Academic Project Management System

By

Akanksha Khanna (ID 102001)
Harsh Vyas (ID 102132)

A Project submitted in

Partial fulfilment of the requirements

For the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER ENGINEERING

Internal Guide
Prof. Malay S. Bhatt
Associate Professor
Dept. of Comp. Engg. DDU

External Guide
Mr.Hemant Joshi
Scientist -SE
Institute for Plasma Research

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2014

CERTIFICATE

This is to certify that the project work titled

Academic Project Management System

is the bonafide work of

Akanksha Khanna (ID 102001)
and
Harsh Vyas (ID 102132)

carried out in the partial fulfilment of the degree of
Bachelor of Technology
in Computer Engineering
at
Dharmsinh Desai University
in the academic session
December 2013 to April 2014.

Prof. Malay Bhatt
Associate Professor
Dept. of Computer Engg.

Prof. C.K.Bhensdadia
Head
Dept. of Computer Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2014

Acknowledgement

Every project big or small is successful largely due to the effort of a number of people who have always given their valuable advice or lent a helping hand. We sincerely appreciate the inspiration; support and guidance of all those people who have been instrumental in making this project a success.

We are extremely grateful to “Institute for Plasma Research” for the confidence bestowed in us. We would like to express our sincere thanks to **Mr.Hemant Joshi** (External guide) for helping us throughout the development of this project; introducing us to ASP.NET technology and for always answering any doubts we ever had. We could not have achieved this without the resources he made available and his valuable insights.

We’d also like to take this opportunity to express our gratitude to **Mr.Amit Srivastava** (Academic Committee, IPR) for his support and encouragement.

We would also like to thank our project guide, **Prof. Malay Bhatt** (Internal guide) for his constant guidance and inputs that helped us improve.

Regards,

Akanksha Khanna

Harsh Vyas

Abstract

The main purpose to develop this system was to facilitate easy communication between IPR and other colleges/institutes regarding internships and project training opportunities. It frees both the organizations from tedious email exchanges, phone calls and piles of paper records that need to be maintained.

We designed this project using ASP.Net technology and Microsoft SQL database. Our project provides effective communication as both, IPR staff and the college fill out forms stating their requirements and students' credentials respectively. These requests are then viewed by the Project Co-ordinator at IPR who assigns available projects to students depending upon the requirements stated.

Without this Academic Project Management System, it was very cumbersome for the Academic Committee at IPR to hire interns and assign them projects. And so there was a need to automate this entire process, which not only ensures that the allocation done doesn't involve any errors but also helps the Committee maintain records without the hassle of paperwork.

This single application can be used by the college to send students' details, by the IPR staff to specify their academic project requirements and by the Academic Committee to map these requests.

This system can however further be developed to include e-mail notifications when allotments have been done and also a request status which will let the requester know the status (in-process, allocated, etc) of his/her requests.

Table of Contents

List of figures

Chapter 3: Analysis

Chapter 5: Implementation

Chapter 7: Screenshots

Appendix A: Initial Learning

Appendix B: Data Flow Diagrams

List of Tables

Chapter 4: Data Dictionary

Chapter 5: Implementation

Chapter 6: Testing

Chapter 1
About the System

Automation of project allotment at IPR has not only made the entire process less cumbersome but also made sure that these records are safely stored in the database for future reference.

Had the system not been automated, piles of papers and/or files would have to be taken care of.

We developed this system in Microsoft Visual Studio 2012, using ASP.NET technology. And used Microsoft SQL Server for our database needs.

The Academic Project Management System broadly has four types of users. Each user has his/her own unique username and password, which ensures that the system cannot be anonymously accessed. The entire system is maintained by the Admin user who:

a) registers other colleges and institutes for further communication
b) carries out the major task of managing roles via the Role Management module provided.

The IPR Staff and college representative can put in their respective academic project related requests through the various forms provided in the system. Also it is necessary for the college contact person (HOD, Training/Placement Officer, etc) to first mail the college details to the Admin for college registration and to obtain a username and password to access the system.

The Project Co-ordinator at IPR gets a list of all the requests, internal and external. Also various views of these requests are available depending on the filter chosen. Reports are generated depending on the Project Co-ordinator's requirement. These reports can be exported to Microsoft Word, Excel or pdf format. And these reports can be easily printed.

This system is easy to use and we are sure that any user having basic knowledge of web browsing, filling online forms and file uploading will be able to efficiently use this system.

Chapter 2
Software Requirement Specification

2.1 INTRODUCTION

2.1.1 Problem Details

Institute for plasma Research – IPR, offers academic projects to students for practical exposure. For such activity there is communication between Institutions and IPR by email and requires many manual efforts to finalise projects for students. Project Management System will help organization in efficiently handling their academic projects related requirements for interns/trainees/students to allocate various projects. This system facilitates easy communication between the IPR and college/institute.

2.1.2 Purpose

The main purpose behind developing this "Academic Project Management System" is to allow colleges/ institutes to submit project requests for their students to IPR and IPR staff can submit their requirements for project students. This system ensures that the mapping between both internal and external requirements is effective and easy. Also this allows IPR to specify the qualifications they are looking for in the students and the duration for which they require the students.

2.1.3 Scope

This is a client server online application having a single database due to which users from different departments can use it at different locations simultaneously.

In Scope:

- The system is less tedious and more efficient compared to the previous manual process.
- To provide easy and fast access to information.
- To provide a user friendly environment and ease of use.

Not In Scope:

- E-mail notifications are not in scope in this version of the system.
- Also in future, provision can be made to display the "Request Status" of the internal and external requests received.

2.1.4 Technology & Tools Used

- Microsoft Asp.net Framework
- Microsoft Visual Studio 2012
- SQL Server Manager 2012

2.1.5 Software Development Model

- Iterative Waterfall Model

2.1.6 Abbreviations, Acronyms & Definitions

- APMS: Academic Project Management System
- IPR: Institute for Plasma Research
- MS: Microsoft
- SQL: Structured Query Language
- VS: Visual Studio
- LINQ: Language-Integrated Query
- SRS: Software Requirement Specification
- CSS: Cascading Style Sheets

2.2 General Description

2.2.1 Hardware Interfaces

- Memory minimum of 1GB RAM
- Hard disk of 40 GB
- Monitor
- Mouse
- Keyboard
- Printer

2.2.2 Software Interfaces

- Operating System: Windows XP / Vista / 7 / 8
- Front End: ASP.Net
- Back End: C#
- Designing: CSS

2.2.3 Communication Interfaces

This system is a Web Application and the communication between the user and the system takes place through a Web Browser.

2.2.4 Software Functions

- User authentication and login
- Receive requests from college and IPR Staff
- Allocation of students to academic projects
- Registration of Institutes and respective contact persons
- Management of Roles (creation of roles and their assignment)

2.2.5 Performance Requirements

- The overall system should be fast, reliable and error free.
- It should have built in error checking facilities.
- The system should be able to handle large amount of data comfortably.

2.2.6 Attributes

- Reliability

  In order to ensure the reliability of this system, we have developed it using MS Visual Studio 2012 which is stable and easy to use.
- Availability

  This system is designed to run as and when the user wishes to use it.

- Security

  The access to the system is given only to the users having a valid username and password. Without proper credentials, the system can't be accessed and the data remains safe. Also added security is provided by Visual Studio when passwords are stored in the database in encrypted format.

2.3 Overall Description

2.3.1 Product Perspective

This software is developed specifically to cater to the IPR employees' academic project requirements, is totally self-contained and works efficiently. It provides a robust database and it provides good and easy graphical user interface to both new as well as experienced user of the computer.

2.3.2 User Characteristics

2.3.2.1 End Users

- End user should have basic idea about internet surfing and filling of online application forms, uploading and downloading documents and photographs.

2.3.2.1 Administrator

- Administrator must be having good knowledge of database management system.
- Administrator must be capable to manage user rights.
- If the network connection does not work properly then our system should not work as intended.
- Also that is assumed that the product is installed properly on the web server.

- This system will not take care of any virus problem, which might occur either on the client or the server system. Avoiding the use of pirated software and ensure that floppies and other removable media are scanned for viruses before use.
- Recovery of data after a system crash will be possible only if backups are taken at regular intervals

2.3.3 Assumptions and Dependencies

Assumptions

- We assume that the client machine must have the environment required for this application to run.
- The client machine should be running 64-bit or 32-bit versions of Windows 7/8
- The operating system requirements for the server is 64-bit versions of Windows Server 2008 R2 with SP1.

Dependencies

- All necessary hardware and software are available for implementation and use of the tool.
- The client machine should have a printer.

2.4 Functional Requirements

Functional requirements define the fundamental actions that must take place in the software in accepting and processing the inputs and generating the outputs. These are generally listed as “shall” statements starting with "The system shall…”

Following are the functional requirements of our “Academic Project Management System” :

R1. Login

R1.1 Authenticate username and password

Input: User (IPR Staff/College Representative/Admin) enters username and password.

Output: System allows user to proceed.

Processing: Verification of username and password.

R1.2 Check type of user & rights given to user

Input: User provides user name and password.

Output: User is allowed access to the system if login is authenticated.

Processing: System verifies the type of user based on the username entered.

R1.3 Password change

Input: User provides old password, username and new password.

Output: System confirms new password.

R1.4 Logout

Input: User clicks on the logout link.

Output: System ends session and logs out the user.

R2. Internal Request

R2.1 Specify employee requirements

Input: Employee enters his/her project requirements in a form.

Output: Message prompt on successful request generation.

Processing: Employee's requirements get added to the database and internal request id is generated.

R2.2 Report Generation

Input: User selects a list of external requests to be viewed based on various filters.

Output: A concise report of all the requests is displayed in the browser and can be printed. This may also be viewed as a pdf, in excel or in Microsoft Word.

R3. External Request

R3.1 Provide student details

Input: User fills the student information form.

Output: Message prompt on successful request generation.

Processing: Student's details get added to the database and external request id is generated.

R3.2 Report Generation

Input: User selects a list of external requests to be viewed based on various filters.

Output: A concise report of all the requests is displayed in the browser and can be printed. This may also be viewed as a pdf, in excel or in Microsoft Word.

R4. Project Allocation Management

R4.1 Detailed view of Internal and External requests

Input: Project Co-ordinator clicks on request menu to view Internal or External requests.

Output: Detailed view of information provided by Internal and External requesters is displayed based on menu selection.

R4.2 Allocation of project requests

Input: Project Co-ordinator selects a particular Internal request and enters an external request id if request granted or rejects it.

Output: The status for respective Internal and External Request changes.

R4.3 Template Generation

Input: User enters the internal request id and clicks on "View Button".

Output: The template gets automatically downloaded as Microsoft Word ".doc" file.

R5. Role Management

R5.1 Creation of New Roles

Input: User enters the new role to be created.

Output: New role gets added to the database.

Processing: Checks for valid role name (eg. No commas) are carried out. Also the role shouldn't already exist.

R5.2 Assigning Users to Roles

Input: User selects the user that needs to be assigned role(s)

Output: The user is assigned to the particular role(s) and "Roles have been successfully assigned" message should be generated.

Processing: The user may be assigned more than one role.

R5.3 Assigning Roles to Users

Input: The user selects a role to which users need to be assigned.

Output: New user(s) is assigned to that particular role and "Roles have been successfully assigned" message should be generated.

Processing: More than one user maybe assigned that role.

R5.4 Removing Roles from Users

Input: Admin unchecks the particular role checked for that user

Output: Corresponding role entry in the database should be changed and "Roles have been successfully assigned" message should be generated.

R5.5 Removing Users from Roles

Input: Admin clicks delete button provided in the Grid View for that user.

Output: Corresponding role entry in the database should be changed.

R6. Course, Branch & Institute Details Management

R6.1 Providing Course and Branch Data

Input: User (admin only) enters the new course and/or branch name.

Output: Course and/or Branch get added to the database.

R6.2 Institute Registration

Input: User (admin only) enters details of the Institute.

Output: The Institute gets successfully registered.

Processing: Institute details are added to the database and a unique Institute ID is generated.

2.5 Non-Functional Requirements

R1. Usability

R1.1 Ease of Use

- The application created by us is user-friendly, so any non-technical user will find it easy to use.
- The application has a very interactive user interface. The nesting level of menu is not more than 2 which facilitates easy navigation.
- The menu titles have clarity and are meaningful.
- Tool tip is also provided in menus

R2. Reliability

R2.1 Error Handling

- The application should not crash
- The application should not have memory leak
- If any error occurs during runtime, it should notify the user about it.

R2.2 Ease of Recovery

- If the application crashes or any error occurs, the information and configuration data about the work done should be restored.

R3. Performance

R3.1 Response Time

- The application will be able to save the data which is submitted by user during session.

R3.2 Resource Usage

- The memory requirement of the application should be as minimum as possible. Also, the memory occupied by the application should not affect the performance of other applications in the PC.

