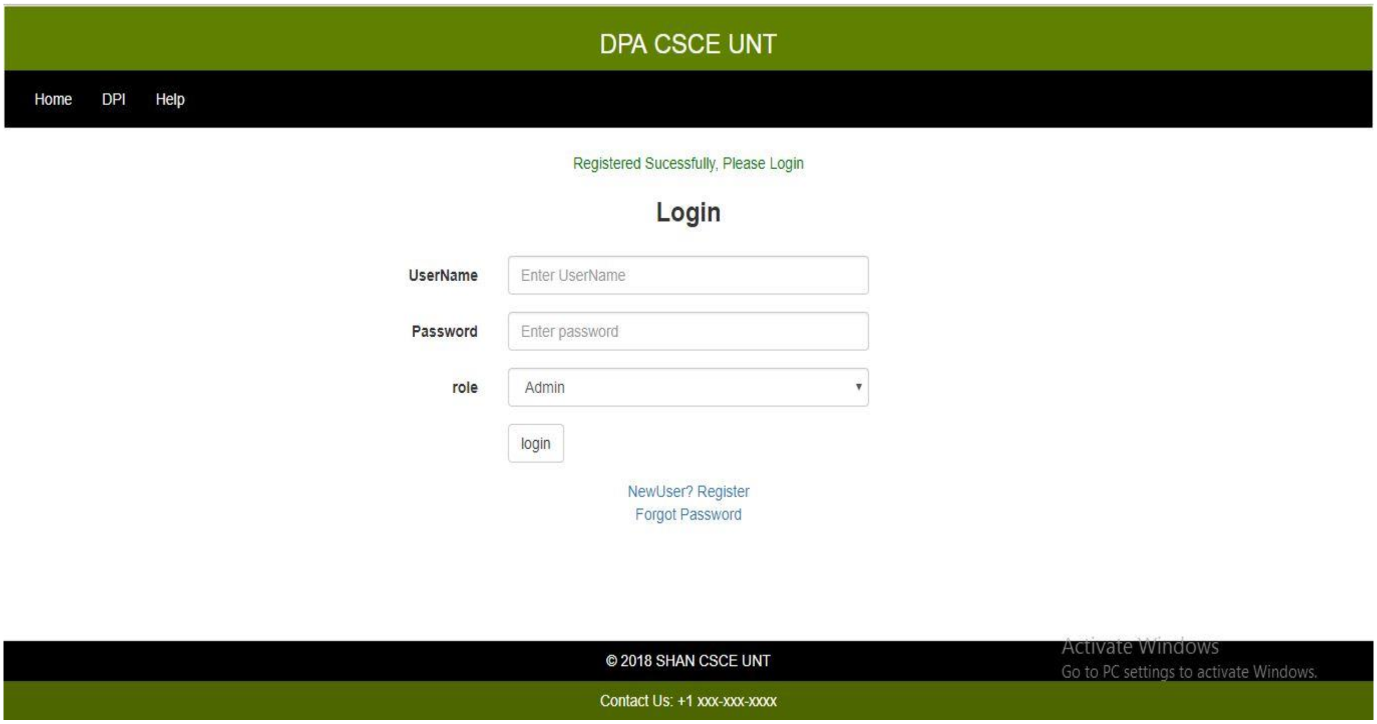


Name should be 5 to 20 characters length

UserName should be 5 to 20 characters length

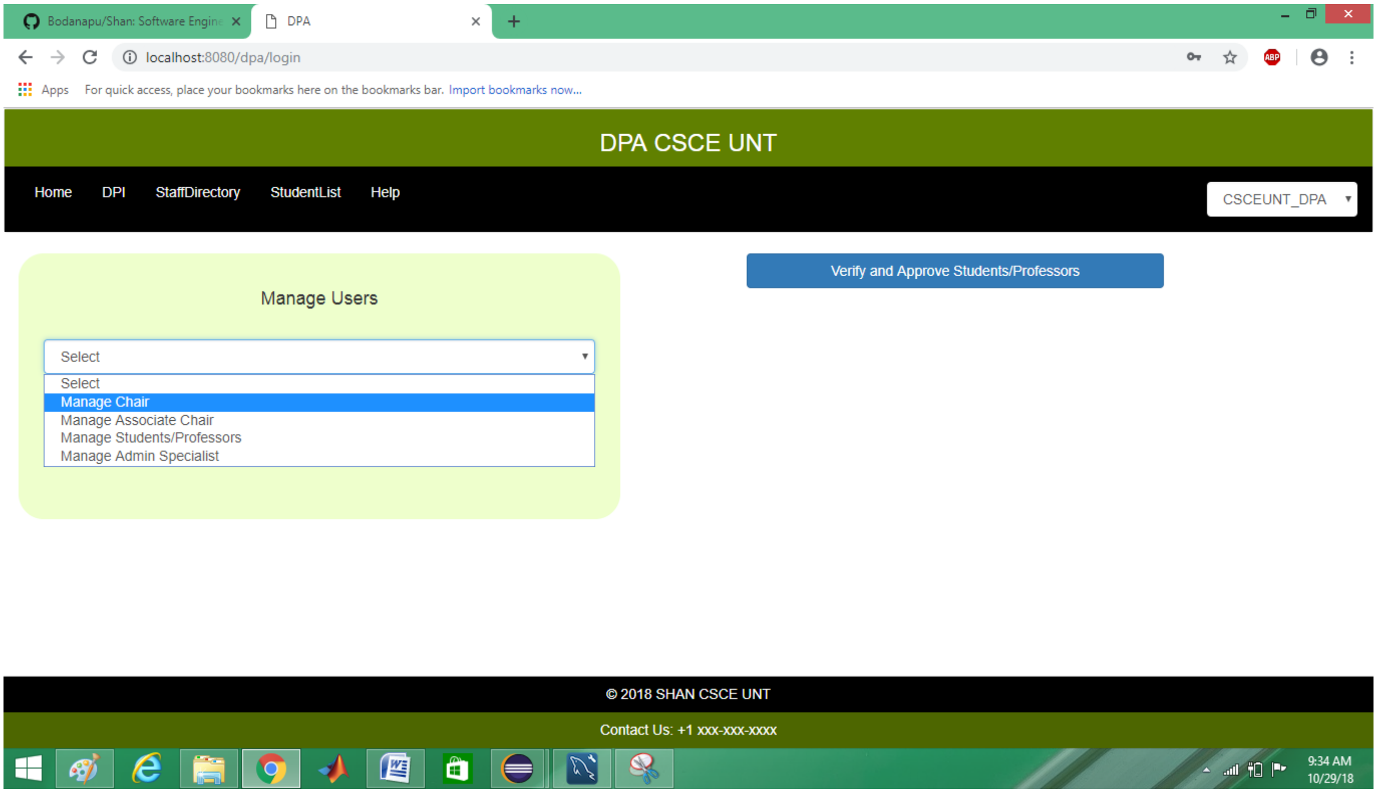
Password length should be between 5 to 15 characters. It should not contain special characters

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

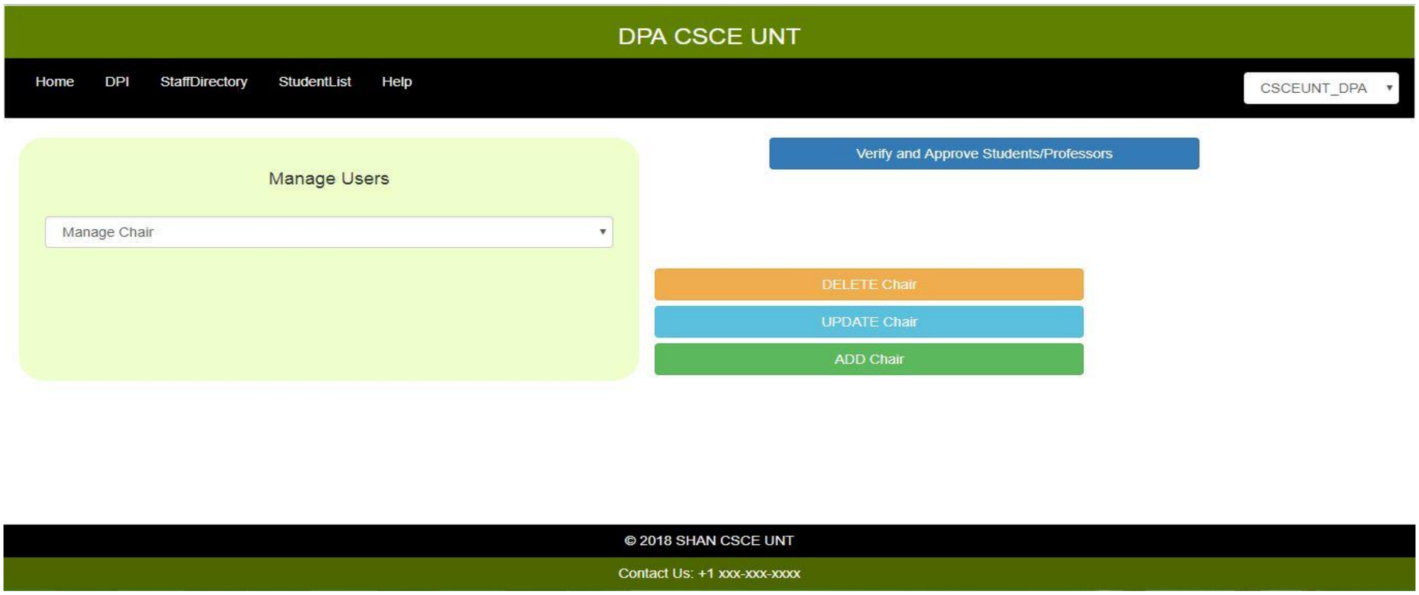


Registered Sucessfully, Please Login

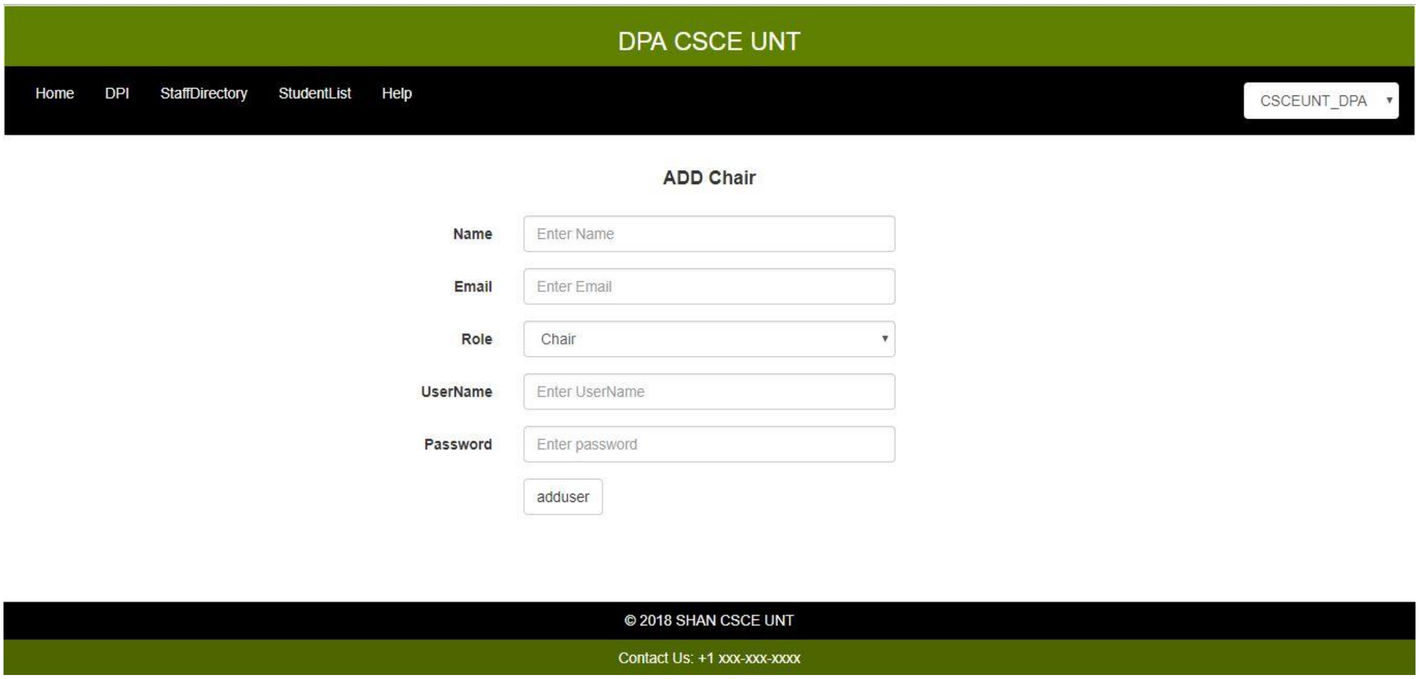
User login as Admin: If your login with role as admin, you will be directed to admin home page where you have options to manage chair, associate chair, students/professors and admin specialist.


Bodanapu/Shan: Software Engine
DPA
localhost:8080/dpa/login
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
DPA CSCE UNT
Home
DPI
StaffDirectory
StudentList
Help
CSCEUNT_DPA
Manage Users
Verify and Approve Students/Professors
Select
Select
Manage Chair
Manage Associate Chair
Manage Students/Professors
Manage Admin Specialist
© 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.



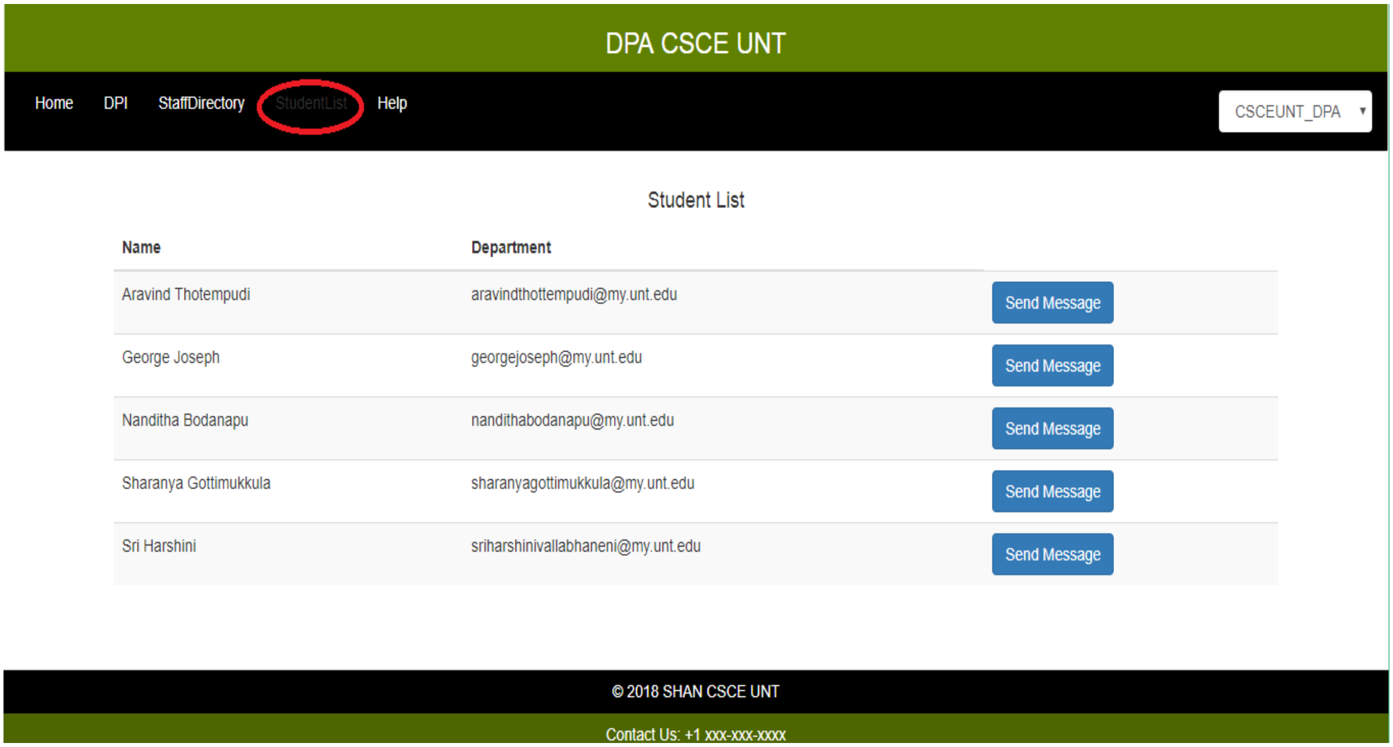
By clicking on ADD chair you will be directed to a add chair page where you can fill in the chair details and add him/her.



ADD Chair

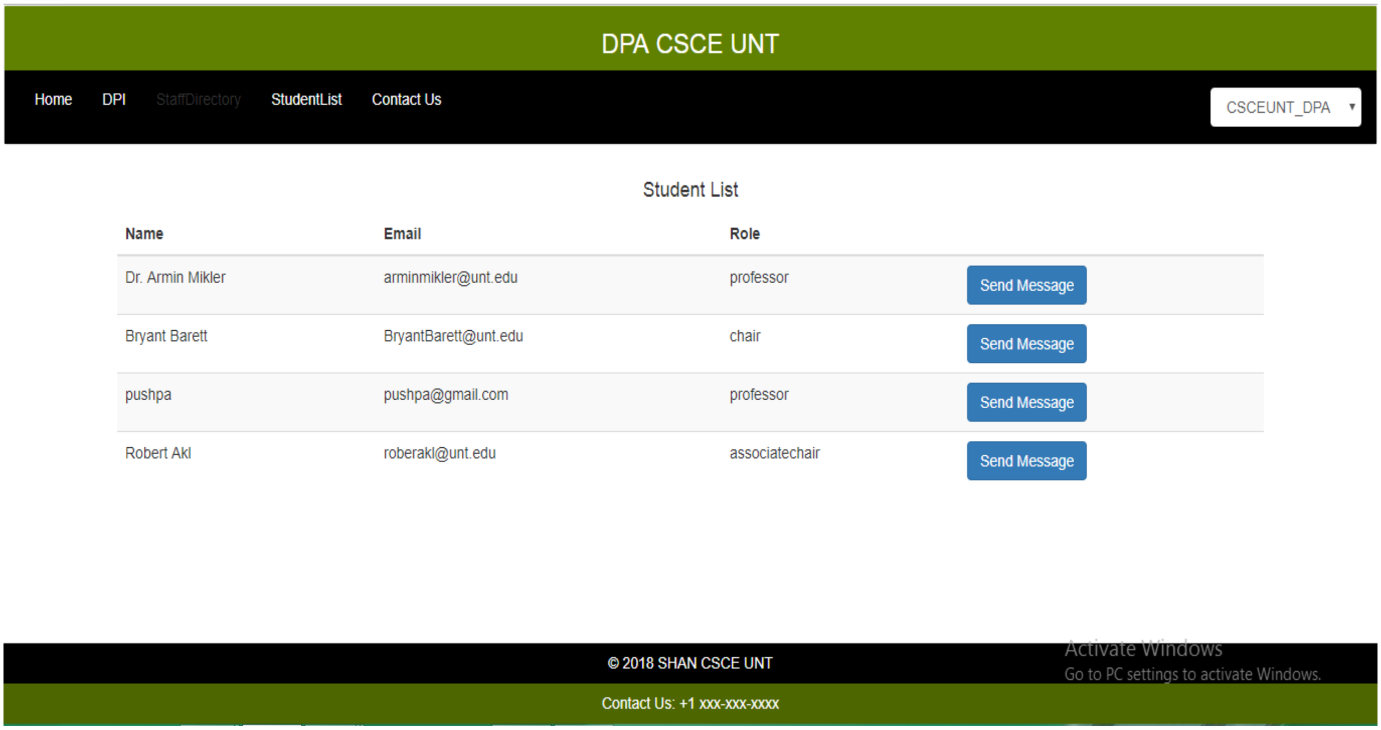
Name	Enter Name
Email	Enter Email
Role	Chair ▾
UserName	Enter UserName
Password	Enter password

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



Student List

Name	Department	
Aravind Thotempudi	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

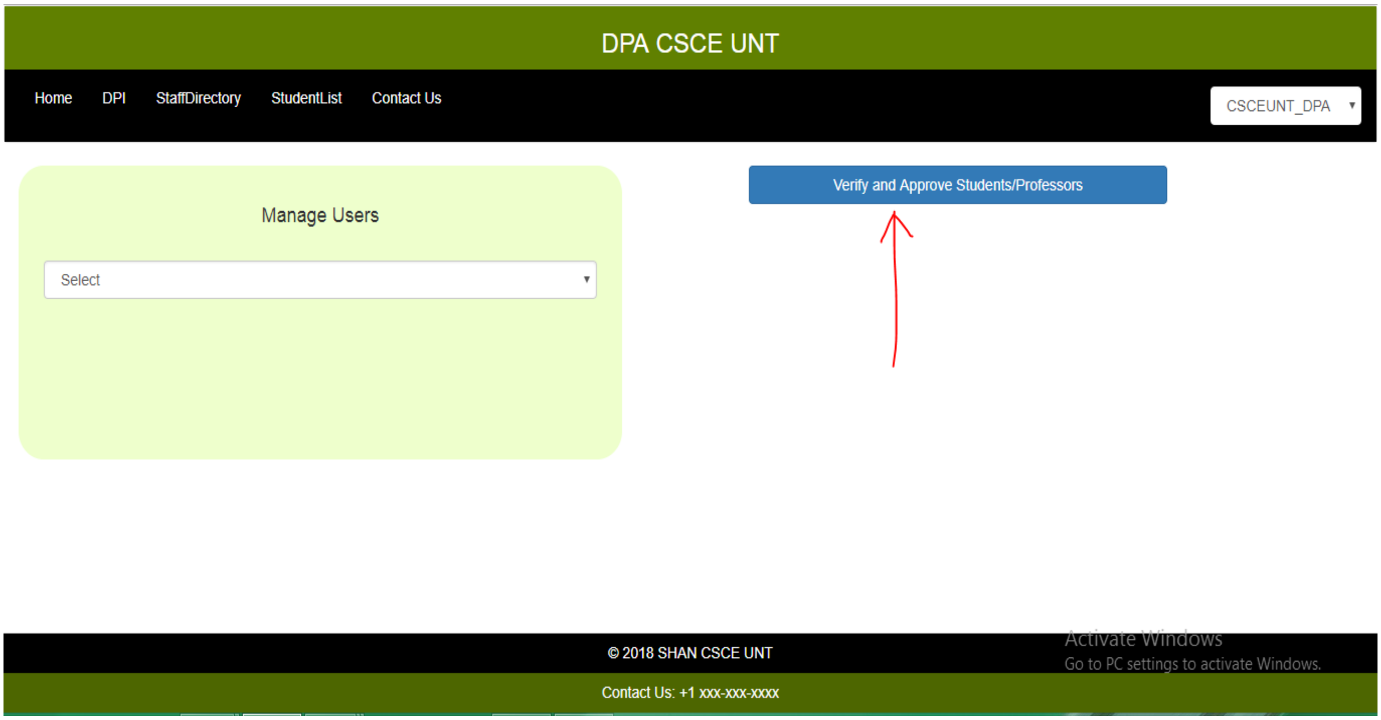


Student List

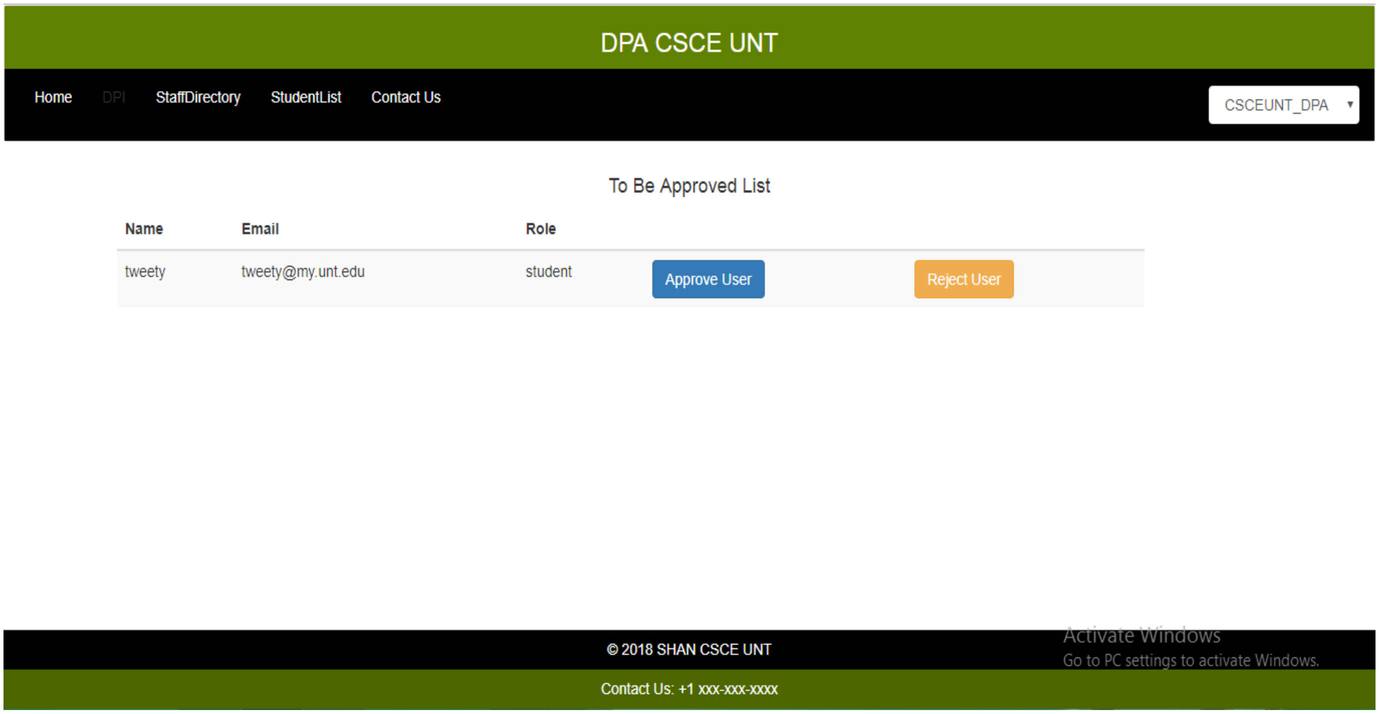
Name	Email	Role	
Dr. Armin Mikler	arminmikler@unt.edu	professor	Send Message
Bryant Barett	BryantBarett@unt.edu	chair	Send Message
pushpa	pushpa@gmail.com	professor	Send Message
Robert Akl	roberakl@unt.edu	associatechair	Send Message

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

You can view the list of students or professors registered to access the website by clicking verify and approve students/professors’ button



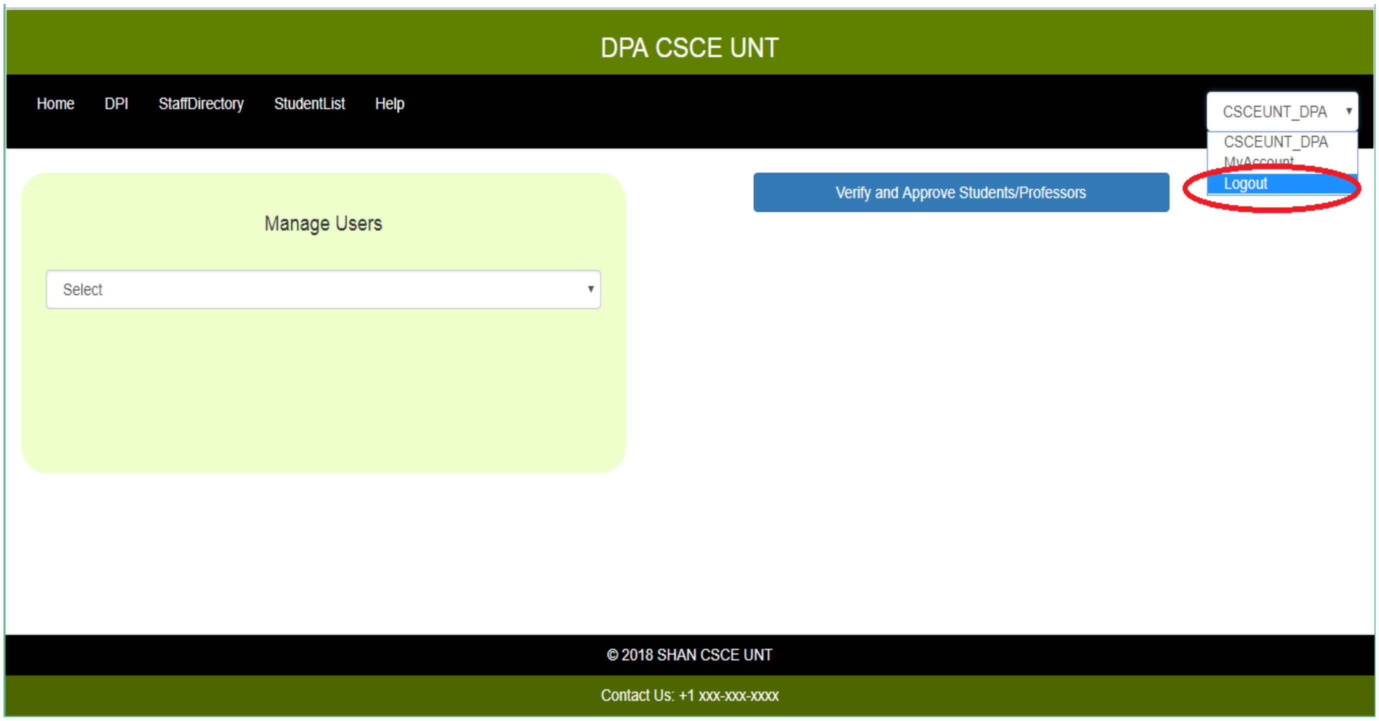
Then by clicking on “approve” button you can approve the user and by clicking on “reject” button reject the user



To Be Approved List

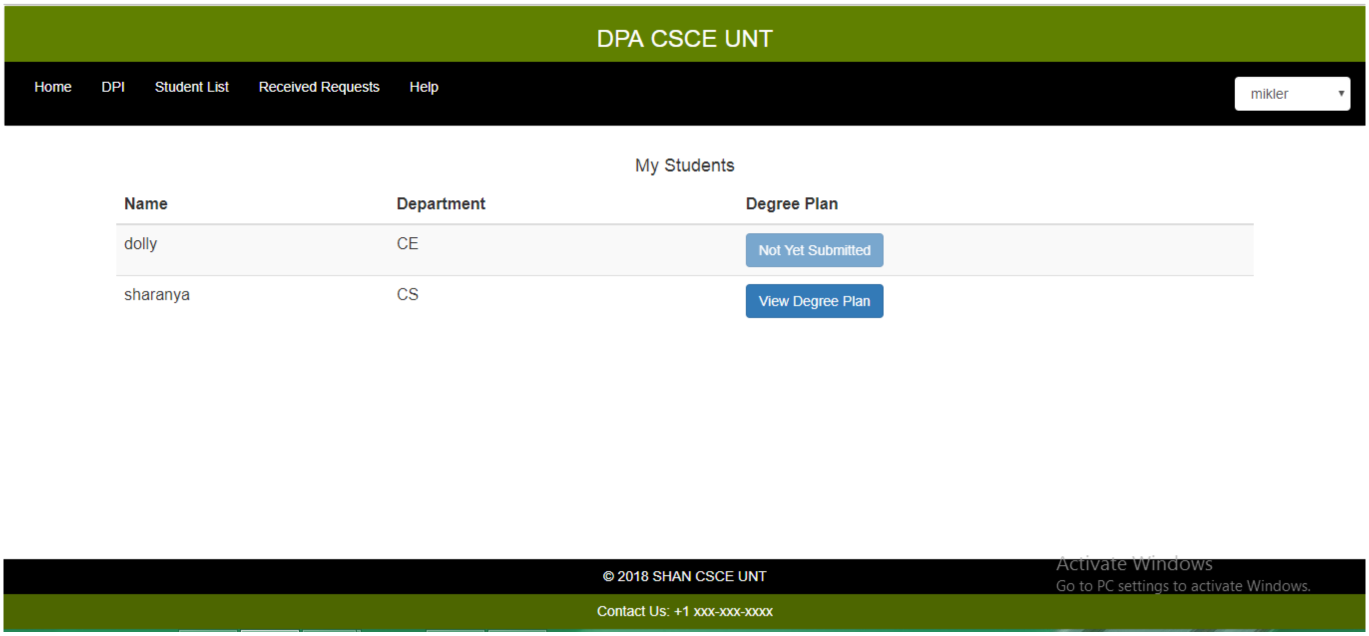
Name	Email	Role		
tweety	tweety@my.unt.edu	student	Approve User	Reject User

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



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



My Students

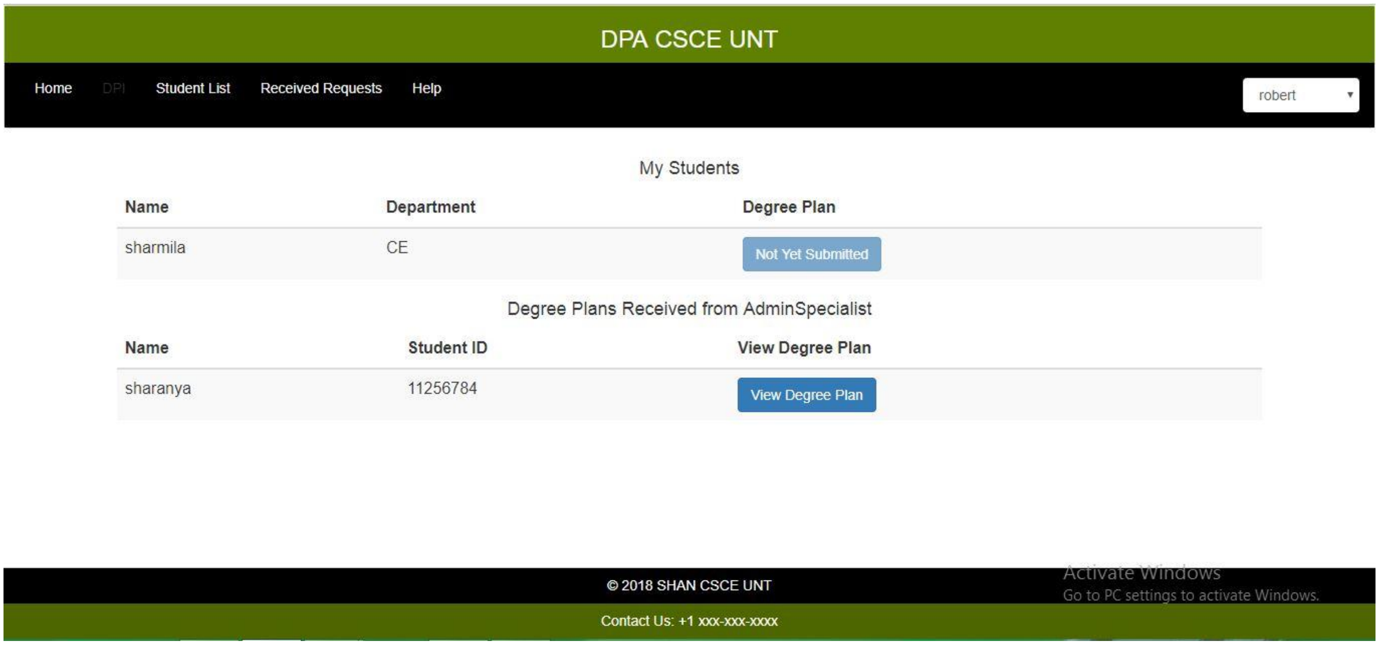
Name	Department	Degree Plan
dolly	CE	Not Yet Submitted
sharanya	CS	View Degree Plan

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

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.



My Students

Name	Department	Degree Plan
sharmila	CE	Not Yet Submitted

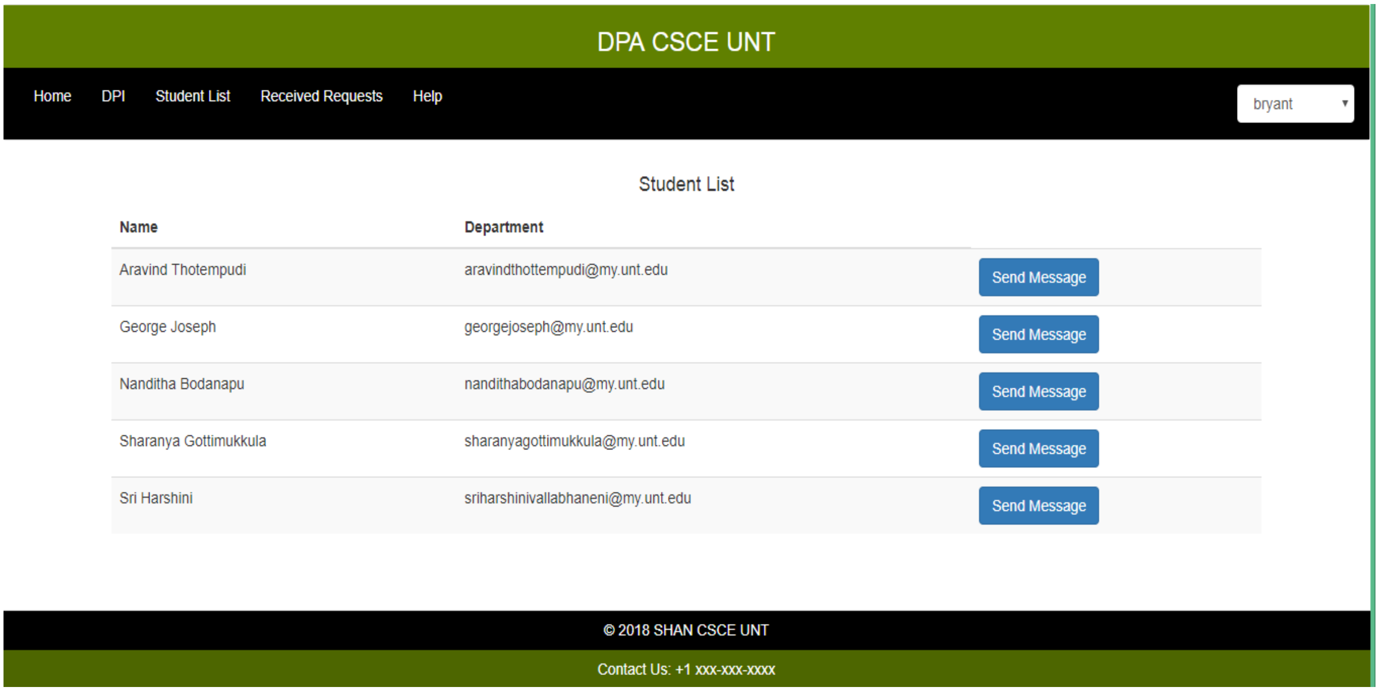
Degree Plans Received from AdminSpecialist

Name	Student ID	View Degree Plan
sharanya	11256784	View Degree Plan

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

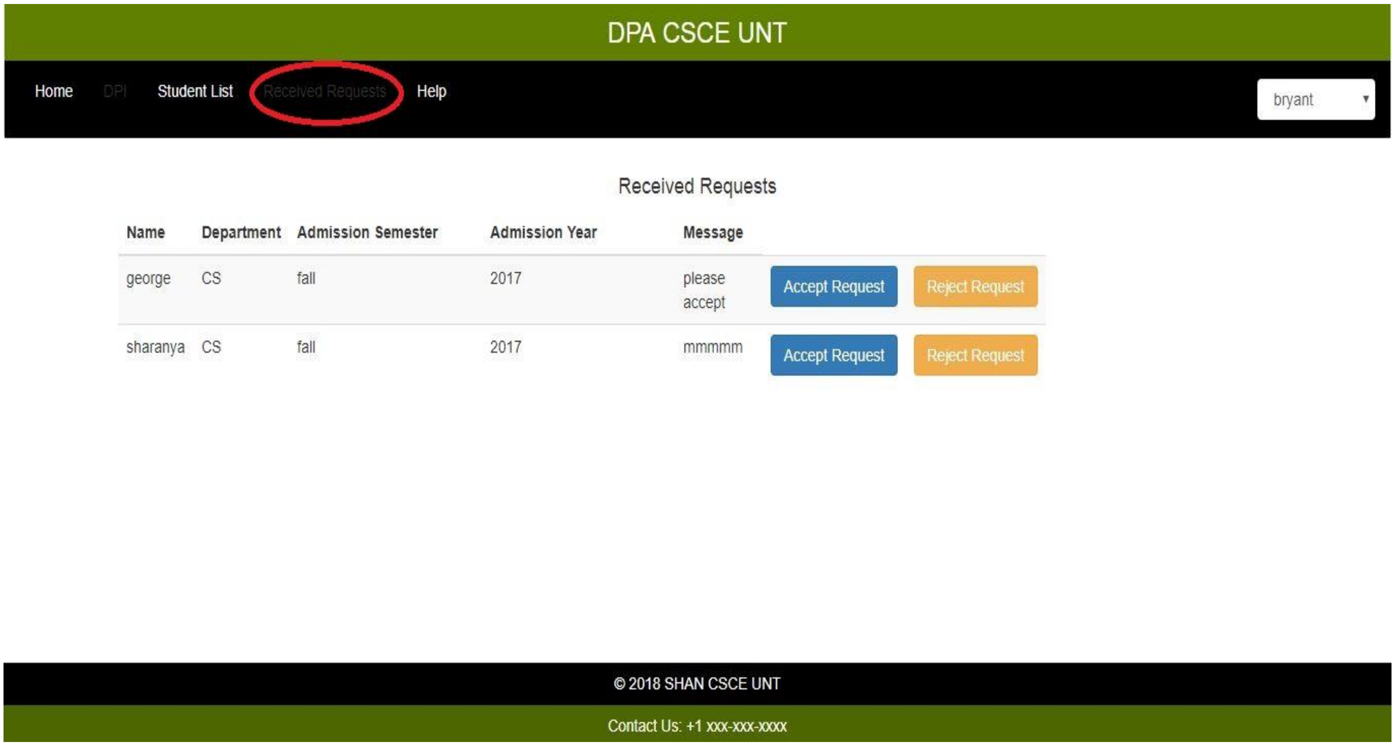
You have options to view Student List and received requests by clicking on the options "StudentList" and "ReceivedRequests" on the navigation bar at the top.



Student List

Name	Department	
Aravind Thotempudi	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

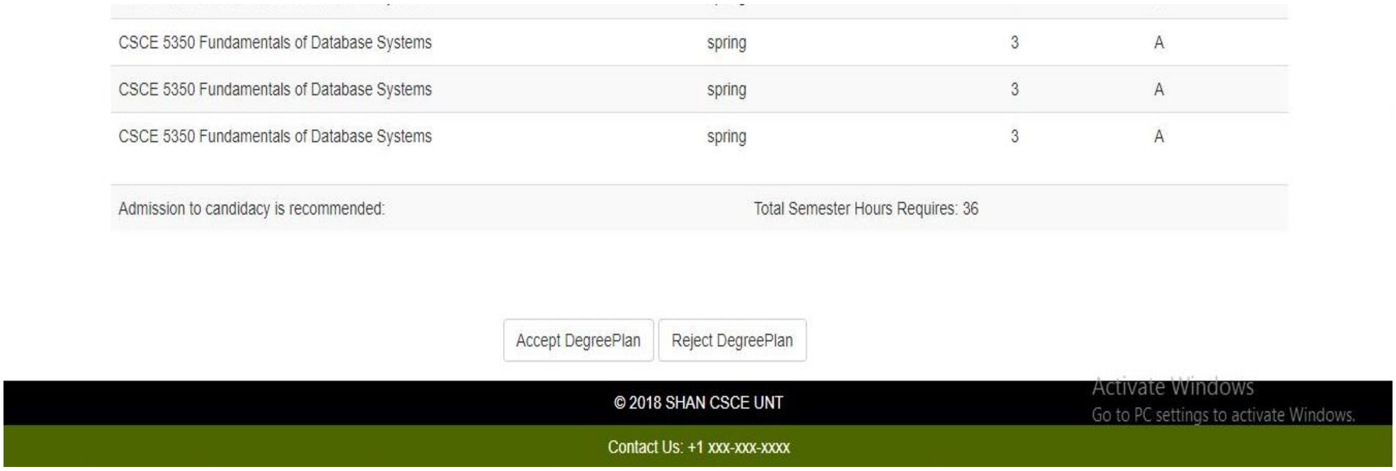
You can accept the request by clicking on the Accept Request button and reject the request by clicking on the Reject Request button



Received Requests

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

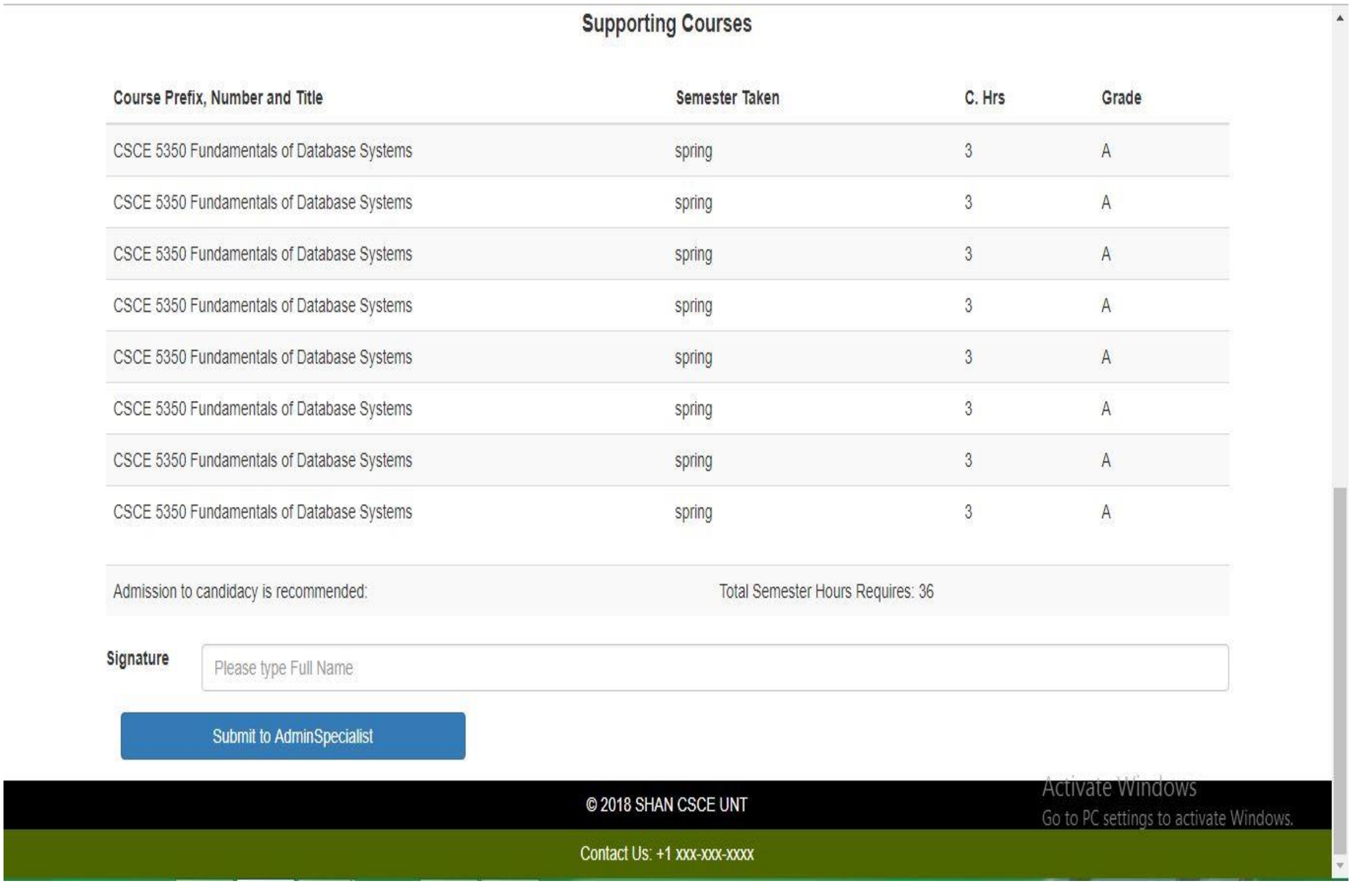
You can approve the degree plan received from students as well as degree plans received from administrative specialist by clicking “Accept Degree Plan” and can reject by clicking “Reject Degree Plan” button.