Chapter 3
Analysis

3.1 Use Case Diagram

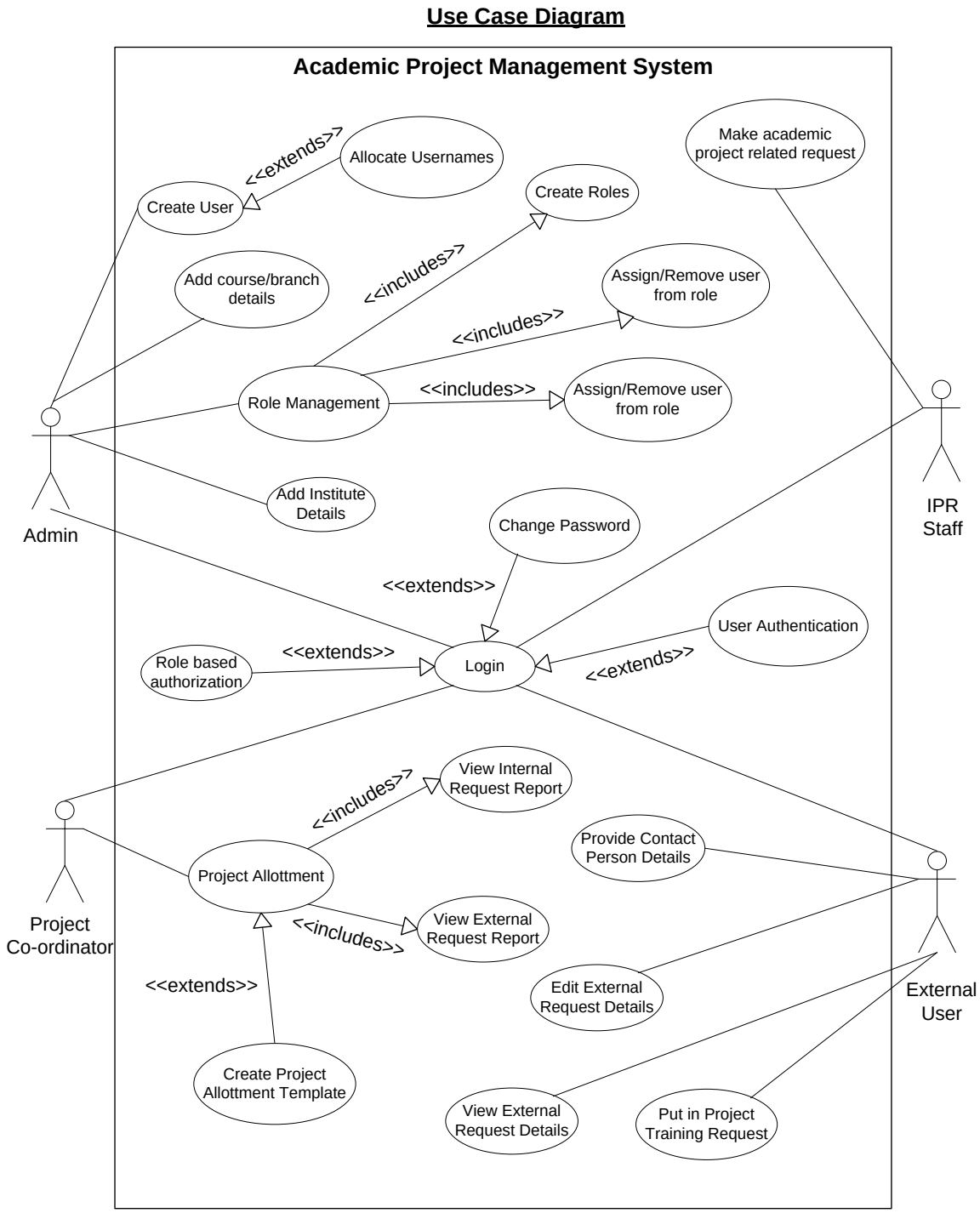


Figure 3.1: Use Case Diagram for APMS

3.2 Sequence Diagram

3.2.1 Sequence Diagram for Project Allocation

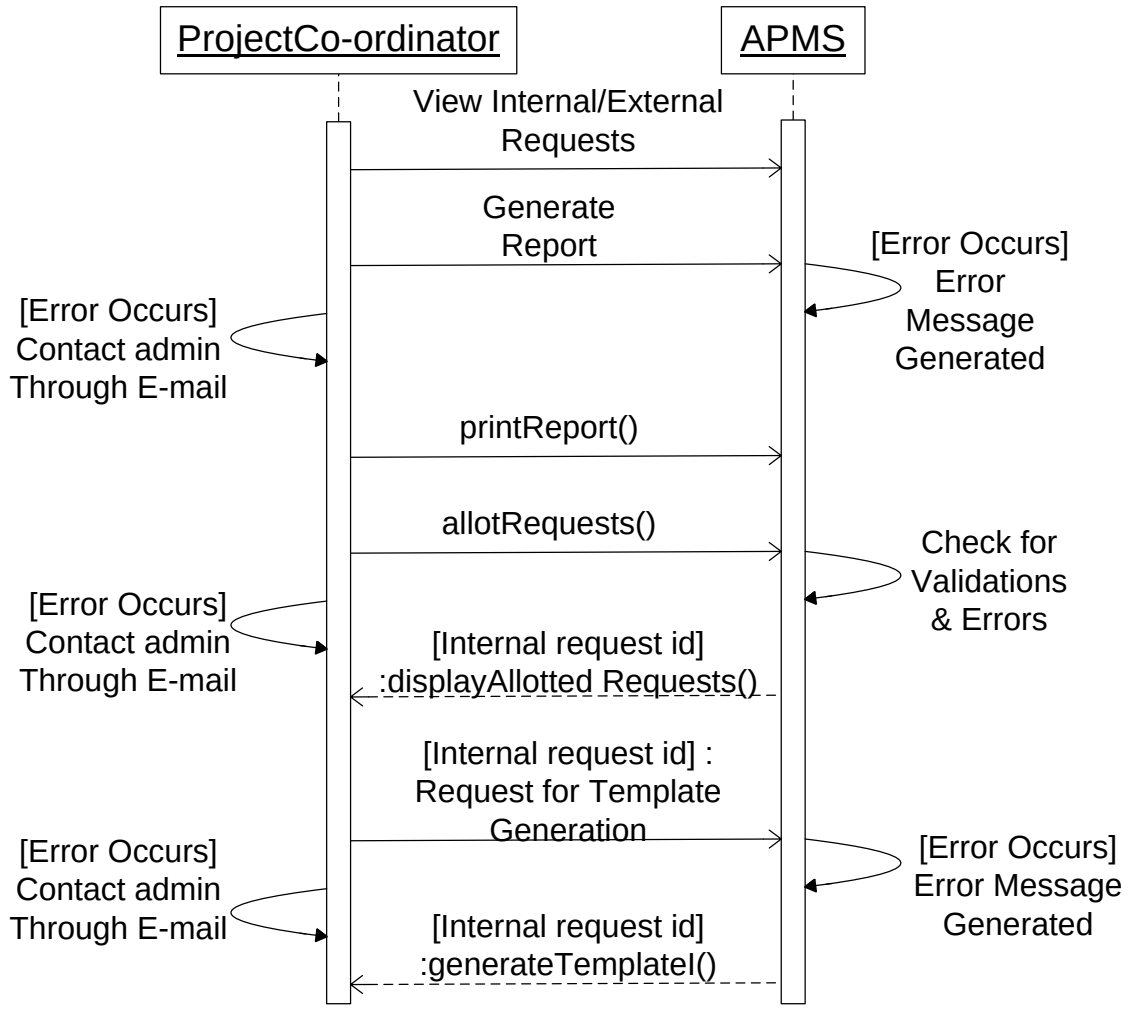


Figure 3.2: Sequence Diagram for Project Allocation

3.2.2 Sequence Diagram for External User Registration

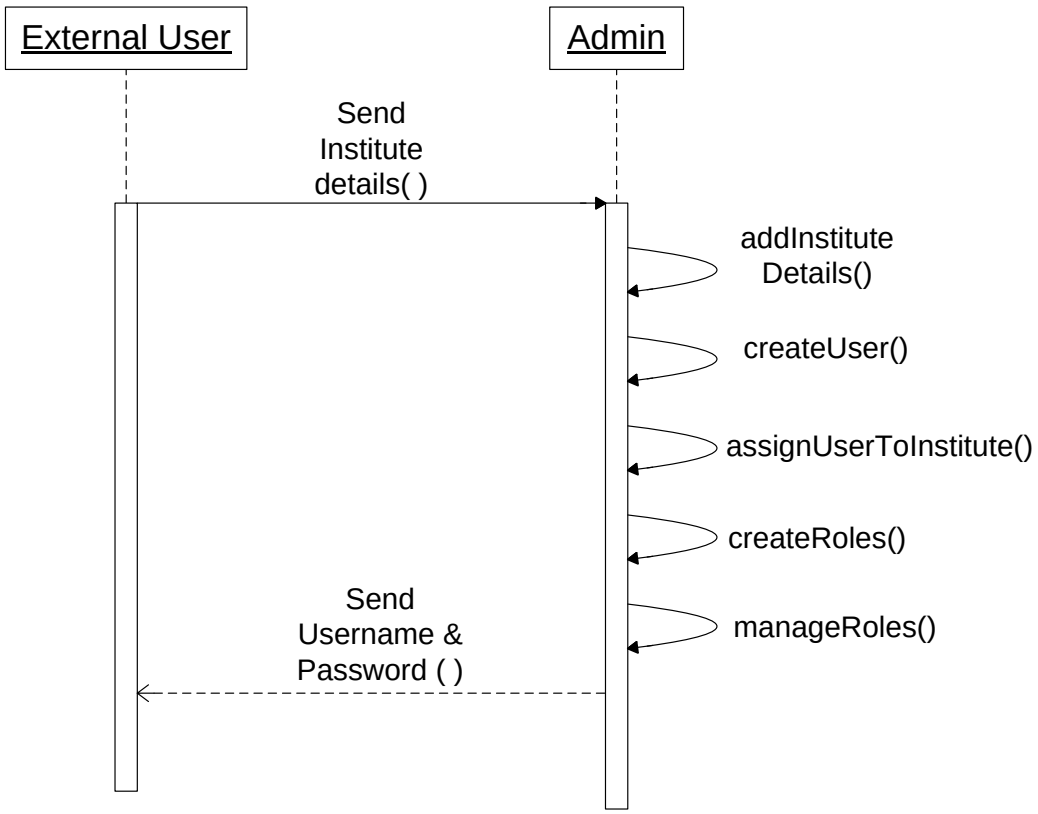


Figure 3.3: Sequence Diagram for External User Registration

3.2.3 Sequence Diagram for External Request

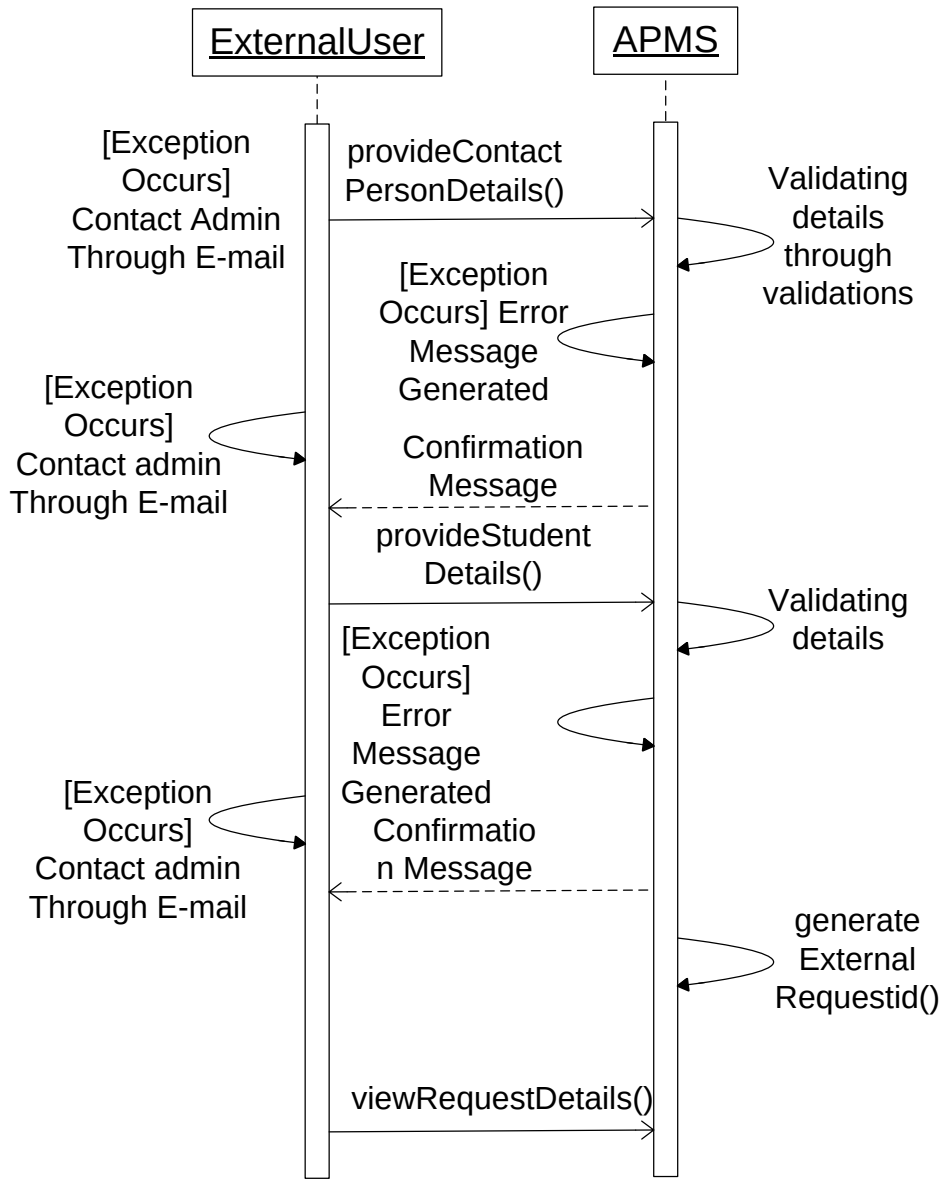


Figure 3.4: Sequence Diagram for External Request

3.2.4 Sequence Diagram for Internal Request

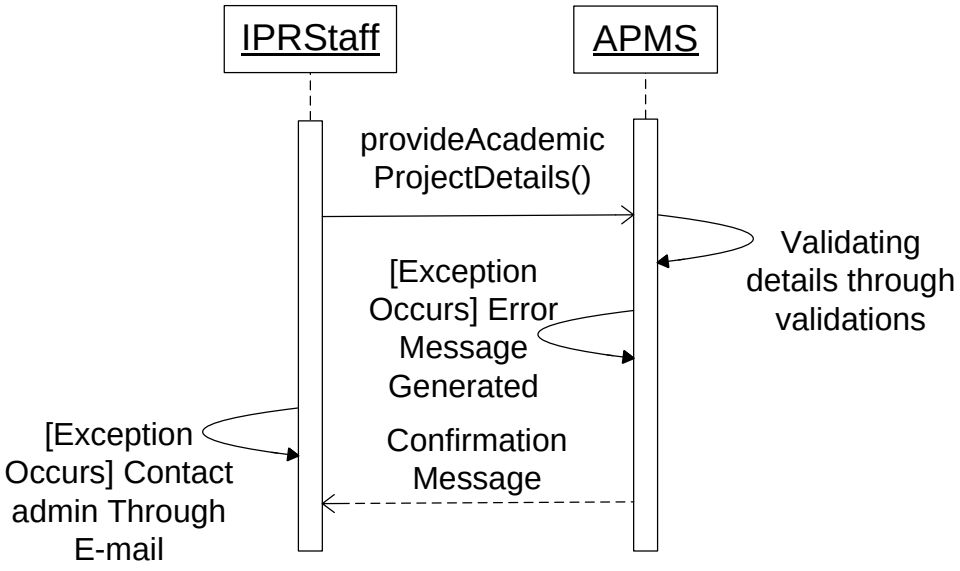


Figure 3.5: Sequence Diagram for Internal Request

3.2.4 Sequence Diagram for Internal Request

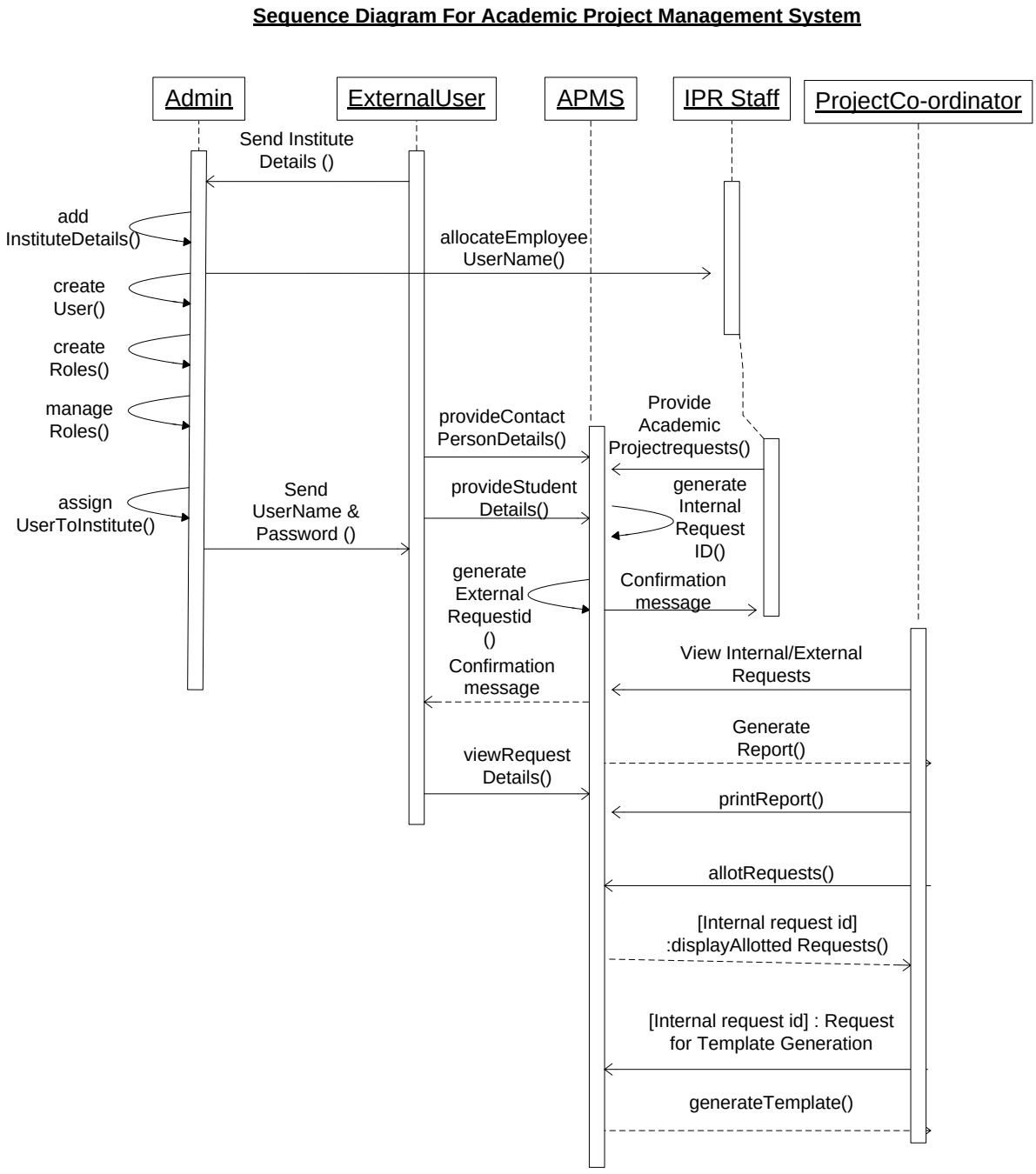


Figure 3.6: Sequence Diagram for APMS

3.3 Class Diagram

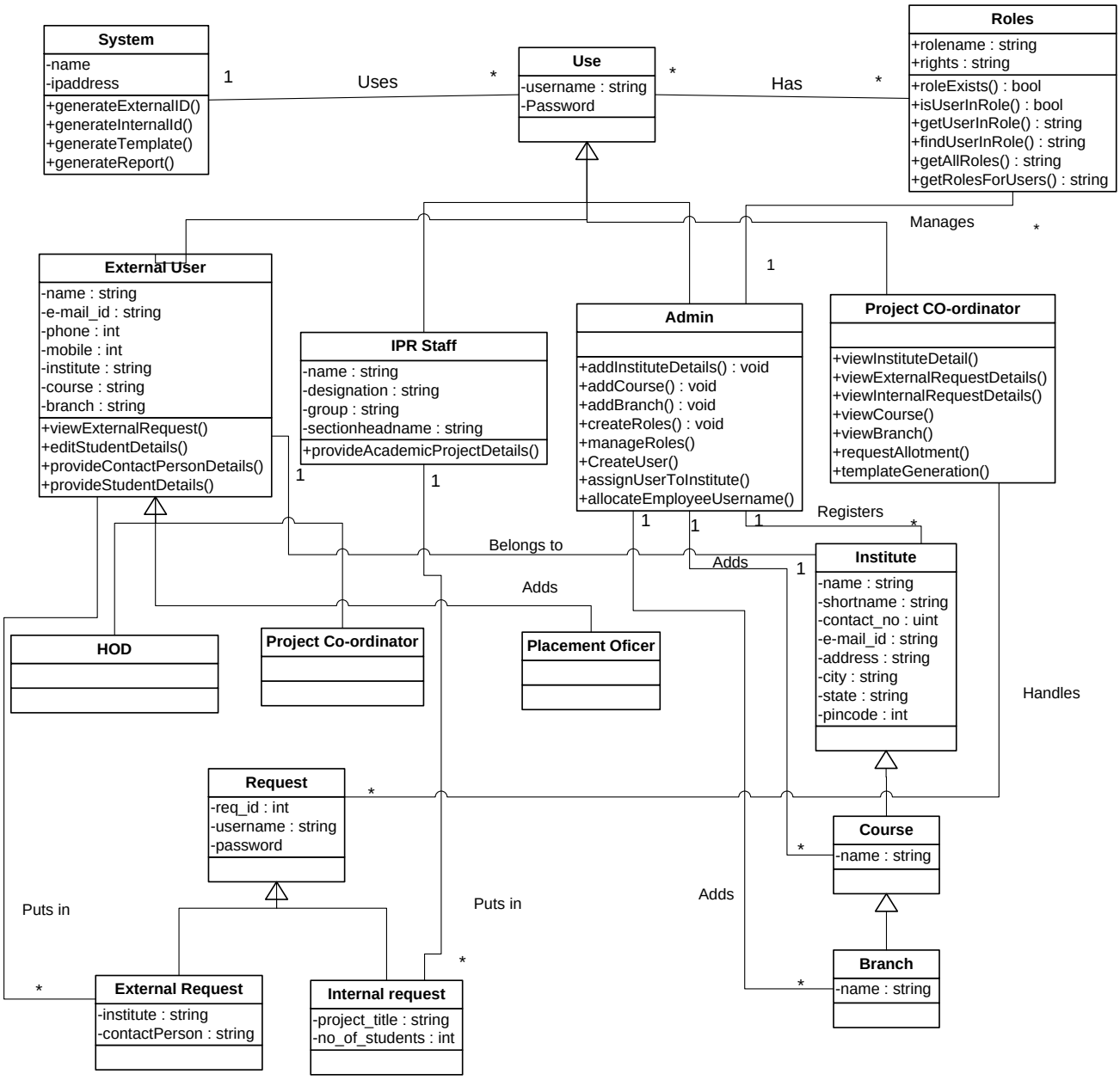


Figure 3.7: Class Diagram for APMS

3.4 Activity Diagram

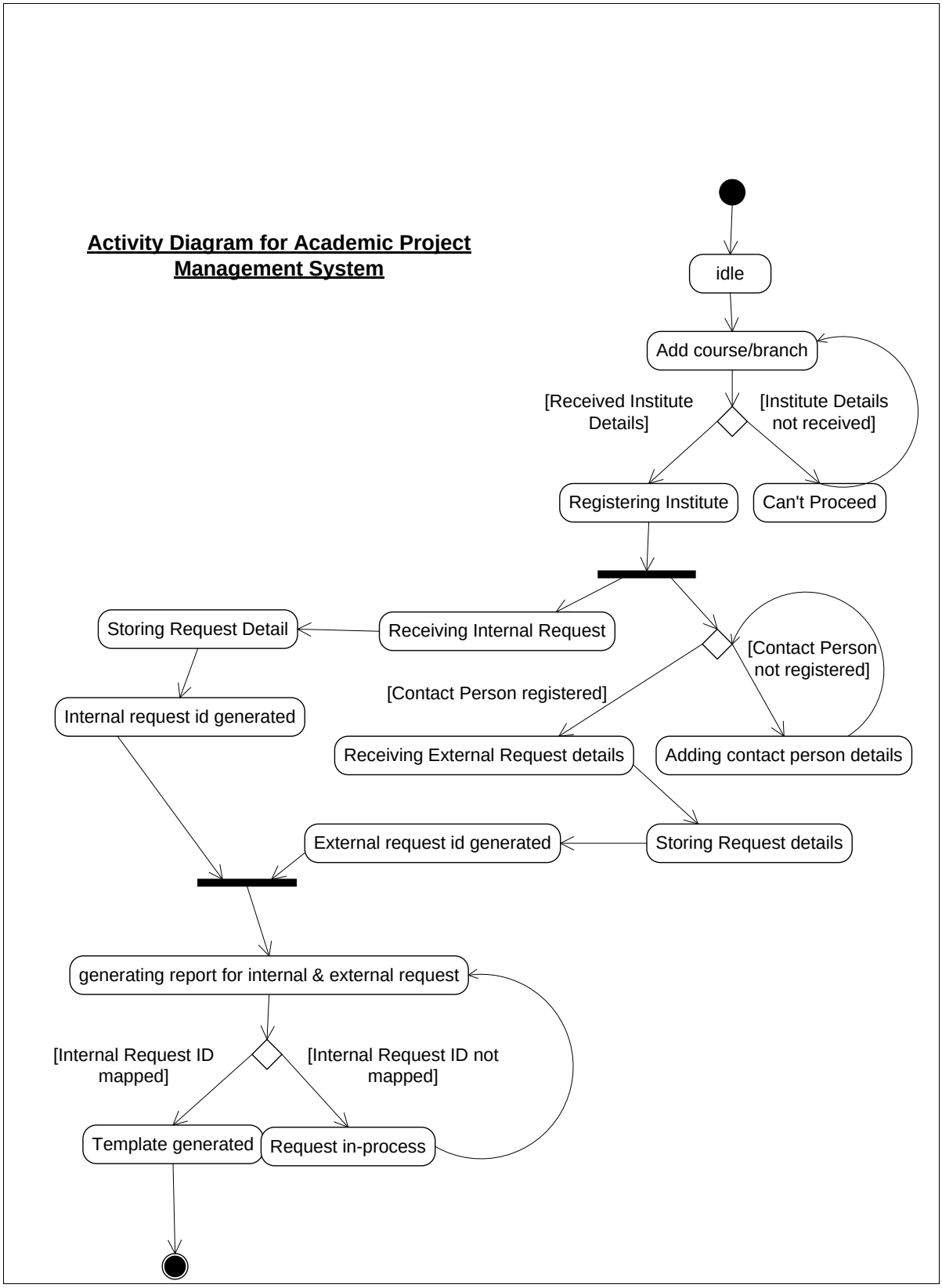


Figure 3.8: Activity Diagram for APMS

3.5 State Chart Diagram for APMS

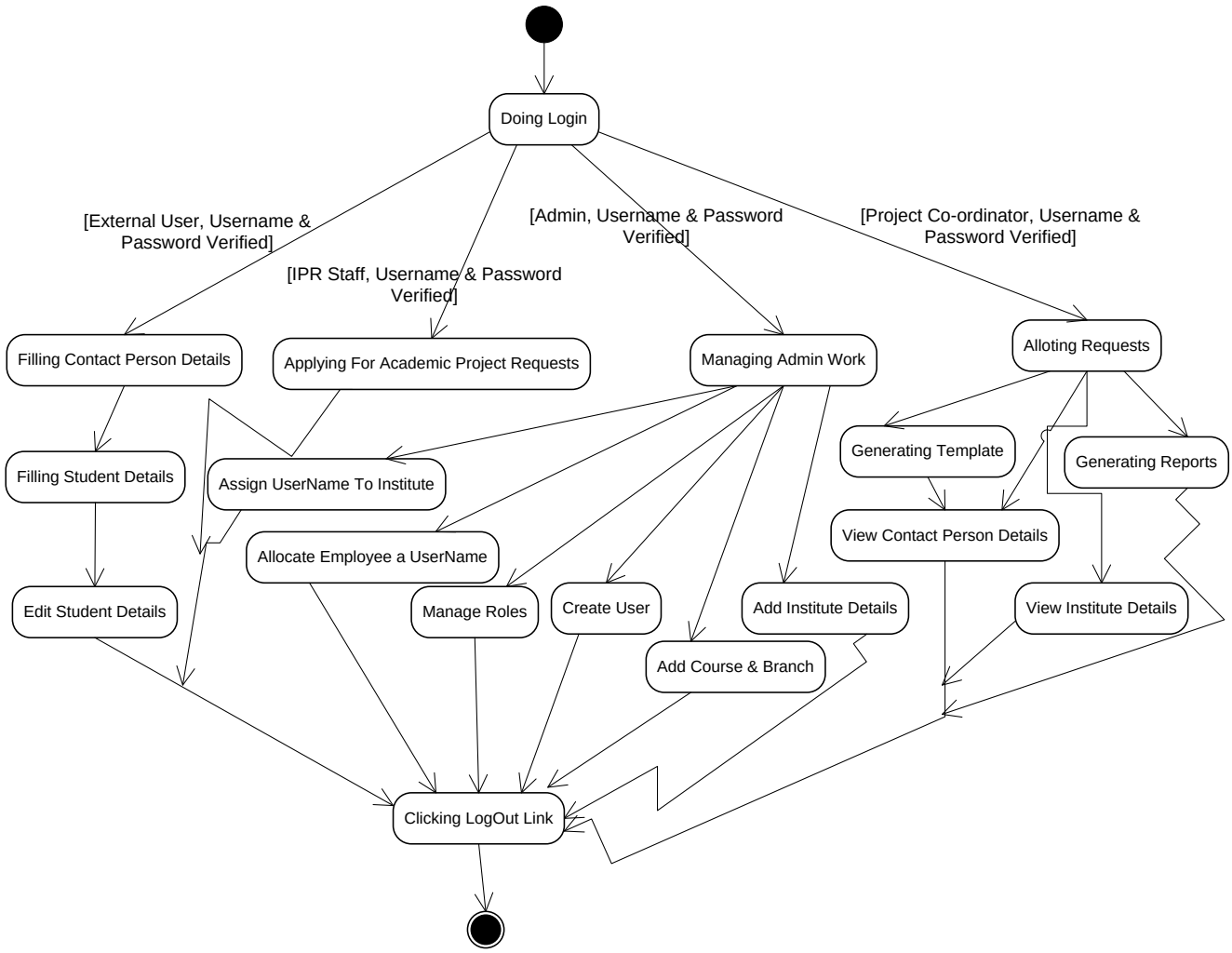


Figure 3.9: State Chart for APMS

3.6 Structure Chart

3. 6.1 Structure Chart for IPR Staff

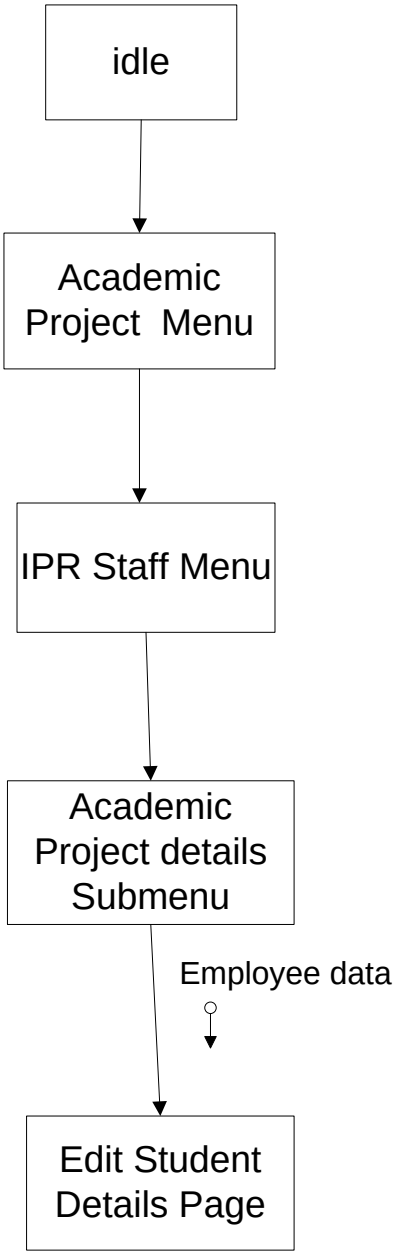


Figure 3.10: Structure Chart for IPR Staff

3.6.2 Structure Chart for Academic Project Co-ordinator

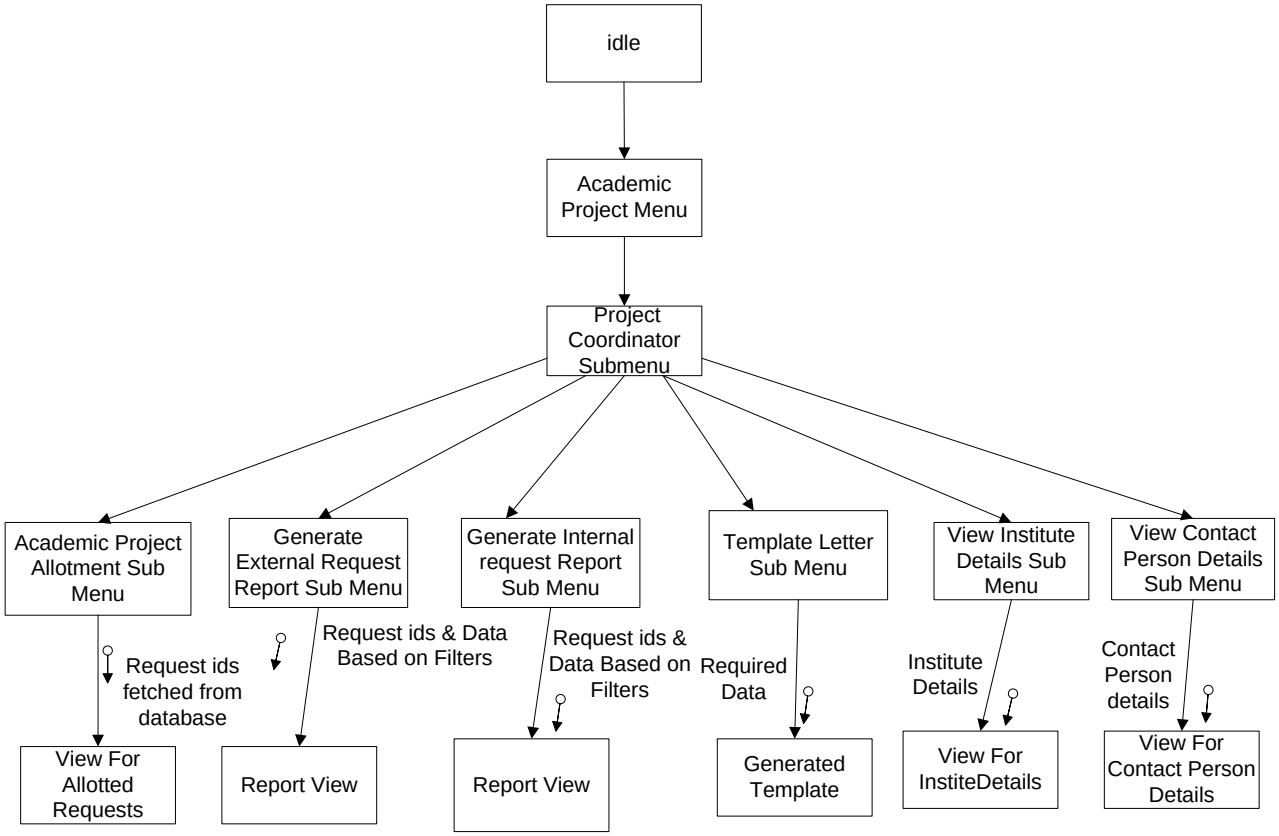


Figure 3.11: Structure Chart for Project Co-ordinator

3.6.3 Structure Chart for Admin

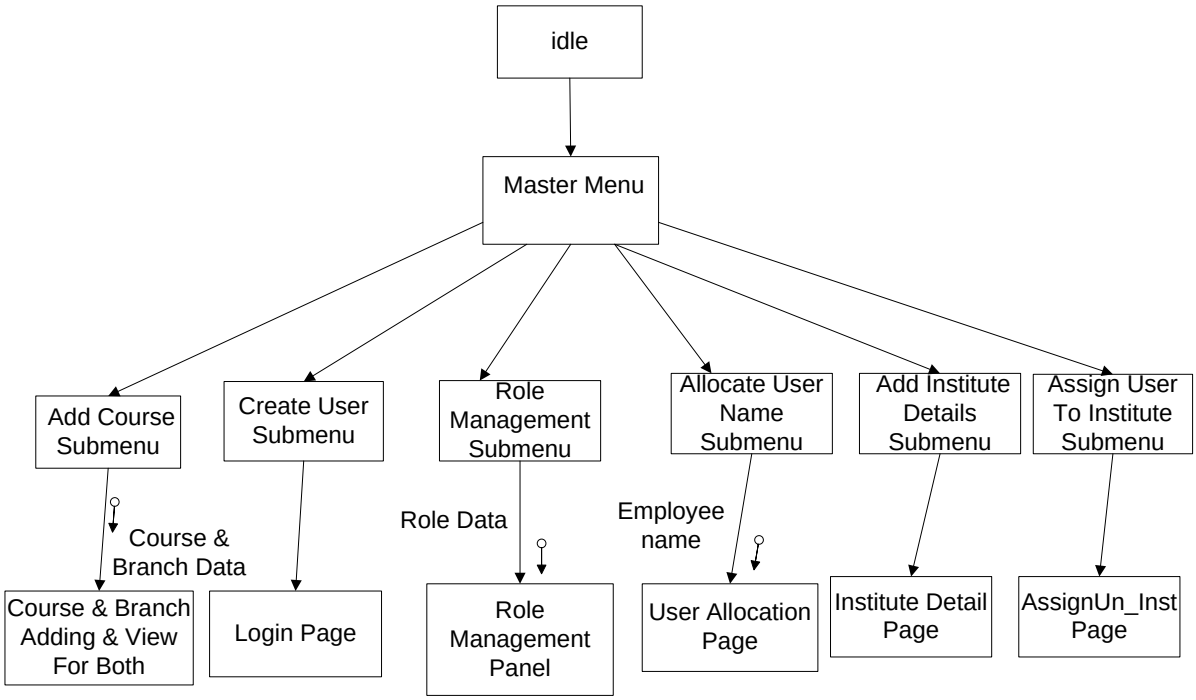


Figure 3.12: Structure Chart for Admin

3.6.5 Structure Chart External User

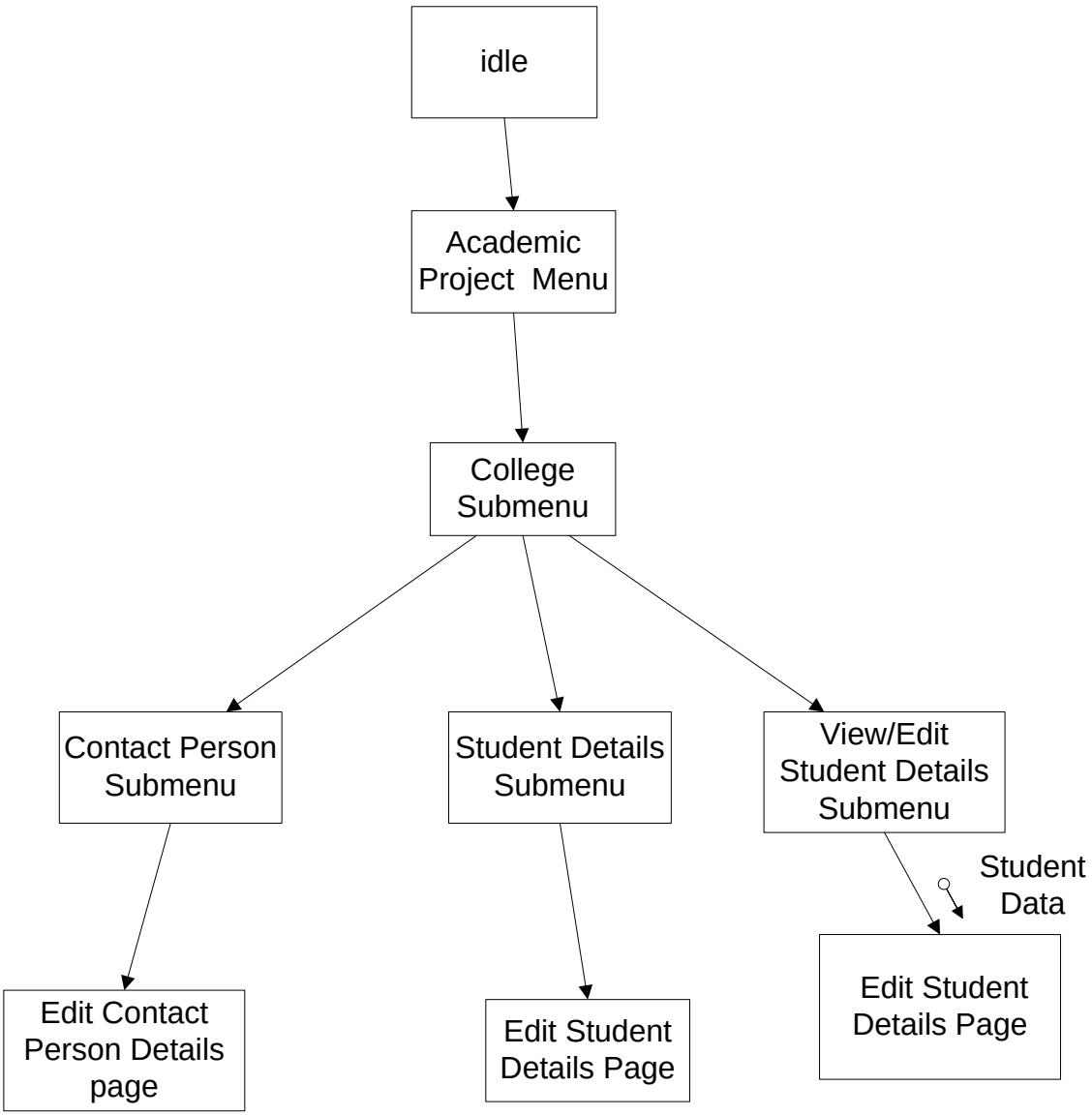


Figure 3.13 State Chart for External User

3.7 Entity Relationship (ER) Diagram

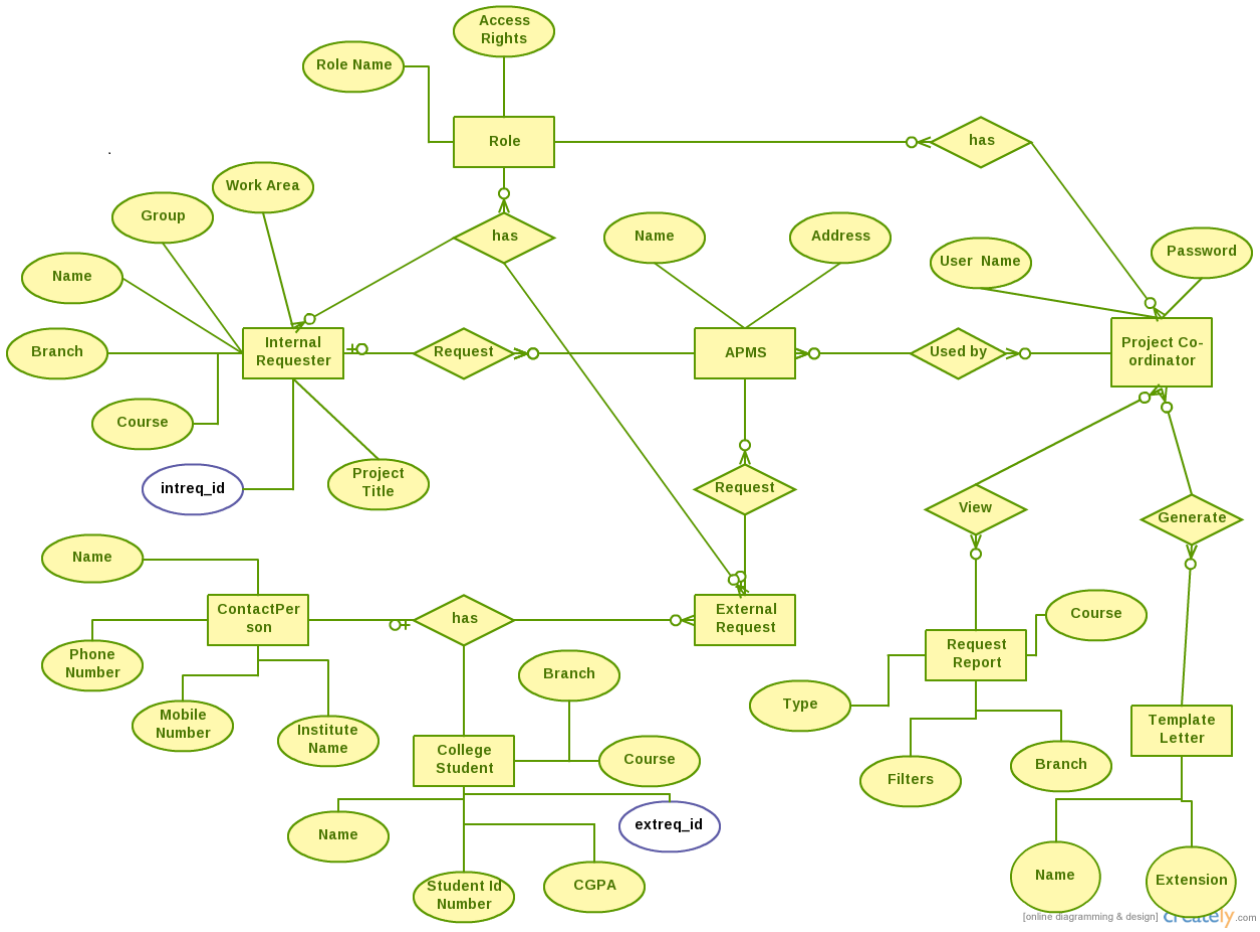


Figure 3.14 Entity Relationship Diagram for APMS

3.8 Table Relationship Diagram

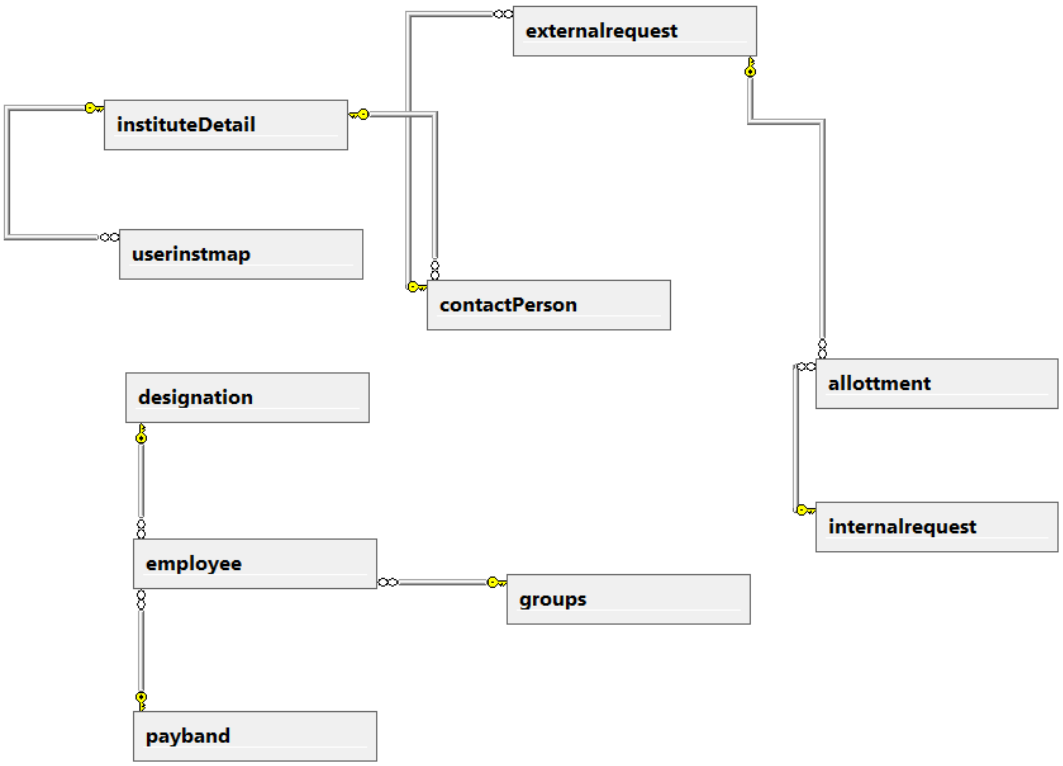


Figure 3.15a Table Relationship Diagram for APMS

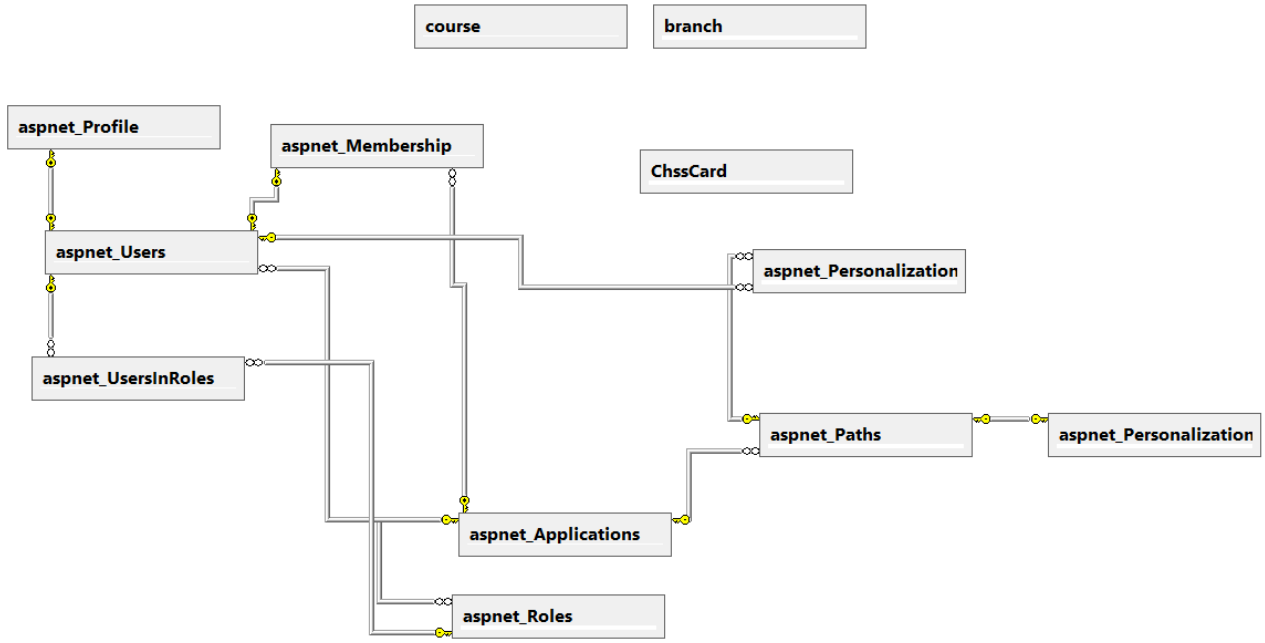


Figure 3.15b Table Relationship Diagram for APMS

3.9 Gantt Chart

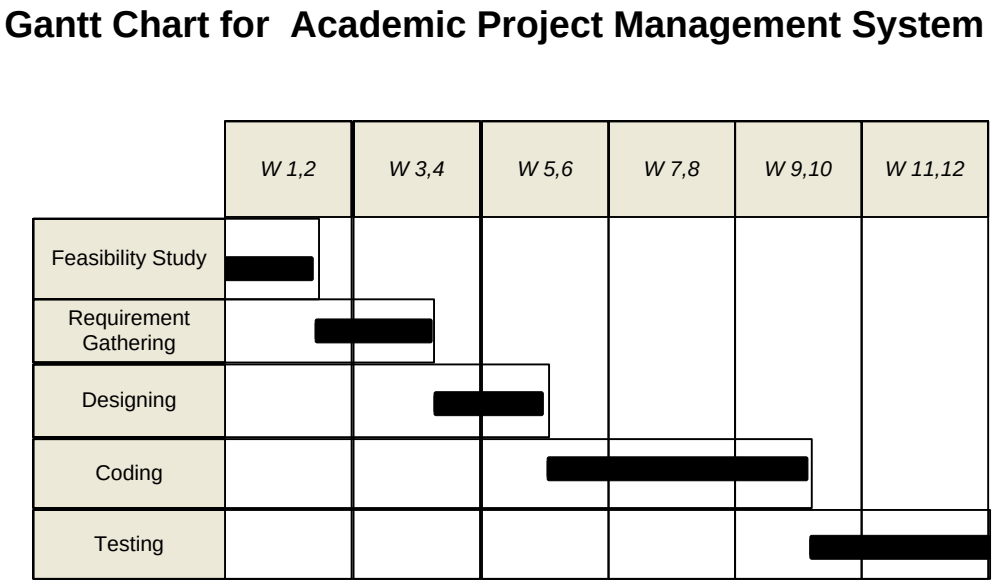


Figure 3.16 Gantt Chart for APMS

3.10 Pert Chart

Pert Chart for Academic Project Management System

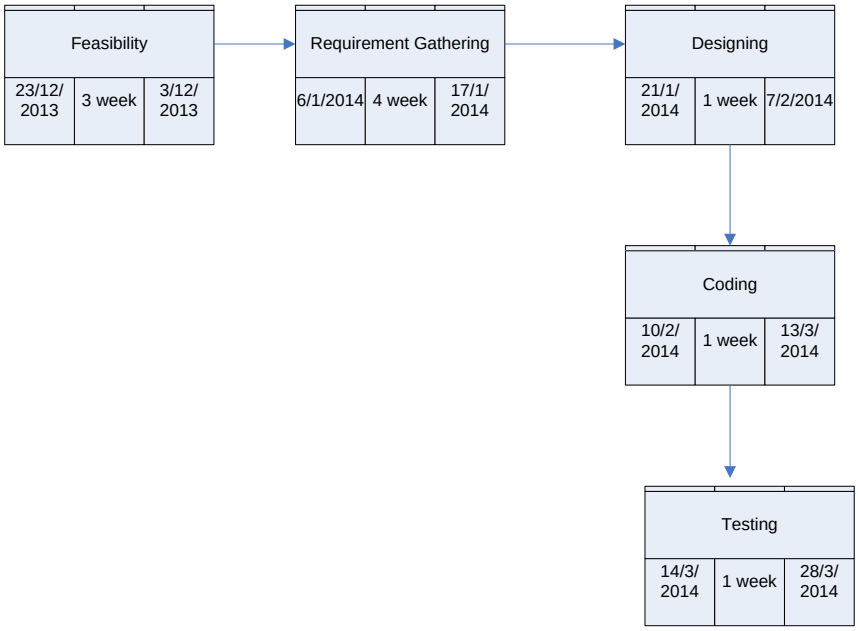


Figure 3.17 Pert Chart for APMS

Chapter 4
Data Dictionary

This section contains detailed information of all the tables that have been used and also the .aspx pages in which these tables have been used.

1. Dbo.allottment:

This table is used in allotment.aspx page under Project Co-ordinator authority.

Table 4.1 Allottment

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Allottment _id	numeric	0	6	NO	NULL	PK_ allottment_1	PRIMAR Y KEY
intreq_id	numeric	0	6	YES	NULL	FK_allottment _ externalrequest	FOREIGN KEY
extreq_id	numeric	0	6	YES	NULL	FK_allottment _ internalrequest	FOREIGN KEY

2. Dbo.branch:

This table is used in the following pages:

- AddCourse.aspx
- ExternalReport.aspx
- InternalReport.aspx
- Internal_req.aspx
- External_req.aspx

- ContactPerson.aspx
- EditStudentDetails.aspx

Table 4.2 branch

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
branch_id	numeric	0	6	NO	NULL	PK_branch	PRIMARY KEY
branchname	varchar	NULL	NULL	NO	50		

3. Dbo.ChssCard:

This table is used in ChssCard.aspx page.

Table 4.3 ChssCard

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Payrollno	numeric	0	20	NO	NULL	PK_ Chss Card	PRIMARY KEY
Spouse Name	varchar	NULL	NULL	YES	50		
Spouse Dob	date	NULL	NULL	YES	NULL		
ch1name	varchar	NULL	NULL	YES	50		
ch1dob	date	NULL	NULL	YES	NULL		
ch2name	varchar	NULL	NULL	YES	50		
ch2dob	date	NULL	NULL	YES	NULL		
Fname	varchar	NULL	NULL	YES	50		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Mname	varchar	NULL	NULL	YES	50		
ch3name	varchar	NULL	NULL	YES	50		
ch3dob	date	NULL	NULL	YES	NULL		
Eimg	varchar	NULL	NULL	YES	50		
Spouse Img	varchar	NULL	NULL	YES	50		
ch1img	varchar	NULL	NULL	YES	50		
ch2img	varchar	NULL	NULL	YES	50		
Fimg	varchar	NULL	NULL	YES	50		
Mimg	varchar	NULL	NULL	YES	50		
ch3img	varchar	NULL	NULL	YES	50		

4. Dbo.contactPerson:

This table is used in ContactPerson.aspx page.

Table 4.4 contactperson

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Contact _id	numeric	0	6	NO	NULL	PK _contact Person	PRIMARY KEY
Contact Person	varchar	NULL	NULL	YES	40		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Name	varchar	NULL	NULL	YES	30		
phone1	numeric	0	18	YES	NULL		
phone2	numeric	0	18	YES	NULL		
Email	varchar	NULL	NULL	YES	30		
Course	varchar	NULL	NULL	YES	20		
Appln Date	date	NULL	NULL	YES	NULL		
User Name	nvarchar	NULL	NULL	YES	256		
fk_inst _id	numeric	0	6	NO	NULL	FK_ Contact Person_ institute Detail	FOREIGN KEY

5. Dbo.course:

This table is used in the following pages:

- AddCourse.aspx
- ExternalReport.aspx
- InternalReport.aspx
- Internal_req.aspx
- External_req.aspx
- ContactPerson.aspx
- EditStudentDetails.aspx

Table 4.5 course

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
course _id	numeric	0	6	NO	NULL	PK_ course	PRIMARY KEY
Course name	varchar	NULL	NULL	NO	50		

6. Dbo.designation:

This table is used in following pages:

- ManageDesig.aspx
- Employee.aspx

Table 4.6 Designation

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
desig_code	numeric	0	6	NO	NULL	PK_Designation	PRIMARY KEY
desig_name	varchar	NULL	NULL	NO	25		

7. Dbo.employee:

This table is used in Employee.aspx page.

Table 4.7 employee

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Payno	numeric	0	6	NO	NULL		
Emp name	varchar	NULL	NULL	YES	30		
Doj	date	NULL	NULL	YES	NULL		
Dob	date	NULL	NULL	YES	NULL		
pay_ inband	numeric	0	6	YES	NULL		
Payband _id	numeric	0	10	NO	NULL		
Design _code	numeric	0	6	NO	NULL		
Group _code	numeric	0	18	NO	NULL		
Emp status	varchar	NULL	NULL	YES	20		
Emailipr	varchar	NULL	NULL	YES	20		
Emailalt	varchar	NULL	NULL	YES	20		
Extno	varchar	NULL	NULL	YES	20		
Room lab	varchar	NULL	NULL	YES	25		
contact1	varchar	NULL	NULL	YES	25		
contact2	Varchar	NULL	NULL	YES	25		
Dor	date	NULL	NULL	YES	NULL		
Extdate	date	NULL	NULL	YES	NULL		
Dol	date	NULL	NULL	YES	NULL		
Doi	date	NULL	NULL	YES	NULL		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
User Name	nvarchar	NULL	NULL	YES	256		

8. Dbo.externalrequest:

This table is used in following pages.

- EditStudentDetails.aspx
- External_req.aspx
- ViewRequestStatus.aspx
- ExternalReport.aspx
- Allotment.aspx

Table 4.8 externalrequest

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Extreq _id	Numeric	0	6	NO	NULL	PK_ Student Detail	PRIMARY KEY
Name	Varchar	NULL	NULL	YES	40		
collegeid	Varchar	NULL	NULL	YES	30		
course	Varchar	NULL	NULL	YES	50		
semester	Numeric	0	3	YES	NULL		
branch	Varchar	NULL	NULL	YES	50		
startdate	Date	NULL	NULL	YES	NULL		
enddate	Date	NULL	NULL	YES	NULL		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
cgpa	Numeric	2	4	YES	NULL		
projectdone	Varchar	NULL	NULL	YES	100		
interest	Varchar	NULL	NULL	YES	100		
reqstatus	Varchar	NULL	NULL	YES	50		
Appln date	Date	NULL	NULL	YES	NULL		
fk_ contact _id	Numeric	0	6	YES	NULL	FK_ External request_ contact Person	FOREIGN KEY

9. Dbo.groups:

This table is used in following pages:

- ManageGroup.aspx
- Employee.aspx

Table 4.9 groups

COLUMN NAME	DATATYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
group_code	numeric	0	18	NO	NULL	PK_Group	PRIMARY KEY
group_name	varchar	NULL	NULL	NO	25		

10. Dbo.instituteDetail:

This table is used in following pages:

- InstituteDetails.aspx
- ContactPerson.aspx
- ViewInstituteDetails.aspx

Table 4.10 instituteDetail

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
inst_id	numeric	0	6	NO	NULL	PK_ Institute Detail	PRIMARY KEY
instname	varchar	NULL	NULL	YES	50		
Short Name	varchar	NULL	NULL	YES	50		
phone1	varchar	NULL	NULL	YES	20		
phone2	varchar	NULL	NULL	YES	20		
Email	varchar	NULL	NULL	YES	50		
instaddr	varchar	NULL	NULL	YES	100		
City	varchar	NULL	NULL	YES	30		
State	varchar	NULL	NULL	YES	300		
pincode	numeric	0	18	YES	NULL		

11. Dbo.internalrequest:

This table is used in following pages.

- Internal_req.aspx
- InternalReport.aspx
- Allotment.aspx

Table 4.11 internalrequest

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
intreq_id	numeric	0	6	NO	NULL	PK_ Internal request	PRIMARY KEY
course	varchar	NULL	NULL	YES	50		
branch	varchar	NULL	NULL	YES	50		
Num students	numeric	0	3	YES	NULL		
title	varchar	NULL	NULL	YES	500		
payno	numeric	0	6	YES	NULL		
Guide name	varchar	NULL	NULL	YES	50		
designation	varchar	NULL	NULL	YES	50		
Group name	varchar	NULL	NULL	YES	50		