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

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



Supporting Courses

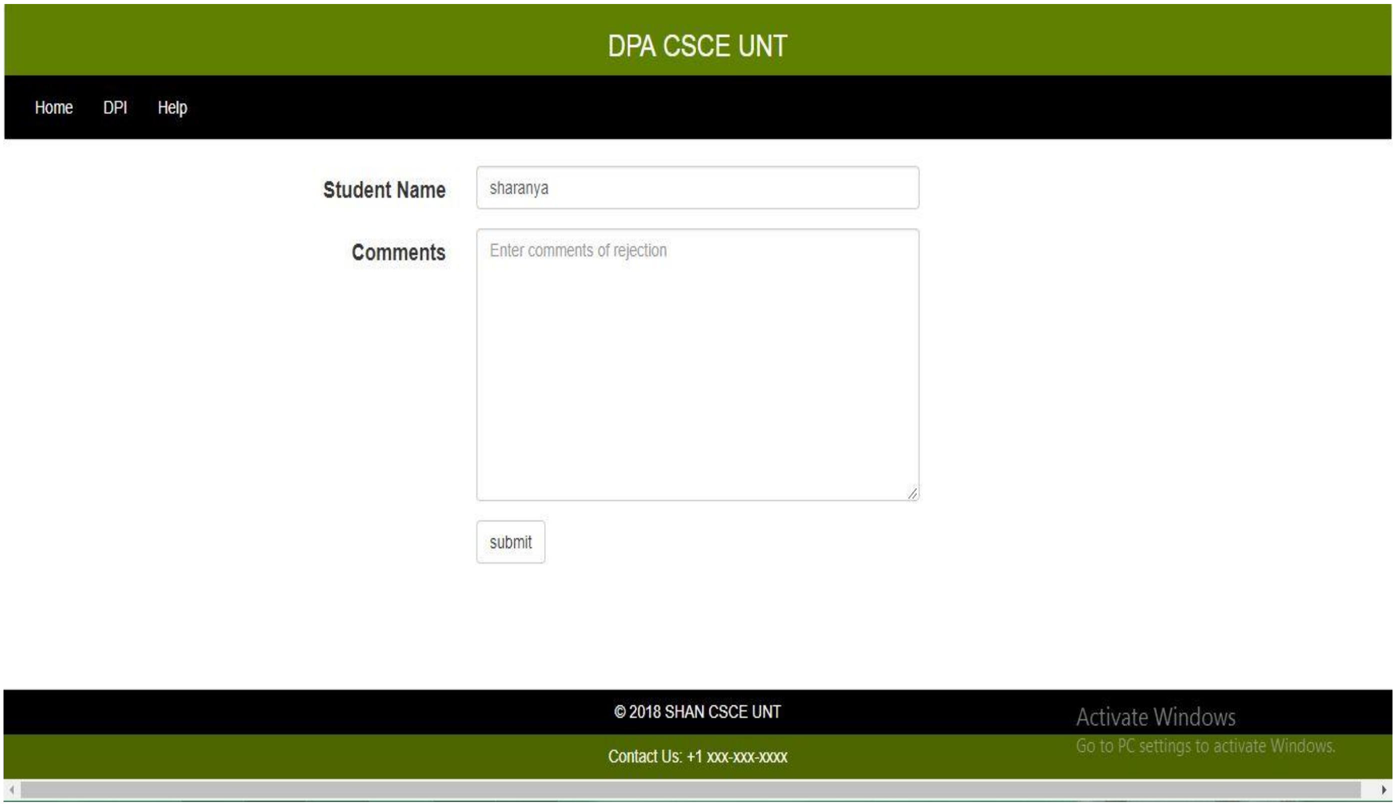
Course Prefix, Number and Title	Semester Taken	C. Hrs	Grade
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of 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
Contact Us: +1 xxx-xxx-xxxx
Activate Windows
Go to PC settings to activate Windows.

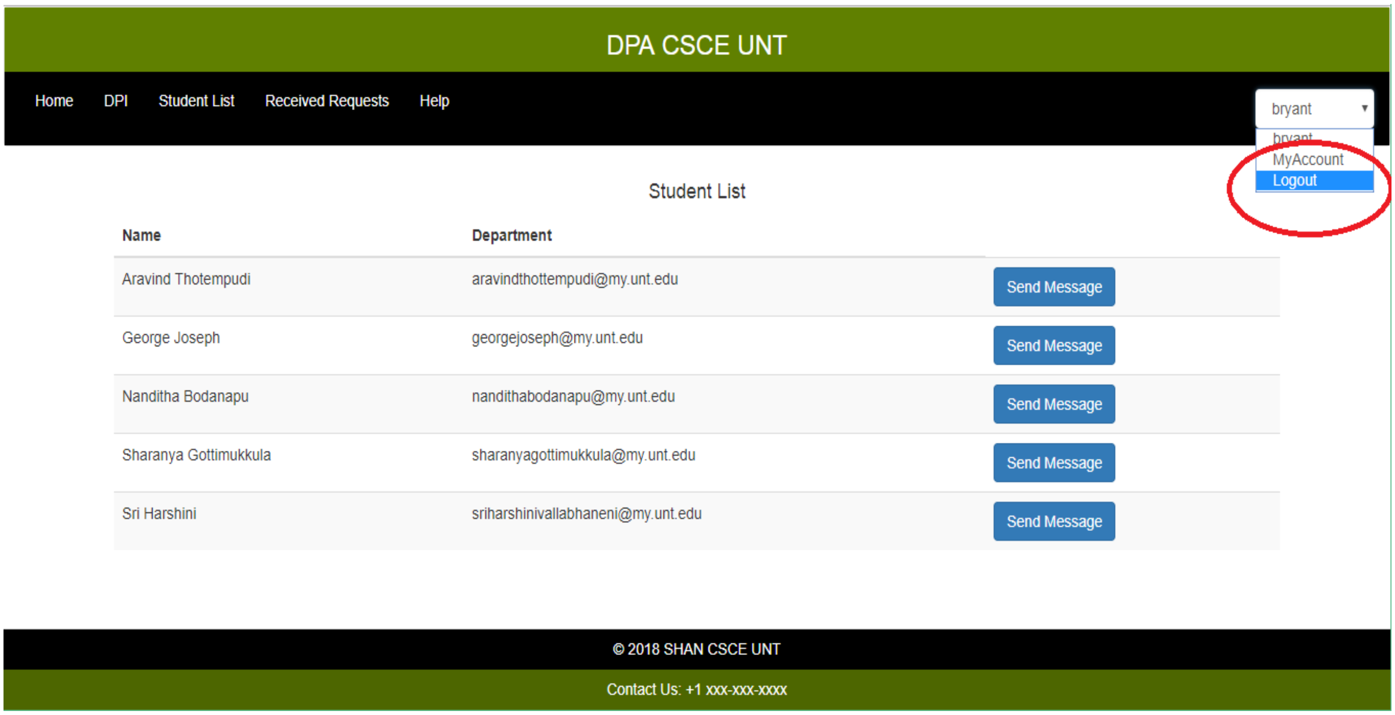
The professor needs to add comments, which allows the student to update the degree plan as per the comments and resubmit it. All this needs to be done when you click “Reject Degree Plan” button.


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

Student Name sharanya

Comments Enter comments of rejection

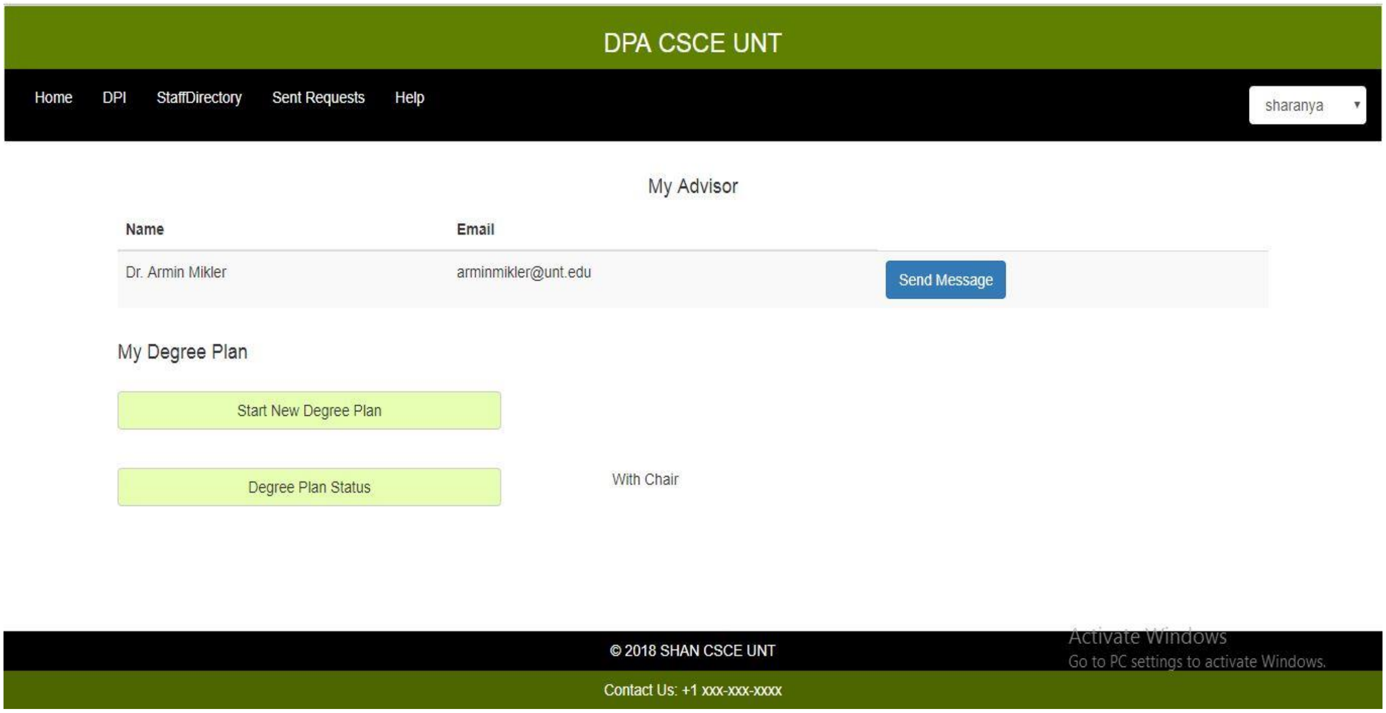
You can logout by clicking the button logout present in the dropdown with your username display.



Student List

Name	Department	
Aravind Thotempudi	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

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.



My Advisor

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

My Degree Plan

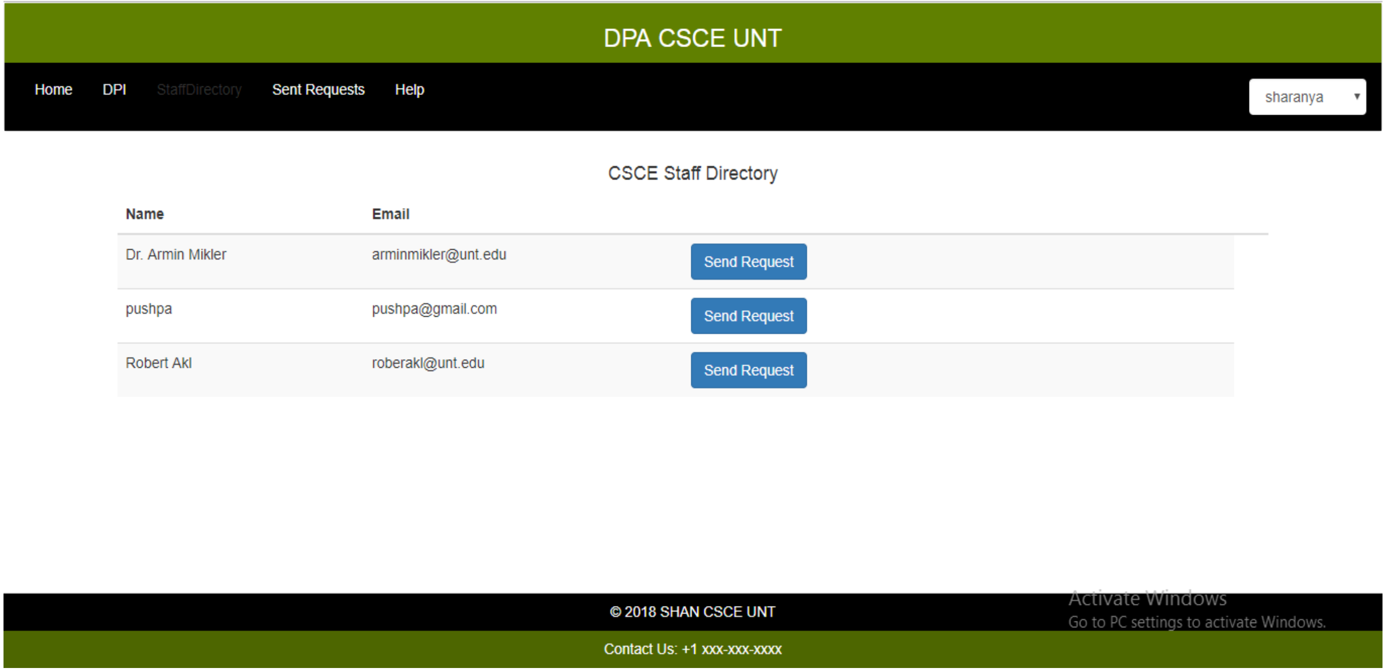
Start New Degree Plan

Degree Plan Status With Chair

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

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.



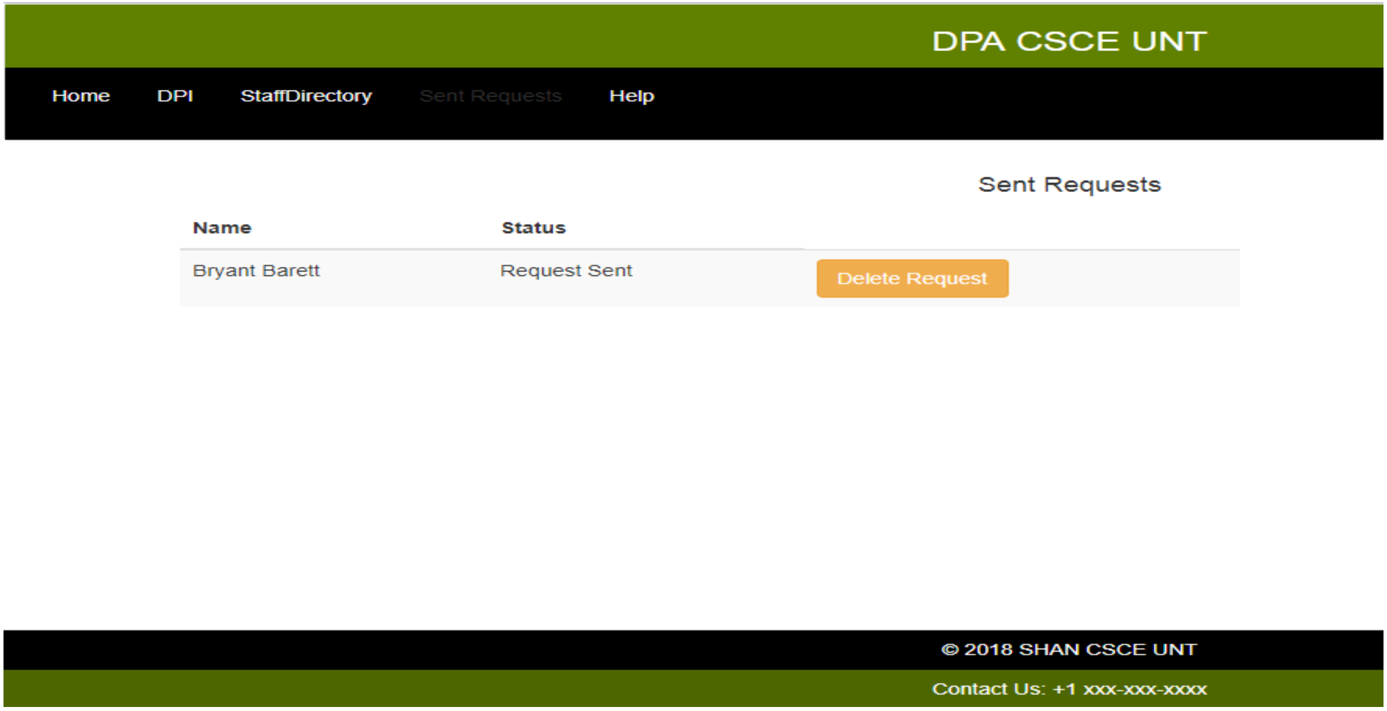
CSCE Staff Directory

Name	Email	
Dr. Armin Mikler	arminmikler@unt.edu	Send Request
pushpa	pushpa@gmail.com	Send Request
Robert Akl	roberakl@unt.edu	Send Request

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

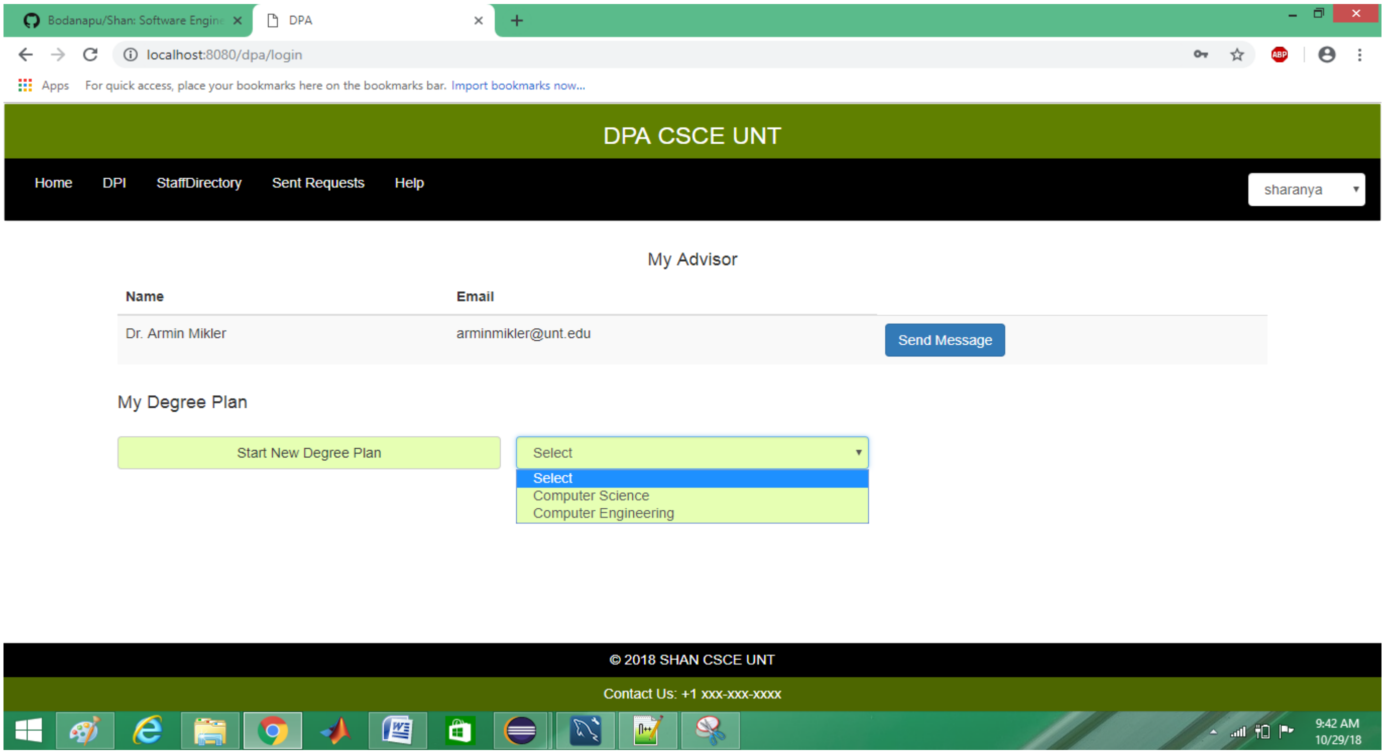
You can view the sent requests by clicking on “sent requests” link in the menu and then click delete if you want to delete the sent request.



Sent Requests

Name	Status	
Bryant Barett	Request Sent	Delete Request

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.

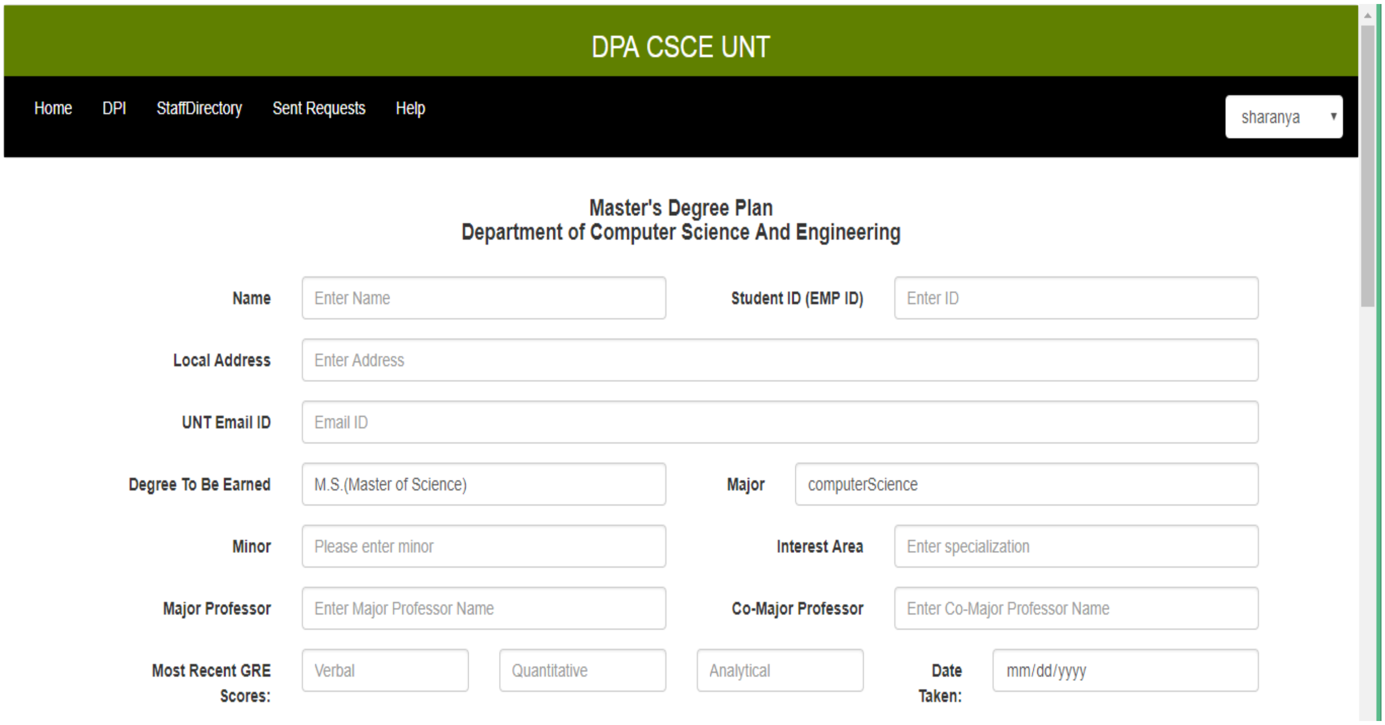


My Advisor

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

My Degree Plan

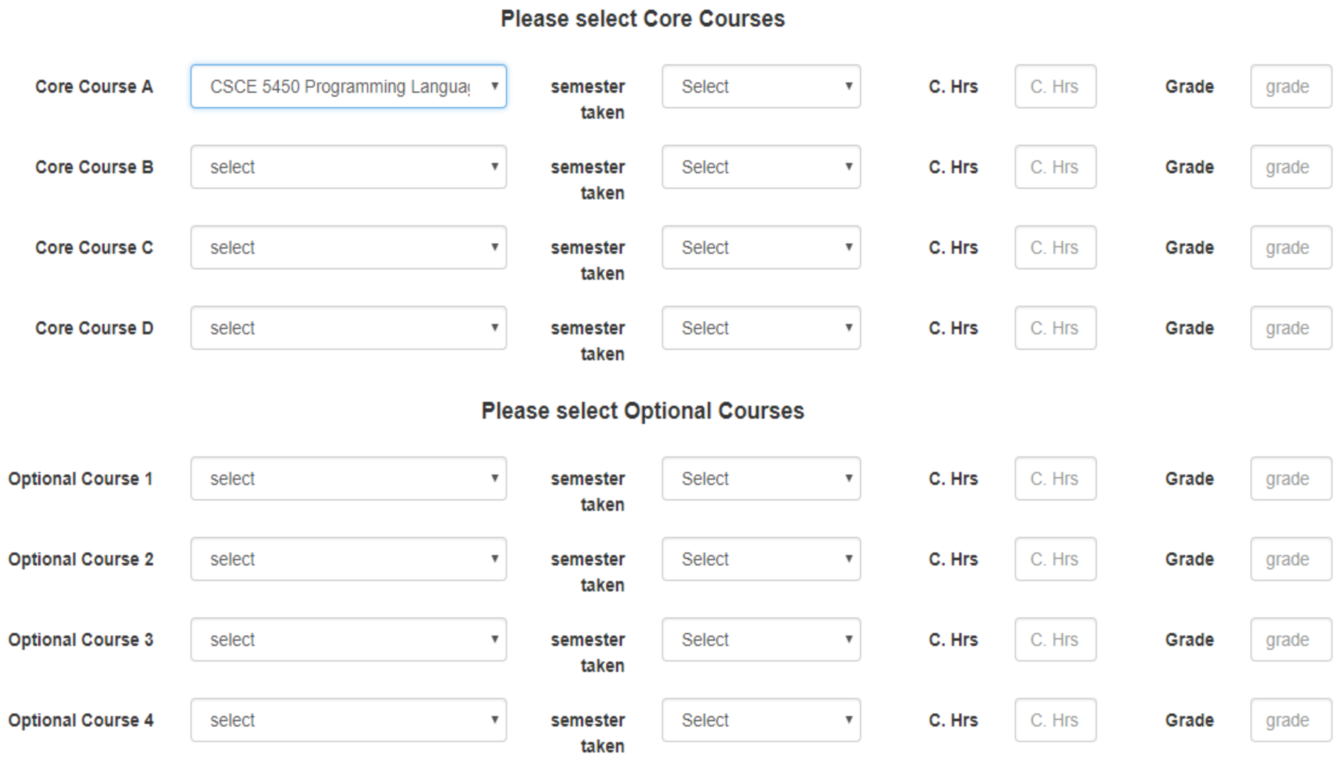
After choosing the major the degree plan form as in below screenshots will be displayed. The major will be auto populated as per your choice of major.



Master's Degree Plan
Department of Computer Science And Engineering

Name	Enter Name	Student ID (EMP ID)	Enter ID
Local Address	Enter Address		
UNT Email ID	Email ID		
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



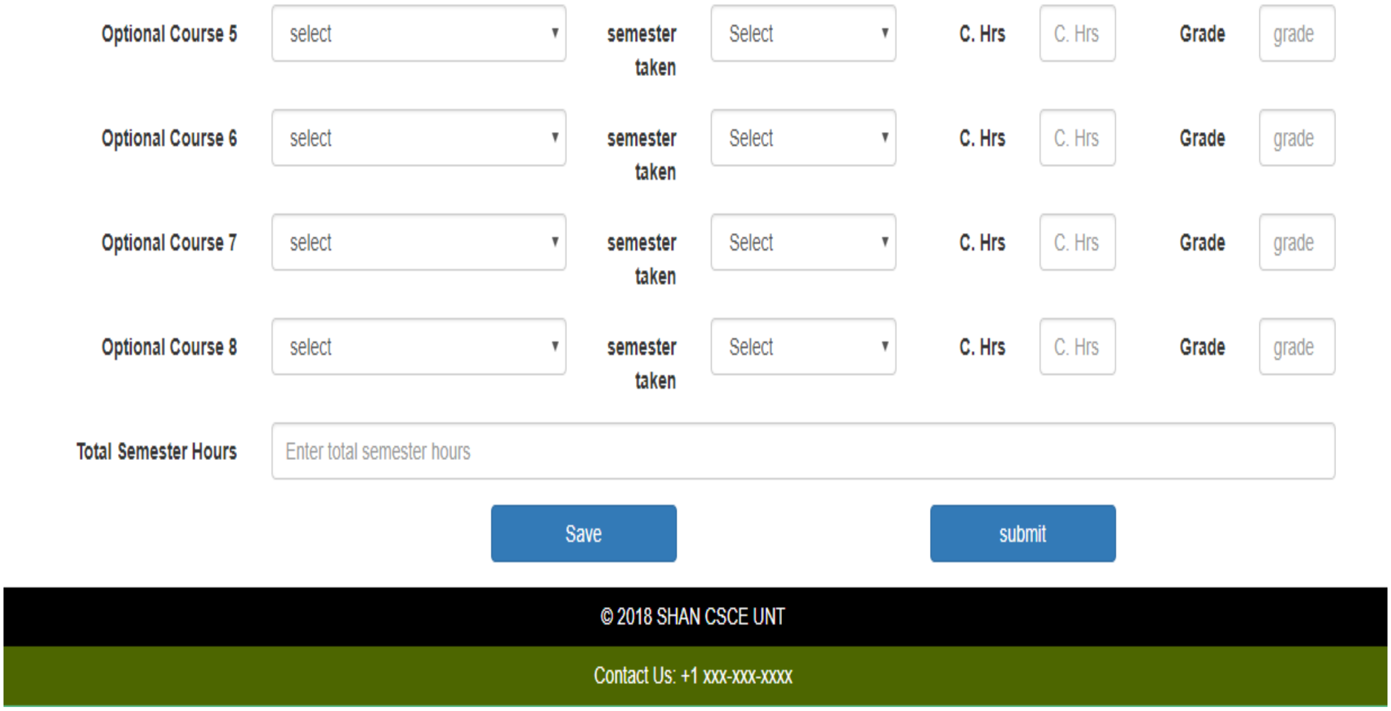
Please select Core Courses

Core Course A	CSCE 5450 Programming Langua▾	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

Please select Optional Courses

Optional Course 1	select ▾	semester taken	Select ▾	C. Hrs	C. Hrs	Grade	grade
Optional Course 2	select ▾	semester taken	Select ▾	C. Hrs	C. Hrs	Grade	grade
Optional Course 3	select ▾	semester taken	Select ▾	C. Hrs	C. Hrs	Grade	grade
Optional Course 4	select ▾	semester taken	Select ▾	C. Hrs	C. Hrs	Grade	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

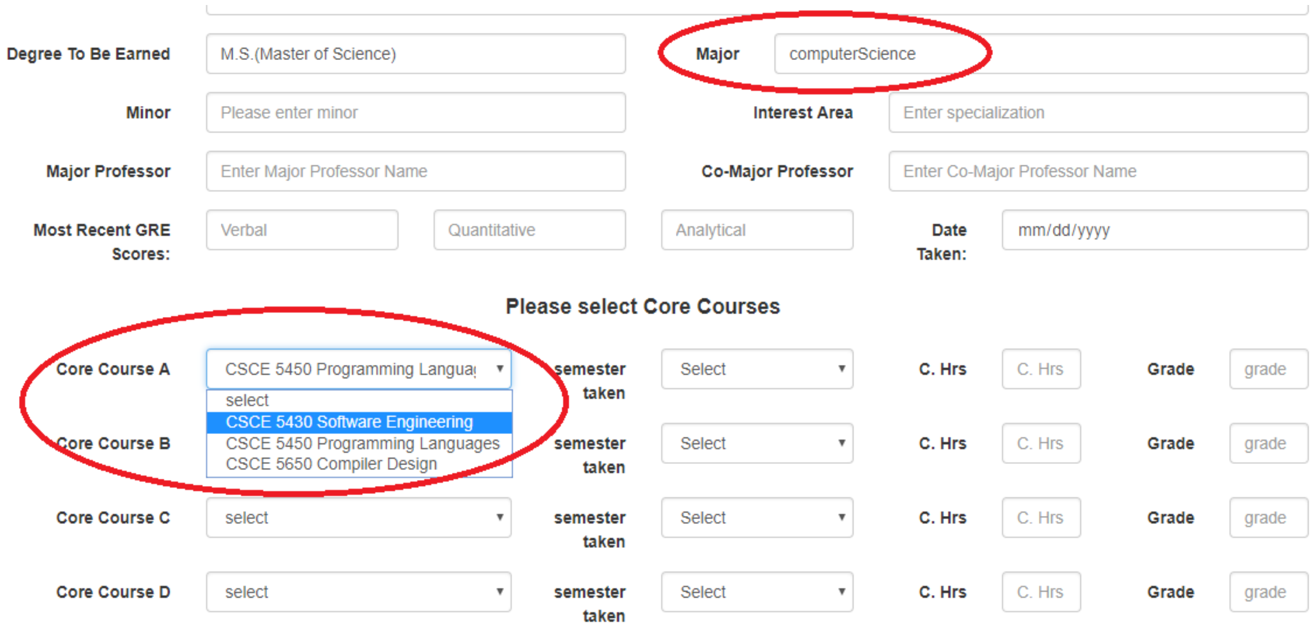
Total Semester Hours: Enter total semester hours

© 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



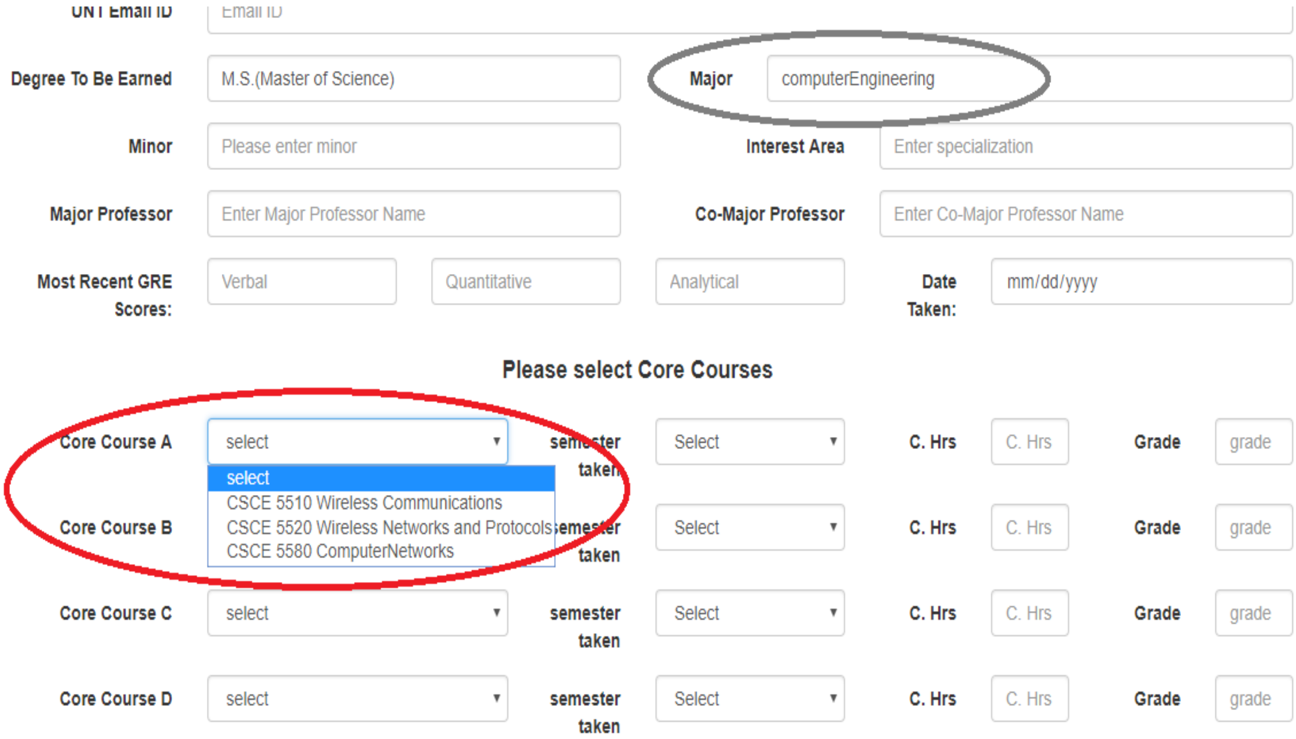
Degree To Be Earned	M.S.(Master of Science)		Major	computerScience	
Minor	Please enter minor		Interest Area	Enter specialization	
Major Professor	Enter Major Professor Name		Co-Major Professor	Enter Co-Major Professor Name	
Most Recent GRE Scores:	Verbal	Quantitative	Analytical	Date Taken:	mm/dd/yyyy

Please select Core Courses

Core Course A	CSCE 5450 Programming Languaį	semester taken	Select	C. Hrs	C. Hrs	Grade	grade
Core Course B		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

select
CSCE 5430 Software Engineering
CSCE 5450 Programming Languages
CSCE 5650 Compiler Design

Core course list when Computer Engineering is chosen



UNT 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

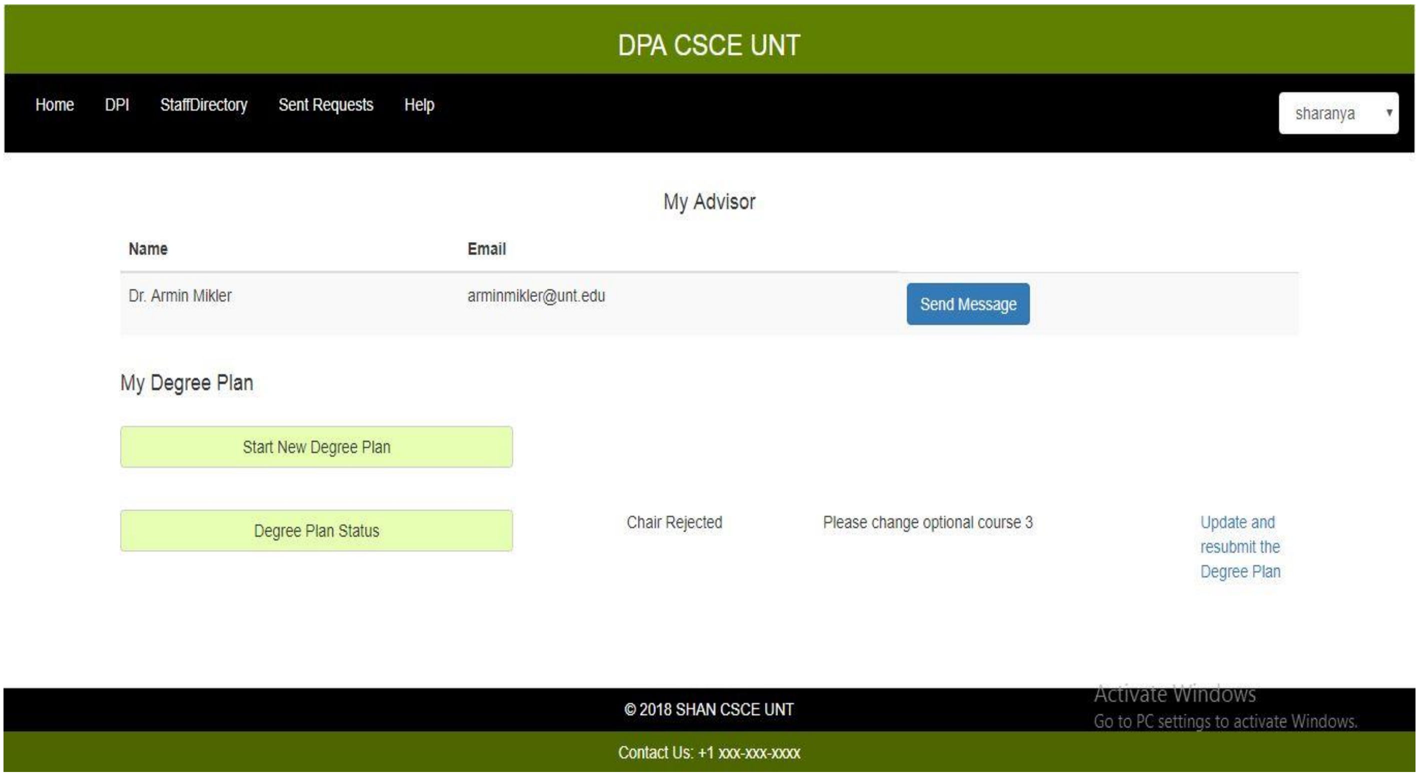
Date Taken: mm/dd/yyyy

Please select Core Courses

Core Course A	select ▾	semester taken	Select ▾	C. Hrs	C. Hrs	Grade	grade
Core Course B		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

select
CSCE 5510 Wireless Communications
CSCE 5520 Wireless Networks and Protocols
CSCE 5580 ComputerNetworks

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

My Advisor

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

My Degree Plan

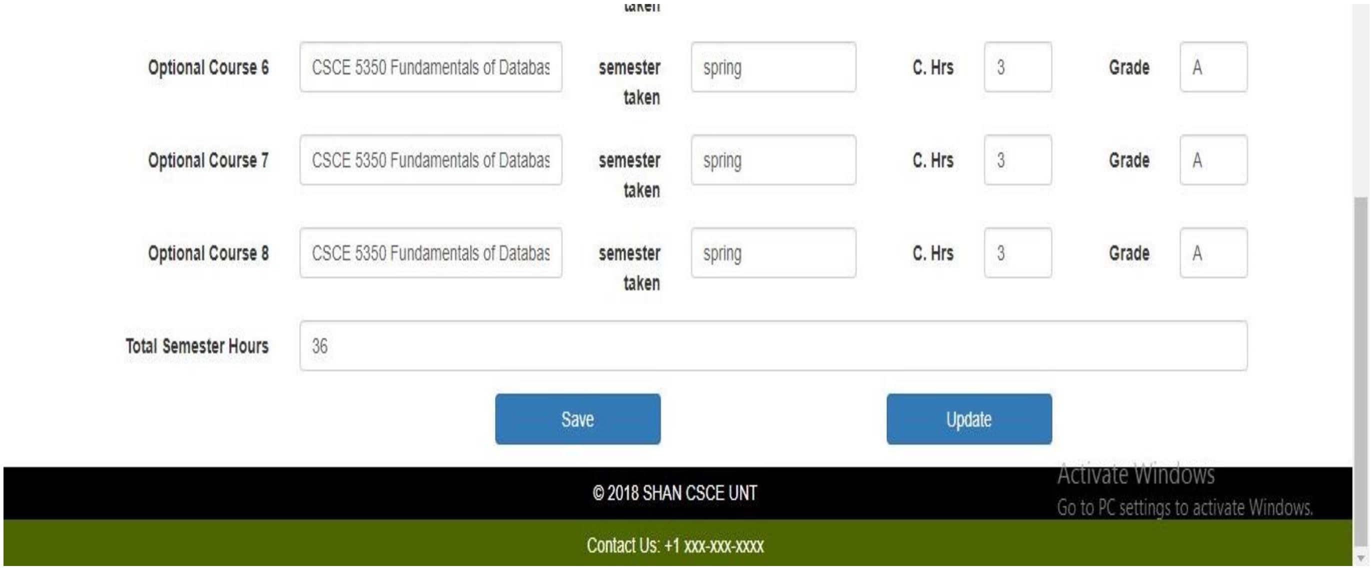
Start New Degree Plan

Degree Plan Status | Chair Rejected | Please change optional course 3 | Update and resubmit the Degree Plan

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

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

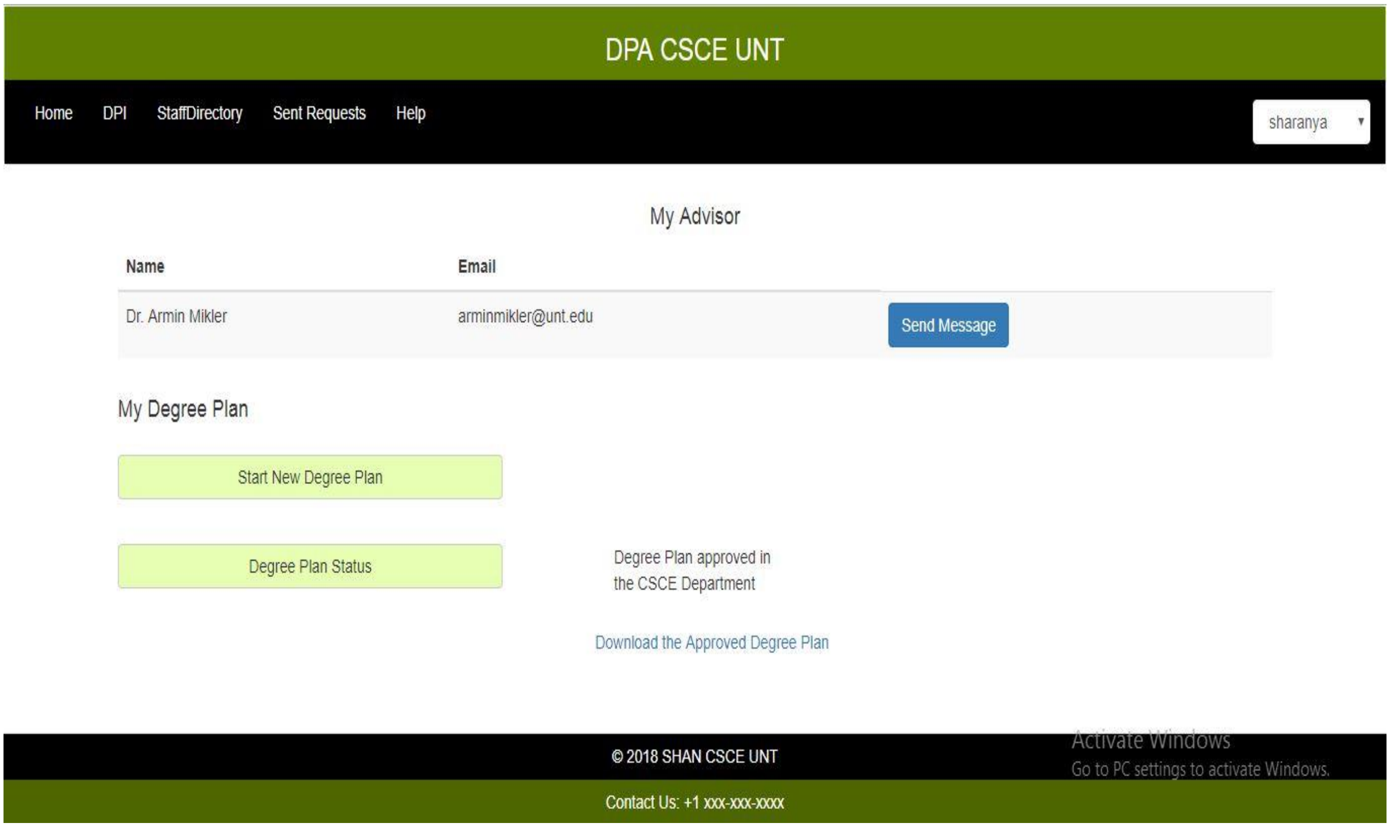


taken

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

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

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



DPA CSCE UNT

My Advisor

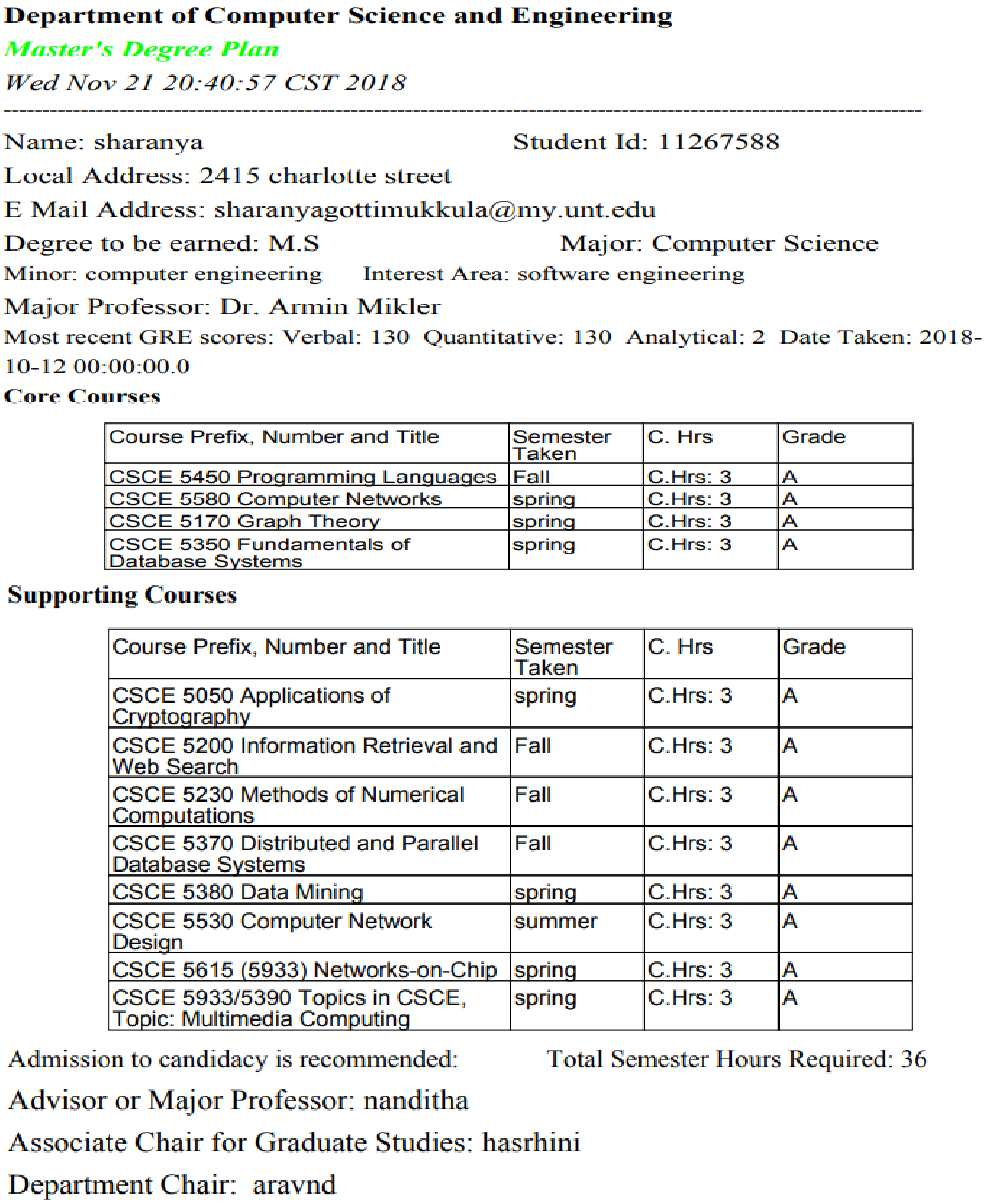
Name	Email	
Dr. Armin Mikler	arminmikler@unt.edu	Send Message