Group head	varchar	NULL	NULL	YES	50		
Proj duration _num	numeric	0	6	YES	NULL		
Proj Duration _unit	varchar	NULL	NULL	YES	20		
publication s	varchar	NULL	NULL	YES	500		
Pastproj details	varchar	NULL	NULL	YES	500		
worklab	varchar	NULL	NULL	YES	50		
applndate	Date	NULL	NULL	YES	NULL		

12. Dbo.payband:

This table is used in following pages.

- ManagePayband.aspx
- Employee.aspx

Table 4.12 payband

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
payband_id	numeric	0	10	NO	NULL	PK_payband	PRIMARY KEY
Payband	varchar	NULL	NULL	YES	50		
grade_pay	numeric	0	18	NO	NULL		

13. Dbo.aspnet_Membership:

This table is used in following pages:

- Login1.aspx
- CreateUserwiz.aspx
- ChangePassword.aspx

Table 4.13 aspnet_Membership

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Application Id	Unique identifier	NULL	NULL	NO	NULL	FK_aspnet_ Me_Appli	FOREIGN KEY
UserId	Unique identifier	NULL	NULL	NO	NULL	PK_aspnet_ M	PRIMARY KEY
Password	nvarchar	NULL	NULL	NO	128		
Password Format	int	0	10	NO	NULL		
Password Salt	nvarchar	NULL	NULL	NO	128		
Mobile PIN	nvarchar	NULL	NULL	YES	16		
Email	nvarchar	NULL	NULL	YES	256		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Lowered Email	nvarchar	NULL	NULL	YES	256		
Password Question	nvarchar	NULL	NULL	YES	256		
Password Answer	nvarchar	NULL	NULL	YES	128		
IsApproved	Bit	NULL	NULL	NO	NULL		
IsLocked Out	Bit	NULL	NULL	NO	NULL		
CreateDate	datetime	NULL	NULL	NO	NULL		
LastLogin Date	datetime	NULL	NULL	NO	NULL		
Last Password Changed Date	datetime	NULL	NULL	NO	NULL		
LastLockout Date	datetime	NULL	NULL	NO	NULL		
Failed Password Attempt Count	Int	0	10	NO	NULL		
Failed Password Attempt WindowStart	datetime	NULL	NULL	NO	NULL		
Failed Password Answer Attempt Count	Int	0	10	NO	NULL		
Failed Password Answer Attempt WindowStart	datetime	NULL	NULL	NO	NULL		
Comment	ntext	NULL	NULL	YES	1073741823		

14. Dbo.aspnet_Roles:

This table is used in RoleMgmt.aspx page.

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Application Id	Unique identifier	NULL	NULL	NO	NULL	FK_aspne _Ro_Appli	FOREIGN KEY
RoleId	Unique identifier	NULL	NULL	NO	NULL	PK _aspnet_R	PRIMARY KEY
RoleName	nvarchar	NULL	NULL	NO	256		
Lowered Role Name	nvarchar	NULL	NULL	NO	256		
Description	nvarchar	NULL	NULL	YES	256		

15. Dbo.aspnet_Users:

This table is used in following pages:

- Login1.aspx
- RoleMgmt.aspx

Table 4.15 aspnet_Users

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Applicatio n Id	Unique identifier	NULL	NULL	NO	NULL	FK_aspnet _Us_Appli	FOREIGN KEY
UserId	Unique identifier	NULL	NULL	NO	NULL	PK_aspnet_ U_	PRIMARY KEY
UserName	nvarchar	NULL	NULL	NO	256		

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Lowered UserName	nvarchar	NULL	NULL	NO	256		
Mobile Alias	nvarchar	NULL	NULL	YES	16		
Is Anonymous	bit	NULL	NULL	NO	NULL		
Last Activity Date	datetime	NULL	NULL	NO	NULL		

16. Dbo.aspnet_UsersInRoles:

This table is used in RoleMgmt.aspx page.

Table 4.16 UsersinRoles

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
User Id	Unique identifier	NULL	NULL	NO	NULL	PK_aspnet_U FK_aspnet_Us_UserI FK_aspnet_Us_UserI	PRIMARY KEY FOREIGN KEY
Role Id	Unique identifier	NULL	NULL	NO	NULL	PK__aspnet_U FK_aspnet_Us_RoleI FK_aspnet_Us_RoleI	PRIMARY KEY FOREIGN KEY

17. Dbo.userinstmap

This table is used in AssignUn_Inst.aspx page.

Table 4.17 userinstmap

COLUMN NAME	DATA TYPE	NUMERIC SCALE	NUMERIC PRECISION	IS NULLABLE	CHARACTER MAXIMUM LENGTH	CONSTRAINT NAME	CONSTRAINT TYPE
Id	numeric	0	6	NO	NULL	PK_userinstmap	PRIMARY KEY
username	varchar	NULL	NULL	NO	50		
fk_instid	varchar	NULL	NULL	NO	50	FK_userinstmap_instituteDetail	

Chapter 5
Implementation

5.1 Methods & their Description

Table 5.1 Methods & their description

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
1.	System	DateTime (Structure)	Now	Gets a DateTIme object that is set to the current date and time on this computer
2.	System.Data. SqlClient	SqlCommand	SqlCommand()	Initializes a new instance of this class.
			ExecuteNonQuery()	Executes a Transact-SQL statement against the connection and returns the number of rows affected.
			ExecuteScalar()	Executes the query, and returns the first column of the first row in the result set returned by the query. Additional columns or rows are ignored.
			ExecuteReader()	Sends the CommandText to the Connection and builds a SqlDataReader.
		SqlConnection	SqlConnection()	Initializes a new instance of this class.
			Open()	Opens a database connection with the property settings specified by the ConnectionStri ng.

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
			Close()	Closes the connection to the database.
		SqlDataReader	Read()	Advances the SqlDataReader to the next record.
3.	System.Configuration	ConfigurationManager	ConnectionStrings	Gets the ConnectionStrings Section data for the current application's default configuration
4.	Microsoft.Reporting. WebForms	Processing Mode	Local	Specifies that the report will be processed and redered using thereporting engine provided by the report viewer.
5.	System.Linq.Xml			Contains the classes for LINQ to XML.
6.	System.Web.UI			This namespace provides classes and interfaces that enable you to create ASP.NET server controls and ASP.NET Web pages.
		HtmlControls		The System.Web.UI.HtmlControls namespace contains classes that allow you to create HTML server controls on a Web Forms page.
7.	System.Web.UI.Web Controls	Button	Button()	Initializes a new instance of the Button Class.
			OnClick()	Raises the Click event of the Button Control.
			CausesValidation	Gets or Sets a value indicating whether

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
				validation is performed when the button control is clicked.
			ID	Gets or sets the programmatic identifier assigned to the server control.
			Text	Gets or sets the text caption displayed in the Button Control.
			Visible	Gets or Sets a value that indicates whether a server control is rendered as UI on the page.
		CheckBoxList	CheckBoxList()	Creates a multi selection checkbox group that can be dynamically created by binding the control to a data source.
			DataBind()	Binds a data source to the invoked server control and all its child controls.
			ID	Gets or sets the programmatic identifier assigned to the server control.
			SelectedValue	Gets the value of the selected item in the list control.
			OnSelectedIndexChanged()	Raises the SelectedIndexChanged event. This allows you to provide a custom handler for the event.
		ContentPlaceHolder		Defines a region for content in an ASP.NET master

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
				page.
		CreateUser Wizard		Provides a user interface for creating new website user accounts.
		DropDownList	DropDownList()	Represents a control that allows the user to select a single item from a dropdown list.
			DataBind()	Binds a data source to the invoked server control and all its child controls.
			ID	Gets or sets the programmatic identifier assigned to the server control.
			SelectedValue	Gets the value of the selected item in the list control.
			OnSelectedIndexChanged( )	Raises the SelectedIndexChanged event. This allows you to provide a custom handler for the event.
		GridView	GridView()	Displays the values of a data source in a table where each column represents a field and each row represents a record. This Control enables you to select, sort and edit this items.
			AllowPaging	Gets or Sets a value indicating whether the paging feature is enabled.
			AllowSorting	Gets or Sets a value indicating whether

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
				the sorting feature is enabled.
			AutoGenerate DeleteButton AutoGenerate EditButton	Gets or Sets a value indicating whether a CommandField field column with a delete/edit button for each data row is automatically added to a gridview control.
			Caption	Gets or Sets the text to render in an HTML caption element in a gridview control.
			DataSource	Gets or Sets the object from which the data bound control retrieves its list of data items.
			FooterRow	Gets a gridview object that represents the footer row in a gridview control.
			ID	Gets or sets the programmatic identifier assigned to the server control.
			Visible	Gets or sets a value that indicates whether a server control is rendered as UI on the page.
			DataBind()	Binds the data source to the gridview control.
		ImageButton		A control that displays an image and responds to mouse clicks on the image.
		Label		Represents a label control, which displays text on a web page.

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
		LinkButton		Displays a hyperlink style button control on web page.
		Login		Provides UI elements for login into a website.
		Menu		Displays a menu in an ASP.NET web page.
		Regular Expression Validator		Validates whether the value of an associated input control matches the pattern specified by a regular expression.
		RequiredField Validator		Makes the associated input control a required field.
		SiteMapData Source		Provides a data source control that webserver controls and other controls can use to bind to hierarchical site map data.
		SqlDataSource		Represents a SQL database to data-bound controls.
		Table		Displays a table on a web page.
		TextBox	TextBox()	Displays a texbox control for user input.
			ID	Gets or sets the programmatic identifier assigned to the server control.
			AutoPostBack	Gets or Sets a value that indicates whether an automatic postback to the server occurs when the textbox control loses focus.

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
			CausesValidation	Gets or Sets a value indicating whether validation is performed when the button control is clicked.
			ReadOnly	Gets or Sets a value indicating whether the contents of the textbox control can be changed.
			OnTextChange ()	Raises the TextChanged event. This allows you to handle the event directly.
8.	System.Web.Security	Roles		Manages user membership in roles for authorisation checking in an ASP.NET application.
			AddUsersToRole()	Adds the specified users to the specified role.
			AddUserToRoles()	Adds the specified user to the specified roles.
			AddUserToRole()	Adds the specified user to the specified role.
			CreateRole()	Adds a new role to the datasource.
			FindUsersInRole()	Gets a list of users in a specified role whether user name contains the specified user name to match.
			GetAllRoles()	Gets a list of all the roles for the application.
			GetRolesForUsers(String)	Gets a list of the roles that a user is in.
			GetUsersInRole()	Gets a list of users in the specified

SR. NO	NAMESPACE	CLASS	METHOD/PROPERTY	DESCRIPTION
				role.
			IsUserInRole (String,String)	Gets a value indicating whether the specified user is in the specified role.
			RemoveUserFromRole()	Removes the specified user from the specified role.
			RoleExists()	Gets a value indicating whether the specified role name already exists in the role datasource.
		SqlMembership Provider		Manages storage of membership information for an ASP.NET application in a SqlServer database.
		SqlRoleProvider		Manages storage of Role membership information for an ASP.NET application in a SqlServer database.
9.	System.Linq			Provides classes and interfaces that support queries that use LINQ.
10.	System.Collections			Contains interfaces and classes that define various collections of objects, such as lists, queues, etc.
11.	IPR	Date_FileClass	Date_FileClass()	Creates an instance of this class.
			ExtractDate(TextBox)	Returns date entered in the textbox in (“yyyy/MM/dd”) format.
			FileUp (FileUpload,TextBox,Text Box)	Renames the uploaded file name.

5.2 Naming Conventions

We have followed the naming convention stated below while coding. We followed this at the time of giving ID names of various Microsoft Visual Studio 2012 controls like Textbox, Checkbox List etc.

Table 5.2 Naming Conventions

Sr. No.	Control Name	Convention followed
1.	Textbox	txtName
2.	Label	lblName
3.	Dropdown List	dmListname
4.	Checkbox List	cblListname
5.	Button	btnButtonName
6.	GridView	gvName
7.	SqlDataSource	SqlDataSource1
8.	RequiredFieldValidator	rfvName

5.3 Error Handling

Error 1:

Server Error in '/' Application.

The parameterized query '(@payno nvarchar(6),@empname nvarchar(3),@doj nvarchar(4000),@do' expects the parameter '@dob', which was not supplied.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: The parameterized query '(@payno nvarchar(6),@empname nvarchar(3),@doj nvarchar(4000),@do' expects the parameter '@dob', which was not supplied.

Source Error:

```
Line 91:
Line 92:
Line 93:                        cmd.ExecuteNonQuery();
Line 94:
Line 95:                        Response.Write("<script>alert('Productinfo entered')</script>");
```

Source File: c:\Users\MEGHAL\Desktop\IPR\IPR\Employee.aspx.cs **Line:** 93

Stack Trace:

```
[SqlException (0x80131904): The parameterized query '(@payno nvarchar(6),@empname nvarchar(3),@doj nvarchar(4000),@do' expects the parameter '@dob', which was not sup
   System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +1753346
   System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) +5295154
   System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose) +242
   System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParser
   System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString) +269
   System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task
   System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, String method, TaskCompletionSource`1
   System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(TaskCompletionSource`1 completion, String methodName, Boolean sendToPipe, Int32 timeout, Boolean asyncWrit
   System.Data.SqlClient.SqlCommand.ExecuteNonQuery() +160
   IPR.Employee.Submit_Click(Object sender, EventArgs e) in c:\Users\MEGHAL\Desktop\IPR\IPR\Employee.aspx.cs:93
```

Figure 5.1 Parameterized Query error

Solution :- In parameterized insert query, If you want to pass null value then write “ ” in Parameters.AddWithValue’s argument instead of null.

Error 2:

Server Error in '/' Application.

Could not find any resources appropriate for the specified culture or the neutral culture. Make sure "AjaxControlToolkit.Properties.Resources.NET4.resources" was correctly embedded or linked into assembly "AjaxControlToolkit" at compile time, or that all the satellite assemblies required are loadable and fully signed.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Resources.MissingManifestResourceException: Could not find any resources appropriate for the specified culture or the neutral culture. Make sure "AjaxControlToolkit.Properties.Resources.NET4.resources" was correctly embedded or linked into assembly "AjaxControlToolkit" at compile time, or that all the satellite assemblies required are loadable and fully signed.

Source Error:

```
An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.
```

Stack Trace:

```
[MissingManifestResourceException: Could not find any resources appropriate for the specified culture or the neutral culture.  Make sure "AjaxControlToolkit.Properties
   System.Resources.ManifestBasedResourceGroveler.HandleResourceStreamMissing(String fileName) +311
   System.Resources.ManifestBasedResourceGroveler.GrovelForResourceSet(CultureInfo culture, Dictionary`2 localResourceSets, Boolean tryParents, Boolean createIfNotExi
   System.Resources.ResourceManager.InternalGetResourceSet(CultureInfo requestedCulture, Boolean createIfNotExists, Boolean tryParents, StackCrawlMark& stackMark) +81
   System.Resources.ResourceManager.InternalGetResourceSet(CultureInfo culture, Boolean createIfNotExists, Boolean tryParents) +34
   System.Resources.ResourceManager.GetString(String name, CultureInfo culture) +427
   AjaxControlToolkit.Properties.Resources_NET4.get_E_NoScriptManager() in f:\TeamCity\buildAgent\work\80acd78aa4c25314\Server\AjaxControlToolkit\Properties\Resources
   AjaxControlToolkit.ScriptObjectBuilder.RegisterCssReferences(Control control) in f:\TeamCity\buildAgent\work\80acd78aa4c25314\Server\AjaxControlToolkit\ExtenderBas
   AjaxControlToolkit.ExtenderControlBase.OnLoad(EventArgs e) in f:\TeamCity\buildAgent\work\80acd78aa4c25314\Server\AjaxControlToolkit\ExtenderBase\ExtenderControlBa
   System.Web.UI.Control.LoadRecursive() +54
   System.Web.UI.Control.LoadRecursive() +145
   System.Web.UI.Control.LoadRecursive() +145
   System.Web.UI.Control.LoadRecursive() +145
   System.Web.UI.Control.LoadRecursive() +145
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +772
```

Figure 5.2 Toolkit ScriptManager error

Solution: To use Ajax toolkit, drag ToolkitScriptManager from AjaxToolKit from toolbox.

Error 3:

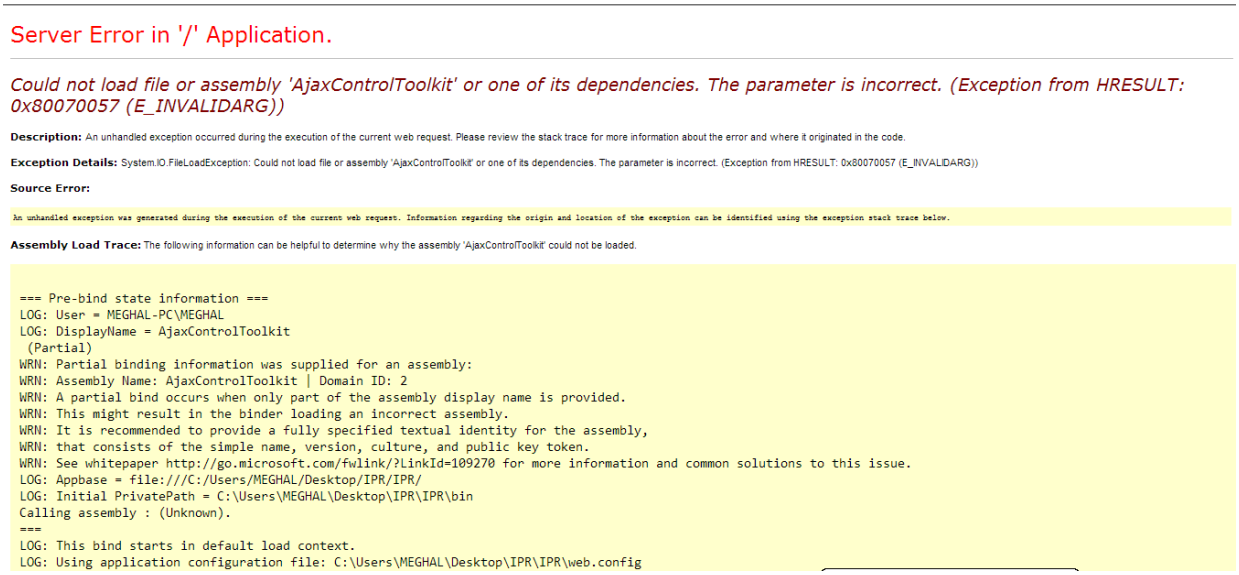


Figure 5.3 Ajax Toolkit error

Solution: To solve it,Drag and drop the .dll file of the ajax toolkit from the bin folder of Ajax toolkit.

Error 4:

When write “localhost” in browser , it gives “Http:404 file not found” error.

Solution: In inet manager , change the port of IIS in binding section.

Error 5:

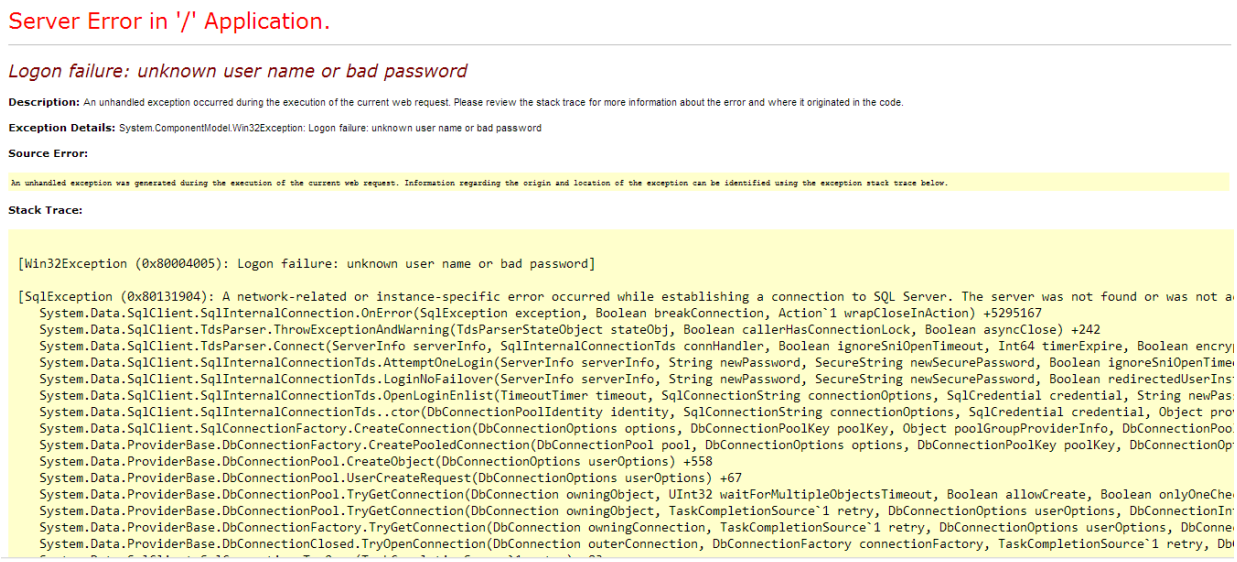


Figure 5.4 MSSQL Service error

Solution: Check your "Data Source" property of connectionstring in web.config file, it must be *your* computer's MSSQL 2012's server name (we are using MSSQL 2012 for database) or you have to start MSSQL service manually if it is not started.

Error 6:

Server Error in '/' Application.

Could not find stored procedure 'dbo.aspnet_CheckSchemaVersion'.

Solution: When we execute sql script of the entire database then we have to run the command given below.

- ➔ Go to this path from command line : "C:\Windows\Microsoft.NET\Framework\v4.0.30319"
- ➔ Run the bellowed command: aspnet_regsql.exe -S DBServerName -U DBLogin -P DBPassword -A all -d DBName

Error 7:

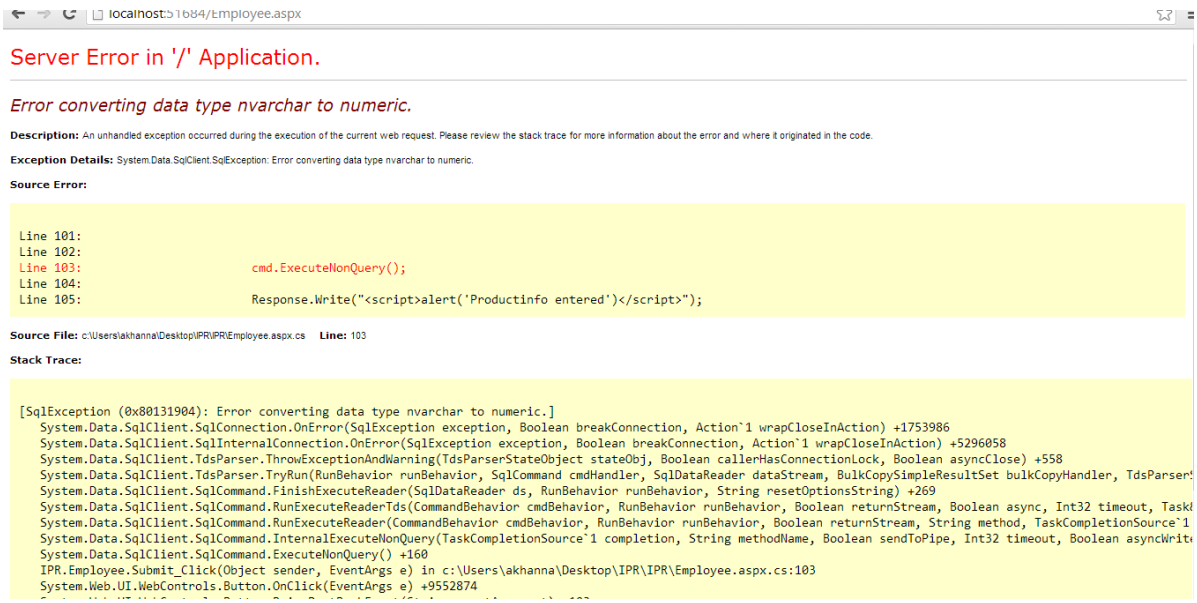


Figure 5.5 Datatype error in SQL Server Manager

Solution: Close and restart the sql server manager.

Error 8: Calendar is not displayed error.

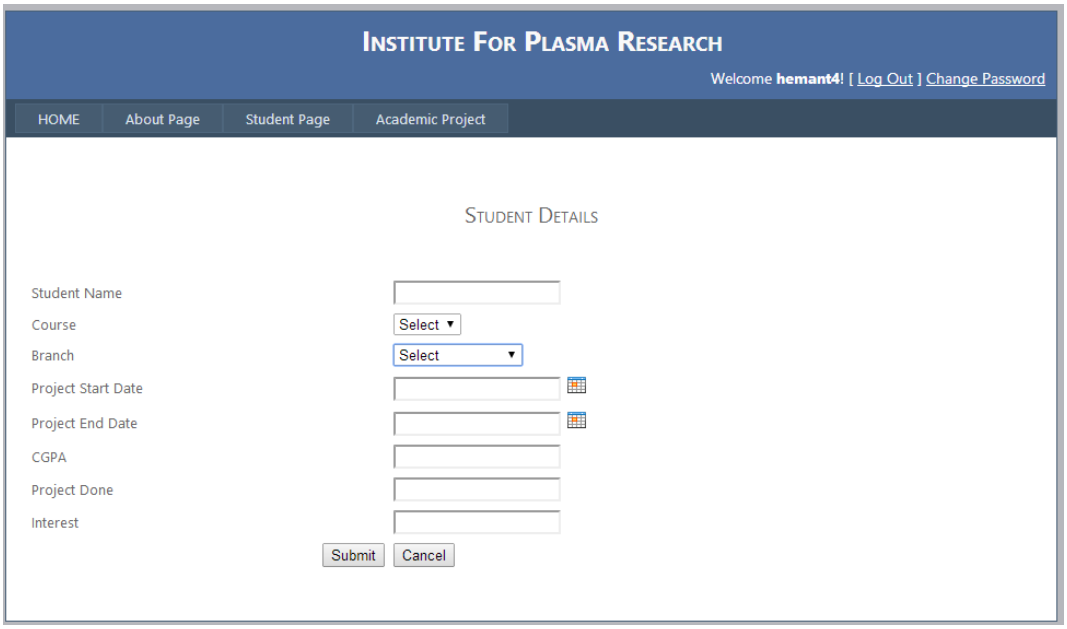


Figure 5.6 Calendar not displayed

Solution: Tags to be added in .aspx page are:

```
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit"
TagPrefix="Ajax" %>

<Ajax:ToolkitScriptManager ID="ToolkitScriptManager1"
runat="server"></Ajax:ToolkitScriptManager>
```

Error 9: Menu is hidden behind the report viewer:

Solution: Put a z-index property in the site.css class under

```
div.menu ul:
{
zindex:1;
}
```

And for report

```
.report
{
z-index:0;
}
```

Chapter 6
Testing

6.1 INTRODUCTION

Testing is the process carried out on software to detect the differences between its behaviour and the desired behaviour as stipulated by the requirements specifications.

Testing is advantageous in several ways:

Firstly, testing gives an idea as to how reliable the software is.

Secondly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

6.2 TEST PLAN:

The testing sub-process includes the following activities in a phase dependent manner:

a) Create Test Plans.

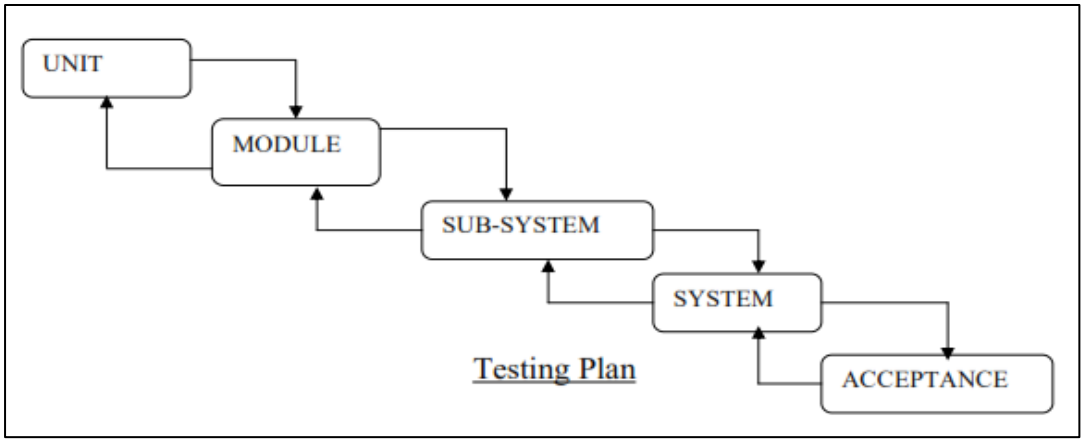
b) Create Test Specifications.

c) Review Test Plans and Test Specifications.

d) Conduct tests according to the Test Specifications, and log the defects.

e) Fix defects, if any.

f) When defects are fixed continue from activity.



6.3 TEST STRATEGY:

A test strategy is an outline that describes the testing approach of the software development life cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment.

Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of test are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used and occasionally, conceptual design documents may be referred to. Design documents describe the functionality of the software to be enabled in the upcoming release.

6.4 TESTING METHODS:

6.4.1 Unit Testing:

Objective:

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

Testing Process:

- Checking for availability of Code Walk-through reports which have documented the existence of and conformance to coding standards.
- Review of Unit Test Specifications.
- Verify the Unit Test Specifications confirm to the program specifications.
- Verify that all boundary and null data conditions are included.

6.5 Test Cases

Table 6.1 Test Cases

Sr. No.	Test Case Name	Test Procedure	Pre-Condition	Expected Result	Output
1.	Login	Enter the valid user name and password and click the login button	Start the Browser & enter the appropriate URL & users should've been created using "CreateUserWizard".	Display the Home page.	Success
		Enter incorrect user name / password	Start the Browser and enter the appropriate URL.	Error message displayed: "Unsuccessful login attempt"	Success
2.	Check User type and rights	Application menus should be visible depending on the type of logged in user	Successful login.	Menus are visible/invisible depending on the type of the user	Success
3.	Password Change	User clicks on Change Password link	Successful login.	Users should enter current and new passwords and also the confirmation of new password should be done	Success
4.	Log Out	User clicks on Log Out link	Successful login.	User should be directed to Login Page	Success
5.	Internal Request	Specify Employee Requirements for the project(s)	Employees' Occupational details required on the page must be entered first in the database.	Message "Request has been submitted" Should appear	Success
6.	Internal Request Report Generation	User may or may not apply filters to view report	Internal Project Request details should be present in the database and Login credentials must be of Project coordinator's	Internal request reports generated based on the applied filters	Success

Sr. No.	Test Case Name	Test Procedure	Pre-Condition	Expected Result	Output
7.	External Request	Specify student(s)/intern(s) basic information	Contact Person Details must be entered and Students' college/institute must be registered.	Message "Request has been submitted" Should appear	Success
8.	External Request Report Generation	User may or may not apply filters to view report	External Request details should be present in the database and Login credentials must be of Project coordinator's	External request reports generated based on the applied filters	Success
9.	Allocation of project request	Enter internal and corresponding external id	Login credentials must be of Project coordinator's and Both internal and external request id must be valid	All allocations made to the specified internal request id are displayed.	Success
10.	Template Generation	Enter internal request id	Login credentials must be of Project coordinator's and internal request id must be valid	Printable reports in the form of .docx file of all allocations made to the specified internal request are displayed with details.	Success
11.	Creation of Roles	Enter the name of the new role to be created	User must be logged in as Admin and the role entered should not already exist	The role is successfully created	Success
12.	Assigning/ Removing Roles from a User	Select a Username and assign new roles or remove existing roles of that user.	User should be logged in as Admin.	Role(s) successfully assigned/ removed from a user	Success
13.	Assigning/Removing Users from a Role	Select a Rolename and add or delete Users of that role.	User should be logged in as Admin.	User(s) successfully added/removed from that role.	Success
14.	Course and Branch	Enter a new course/branch	User should be logged in as	Course/Branch name is	Success

Sr. No.	Test Case Name	Test Procedure	Pre-Condition	Expected Result	Output
	Management	name to be entered into the database	Admin.	successfully added to the database	
15.	Institute Registration	Enter the details required in the “Institute Detail” form	Can only be added by the Admin	Institute is successfully registered with IPR	Success