My Degree Plan

Degree Plan approved in the CSCE Department

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

The downloaded degree plan looks like below:



Department of Computer Science and Engineering

Master's Degree Plan

Wed Nov 21 20:40:57 CST 2018

Name: sharanya Student Id: 11267588

Local Address: 2415 charlotte street

E Mail Address: sharanyagottimukkula@my.unt.edu

Degree to be earned: M.S Major: Computer Science

Minor: computer engineering Interest Area: software engineering

Major Professor: Dr. Armin Mikler

Most recent GRE scores: Verbal: 130 Quantitative: 130 Analytical: 2 Date Taken: 2018-10-12 00:00:00.0

Core Courses

Course Prefix, Number and Title	Semester Taken	C. Hrs	Grade
CSCE 5450 Programming Languages	Fall	C.Hrs: 3	A
CSCE 5580 Computer Networks	spring	C.Hrs: 3	A
CSCE 5170 Graph Theory	spring	C.Hrs: 3	A
CSCE 5350 Fundamentals of Database Systems	spring	C.Hrs: 3	A

Supporting Courses

Course Prefix, Number and Title	Semester Taken	C. Hrs	Grade
CSCE 5050 Applications of Cryptography	spring	C.Hrs: 3	A
CSCE 5200 Information Retrieval and Web Search	Fall	C.Hrs: 3	A
CSCE 5230 Methods of Numerical Computations	Fall	C.Hrs: 3	A
CSCE 5370 Distributed and Parallel Database Systems	Fall	C.Hrs: 3	A
CSCE 5380 Data Mining	spring	C.Hrs: 3	A
CSCE 5530 Computer Network Design	summer	C.Hrs: 3	A
CSCE 5615 (5933) Networks-on-Chip	spring	C.Hrs: 3	A
CSCE 5933/5390 Topics in CSCE, Topic: Multimedia Computing	spring	C.Hrs: 3	A

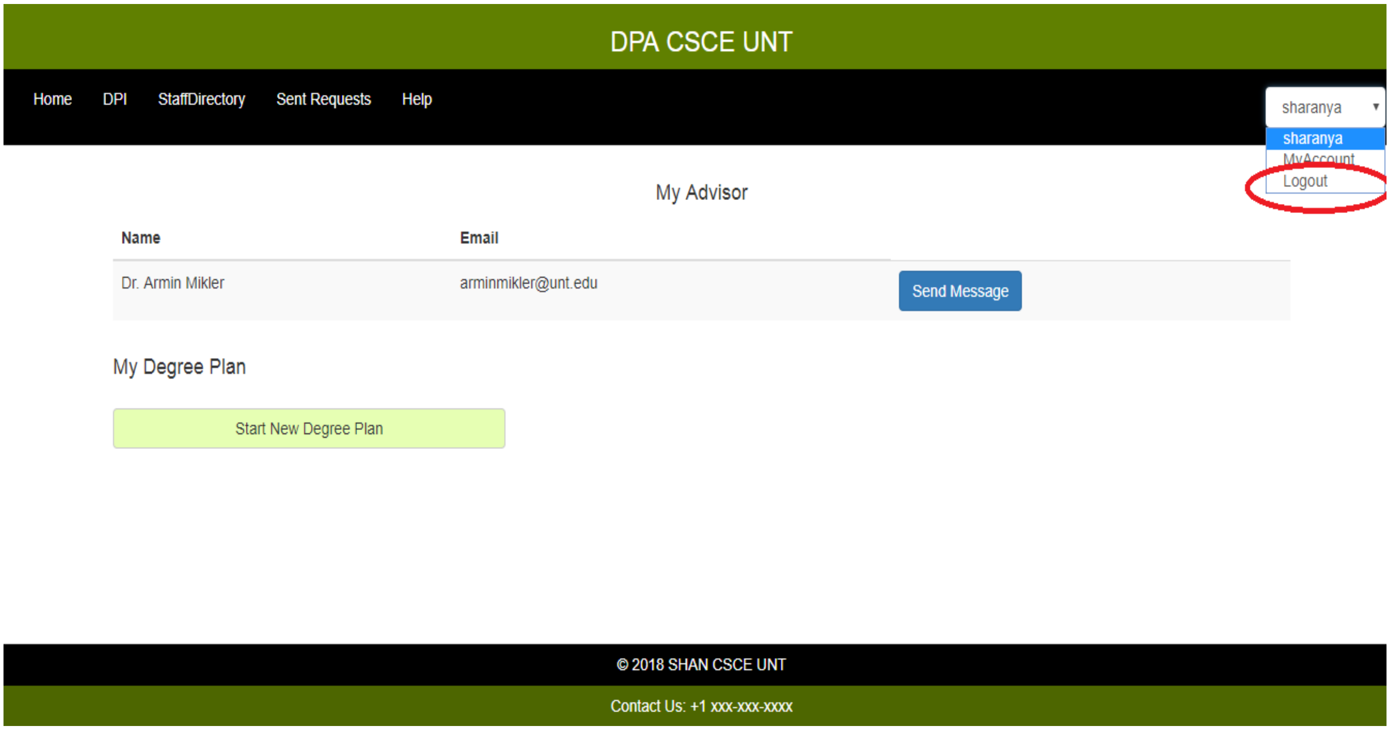
Admission to candidacy is recommended: Total Semester Hours Required: 36

Advisor or Major Professor: nanditha

Associate Chair for Graduate Studies: hasrhini

Department Chair: aravnd

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



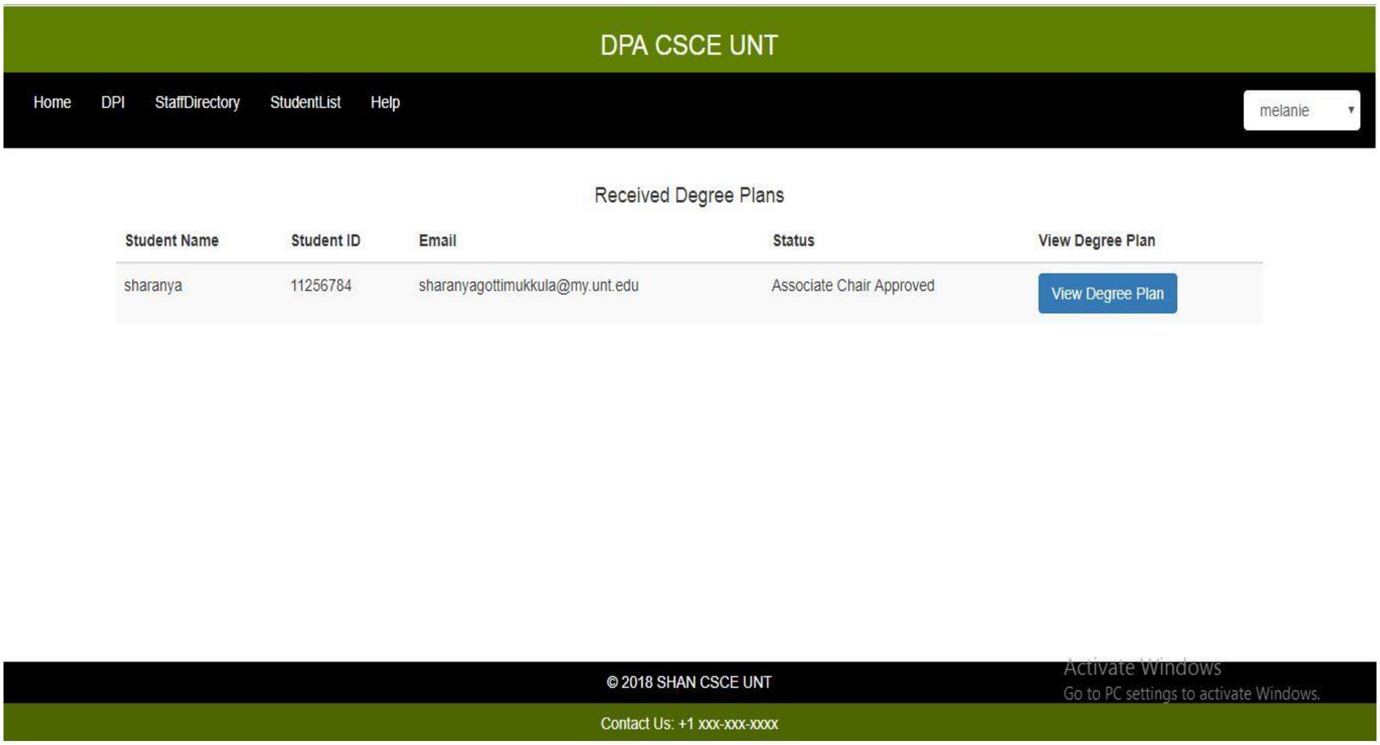
My Advisor

Name	Email
Dr. Armin Mikler	arminmikler@unt.edu

My Degree Plan

User as Admin Specialist:

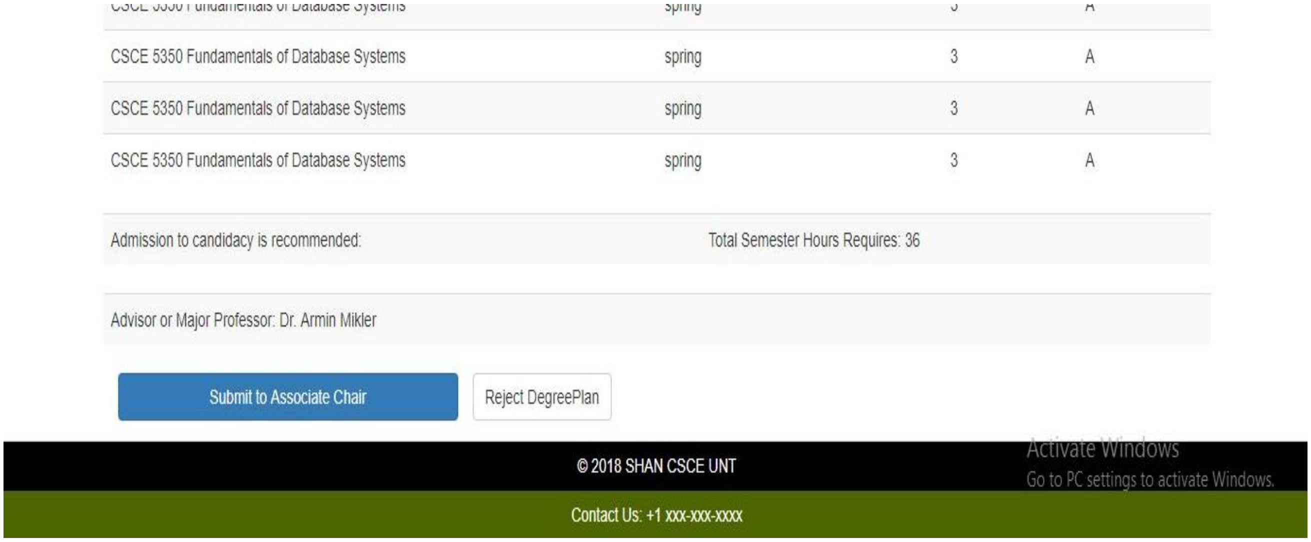
If the user logs in with role as a admin specialist the user will be redirected to the admin specialist home. Admin specialist home displays the degree plans received from the professors.



Received Degree Plans

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

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.

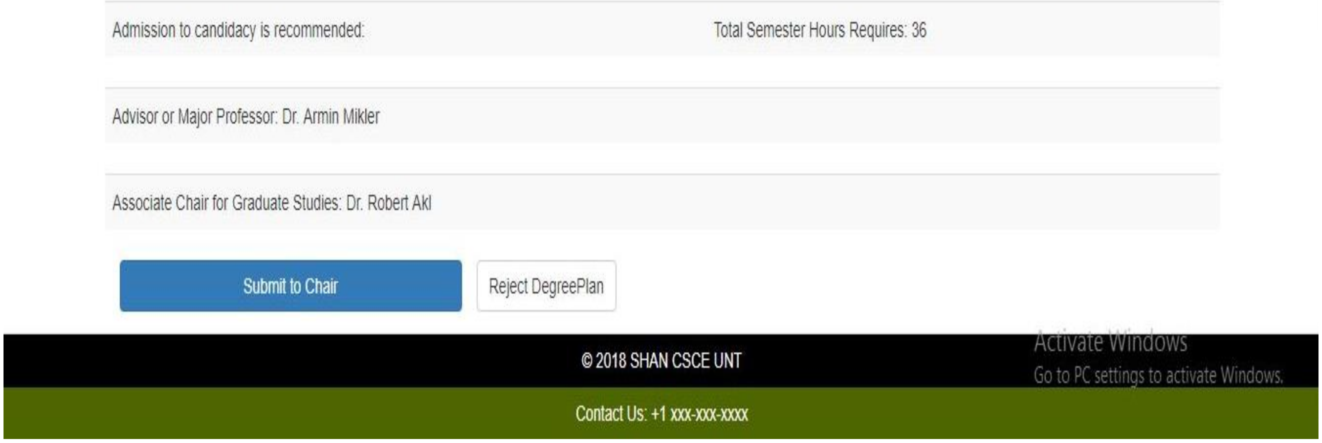


CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A
CSCE 5350 Fundamentals of Database Systems	spring	3	A

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

Advisor or Major Professor: Dr. Armin Mikler

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

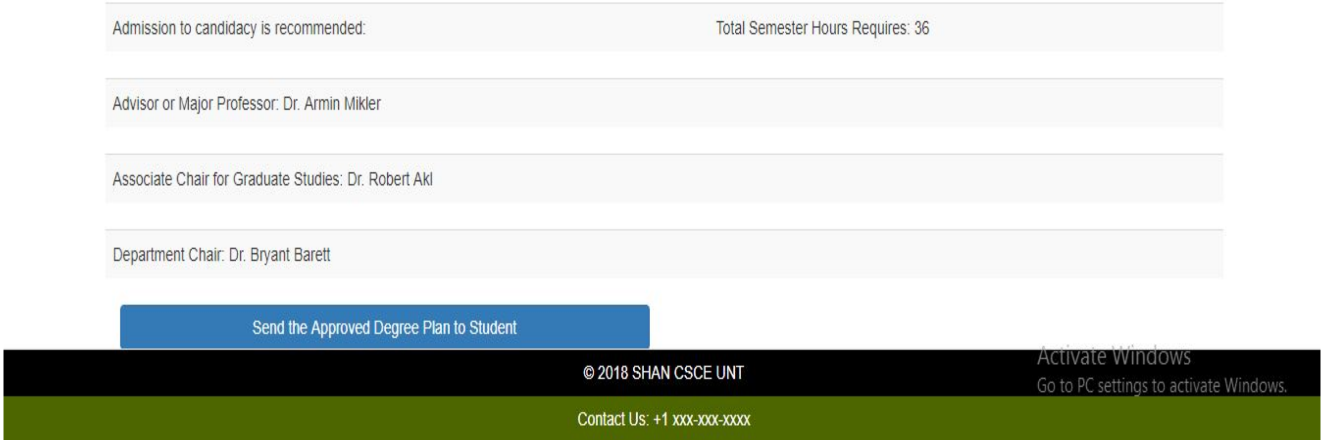


Admission to candidacy is recommended: Total Semester Hours Requires: 36

Advisor or Major Professor: Dr. Armin Mikler

Associate Chair for Graduate Studies: Dr. Robert Akl

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.



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

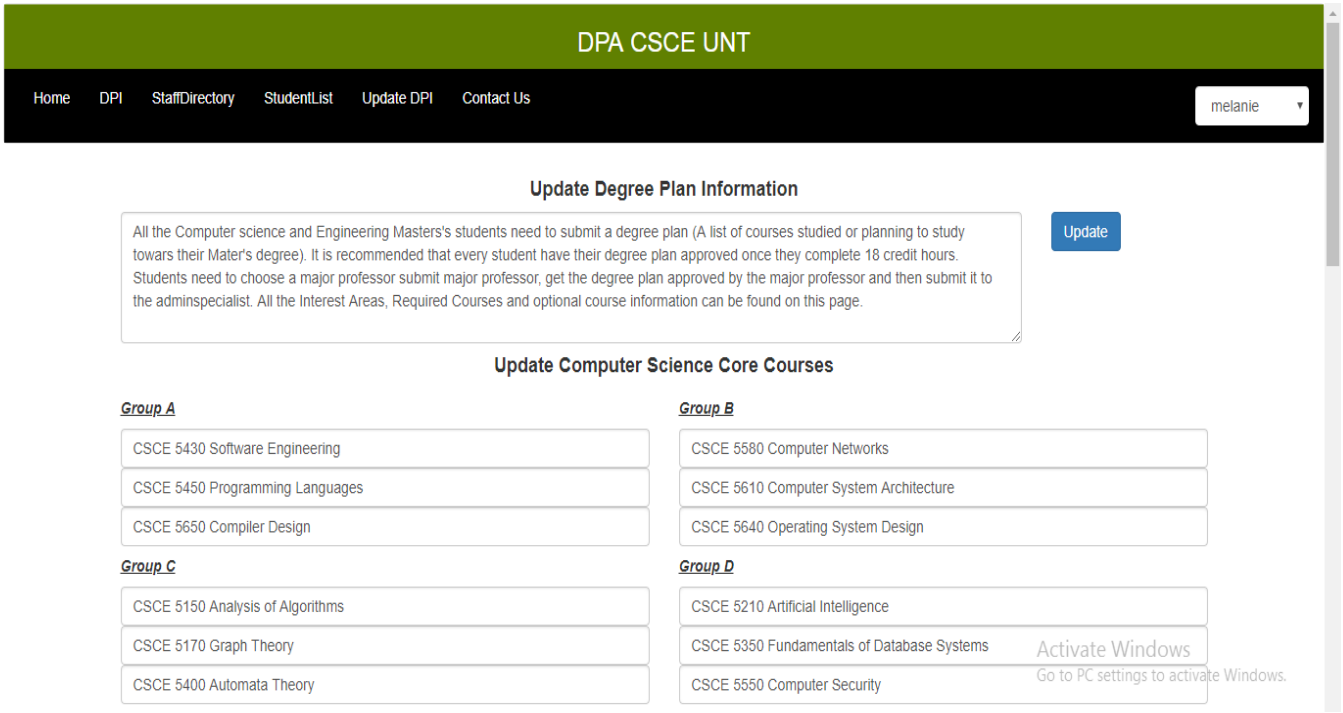
Advisor or Major Professor: Dr. Armin Mikler

Associate Chair for Graduate Studies: Dr. Robert Akl

Department Chair: Dr. Bryant Barett

Send the Approved Degree Plan to Student
© 2018 SHAN CSCE UNT
Contact Us: +1 xxx-xxx-xxxx
Activate Windows
Go to PC settings to activate Windows.

You can update the degree plan information by clicking on the “updateDPI” link present in the menu bar



Update Degree Plan Information

All the Computer science and Engineering Masters's students need to submit a degree plan (A list of courses studied or planning to study towars their Mater's degree). It is recommended that every student have their degree plan approved once they complete 18 credit hours. Students need to choose a major professor submit major professor, get the degree plan approved by the major professor and then submit it to the adminspecialist. All the Interest Areas, Required Courses and optional course information can be found on this page.

Update Computer Science Core Courses

Group A

CSCE 5430 Software Engineering
CSCE 5450 Programming Languages
CSCE 5650 Compiler Design

Group B

CSCE 5580 Computer Networks
CSCE 5610 Computer System Architecture
CSCE 5640 Operating System Design

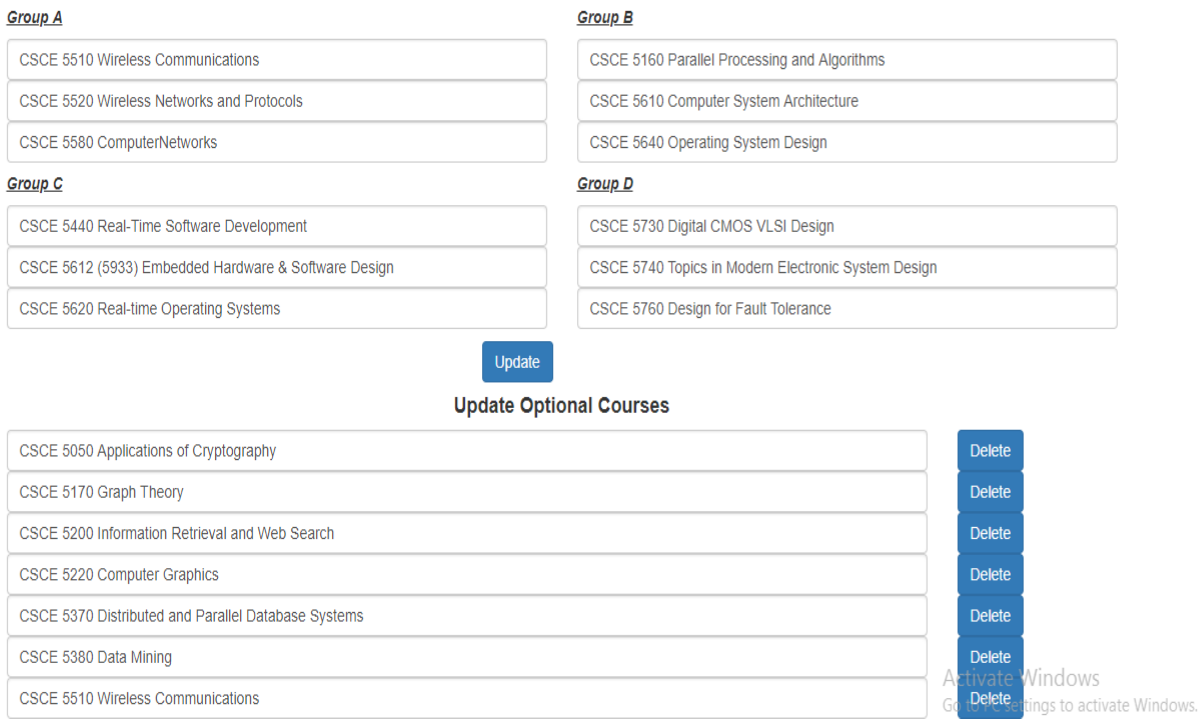
Group C

CSCE 5150 Analysis of Algorithms
CSCE 5170 Graph Theory
CSCE 5400 Automata Theory

Group D

CSCE 5210 Artificial Intelligence
CSCE 5350 Fundamentals of Database Systems
CSCE 5550 Computer Security

You can update the degree plan information, core courses and add or delete optional courses



Group A

CSCE 5510 Wireless Communications
CSCE 5520 Wireless Networks and Protocols
CSCE 5580 ComputerNetworks

Group B

CSCE 5160 Parallel Processing and Algorithms
CSCE 5610 Computer System Architecture
CSCE 5640 Operating System Design

Group C

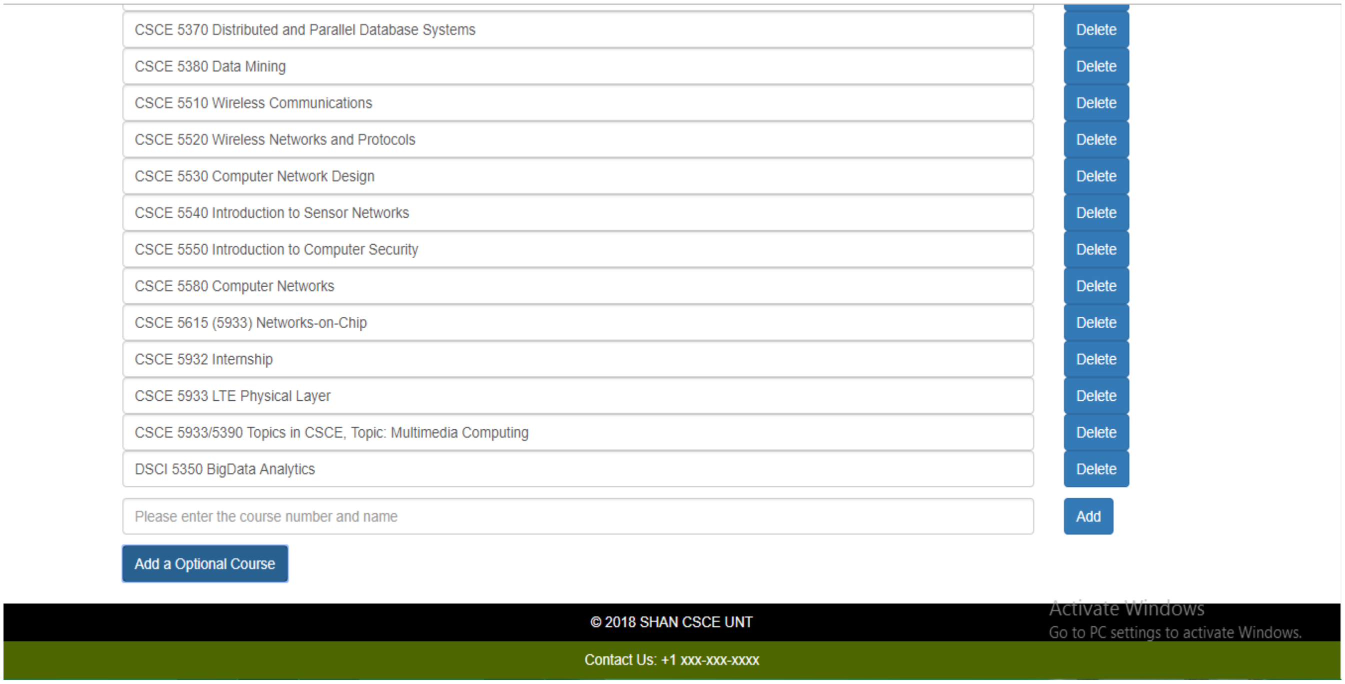
CSCE 5440 Real-Time Software Development
CSCE 5612 (5933) Embedded Hardware & Software Design
CSCE 5620 Real-time Operating Systems

Group D

CSCE 5730 Digital CMOS VLSI Design
CSCE 5740 Topics in Modern Electronic System Design
CSCE 5760 Design for Fault Tolerance

Update Optional Courses

CSCE 5050 Applications of Cryptography
CSCE 5170 Graph Theory
CSCE 5200 Information Retrieval and Web Search
CSCE 5220 Computer Graphics
CSCE 5370 Distributed and Parallel Database Systems
CSCE 5380 Data Mining
CSCE 5510 Wireless Communications


CSCE 5370 Distributed and Parallel Database Systems

CSCE 5380 Data Mining

CSCE 5510 Wireless Communications

CSCE 5520 Wireless Networks and Protocols

CSCE 5530 Computer Network Design

CSCE 5540 Introduction to Sensor Networks

CSCE 5550 Introduction to Computer Security

CSCE 5580 Computer Networks

CSCE 5615 (5933) Networks-on-Chip

CSCE 5932 Internship

CSCE 5933 LTE Physical Layer

CSCE 5933/5390 Topics in CSCE, Topic: Multimedia Computing

DSCI 5350 BigData Analytics

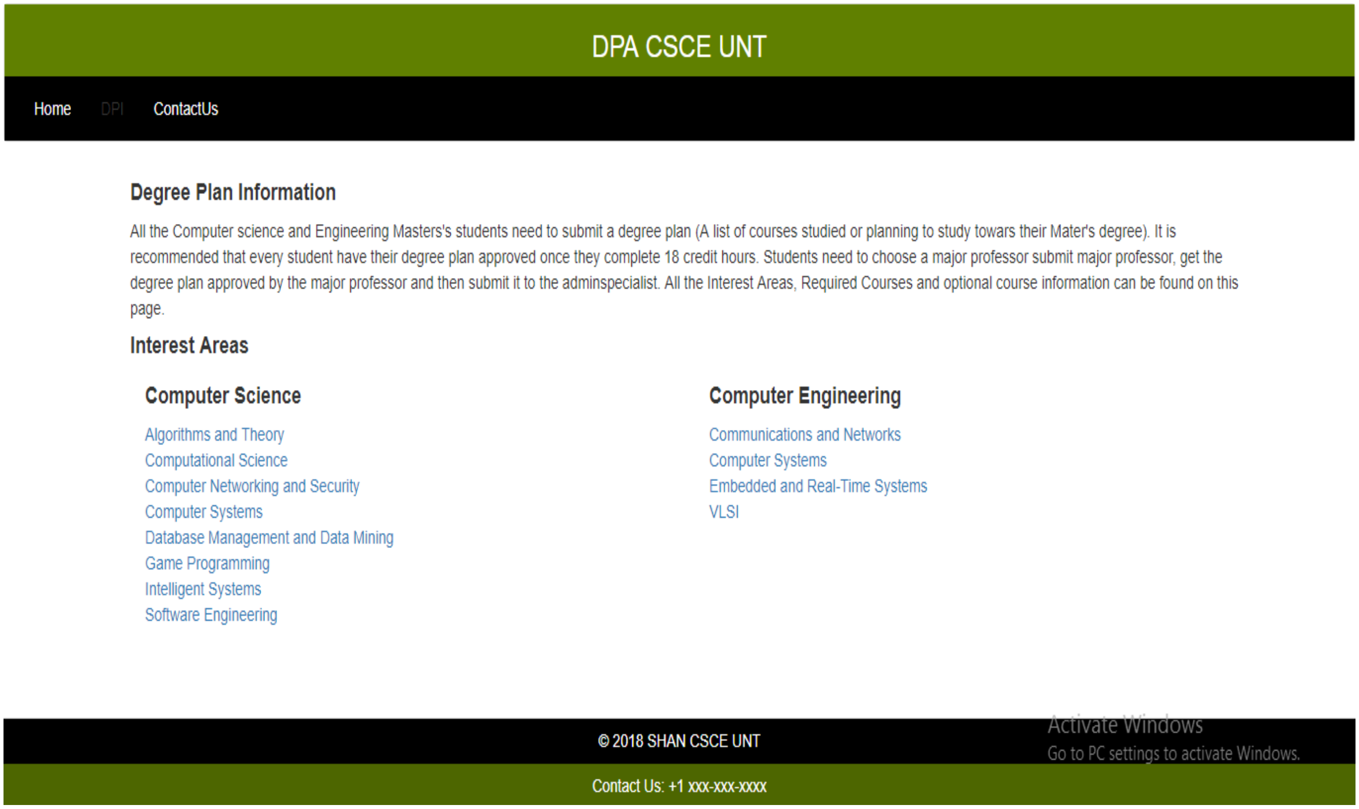
Please enter the course number and name

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

Common to All the Users:

All the users can view the degree plan information by clicking on the DPI link present in the menu bar


DPA CSCE UNT

Degree Plan Information

All the Computer science and Engineering Masters's students need to submit a degree plan (A list of courses studied or planning to study towars their Mater's degree). It is recommended that every student have their degree plan approved once they complete 18 credit hours. Students need to choose a major professor submit major professor, get the degree plan approved by the major professor and then submit it to the adminspecialist. All the Interest Areas, Required Courses and optional course information can be found on this page.

Interest Areas

Computer Science

- Algorithms and Theory
- Computational Science
- Computer Networking and Security
- Computer Systems
- Database Management and Data Mining
- Game Programming
- Intelligent Systems
- Software Engineering

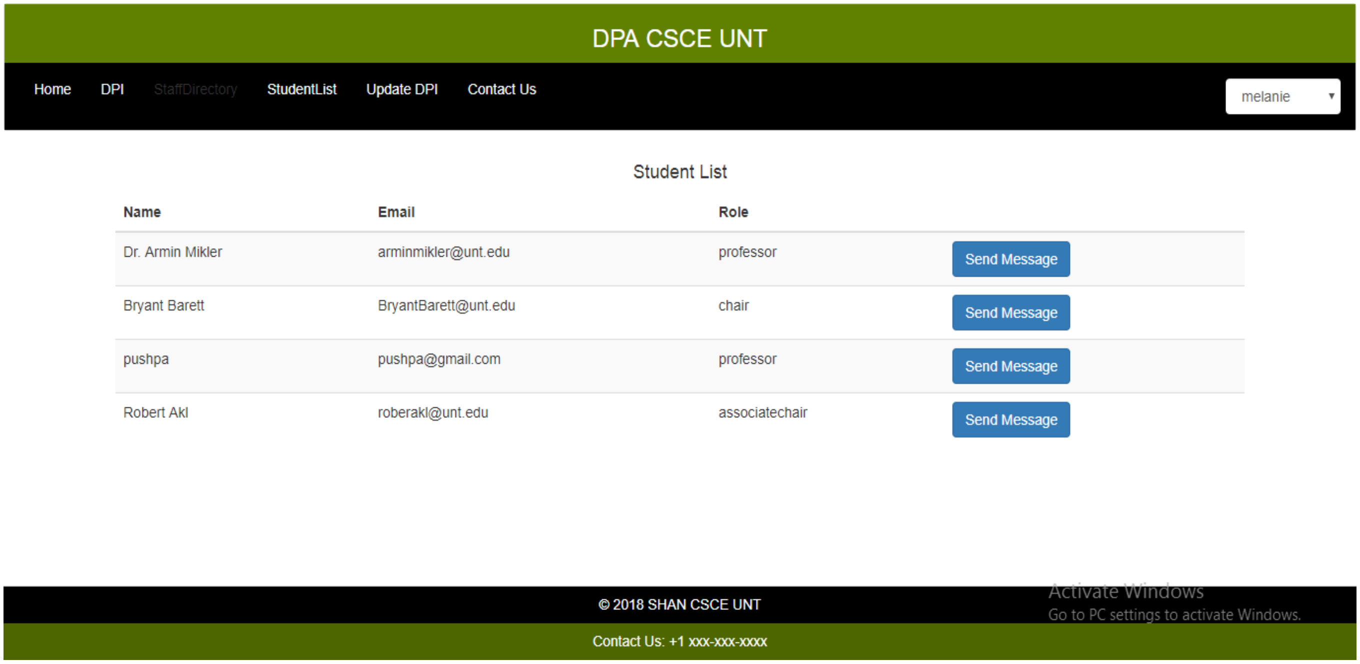
Computer Engineering

- Communications and Networks
- Computer Systems
- Embedded and Real-Time Systems
- VLSI

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

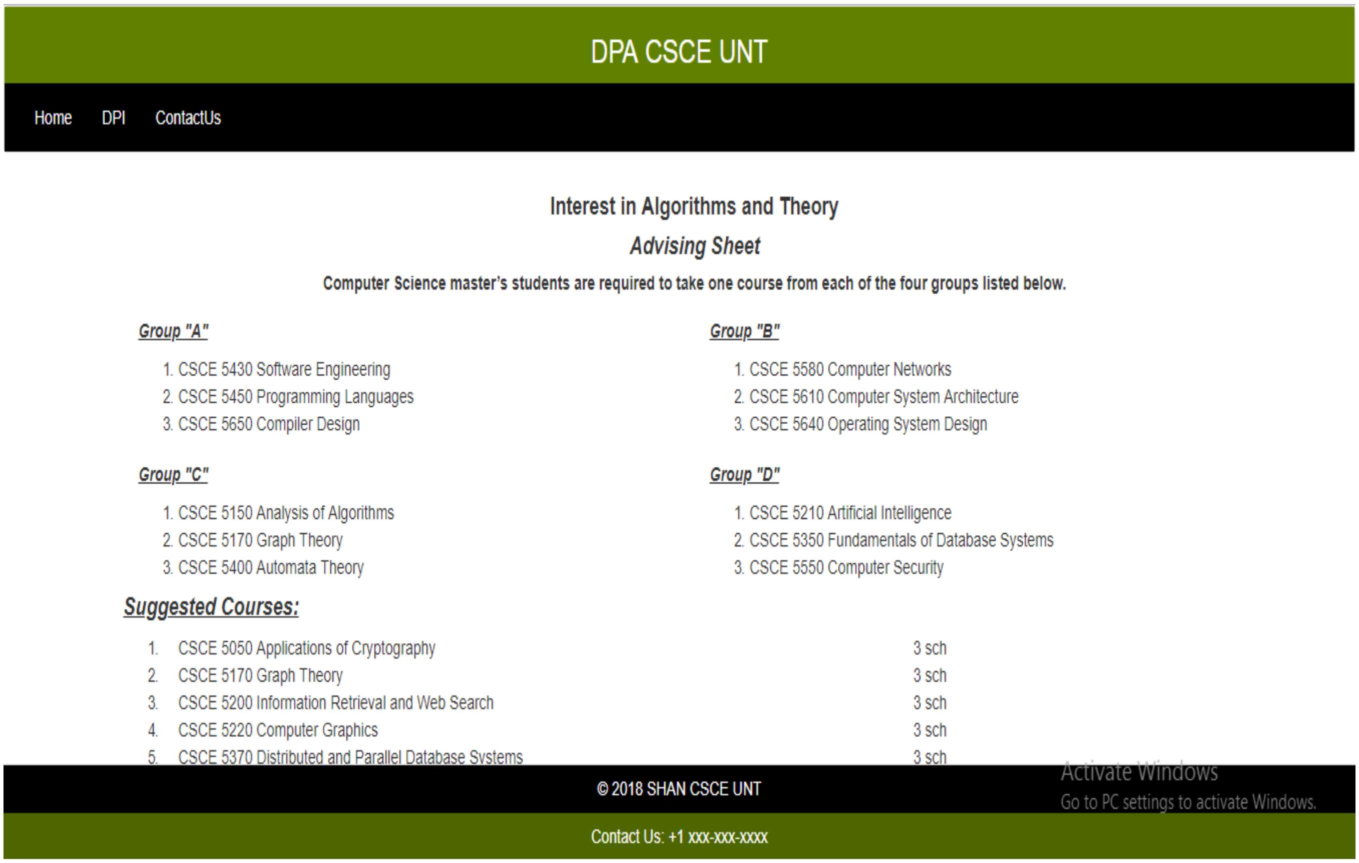
You can view staff list by clicking the “staffdirectory” link in the menu bar



Student List

Name	Email	Role	
Dr. Armin Mikler	arminmikler@unt.edu	professor	Send Message
Bryant Barett	BryantBarett@unt.edu	chair	Send Message
pushpa	pushpa@gmail.com	professor	Send Message
Robert Akl	roberakl@unt.edu	associatechair	Send Message

Users can click on a specific interest area to view requirements of that particular interest area


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

Interest in Algorithms and Theory

Advising Sheet

Computer Science master's students are required to take one course from each of the four groups listed below.

Group "A"

1. CSCE 5430 Software Engineering
2. CSCE 5450 Programming Languages
3. CSCE 5650 Compiler Design

Group "B"

1. CSCE 5580 Computer Networks
2. CSCE 5610 Computer System Architecture
3. CSCE 5640 Operating System Design

Group "C"

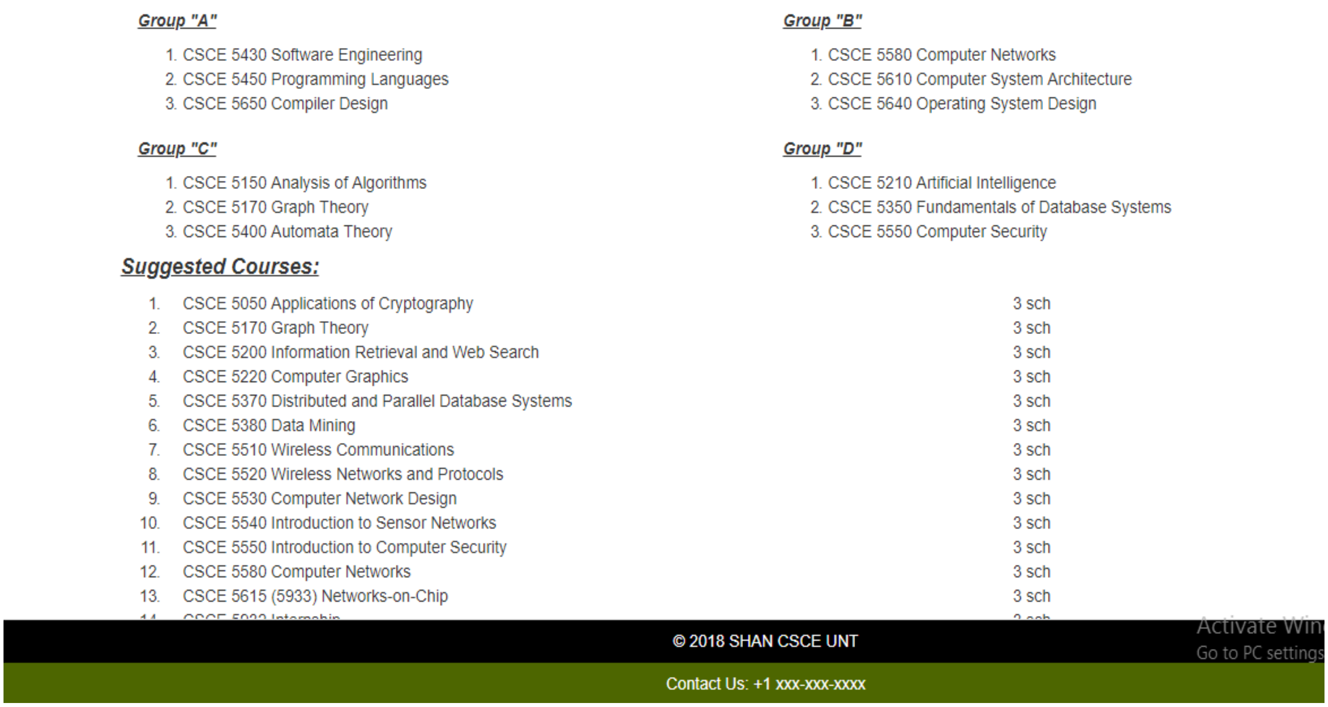
1. CSCE 5150 Analysis of Algorithms
2. CSCE 5170 Graph Theory
3. CSCE 5400 Automata Theory

Group "D"

1. CSCE 5210 Artificial Intelligence
2. CSCE 5350 Fundamentals of Database Systems
3. CSCE 5550 Computer Security

Suggested Courses:

1.	CSCE 5050 Applications of Cryptography	3 sch
2.	CSCE 5170 Graph Theory	3 sch
3.	CSCE 5200 Information Retrieval and Web Search	3 sch
4.	CSCE 5220 Computer Graphics	3 sch
5.	CSCE 5370 Distributed and Parallel Database Systems	3 sch



Group "A"

1. CSCE 5430 Software Engineering
2. CSCE 5450 Programming Languages
3. CSCE 5650 Compiler Design

Group "B"

1. CSCE 5580 Computer Networks
2. CSCE 5610 Computer System Architecture
3. CSCE 5640 Operating System Design

Group "C"

1. CSCE 5150 Analysis of Algorithms
2. CSCE 5170 Graph Theory
3. CSCE 5400 Automata Theory

Group "D"

1. CSCE 5210 Artificial Intelligence
2. CSCE 5350 Fundamentals of Database Systems
3. CSCE 5550 Computer Security

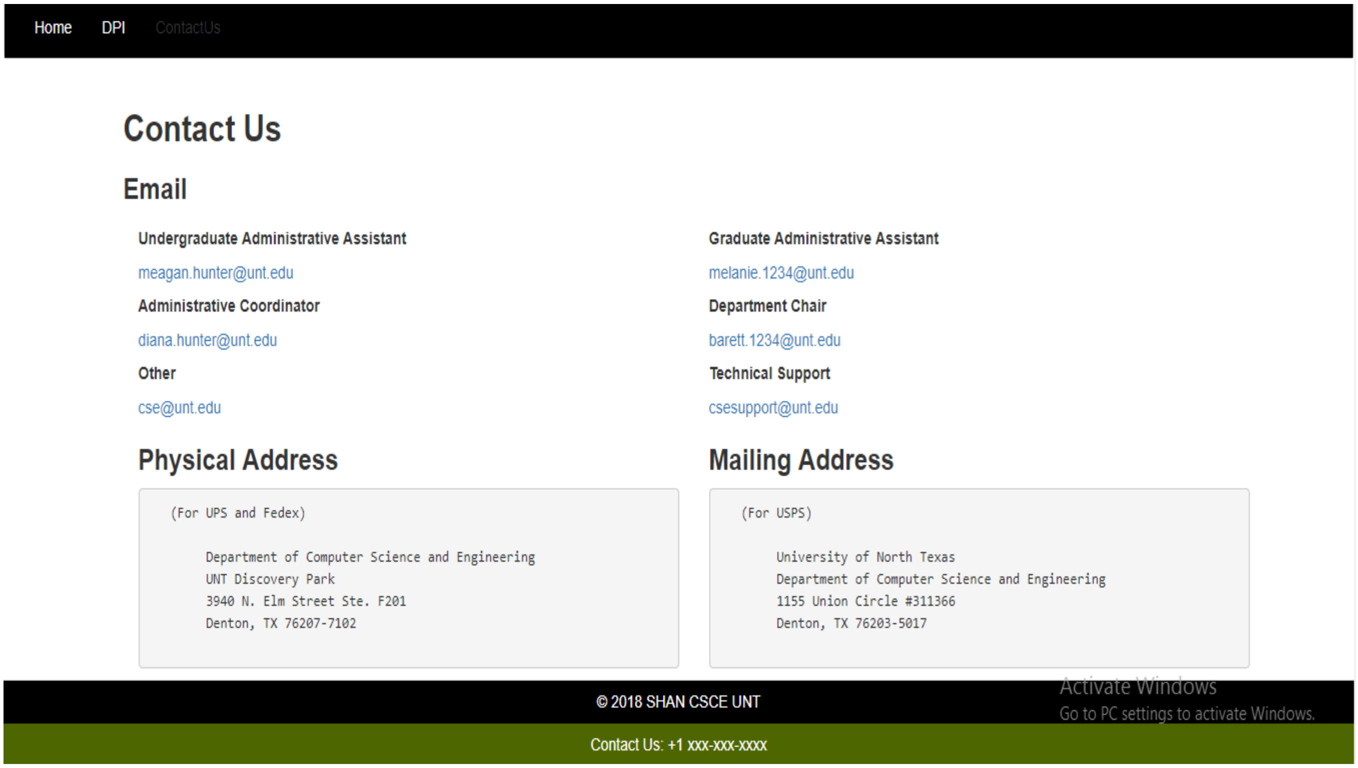
Suggested Courses:

1.	CSCE 5050 Applications of Cryptography	3 sch
2.	CSCE 5170 Graph Theory	3 sch
3.	CSCE 5200 Information Retrieval and Web Search	3 sch
4.	CSCE 5220 Computer Graphics	3 sch
5.	CSCE 5370 Distributed and Parallel Database Systems	3 sch
6.	CSCE 5380 Data Mining	3 sch
7.	CSCE 5510 Wireless Communications	3 sch
8.	CSCE 5520 Wireless Networks and Protocols	3 sch
9.	CSCE 5530 Computer Network Design	3 sch
10.	CSCE 5540 Introduction to Sensor Networks	3 sch
11.	CSCE 5550 Introduction to Computer Security	3 sch
12.	CSCE 5580 Computer Networks	3 sch
13.	CSCE 5615 (5933) Networks-on-Chip	3 sch
14.	CSCE 5933 Internship	3 sch

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

All the users can view the contact information of the CSCE department by clicking on the Contact Us link present in the menu bar.