Chapter 7
Screenshots

7.1 Login Page

Figure 7.1 Login Page

7.2 Home Page

Figure 7.2 Home Page

7.3 Academic Master Menu

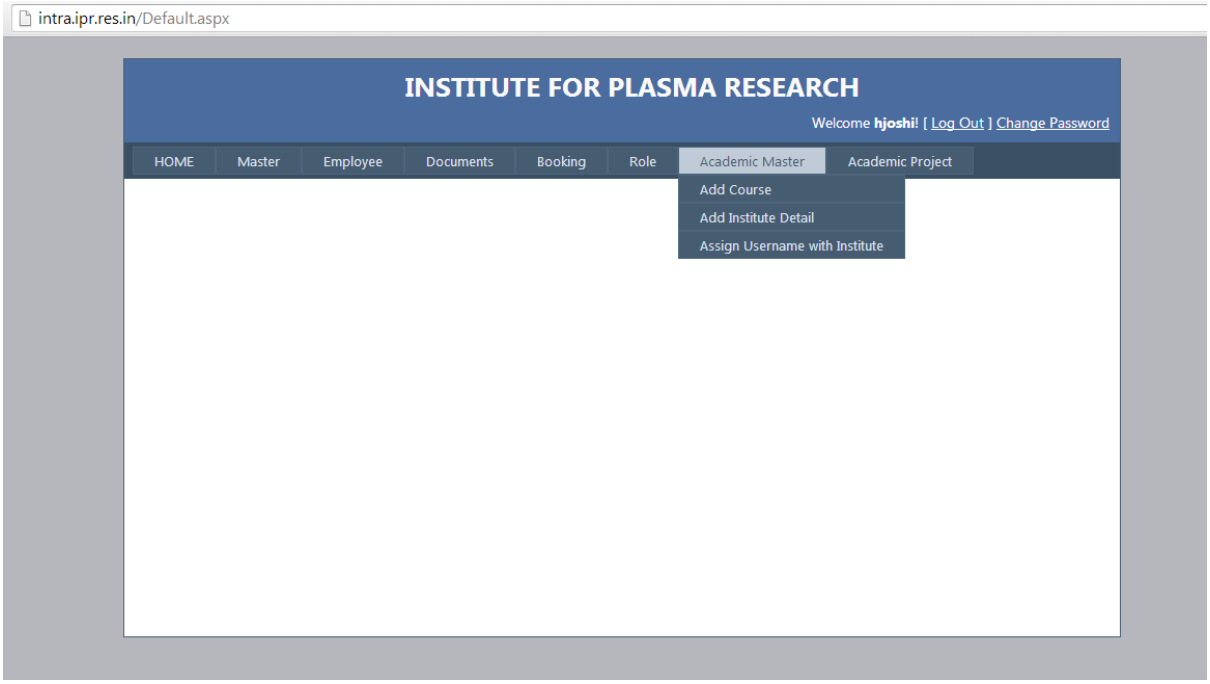


Figure 7.3 Menu for Admin

7.4 Add Course & Branch Page

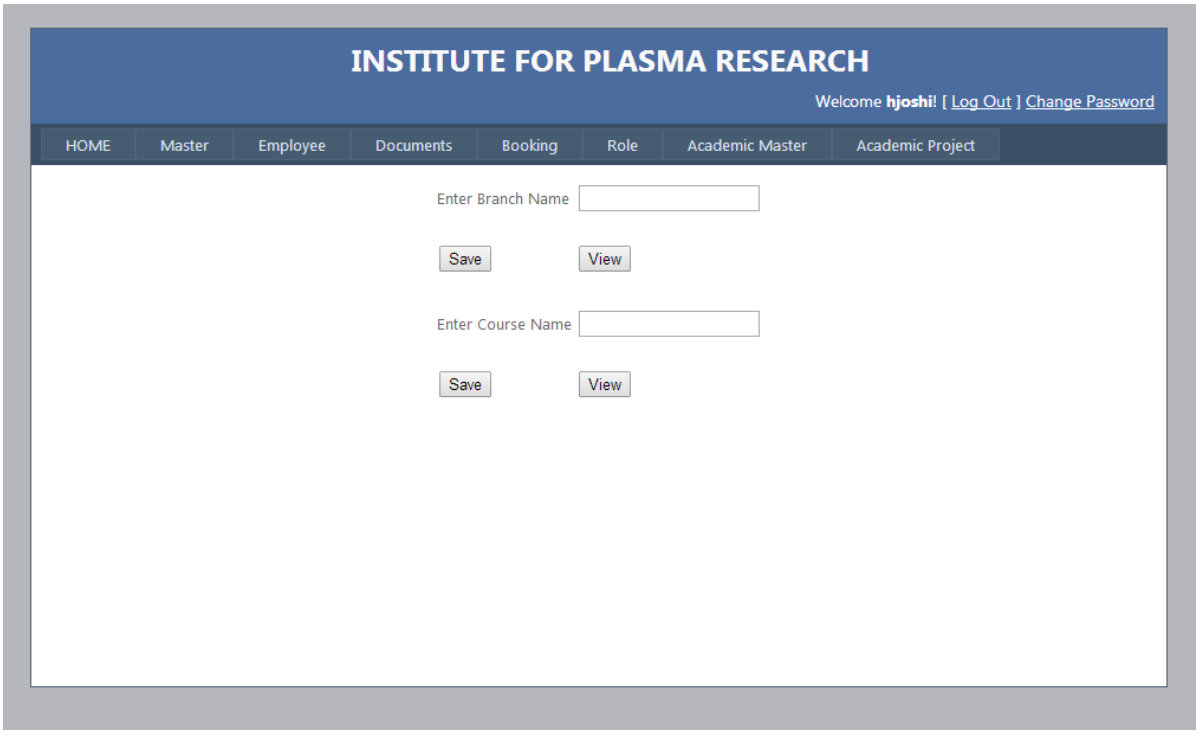


Figure 7.4 Page to add new Course and Branch to the database

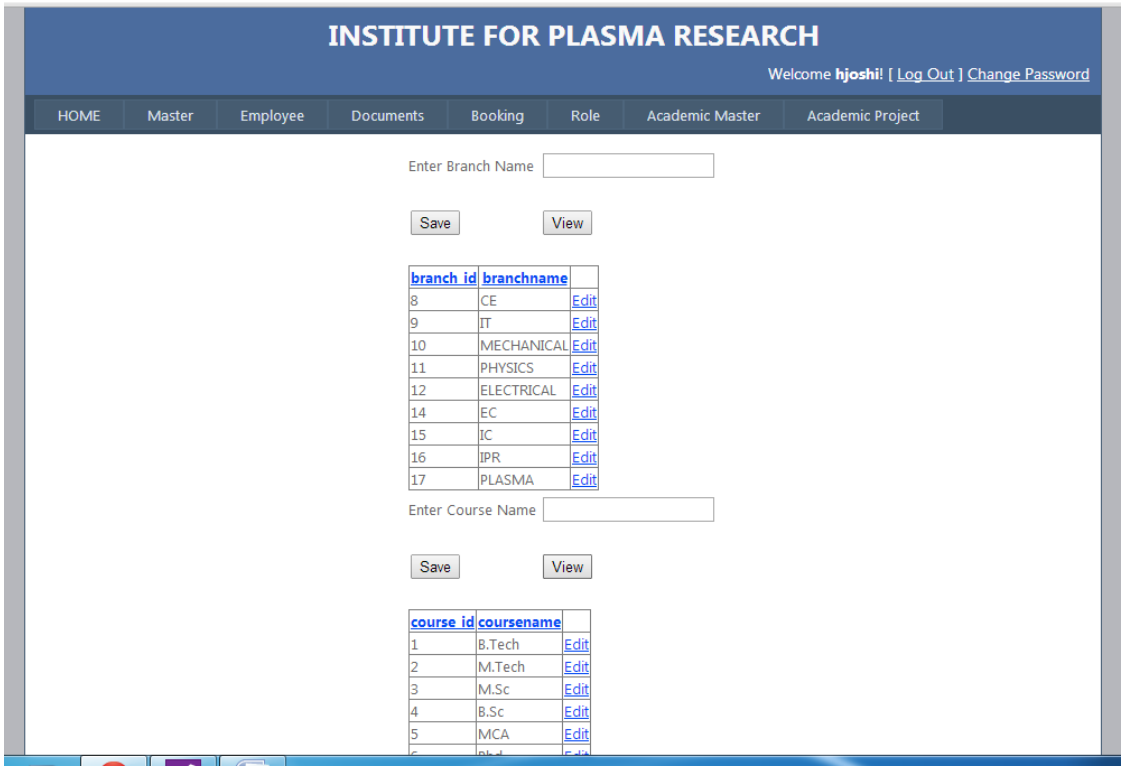


Figure 7.5 Displaying the existing and newly added courses & branch in the database

7.5 Institute Registration

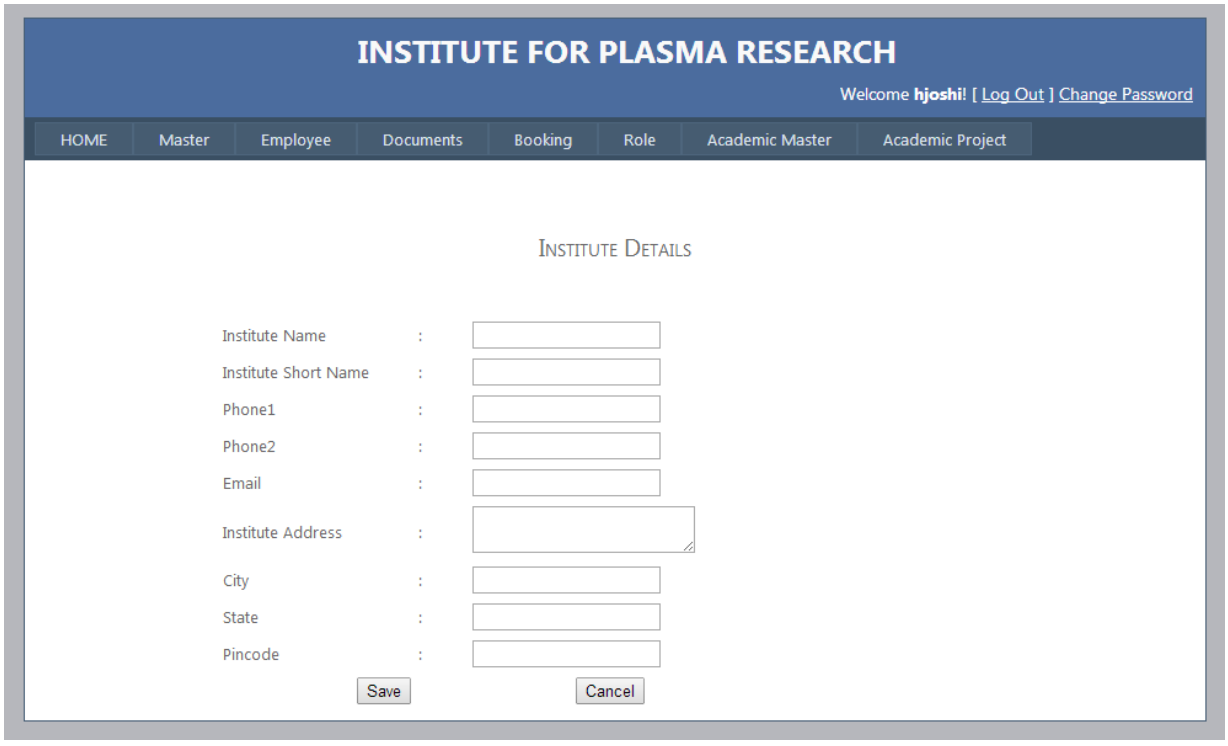


Figure 7.6 Form for entering Institute Details for its registration

7.6 Assign Username to Institute

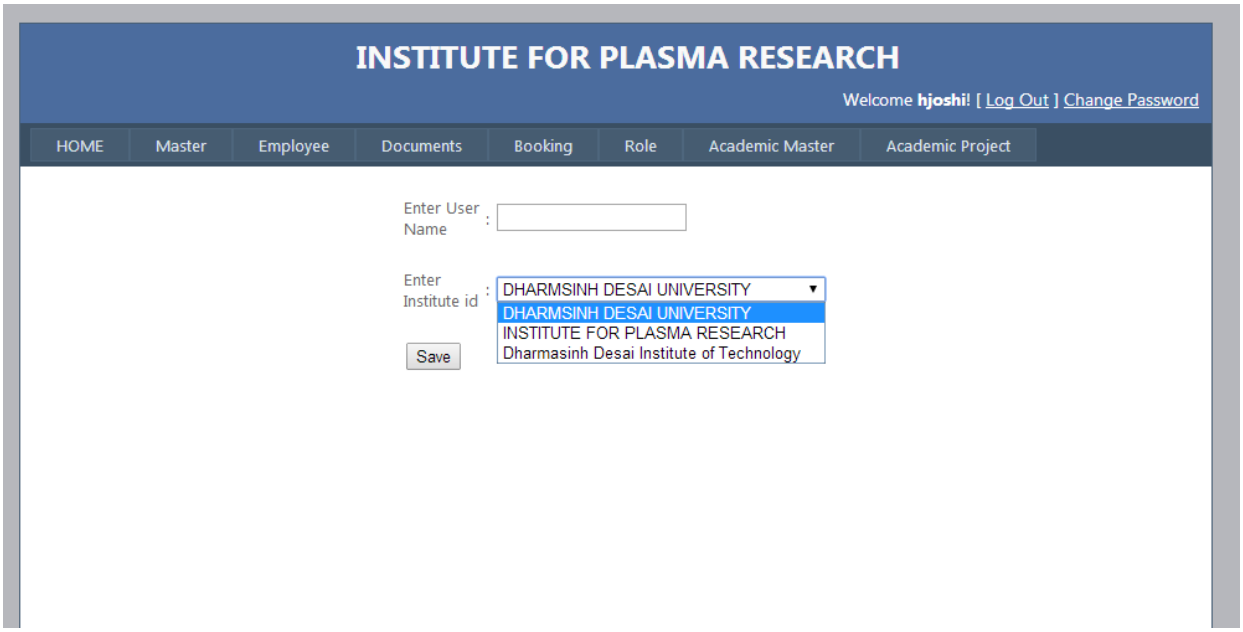


Figure 7.7 College Username is mapped to a registered Institute name

7.7 College Menu

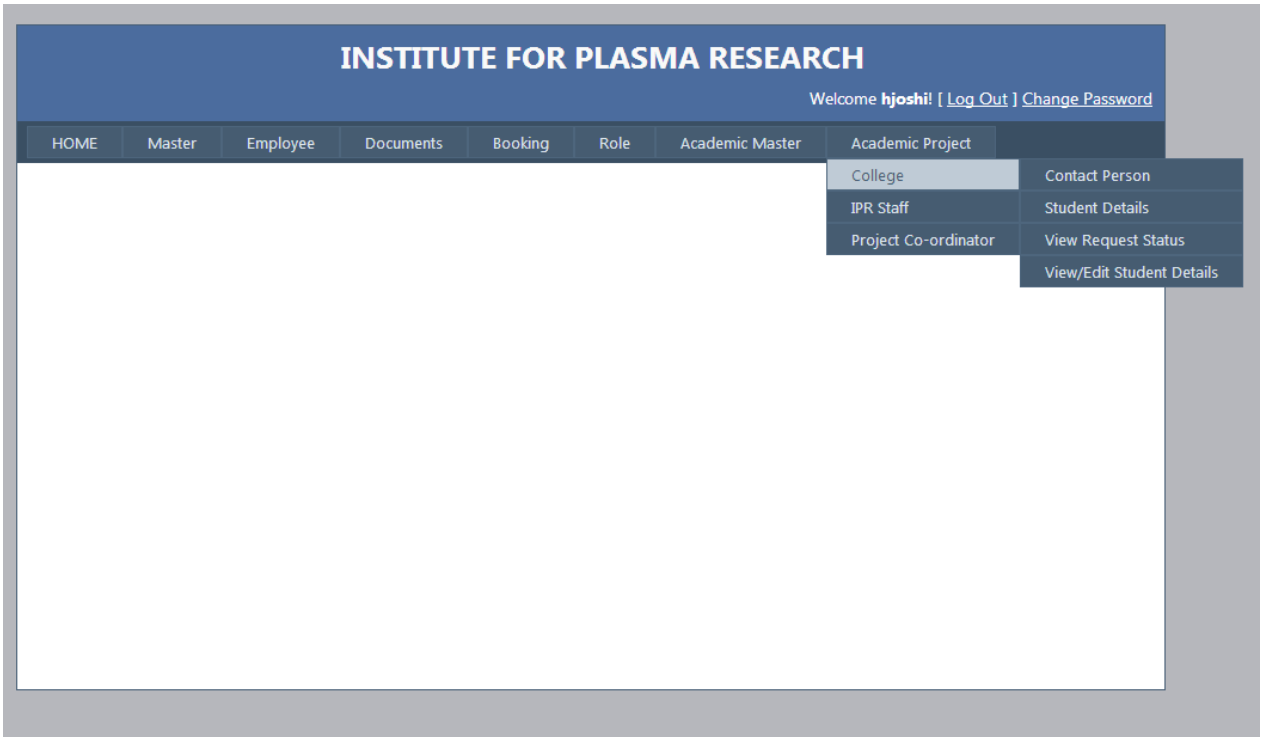


Figure 7.8 Menu displayed to the College User

7.8 Contact Person Registration

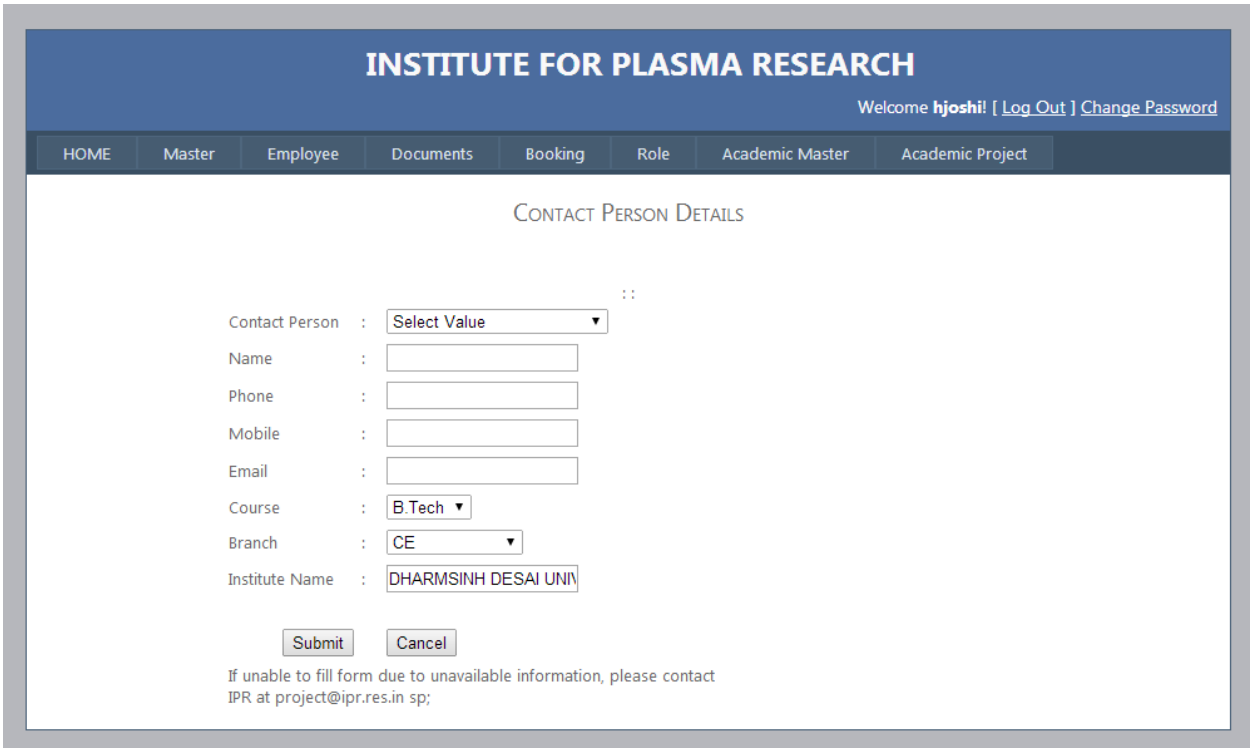


Figure 7.8 Form to enter details of Contact Person

7.9 External Request Form

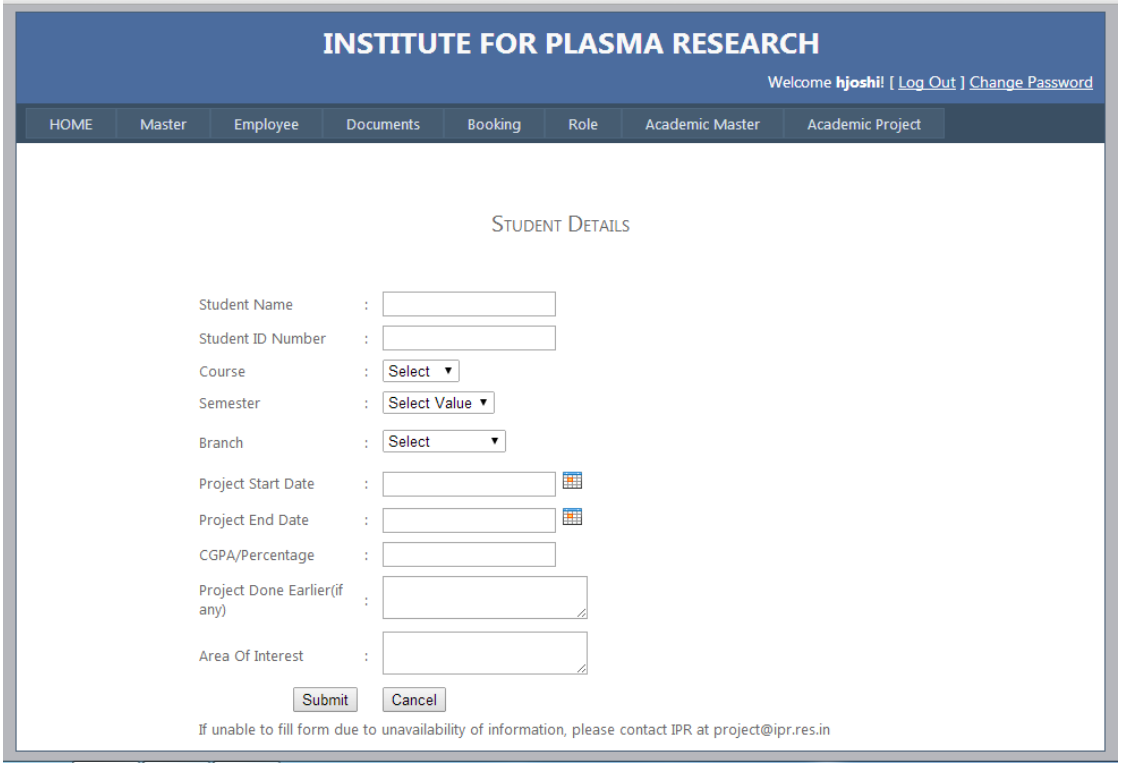


Figure 7.9 Form to enter student details for project training(to be filled by contact person)

7.10 View Request Status

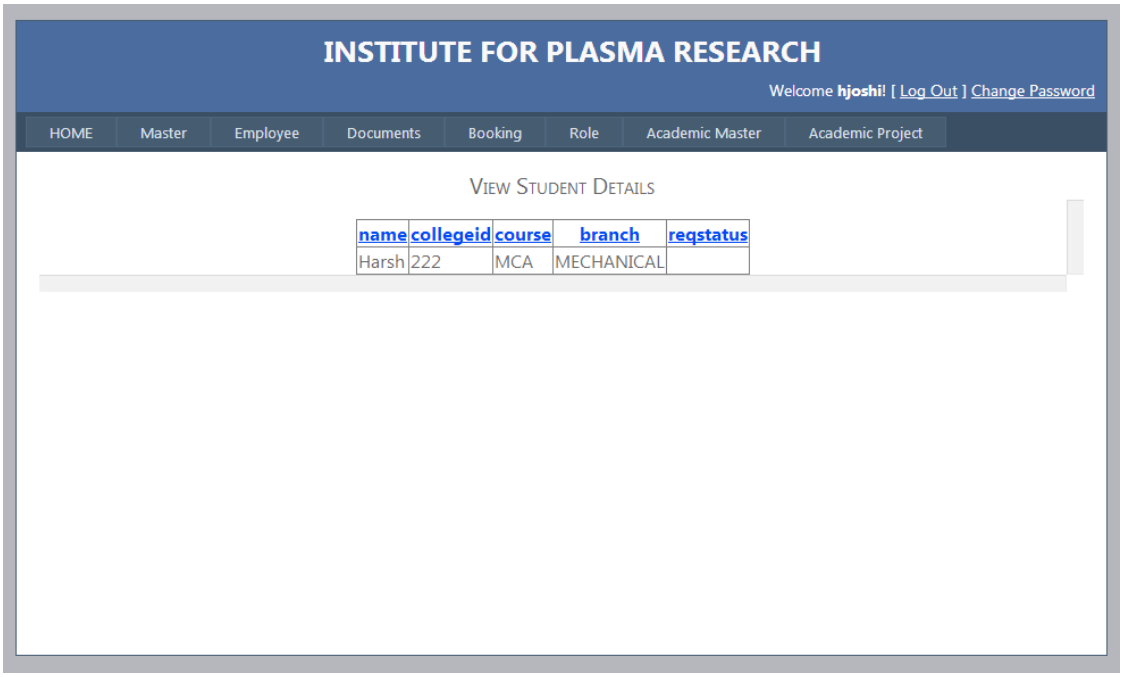


Figure 7.10 Existing Student Details displayed ("request status"- future extension)

7.11 View & Edit the Student Details

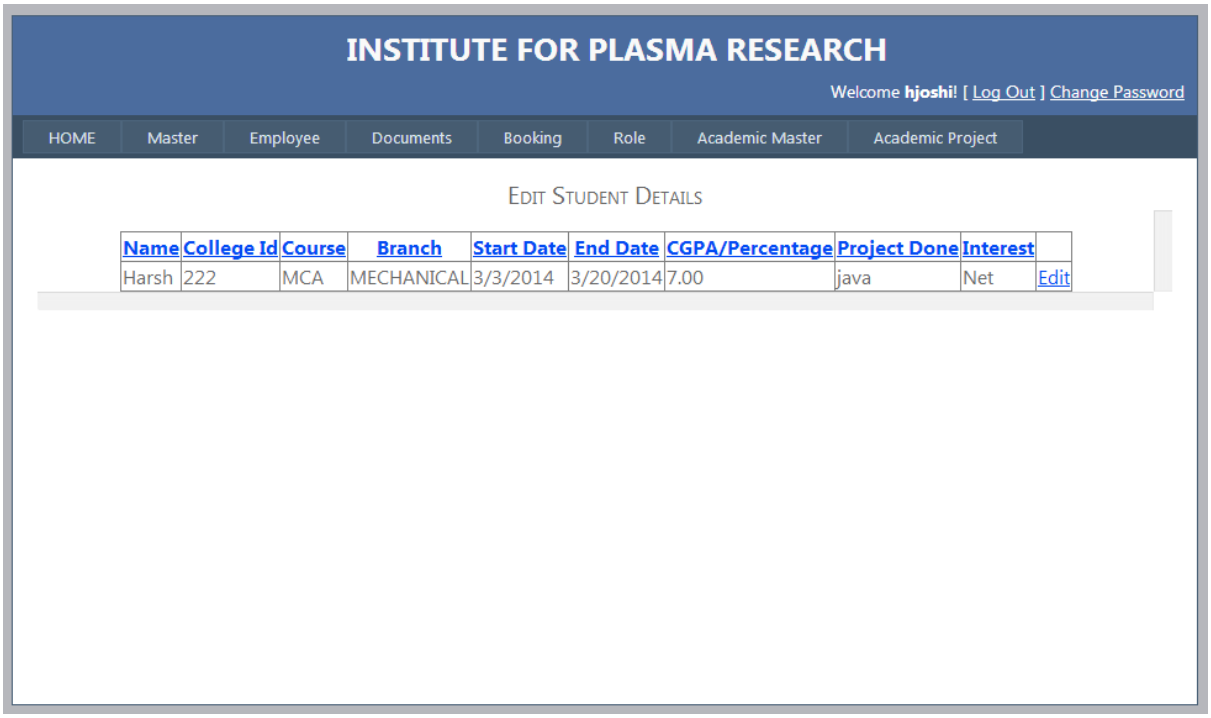


Figure 7.11 Edit submitted details of students (to be performed by Contact Person)

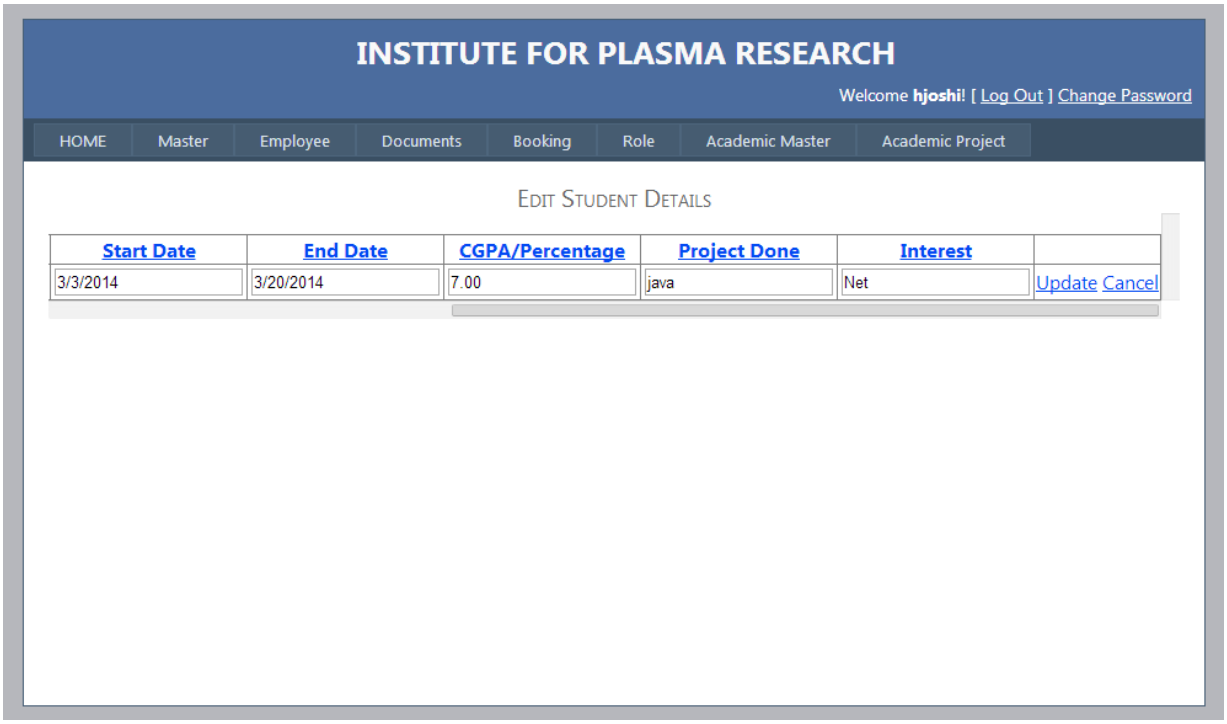


Figure 7.12 Edit submitted details of students (to be performed by Contact Person)

7.12 IPR Staff Menu

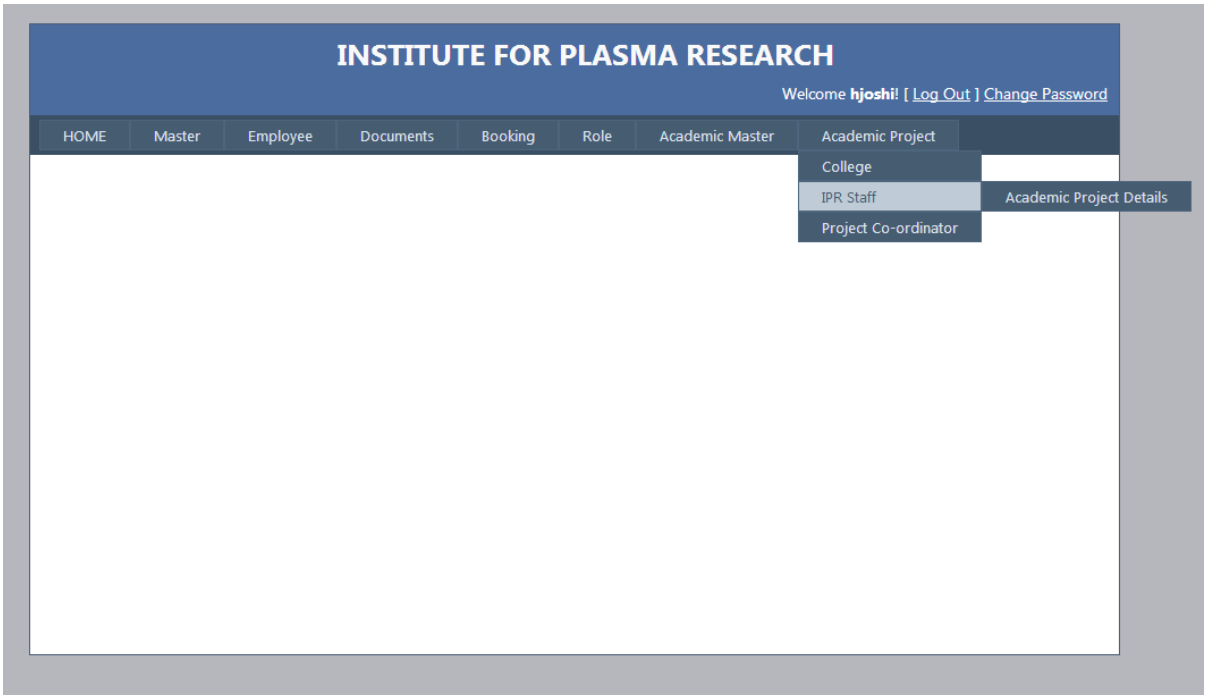


Figure 7.13 Menu displayed to IPR Staff

7.13 IPR Staff Request Form

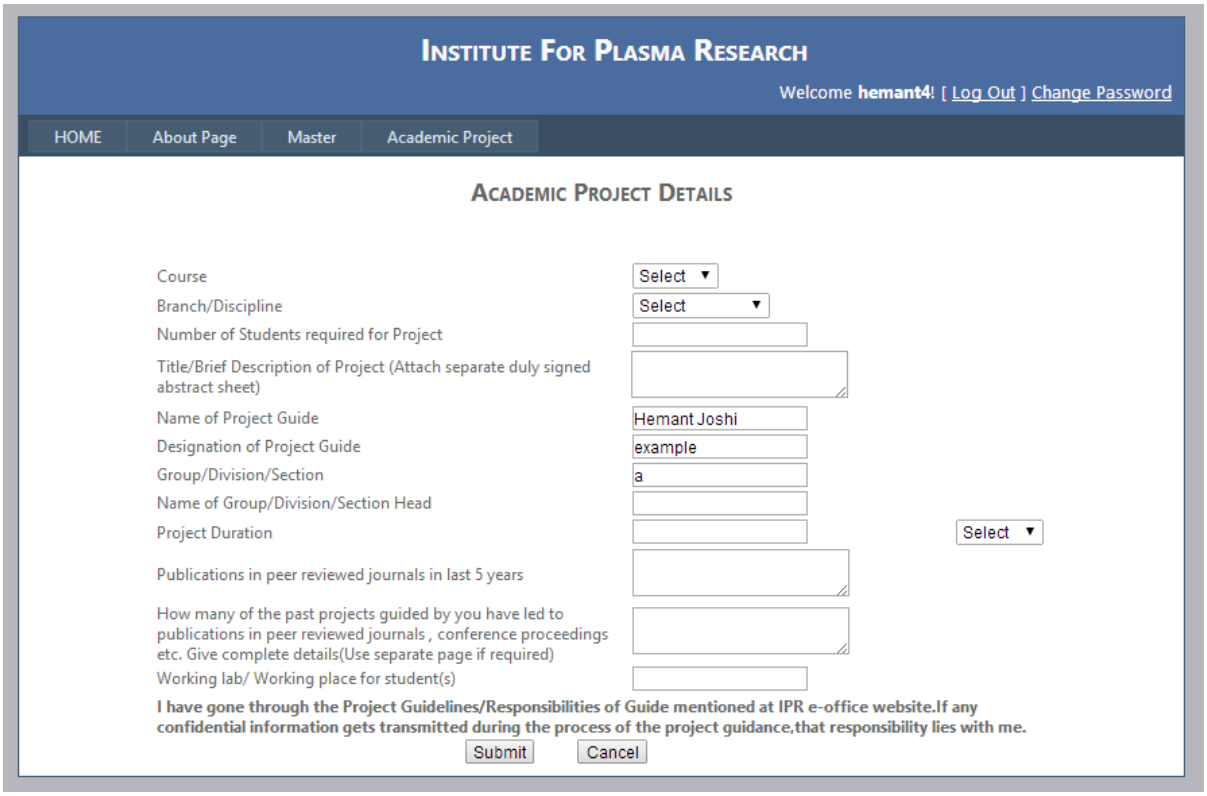


Figure 7.14 Form to entire Project Student Requirements (to be filled by IPR Staff)

7.14 Project Co-ordinator Menu

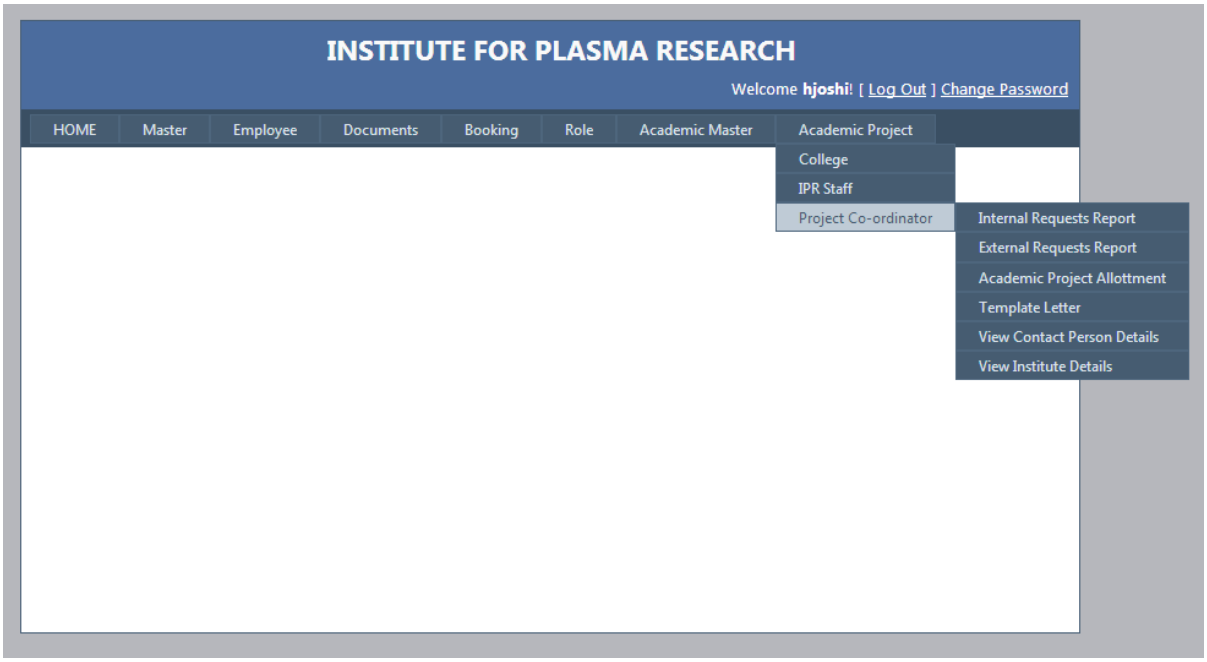


Figure 7.15 Menu displayed to Project Co-ordinator

7.15 Internal Report

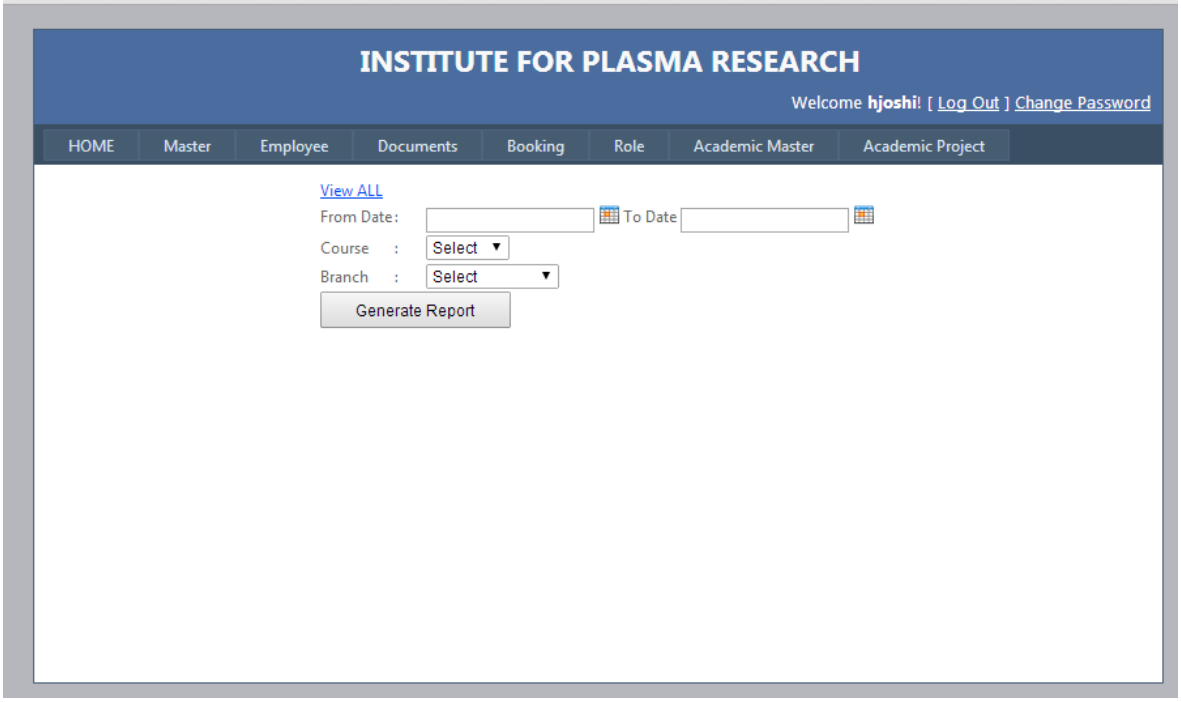


Figure 7.16 Filters available for viewing various Internal Requests Reports

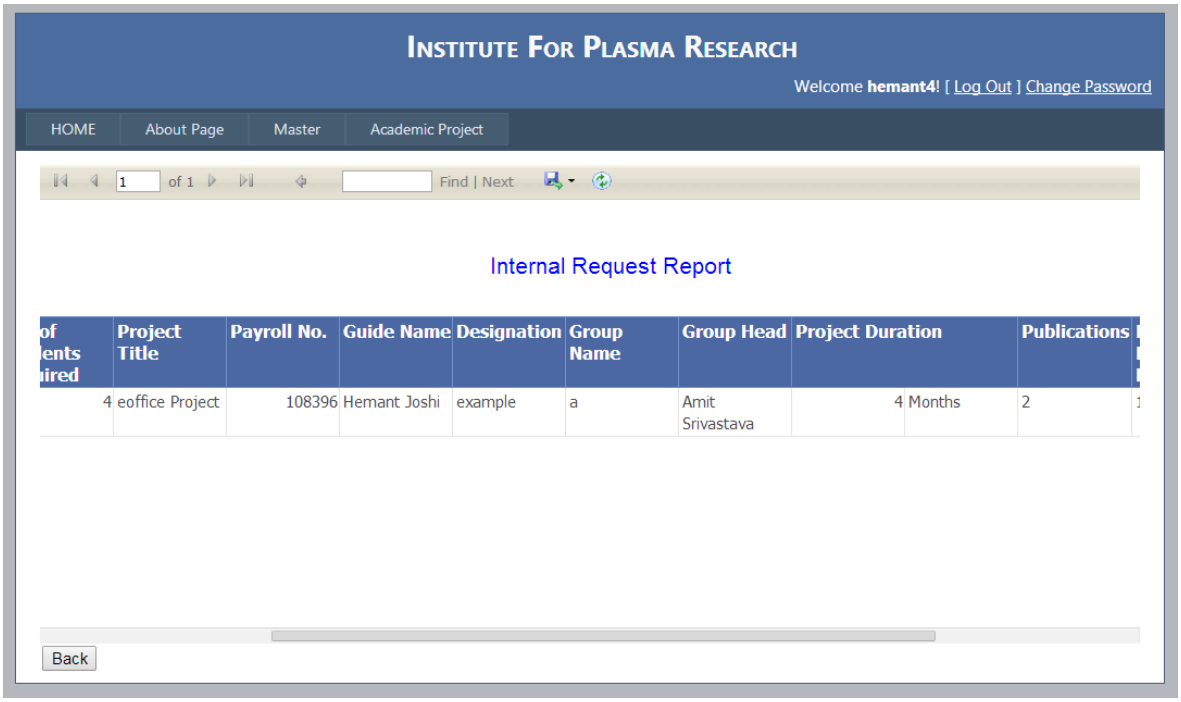


Figure 7.17 Internal Request Report displayed (can be exported to Word, Excel or PDF)

7.16 External Report

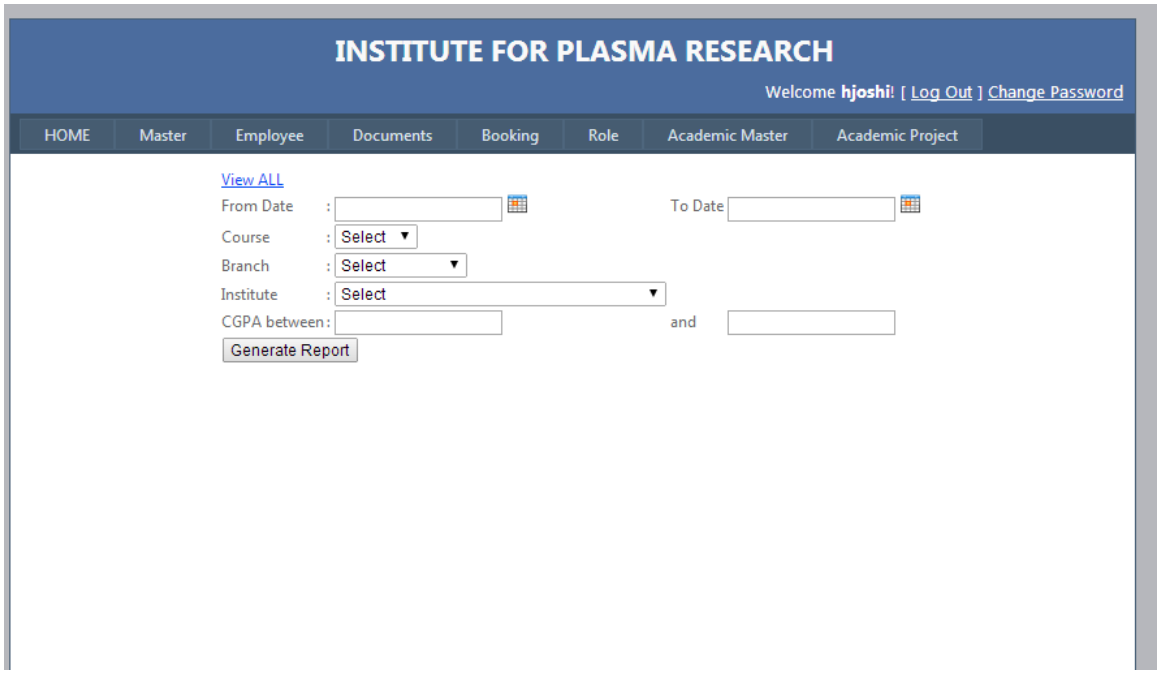


Figure 7.18 Filters available for viewing various External Requests Reports

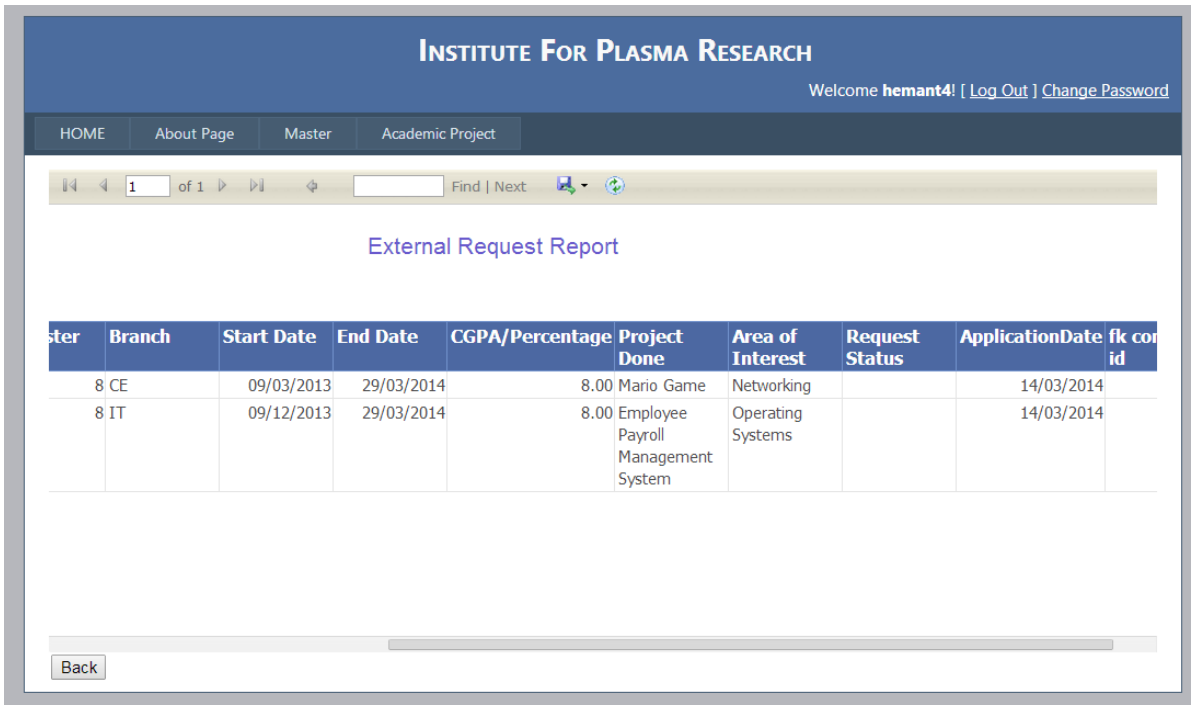


Figure 7.19 External Request Report displayed (can be exported to Word, Excel or PDF)

7.17 Allotment Page

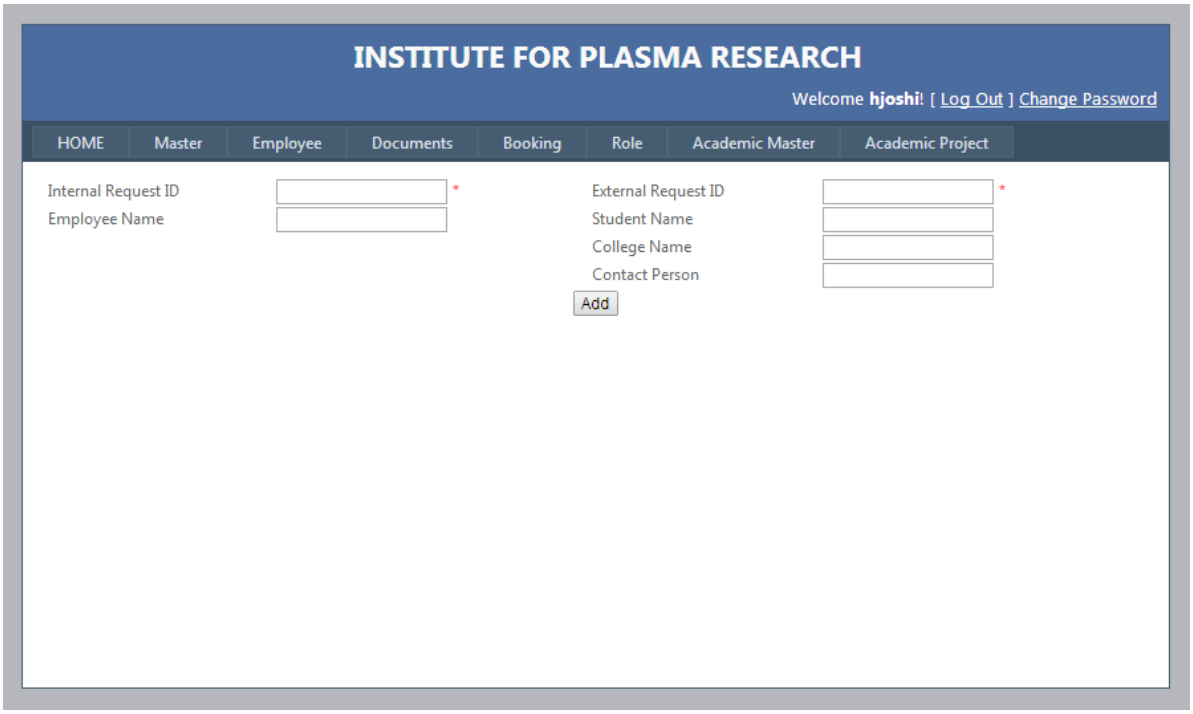


Figure 7.20 Internal Request ID is mapped to External Request ID (done by Project Co-ordinator)

7.18 Template Letter

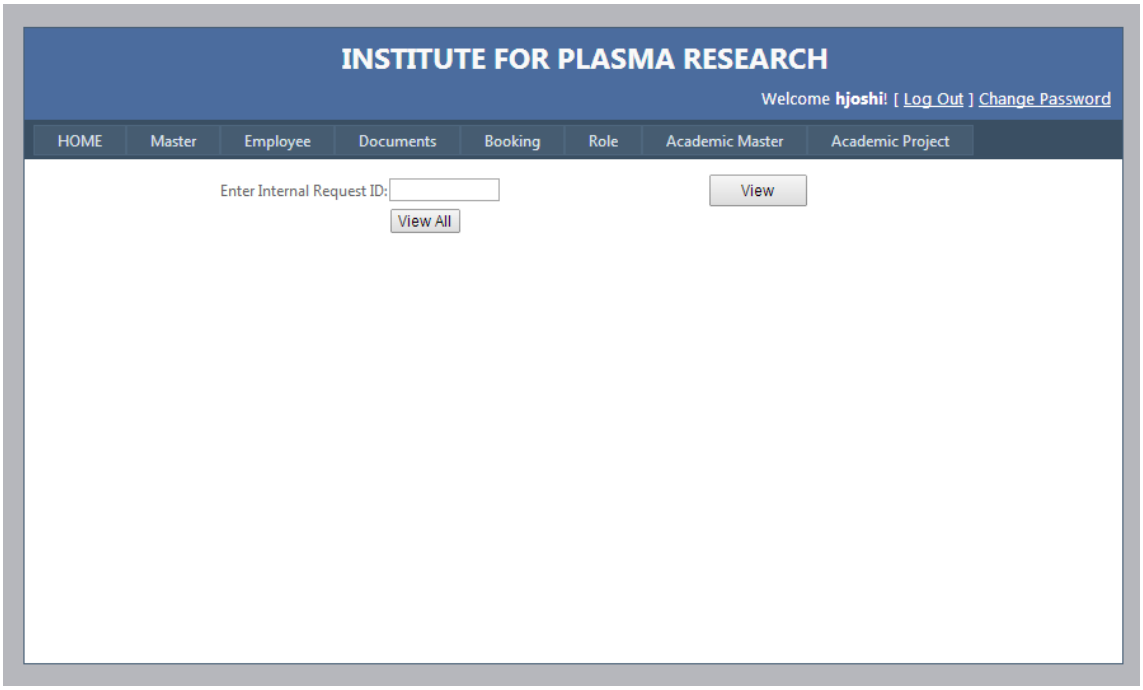


Figure 7.21 Template Letter generation

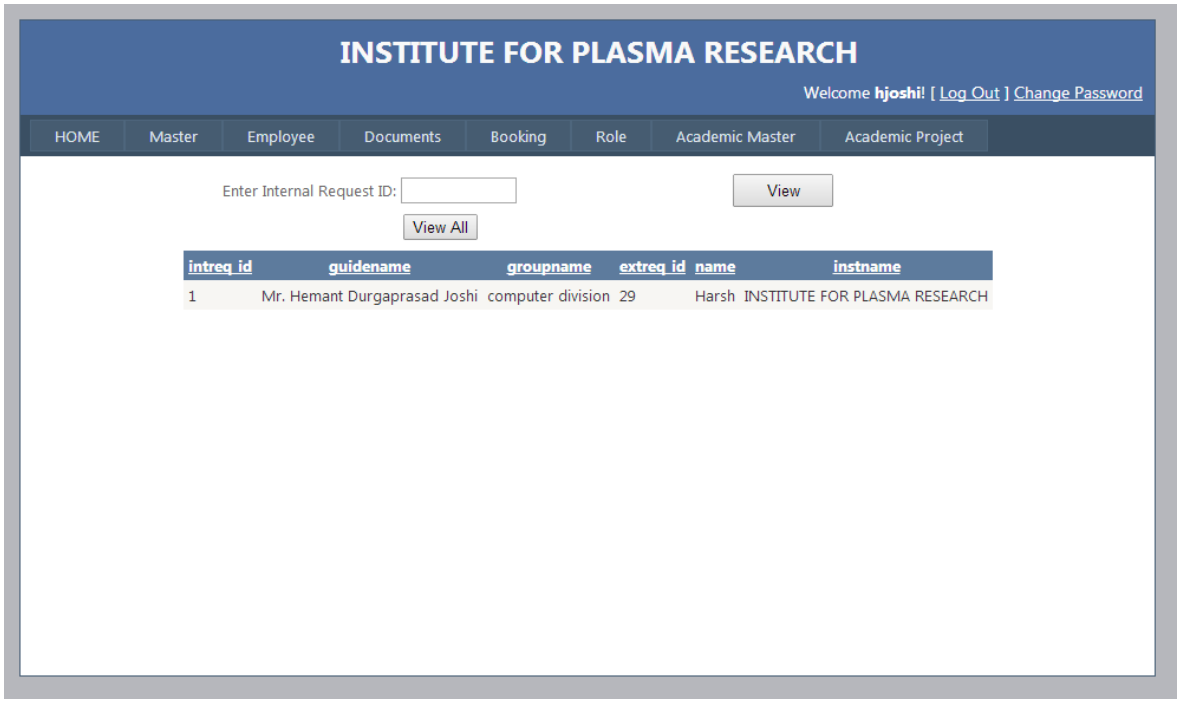


Figure 7.22 Template generated when "View All" is clicked

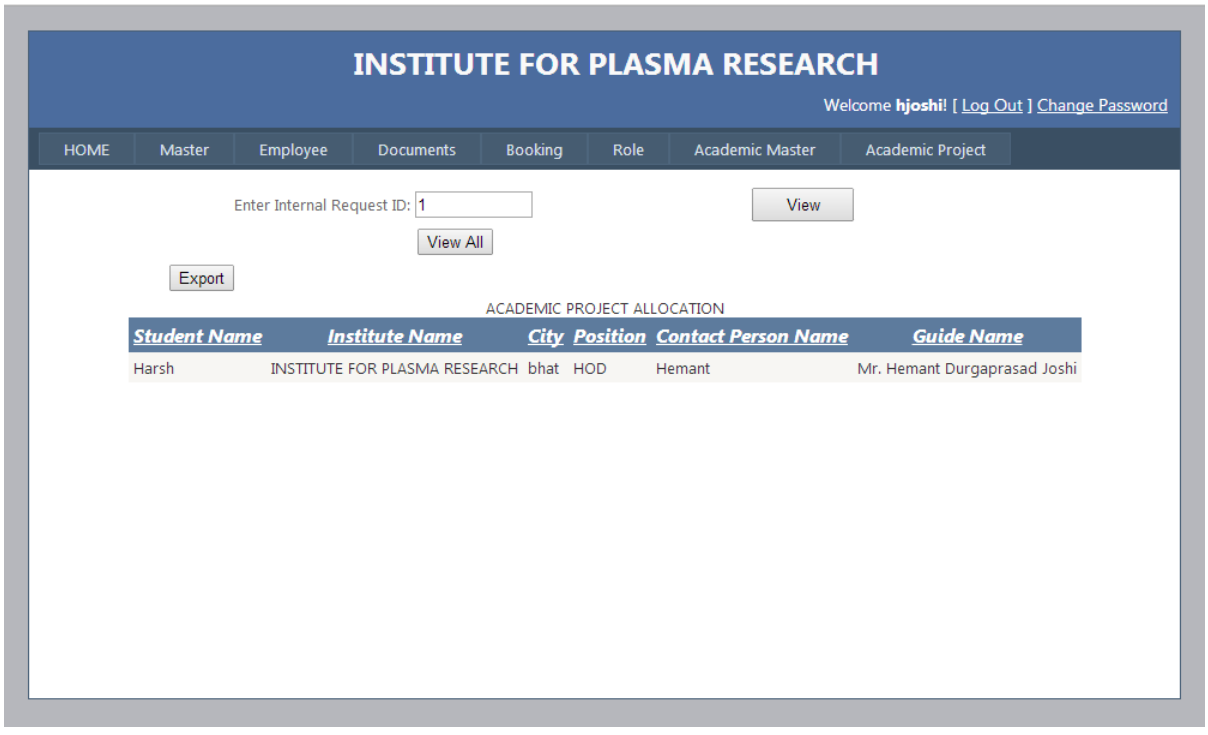


Figure 7.23 Template generated when Internal Request ID is entered and "View" button clicked