Contact Us

Email

Undergraduate Administrative Assistant

meagan.hunter@unt.edu

Administrative Coordinator

diana.hunter@unt.edu

Other

cse@unt.edu

Graduate Administrative Assistant

melanie.1234@unt.edu

Department Chair

barett.1234@unt.edu

Technical Support

csesupport@unt.edu

Physical Address

```
(For UPS and Fedex)

     Department of Computer Science and Engineering
     UNT Discovery Park
     3940 N. Elm Street Ste. F201
     Denton, TX 76207-7102
```

Mailing Address

```
(For USPS)

     University of North Texas
     Department of Computer Science and Engineering
     1155 Union Circle #311366
     Denton, TX 76203-5017
```

© 2018 SHAN CSCE UNT

Contact Us: +1 xxx-xxx-xxxx

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

Software Required:

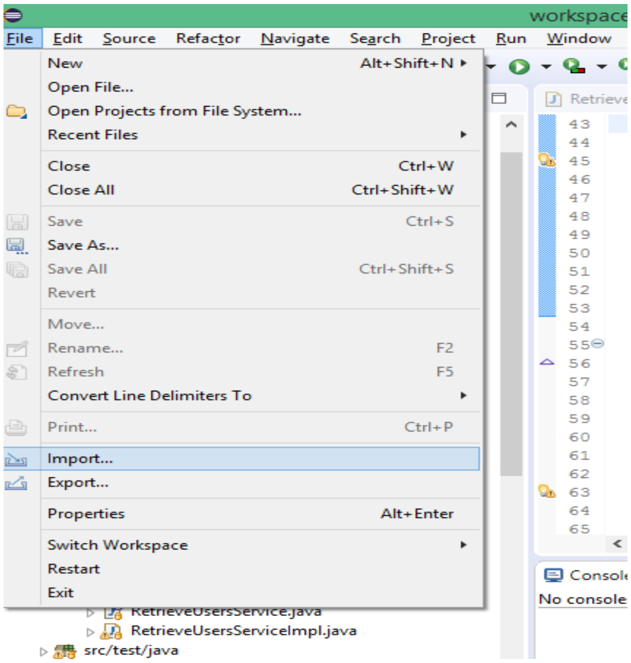
Eclipse

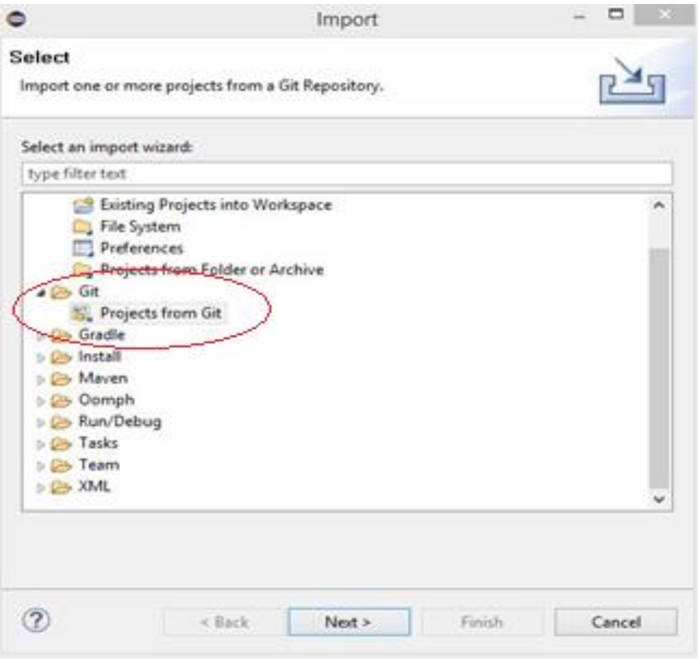
MYSQL workbench and server

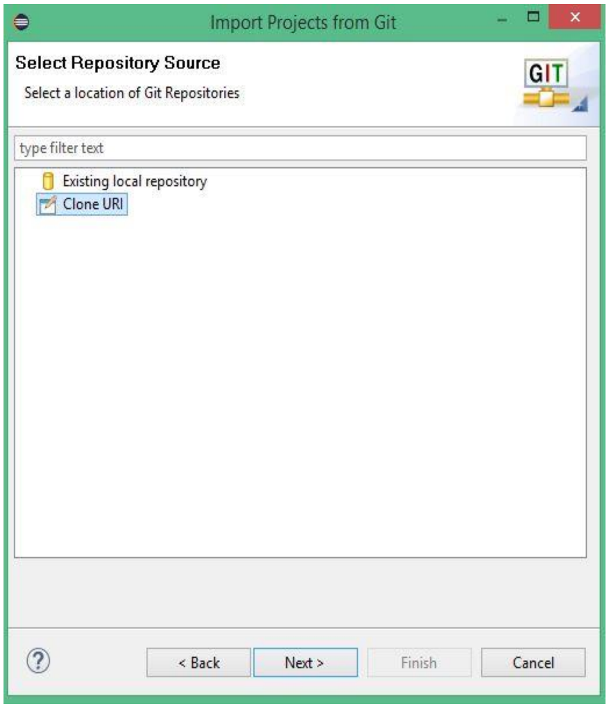
Import the code in to eclipse[3]:

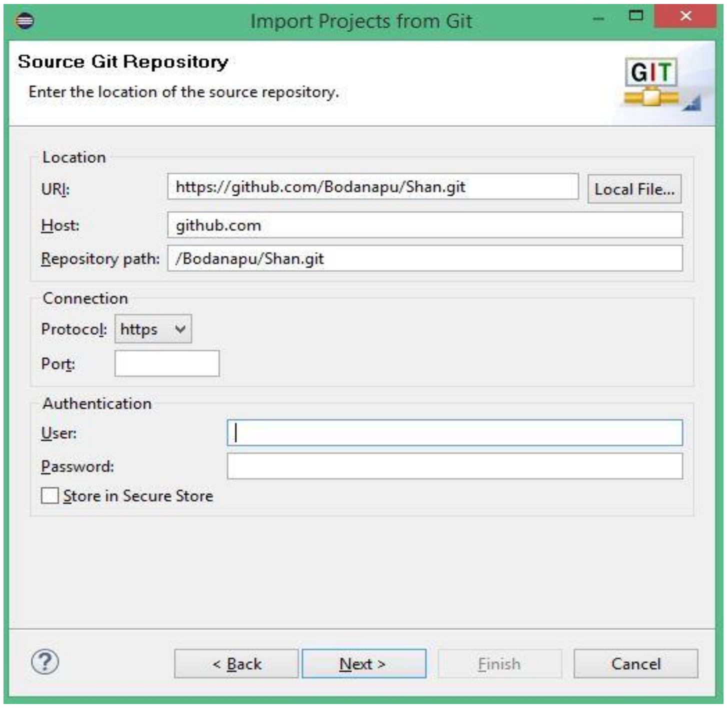
Open the eclipse workspace and import the code from GIT repository by clicking:

File -> import->Git->Projects from Git->clone URI









URI: https://github.com/Bodanapu/Shan.git

Host: github.com

Repository path: /Bodanapu/Shan.git

Protocol: https

Port:

User:

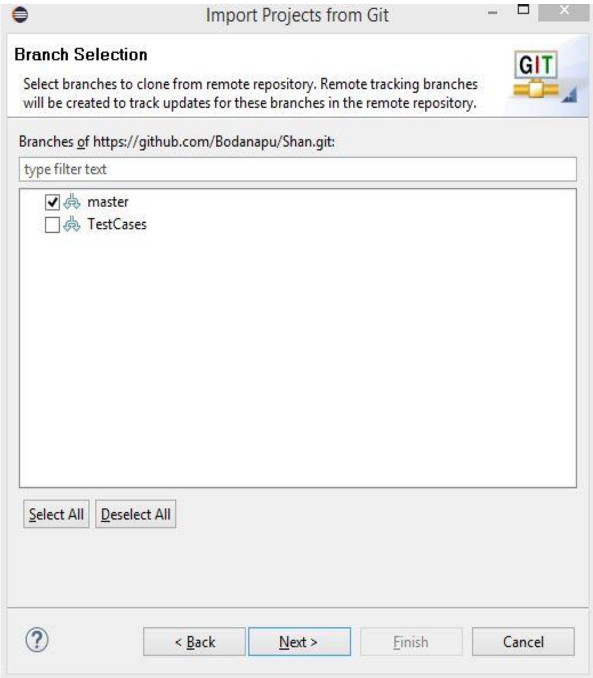
Password:

☐ Store in Secure Store

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 choose a folder to store the project and import it a as import existing eclipse project and then click next, you will see three choices to clone 1.DPA, 2. DPA_Phase2, 3. DPA_Phase3, 4. SHAN choose both option 3, 4 and click fetch. You will now find the DPA_Phase3 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 Projects from Git

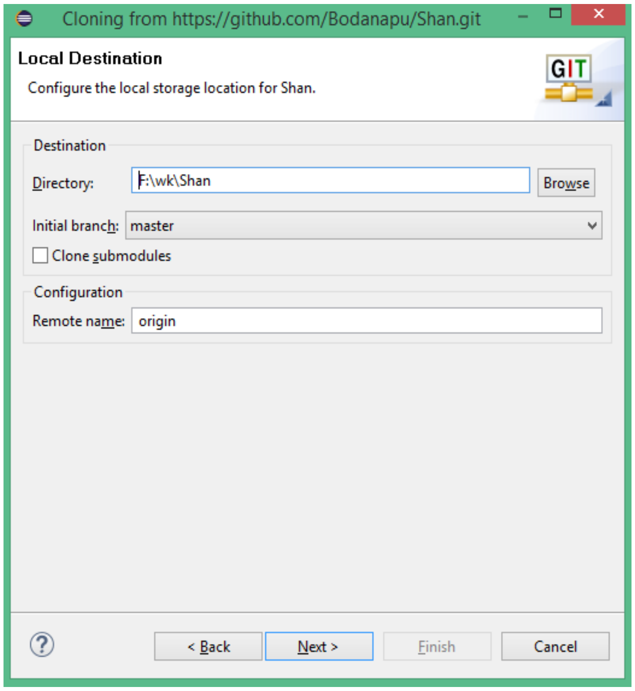
Branch Selection

Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

Branches of https://github.com/Bodanapu/Shan.git:

type filter text

☑ master
☐ TestCases



Cloning from https://github.com/Bodanapu/Shan.git

Local Destination

Configure the local storage location for Shan.

Destination

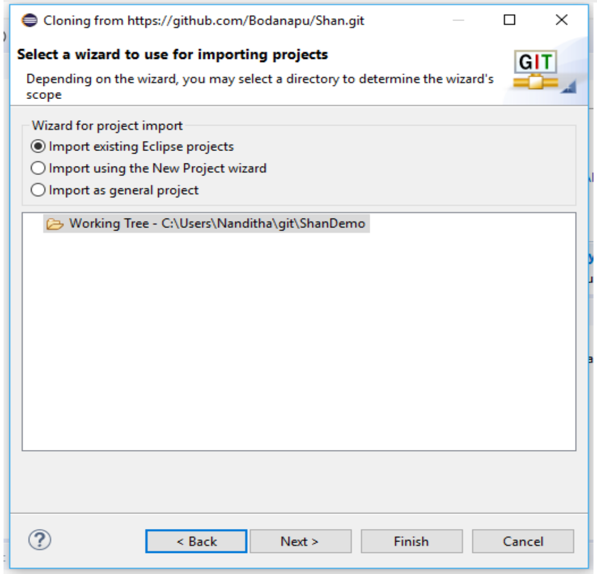
Directory: F:\wk\Shan

Initial branch: master

☐ Clone submodules

Configuration

Remote name: origin



Cloning from https://github.com/Bodanapu/Shan.git

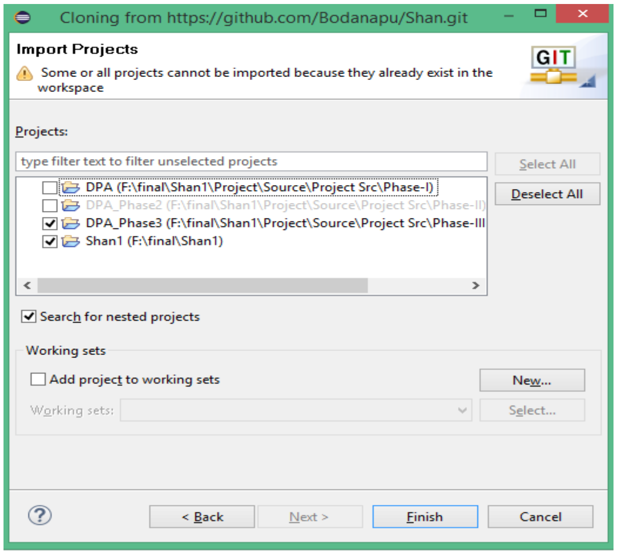
Select a wizard to use for importing projects

Depending on the wizard, you may select a directory to determine the wizard's scope

Wizard for project import

● Import existing Eclipse projects
○ Import using the New Project wizard
○ Import as general project

Working Tree - C:\Users\Nanditha\git\ShanDemo



Cloning from https://github.com/Bodanapu/Shan.git

Import Projects

Some or all projects cannot be imported because they already exist in the workspace

Projects:

type filter text to filter unselected projects

☐ DPA (F:\final\Shan1\Project\Source\Project Src\Phase-I)
☐ DPA_Phase2 (F:\final\Shan1\Project\Source\Project Src\Phase-II)
☑ DPA_Phase3 (F:\final\Shan1\Project\Source\Project Src\Phase-III
☑ Shan1 (F:\final\Shan1)

☑ Search for nested projects

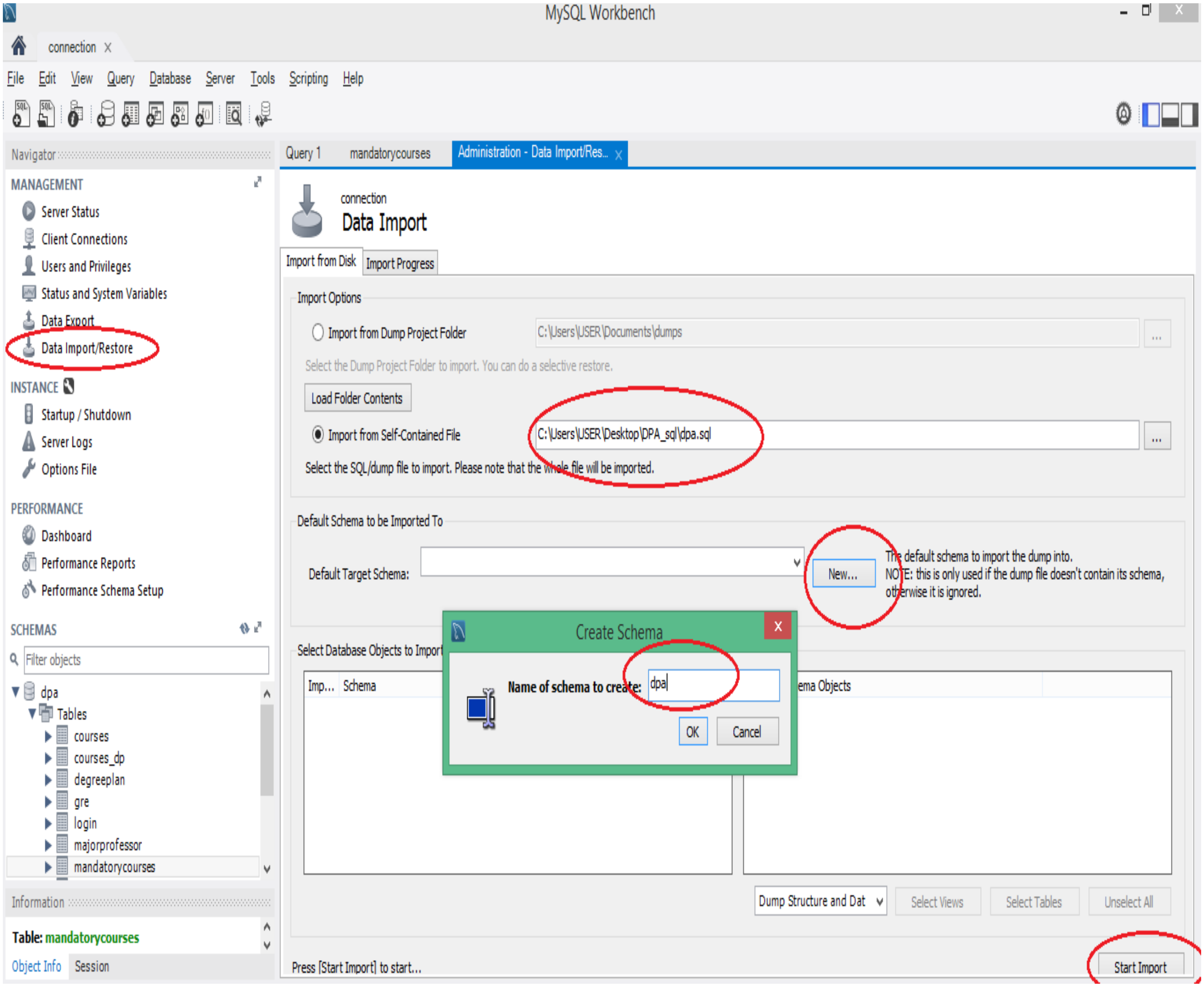
Working sets

☐ Add project to working sets

Working sets:

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.



connection

Data Import

Import Options

○ Import from Dump Project Folder: C:\Users\USER\Documents\dumps

Select the Dump Project Folder to import. You can do a selective restore.

● Import from Self-Contained File: C:\Users\USER\Desktop\DPA_sql\dpa.sql

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema:

The default schema to import the dump into.
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Create Schema

Name of schema to create: dpa

Select Database Objects to Import

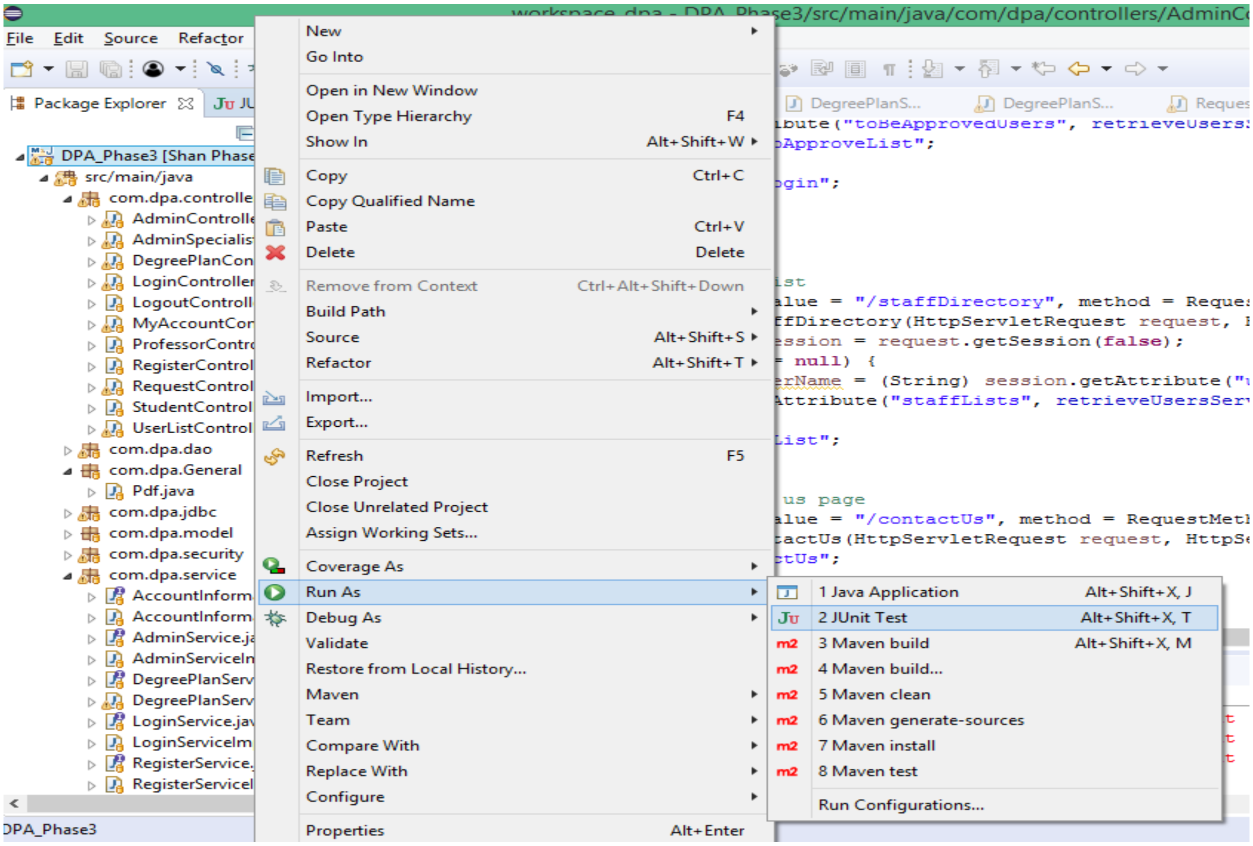
Dump Structure and Dat

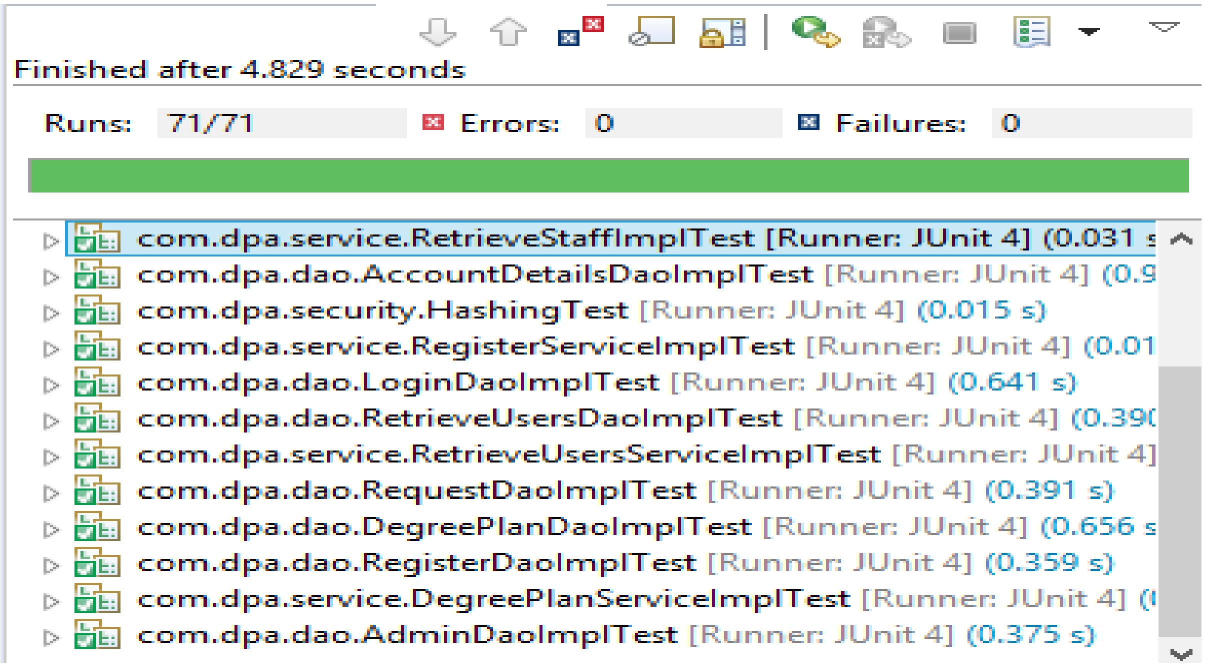
Press [Start Import] to start...