7.19 View Contact Person Details

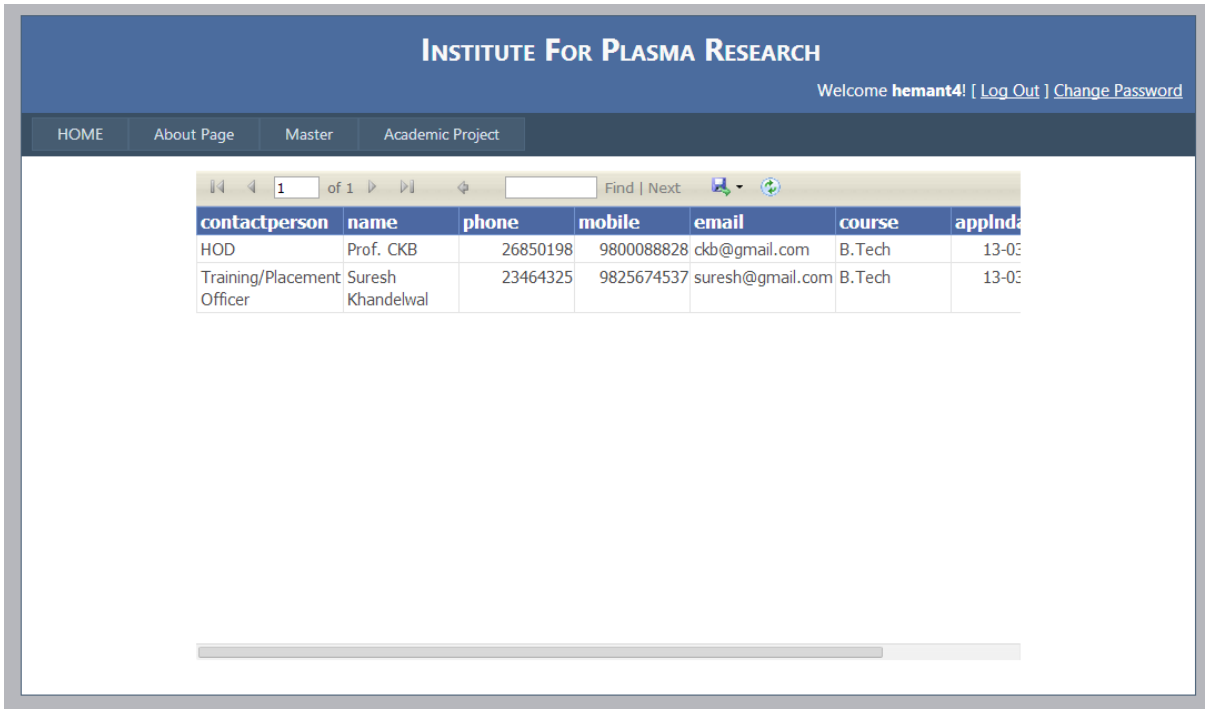


Figure 7.24 All details of Contact Person displayed

7.20 View Institute Detail

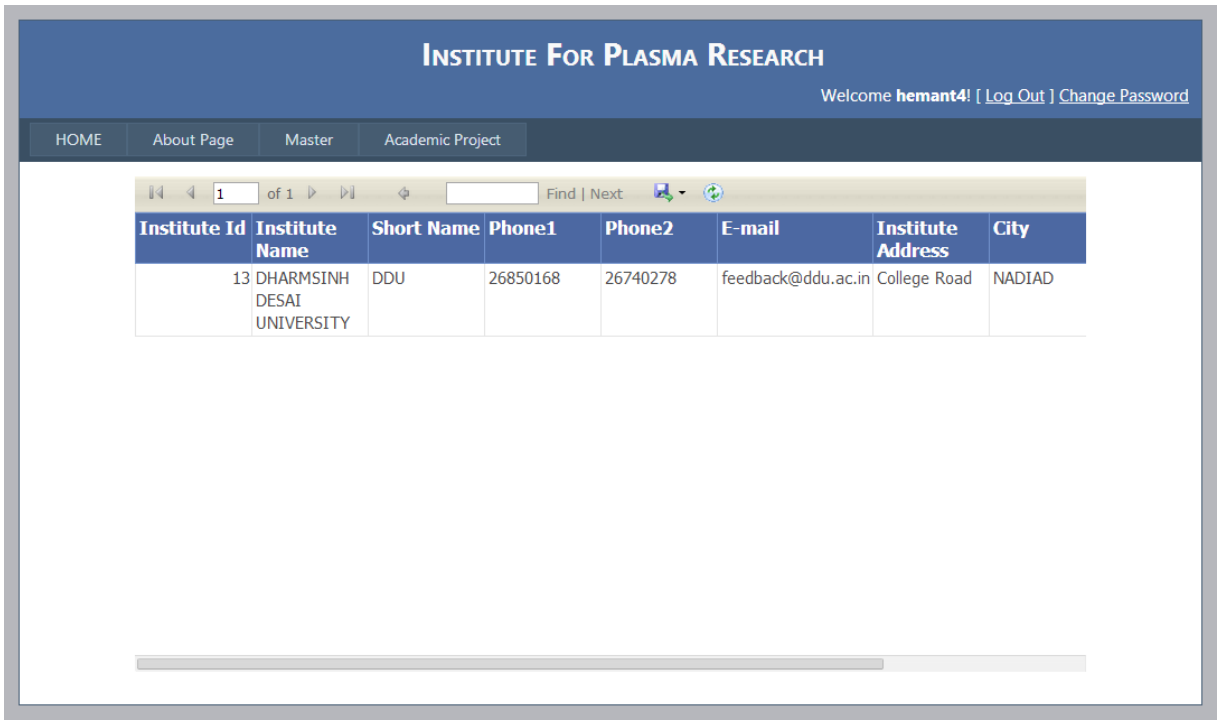


Figure 7.25 All details of registered institutes displayed

7.21 Role Management

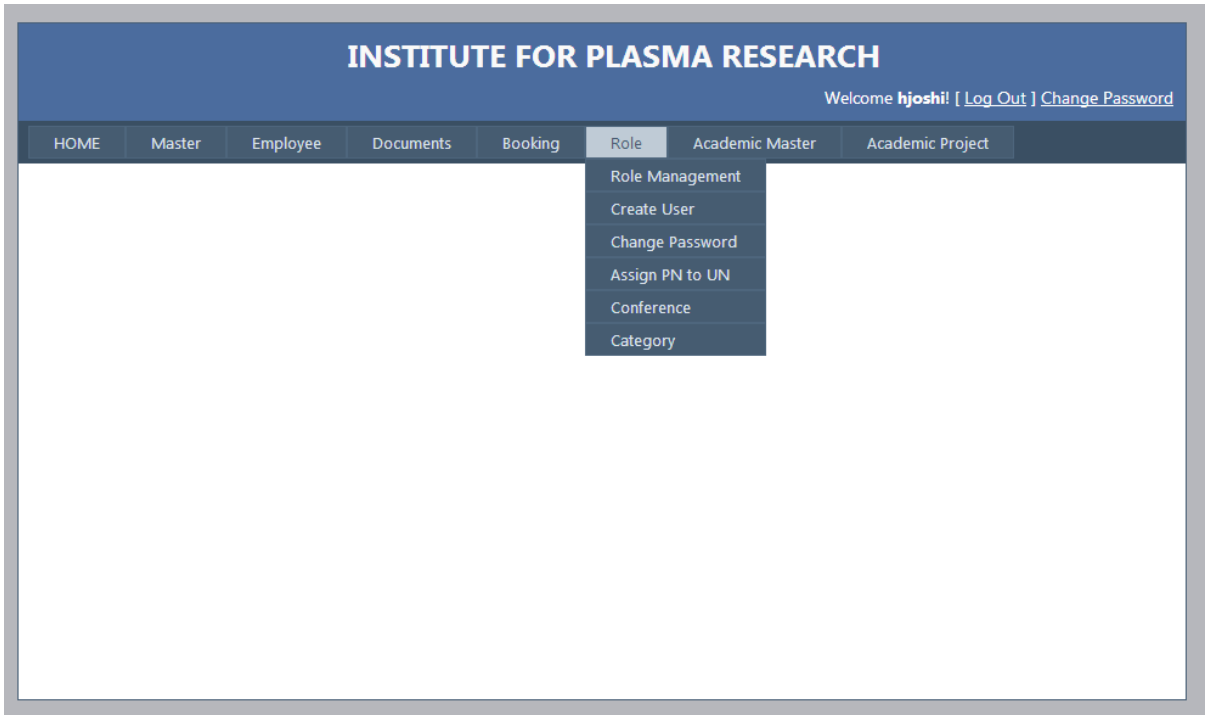


Figure 7.26 Role Management menu

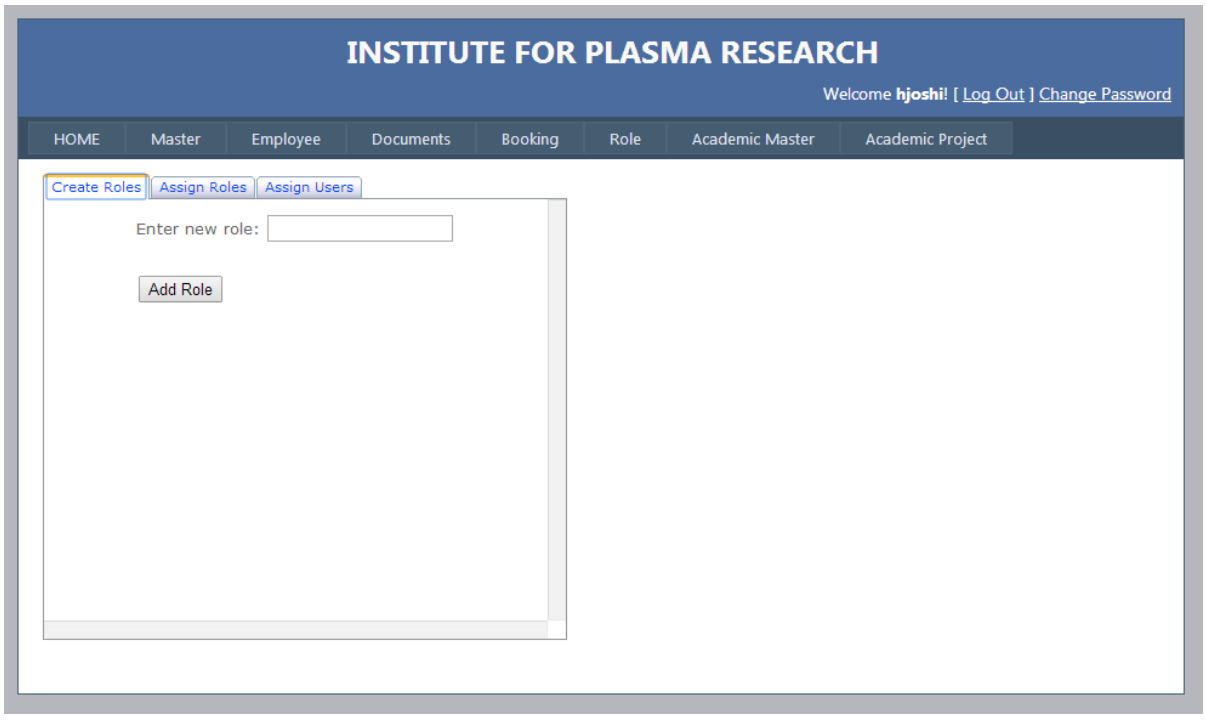


Figure 7.27 Page to add new role (done by Admin)

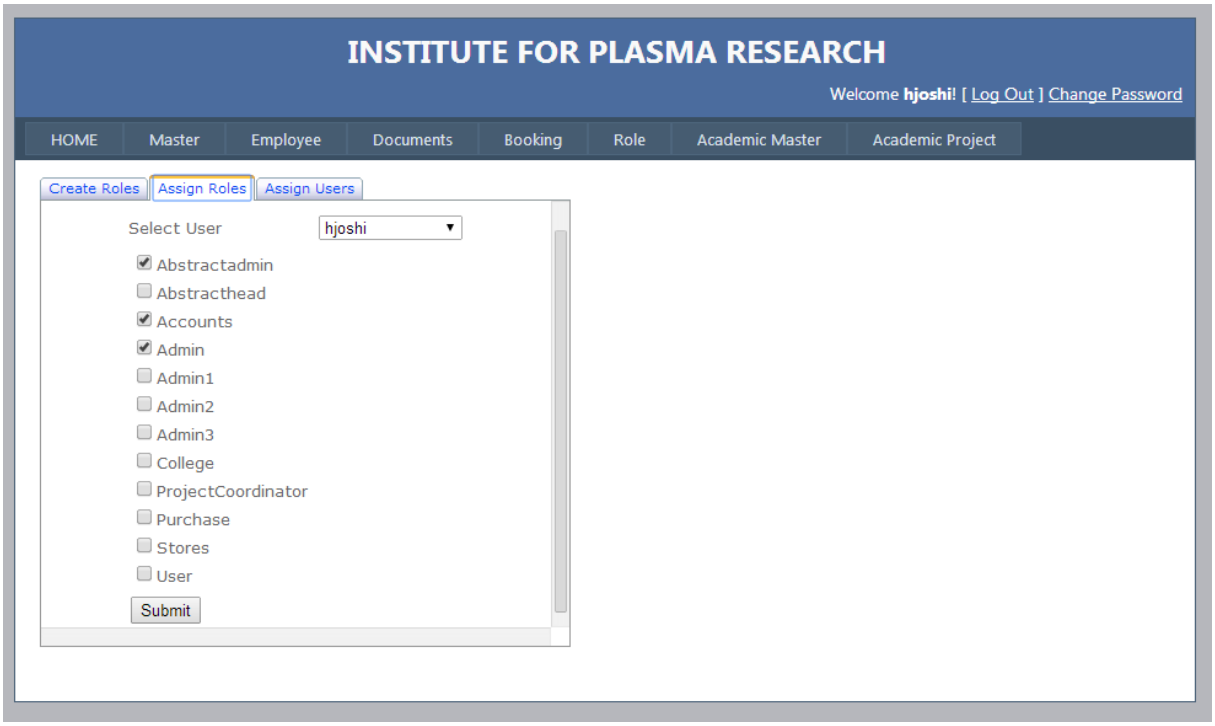


Figure 7.28 Page to assign role(s) to users (done by Admin)

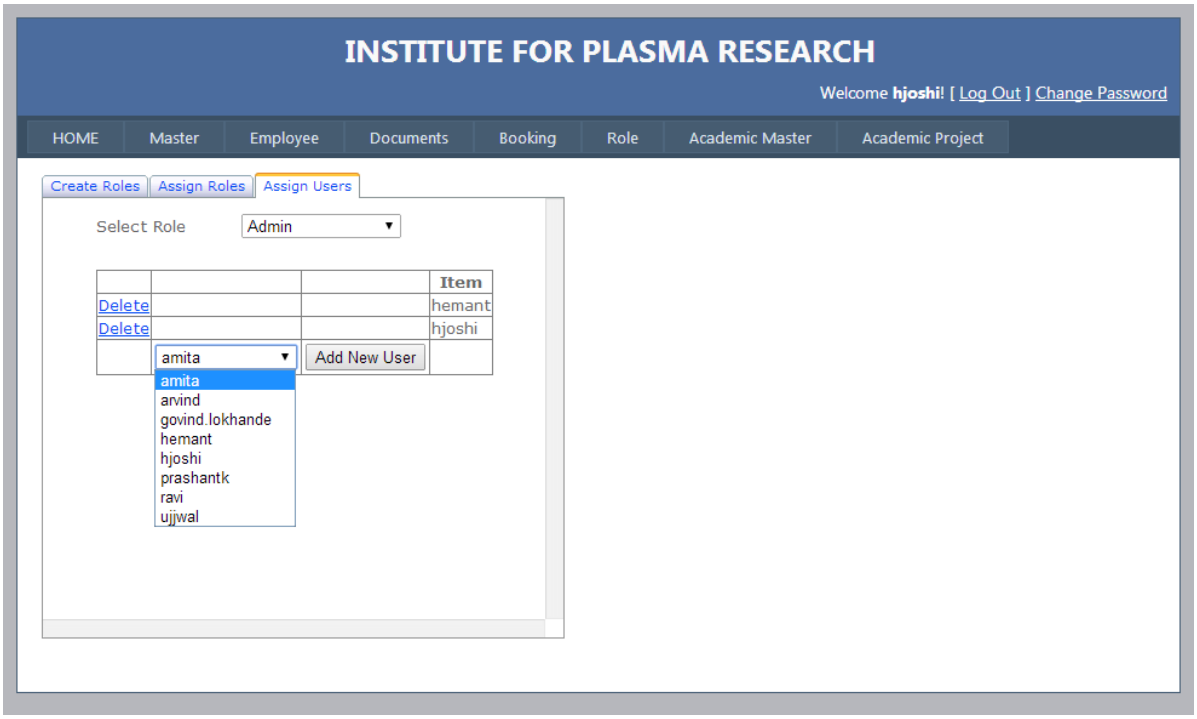


Figure 7.28 Page to assign user(s) to a role (done by Admin)

7.22 Create User Wizard

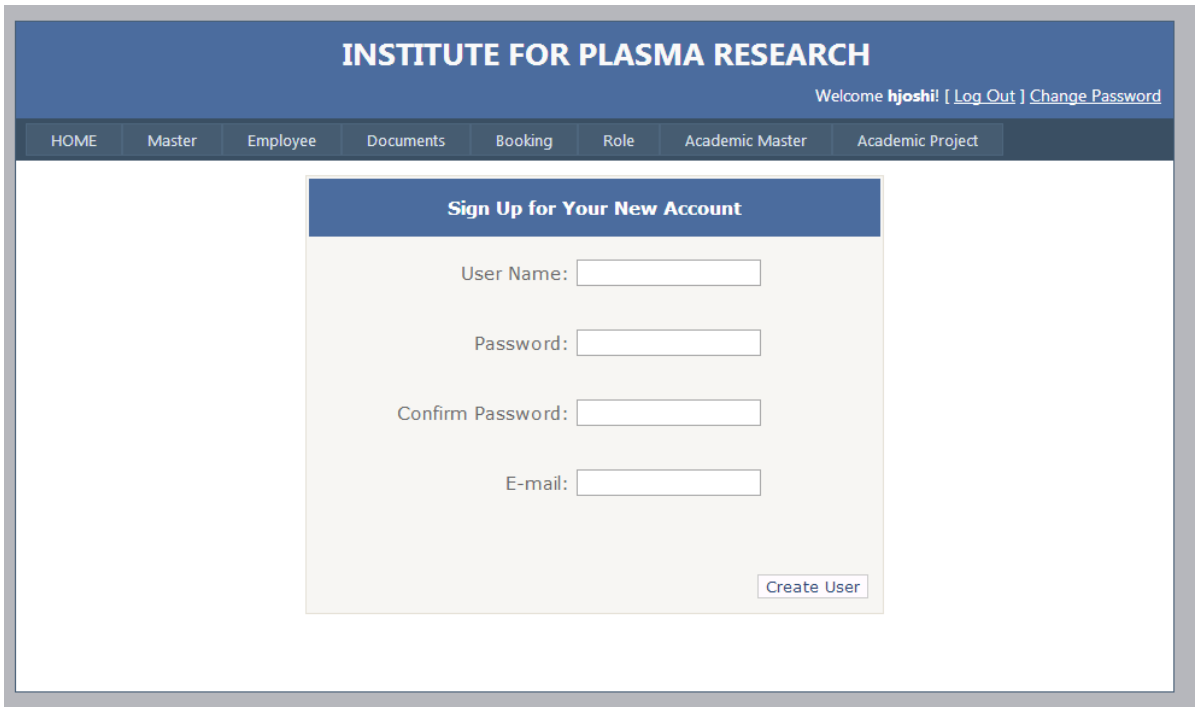


Figure 7.29 Wizard to create new user (done by Admin)

Figure 7.30 Successful User Creation

7.23 Password Change Wizard

Figure 7.31 Change password

7.24 Username assignment to Payroll Number

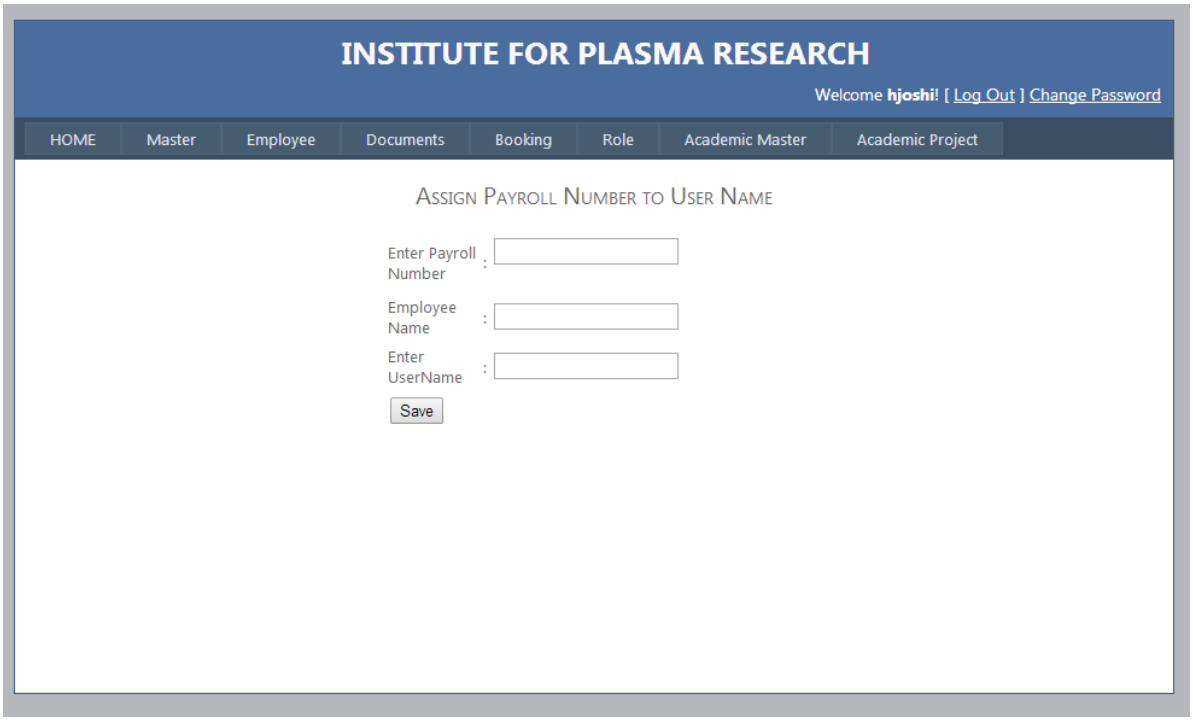


Figure 7.32 Username is assigned to Employee via Payroll Number (done by Admin)

Chapter 8
Conclusion & Future Extensions

Our main goal while developing this Academic Project Management System was to fulfil the main functional requirements that the Academic Committee of IPR had. We had to make sure that our system was easy to navigate through and we developed its interface keeping in mind the ease of use for the user and also that the user may/may not belong from a technical background.

Possible future extensions to this "Academic Project Management System" are listed as follows:

- In future, "E-mail Notification" as a functionality can be added to this system.
  This would include sending emails to the employees (internal requesters) as well as the contact person in the colleges. This facility could be used to provide them with information regarding the request that has been submitted by them.

- Another possible future extension could be displaying a "Request Status" field.
  So the IPR Staff and also the college contact person could know the status of the request submitted by them i.e. if it's in-process or allotted etc.

Appendix A
Initial Learning

Installation and Environment Setup:

We have created this system using Microsoft Visual Studio and Microsoft SQL Server (For our database needs). We decided to use Microsoft Visual Studio and Microsoft SQL Server as our development platform because IPR wanted an application that could be created easily.

Visual Studio offers a comprehensive development environment for building standards-based web applications and services. It improves productivity by allowing users to rapidly develop, test and deploy web solutions. Microsoft SQL Server is a very stable and fast database engine. There is a tremendous amount of support and resources available on the web. It is tightly integrated with the Windows Server security settings. This allows for a quick installation on your existing server.

Installation Specification:

- Windows OS 7, 64 bit & Service Pack 1
- Microsoft SQL Server 2012, 64 bit
- .net Framework 4.0 & IIS 7 or higher
- Microsoft Visual Studio 2012

In order to get familiar with MS Visual Studio and MS SQL database, we developed two modules:

- CHSS card
- Employee

  These helped us to understand the following concepts.

1. Master page and Site Map:

- Master Pages provide the mechanism for creating consisting web pages and give the developer a centralised way of changing the UI elements that need to be changed across all the pages of the web application.

  Following is the master page which we had developed.

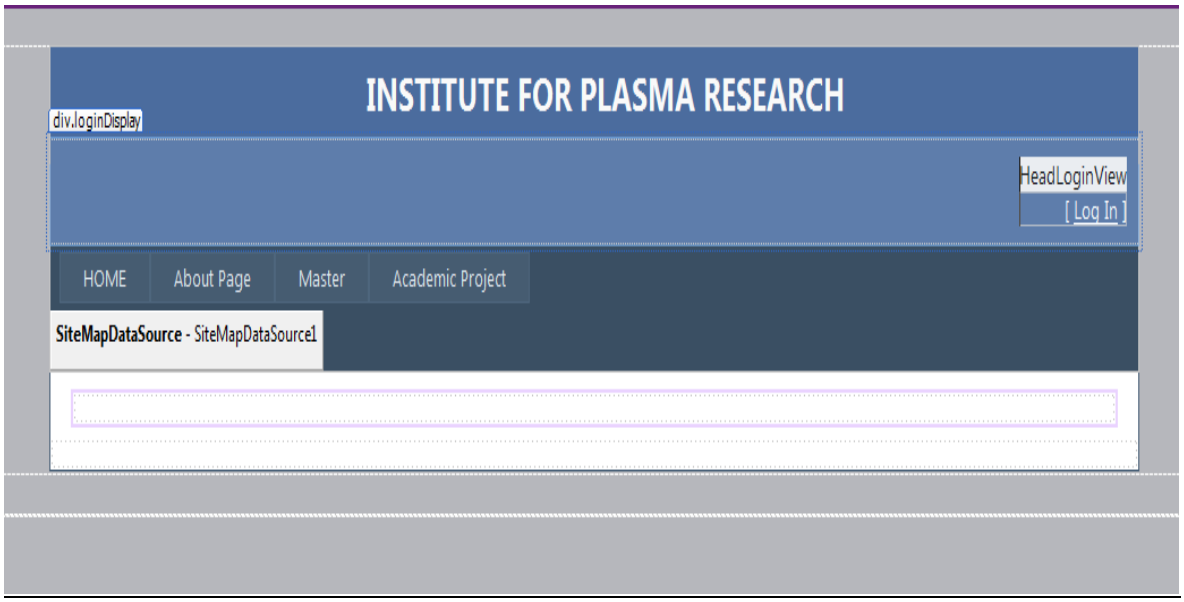


Figure 1 Master Page

- Maintaining the menu of a large web application is difficult and time consuming. In ASP.NET, the site navigation system uses an XML file that contains the site hierarchy.
  This sitemap file must be located in the application root directory, though it can reference other sitemap providers or other sitemap files in other directories as long as those files are in the same application.

2. WSAT(Web Site Administration Tool):

   This tool is a utility provided along with MS Visual Studio which assists in configuration and administration of a web site/application. The security tab is used to create users and roles, group users under different roles and assign access rules either at the role-level or user-level. The database that stores ASP.NET membership-related information, by default, is called ASPNETDB . The security tab simplifies and optimizes user authentication and authorization. Major drawback of this tool is that access rules could be defined only at the folder-level and not at the page level.

3. Ajax Control Toolkit:

   The ASP.NET Ajax control kit is an open source project built on top of Microsoft ASP.NET Ajax Framework. It provides a powerful infrastructure to write reusable, customizable and extensible ASP.NET Ajax extenders and controls, as well as a rich array of controls that can be used out of the box create an interactive web experience.

The first step is to download the Ajax Control Toolkit. The next step is to add it to the Visual Studio Toolbox. Follow these steps:

- Create a new Tool Box tab by right clicking the toolbox and selecting "Add Tab".
- Name the new tab "Ajax".
- Right click beneath the new tab and select the menu option "Choose Items".
- Click the Browse button and browse to the folder where you extracted the Ajax Control Toolkit.
- Pick the AjaxControlToolkit.dll and click the "ok" button to close the "Choose Toolbox Items" dialog.
  From this toolkit, we have majorly used "CalendarExtender " control.

4. Validations:

In the modules that we created, we had to ensure that the data entered through our forms was validated that is we had to make sure that the user did not provide any incorrect data. So to fulfil this requirement we used the "Validation control" provided in the Microsoft visual studio tool box.
We learnt about various types of validators:

- *Required Field Validator:*
  This ensures that the field on which this validator is applied is not left blank.
- *Regular Expression Validator:*
  This ensures that the field on which this validator is applied only accepts that data which matches the specified regular expression.
- *Custom validator:*
  This ensures that the field on which this validator is applied only accepts that data which matches the conditions specified in the custom defined method.

5. File Uploading:

The "file upload" control enables you to provide user with a way to send a file from their computer to the server at a specific location. You can also limit the size of the file and examine the properties of the uploaded file before storing.
We used this control to upload a file to the server. In addition to this, we also created a class "Date_FileClass.cs" which renamed the uploaded file as per IPR's requirement. Also to limit the size of the uploaded file to 1 MB, we used a custom validator.

6. Data Migration:

After installing the Microsoft SQL Server Management Studio and creating the desired tables, the next important task was to populate these tables with the available data.

We achieved this task in 2 ways:

- *From text format (.csv file)*

Initially, our data was in MS Excel and needed to be transferred to the respective tables. We did this as follows:

I. We first separated all the columns by placing a delimiter (comma, space) between them.
II. This delimited file was saved in .csv format.
III. The data from this .csv file was copied to a .txt file (in Notepad).
IV. Open SQL Server Management Studio, then right click on your database and select "Import Data".
V. In the "Import & Export Wizard", follow the steps as below:

(1) Choose the "Data Source" as "Flat File Source".
(2) Select the delimiter as "Comma" (,).
(3) And change the data types and lengths of the columns as defined in the tables.

- *From SQL Scripts*

This method was useful when we wanted to transfer selected tables (schema and data) from one laptop to another and also between different versions of MS SQL Server Management Studio.

I. In SQL Server Management Studio, right click on the database name & select "Task" then "Generate Scripts".
II. Then follow the steps in the Wizard and select the tables you wish to generate the scripts for.
III. In "Advanced Scripting Options", choose "Schema & Data" for "Types of data to Script" property.
IV. Save the generated script at the desired location.
V. Now execute this script on the computer where you want to transfer the data.

7. User & Role Creation in SQL Server Manager:

This enabled us to create different database users having different roles. Its advantage is that it provides added security as we can give table/column-level access permissions to the user.
Order of creation: Login -> Role -> User

- *LOGIN:*
  Allows you to enter into the SQL Server. Many USERS may be mapped to the same login BUT each of these users should be linked with a different database.
  Steps to follow:
  I. Security -> new login -> untick "enforce password expiration"
  II. Select "default database" as your database name.
  III. In the "user mapping" tab, the username is automatically generated, may change the schema (ex. to "dbo")
  IV. Tick your database name as the database.
  V. For "database role membership" option, select public or if a role is created then may select that.
  VI. Rest need not be changed.

- *ROLES:*
  Steps to follow:
  I. Rt. click on your database -> new database role
  II. Role name: Give a role name.
  III. "owner" and "schemas owned by this role" may be left blank
  IV. Role Members: can add a user to this role from the browse option.
  V. "Securable" section: Choose "All objects of the type" - then may select database/tables etc. depending on the requirement.
  VI. If we choose tables, then select the table you want to give permissions from the “grant/deny/don’t grant” columns.

8. Encryption and Decryption:

Protected Configuration helps improve the security of an application by enabling you to encrypt sensitive information that is stored in a configuration file. The .NET framework automatically decrypts the configuration file when the configuration file is processed, and decryption does not require any additional code.
Steps:

- Create a custom RSA Key container. The following command will create the “MyKeys” Key container.
  aspnet_regiis –pc “MyKeys” –exp

- Granting read access to an RSA Encryption Key. Before ASP.NET can decrypt encrypted information that is in the web.config file, the identity of your application must have read access to the encryption key that is used to encrypt and decrypt the encrypted sections.
  aspnet_regiis –pa "MyKeys" "NT AUTHORITY\NETWORK SERVICE"

- Next specify an instance of a Protected Configuration provider in the web.config file.
- The following command encrypts the < connectionStrings> element of the web.config file.
  aspnet_regiis –pe "connectionStrings" –app "/MyApplication" –prov "MyProvider"

- To access the decrypted configuration settings give the following command. aspnet_regiis –pd "connectionStrings" –app "/MyApplication" –prov "MyProvider"

9. Publishing our Web Application:
   For publishing a Web Application, right click on the root folder and then click on Publish carry out the following steps.

   I. Publish Web Wizard:
   You can choose or create a profile for your web application. A publish profile specifies the server you are deploying to, the credentials needed to log on to the server, the databases to deploy, and other deployment options.

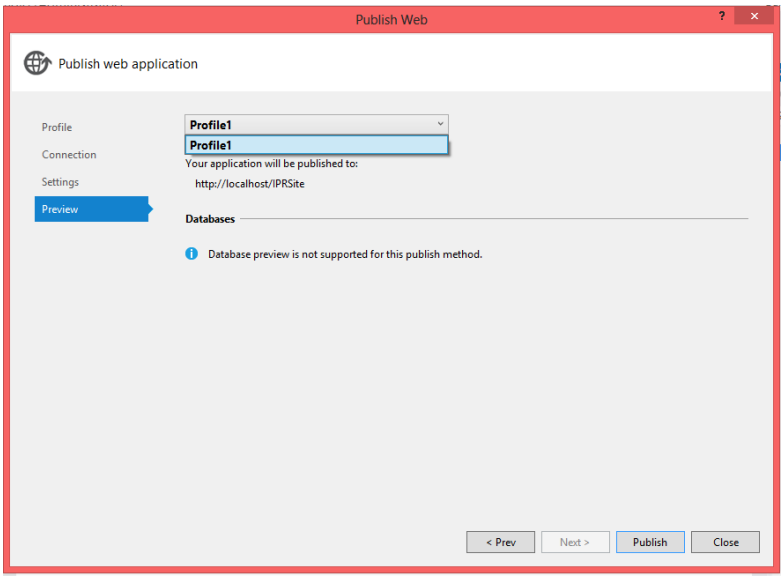


Figure 2 Publish Profile

II. Configuring the Connection Tab:

Select a value from the "Publish method" list. The "File System" publish method publishes the web application to a folder that you specify.

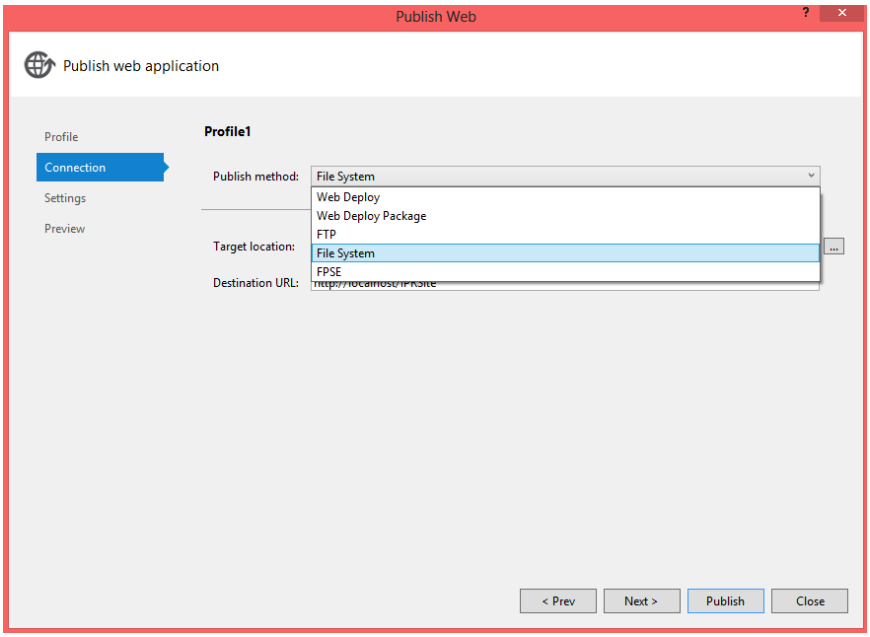


Figure 3: Connection Tab

III. Configuring the Settings Tab:

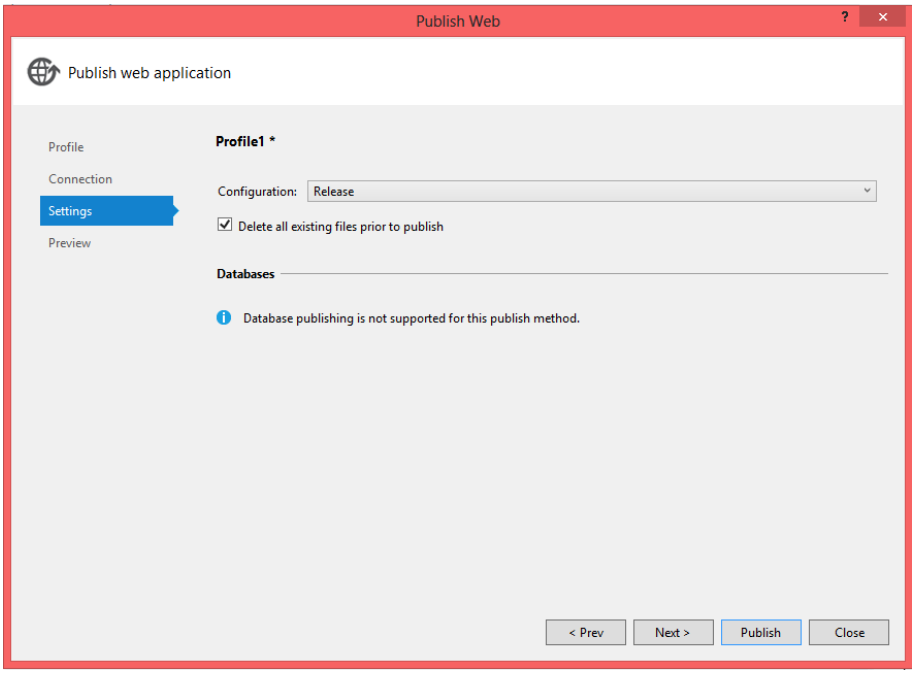


Figure 4: Settings Tab

IV. Previewing Changes & Publishing the Project:

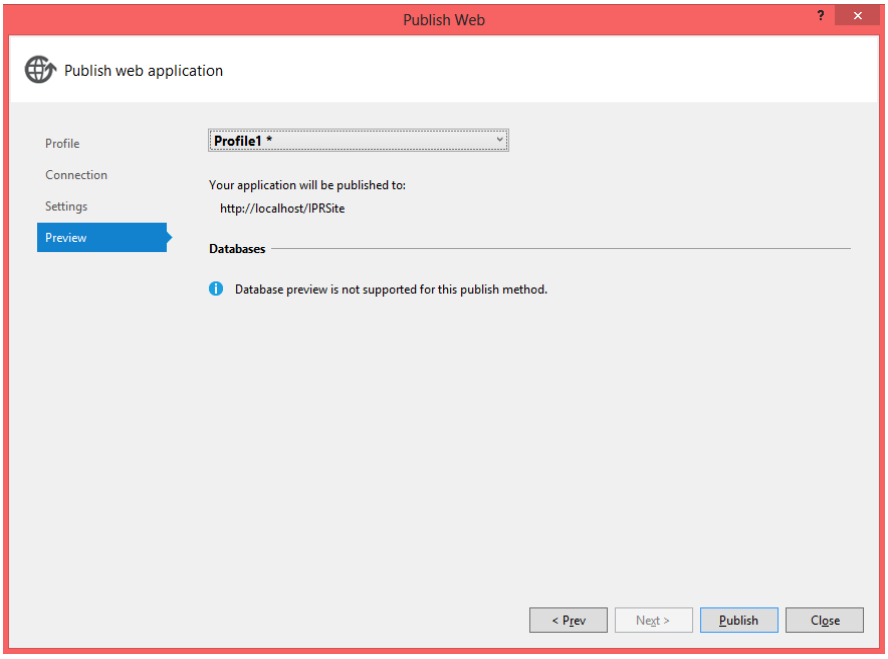


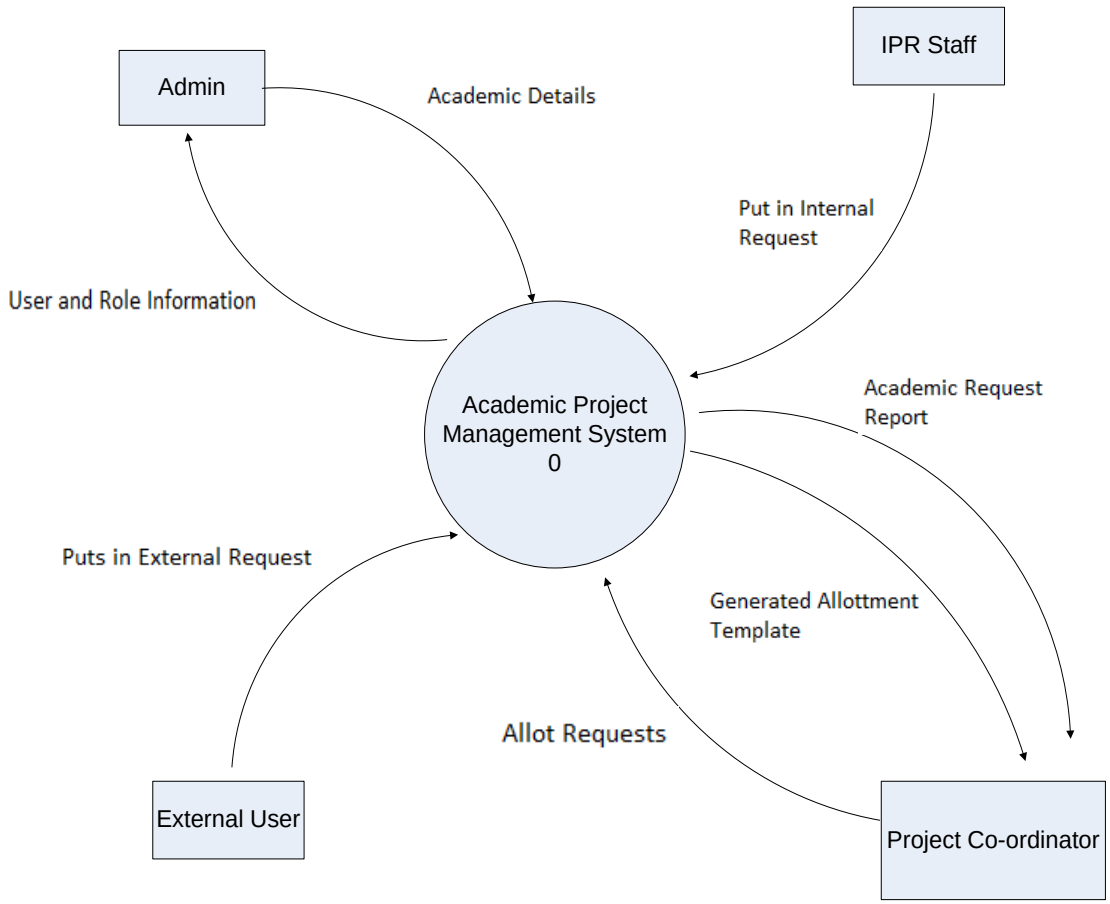
Figure 5: Preview

Appendix B

Data Flow Diagrams

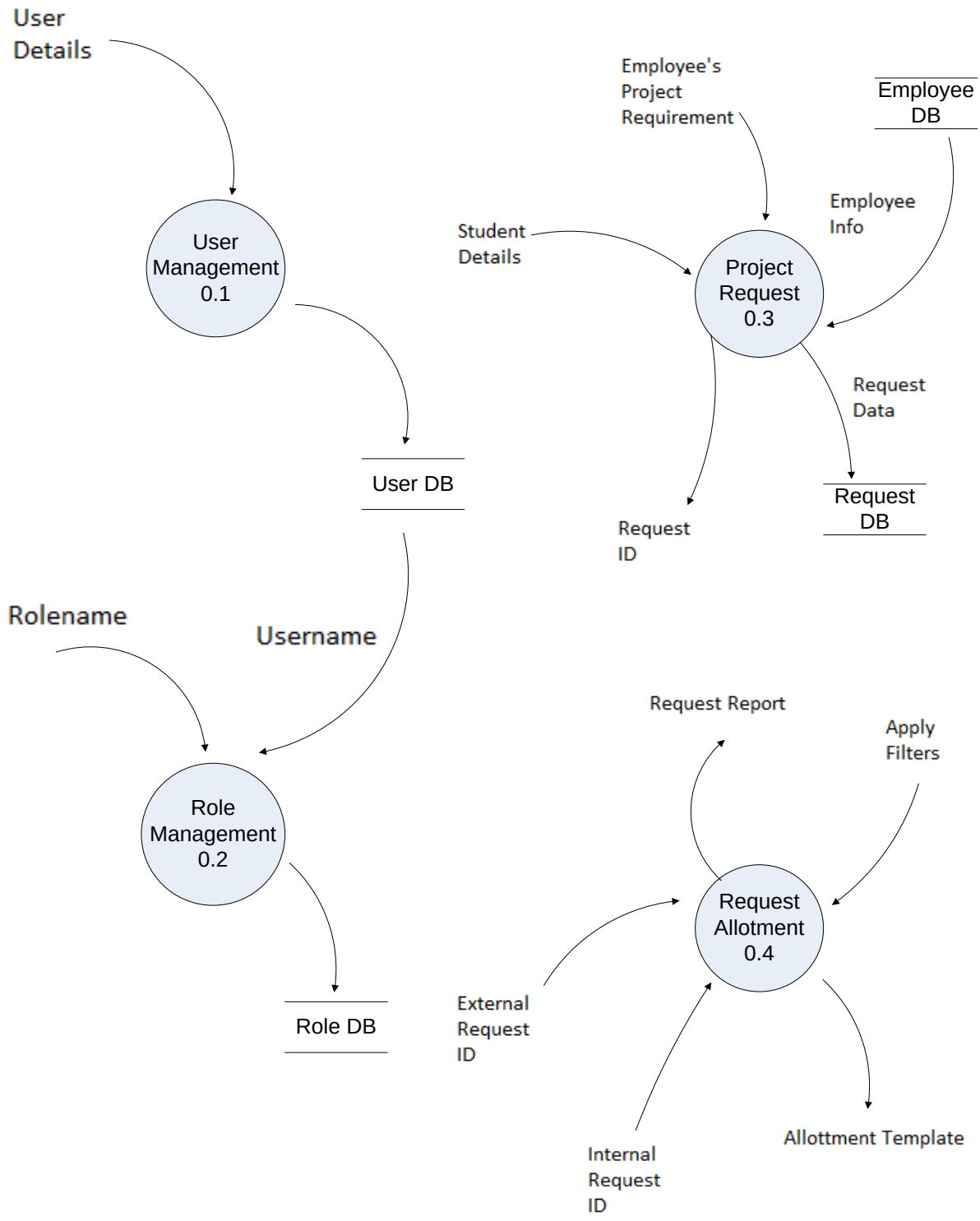
Data Flow Diagram Level – 0

Level 0 : Context Diagram

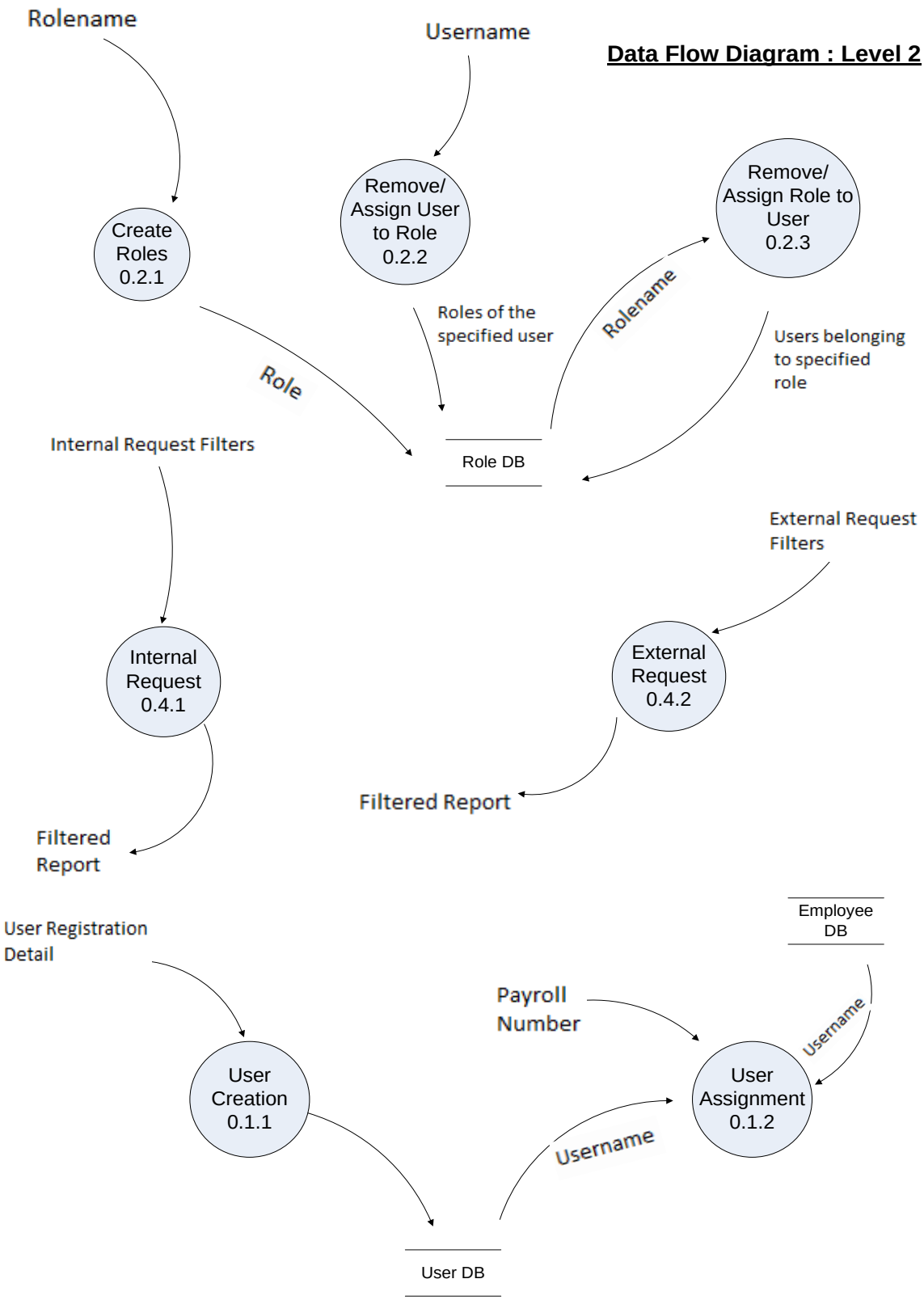


Data Flow Diagram Level – 1

Data Flow Diagram : Level 1



Data Flow Diagram Level – 2



Bibliography

The following resources and websites have been of immense help in understanding .NET technology.

- msdn.microsoft.com
- www.asp.net/webforms
- www.stackoverflow.com
- www.codeproject.com
- weblogs.asp.net
- www.youtube.com
- "Beginning ASP.NET 4.5 in C# and VB" by Imar Spaanjaars
- The large number of videos available on the Microsoft site helped us to get our basics right.