Table: mandatorycourses

Procedure to Run the Test Cases:

Once you find the maven project DPA_Phase3 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 4.829 seconds

Runs: 71/71     Errors: 0     Failures: 0

com.dpa.service.RetrieveStaffImplTest [Runner: JUnit 4] (0.031 s
com.dpa.dao.AccountDetailsDaoImplTest [Runner: JUnit 4] (0.9
com.dpa.security.HashingTest [Runner: JUnit 4] (0.015 s)
com.dpa.service.RegisterServiceImplTest [Runner: JUnit 4] (0.01
com.dpa.dao.LoginDaoImplTest [Runner: JUnit 4] (0.641 s)
com.dpa.dao.RetrieveUsersDaoImplTest [Runner: JUnit 4] (0.39
com.dpa.service.RetrieveUsersServiceImplTest [Runner: JUnit 4]
com.dpa.dao.RequestDaoImplTest [Runner: JUnit 4] (0.391 s)
com.dpa.dao.DegreePlanDaoImplTest [Runner: JUnit 4] (0.656 s
com.dpa.dao.RegisterDaoImplTest [Runner: JUnit 4] (0.359 s)
com.dpa.service.DegreePlanServiceImplTest [Runner: JUnit 4] (
com.dpa.dao.AdminDaoImplTest [Runner: JUnit 4] (0.375 s)
```

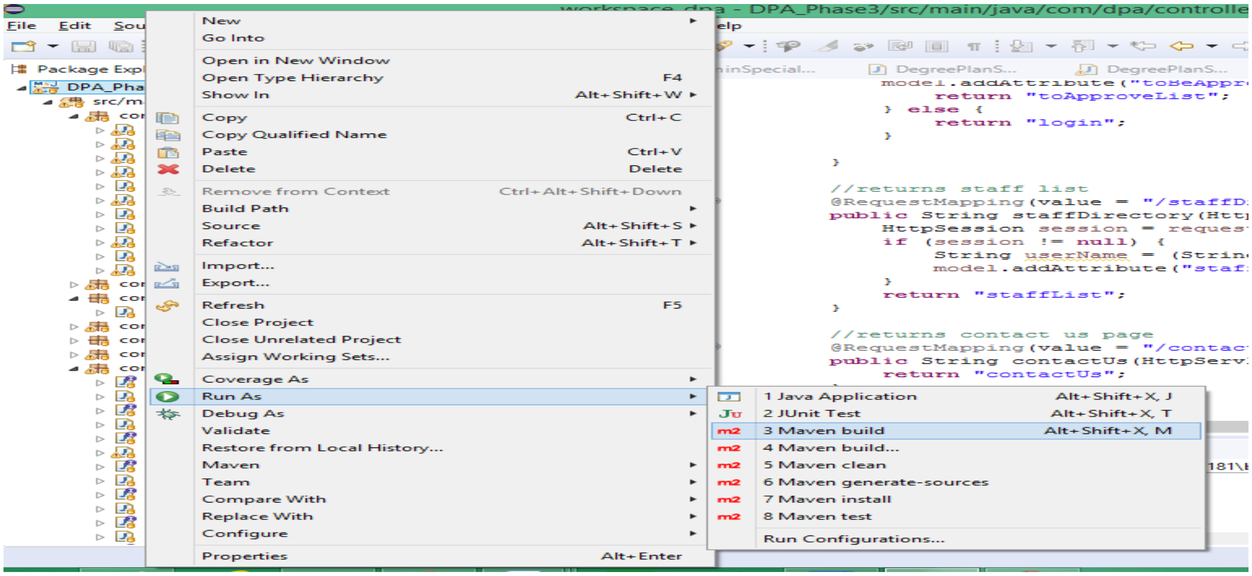
Procedure to Run the Program and check using web browser:

Right click on the maven project DPA_Phase3 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.

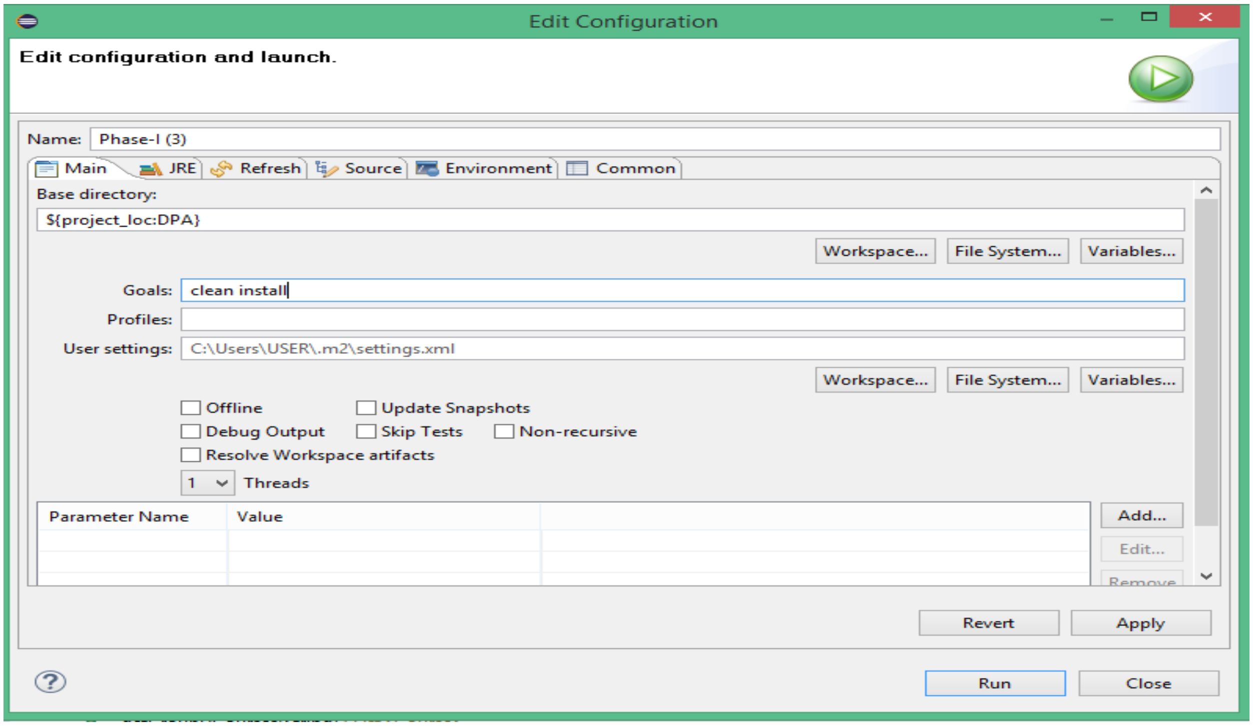
The maven will download and build all the dependencies of the project present in the “pom.xml” file.

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.



```
        model.addAttribute("toBeAppr
        return "toApproveList";
    } else {
        return "login";
    }
}
//returns staff list
@RequestMapping(value = "/staffD
public String staffDirectory(Htt
    HttpSession session = reques
    if (session != null) {
        String userName = (Strin
        model.addAttribute("staf
    }
    return "staffList";
}
//returns contact us page
@RequestMapping(value = "/contac
public String contactUs(HttpServ
    return "contactUs";
```



Edit configuration and launch.

Name: Phase-I (3)

Base directory: ${project_loc:DPA}

Goals: clean install

Profiles:

User settings: C:\Users\USER\.m2\settings.xml

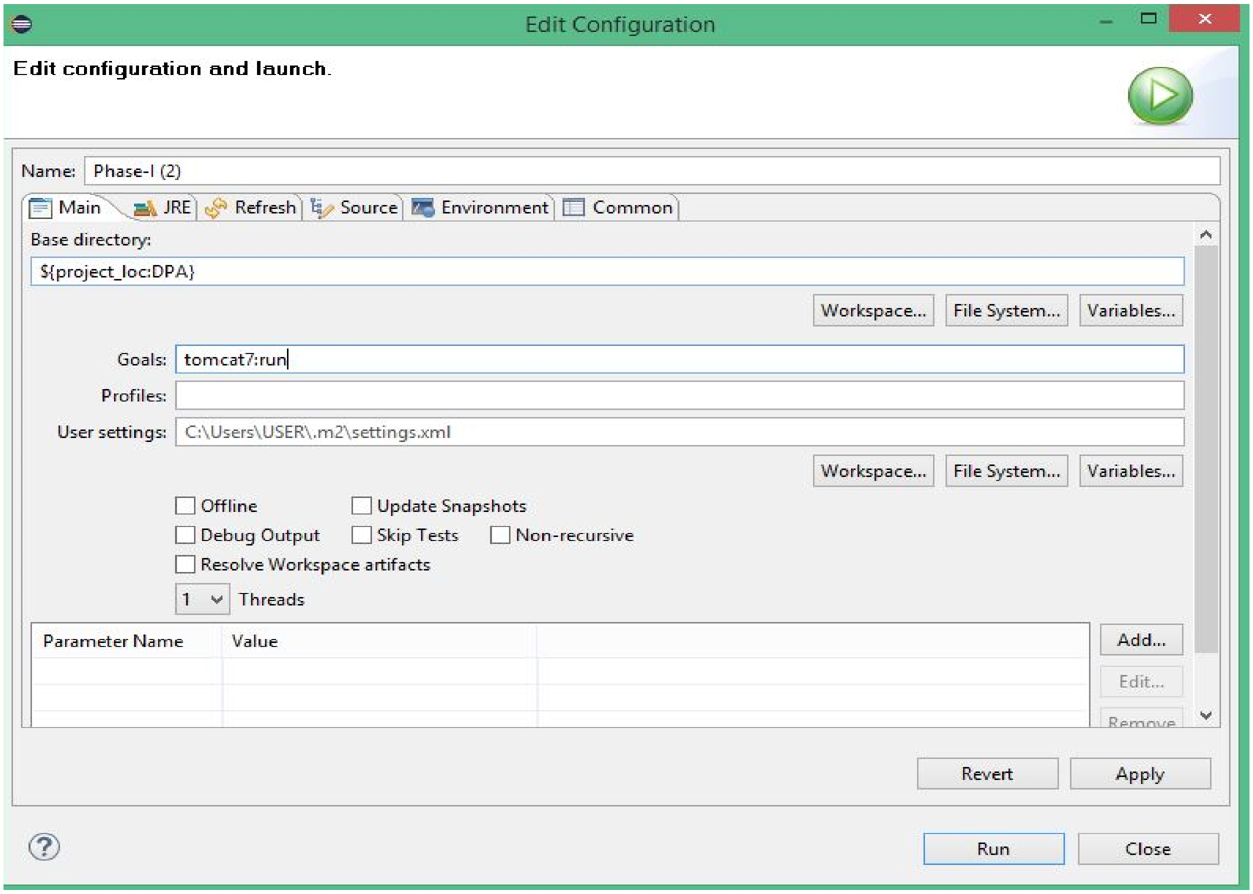
□ Offline □ Update Snapshots

□ Debug Output □ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

1 Threads

Parameter Name	Value



Edit Configuration

Edit configuration and launch.

Name: Phase-I (2)

Base directory:
${project_loc:DPA}

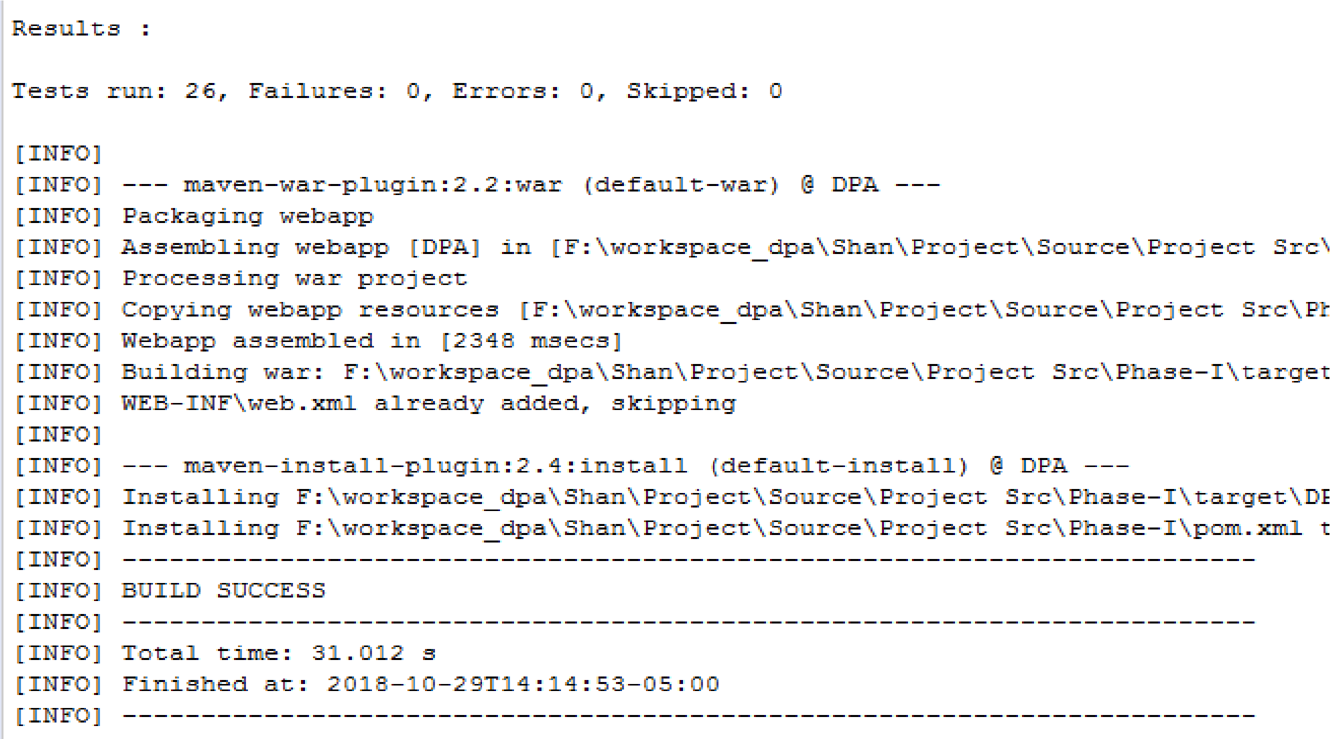
Goals: tomcat7:run

Profiles:

User settings: C:\Users\USER\.m2\settings.xml

□ Offline □ Update Snapshots
□ Debug Output □ Skip Tests □ Non-recursive
□ Resolve Workspace artifacts
1 Threads

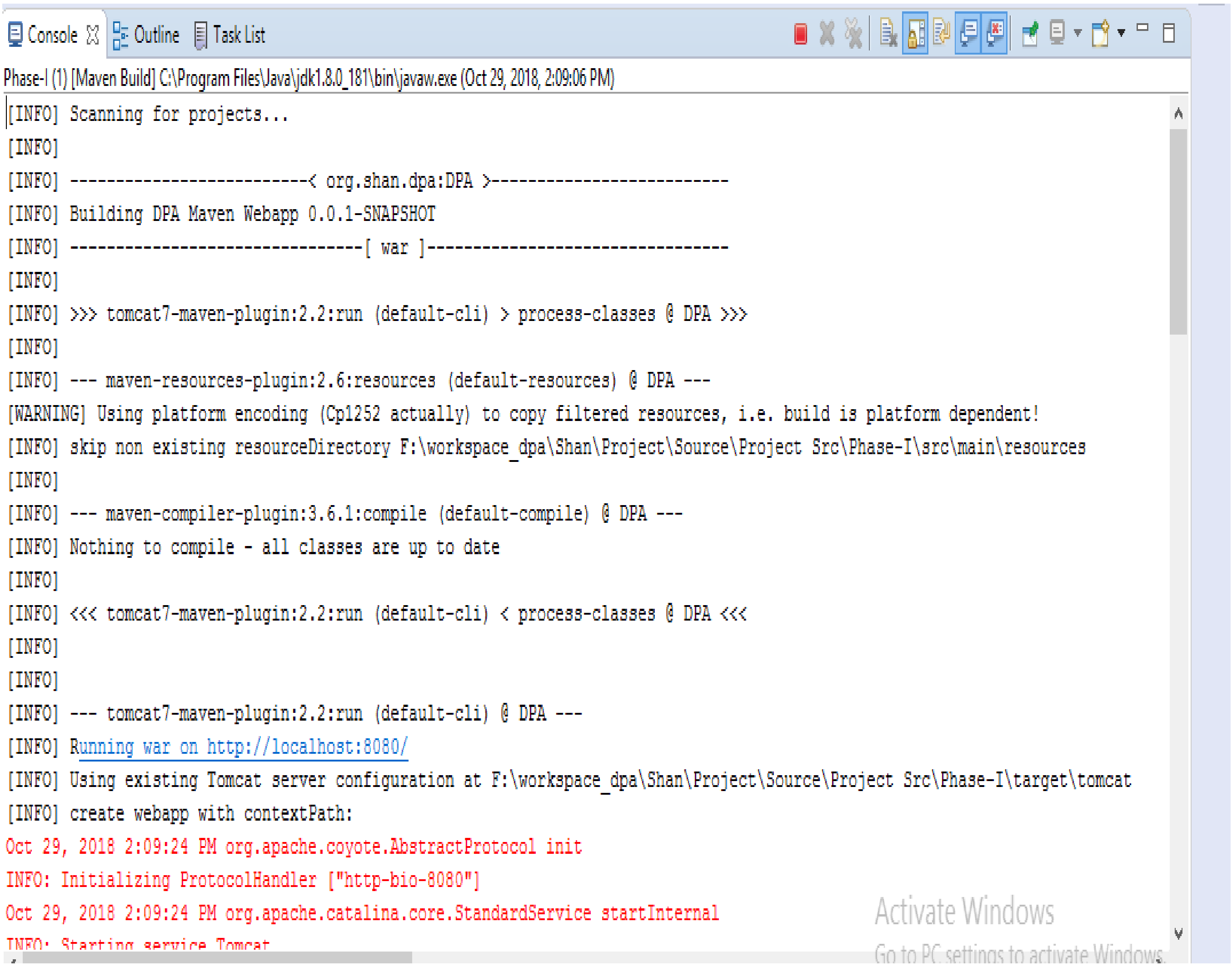
Parameter Name	Value



```
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\Ph
[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] ------------------------------------------------------------------------
```



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
```

g. Feedback received during code inspection session and actions taken

We received the feedback from the Mad Magicians team. They went through the code inspection document thoroughly and raised two questions that were regarding the SQL injection while checking for the login credentials and other is regarding the component name, writer and date written in the description of code. We forgot writing the component name, writer name and date the code is written in the description of component in the code inspection document.
We used prepared statements while checking the login credentials that is we are hashing the password and SQL injection with password is not possible. The only possibility is by using username. We applied primary validations at the front end by not allowing the special characters in the username field and the other is we used prepared statement SQL query which sets the values after the query is read there by making SQL injection not possible.
We have included the component name, writer name and date the code is written in the description of component in the code inspection document we prepared while submitting deliverable-4.

h. Features implemented successfully, limitations and future scope

Features Implemented successfully: We have implemented the following requirements successfully:

FREQ-1	Login
FREQ-2	Registration
FREQ-3	Manage users
FREQ-5	Send Requests to professors and withdraw
FREQ-8	Accept/decline requests (Professor)
FREQ-6	Fill in the Degree Plan (Student)
FREQ-7	Submit the Degree Plan to Professor for Signature (student)
FREQ-9	Approve/Decline Degree Plan
FREQ-11	Manage Degree Plan
FREQ-13	Account Management
FREQ-10	Manage CSCE Department degree plan information
FREQ-12	Send approval notice to the student
FREQ-4	Verify and approve students/professors
FREQ-14	Contact Us
FREQ-15	Notifications

Limitations:

We could not implement the degree plan print option at each of the stage before the approval, which is requirement we received during the peer review session and forgot password option due to time complexity. There are few SQL exceptions left unhandled, these were left unknowingly and are found during the system testing and we are working on it to handle all the SQL exceptions and will complete it in 2 days.

Plan for next phase or future scope:

The features to be added or improvements to be made to the application for the next phase of the project include:
FREQ 16: Forgot Password
FREQ 17: Email service to notify users of information at each stage
FREQ 18: Print Degree Plan option at each step before approval
FREQ 19: Delete user and Update User as a part of Manage users – admin side
FREQ 22: Improve the UI of the website

FREQ 20: Help (to guide users about application usage and message option to approach with doubts)
FREQ 21: Update the Contact Us information.

i. Member Contribution Table

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Sharanya Gottimukkula	FREQ-5, FPREQ-8, FREQ-11, FREQ-10 Designed HTML pages for showing the professor list, student list, sent requests, received requests and accepted requests, verify the student’s degree plan and send it to chair and associate chair. Involved in writing AdminSpecialistController, RequestController, DegreePlanServiceImpl, DegreePlanDaoImpl, RequestServiceImpl and RequestDaoImpl classes. Wrote user manual, a section describing features successfully implemented, limitations and future scope parts of the report.	25%	
Nanditha Bodanapu	FREQ-6, FREQ-7, FREQ-13, FREQ-12 Designed HTML degree plan form. Created Database tables required to store the degree plan related data. Involved in writing updateAccountInformation() method in all the user controllers. Also involved in writing DegrePlanController, DegreePlanServiceImpl and DegreePlanDaoImpl classes which handles all the degree plan between HTML and database. Wrote Test cases, Feedback during	25%	

	code inspection sessions of report		
Aravind Swamy Thottempudi	FREQ-1, FREQ-2, FREQ-11, FREQ-4 Designed HTML pages for login and registration. Involved in writing LoginController, RegisterController, RegisterServiceImpl and LoginServiceImpl classes. Wrote requirements and did UML diagrams part of report	25%	
Sri Harshini Vallabhaneni	FREQ-3, FREQ-9, FREQ-14, FREQ-15 Designed HTML page for administrator and his functionality manage users. Involved in writing AdminController, Student Controller, RetrieveStaffImpl and RetrieveUsersService classes Involved in writing the approveDegreePlan() and rejectDegreePlan() methods in the. Wrote instructions to compile and run code and tests, member contribution table part of report	25%	

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/