

JOIN

**CONNECTING STUDENT AND ORGANIZATION
FOR COMMUNITY SERVICES**

JOIN
CONNECTING STUDENT AND ORGANIZATION FOR COMMUNITY SERVICES

ABSTRACT

This document is a final report of a web based application which is named as “JOIN” comprised of services for both the work seeker at Bahria University and the Organization looking for students for their community support work and interneers to boost up their workforce. Community Support work is essential for students of Bahria university. Currently Bahria University students facing difficulties in completion of their CSP hours. NGOs contact student advisor or PO SRC for community support program then they announce the task in class rooms and display notification on notice board. Same is in the case of internship. Unlike this manual procedure JOIN provides internships and community support work to students to complete their internship and CSP hours for completion of their degree. This application also able organization to post a job that require multiple students and handle all the process of assigning job to multiple persons.

This application is designed using PHP Framework “Code ignitor”. Front end of this application developed using HTML, CSS and JavaScript/JQuery. Algorithms for crucial requirements are also designed. The roll out of this application onto the World Wide Web shows the project is successful. This application could be developed further to include additional functionality and allow the users greater control.

TABLE OF CONTENTS

DECLARATION	Error! Bookmark not defined.
ACKNOWLEDGEMENTS	Error! Bookmark not defined.
ABSTRACT	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	vii
LIST OF SYMBOLS / ABBREVIATIONS	xii
LIST OF APPENDICES	xiii

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	1
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
2	SOFTWARE REQUIREMENT SPECIFICATION (SRS)	4
	2.1 User Classes and Characteristics	4
	2.2 Operating Environment	5
	2.3 Design and Implementation Constraints	5
	2.4 Assumptions and Dependencies	5
	2.5 System Requirement Chart	6
3	DESIGN AND METHODOLOGY	10

3.1	System Use Case Diagram	11
3.2	Use Case Description	12
3.3	Domain Model	38
3.4	Class Diagrams	42
3.5	Sequence Diagrams	47
3.6	Collaboration Diagrams	57
3.7	Data Model	74
4	METHODOLOGY & IMPLEMENTATION	82
4.1	Methodology	82
4.2	Implementation	85
5	USER MANUAL	87
5.1	Getting started	87
5.2	JOIN for Student	94
5.3	JOIN for Organization	114
5.4	Join for Admin	119
6	CONCLUSION AND RECOMMENDATIONS	127
6.1	Conclusion	127
	REFERENCES	128
	APPENDICES	129

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1	System Requirement Chart	6
Table 2	Use Case User registration	12
Table 3	Use Case User login	13
Table 4	Use Case Reset User Password	14
Table 5	Use Case Edit Group Permissions	15
Table 6	Use Case Search user	16
Table 7	Use Case Block user	17
Table 8	Use Case User logout	18
Table 9	Use Case Job Posting	19
Table 10	Use Case View Internship	20
Table 11	Use Case Apply for Internship	21
Table 12	Use Case View CSP Jobs	22
Table 13	Use Case Join CSP Job	23
Table 14	Use Case View My Jobs	24
Table 15	Use Case Edit Job	25
Table 16	Use Case Discussion	26
Table 17	Use Case Delete Post	27
Table 18	Use Case User rating	28
Table 19	Use Case Friend Circle	29
Table 20	Use Case Job Search	30
Table 21	Use Case Uploading resume	31

Table 22 Use Case User Search	32
Table 23 Use Case Advance Search	33
Table 24 Use Case Close Job	34
Table 25 Use Case Update Profile	35
Table 26 Use Case Update Profile Picture	36
Table 27 Use Case Change Password	37

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1	System use case diagram	11
Figure 2	Domain Model User Side 1	38
Figure 3	Domain Model User Side 2	39
Figure 4	Domain Model Admin Side 1	40
Figure 5	Domain Model Admin Side 2	41
Figure 6	User Side Controller Class Diagram	42
Figure 7	User Side Model Class Diagram	43
Figure 8	Admin Side Controller Class Diagram	44
Figure 9	Admin Side Model Class Diagram	45
Figure 10	User Registration	47
Figure 11	User login	48
Figure 12	Reset password	48
Figure 13	Edit User Permissions	49
Figure 14	User search	49
Figure 15	Block user	50
Figure 16	Logout	50
Figure 17	Job posting	51
Figure 18	View internship	51
Figure 19	Apply for internship	52

Figure 20 View CSP job	52
Figure 21 Apply for CSP job	53
Figure 22 View my jobs	53
Figure 23 Delete Job	54
Figure 24 User rating	54
Figure 25 Discussion	55
Figure 26 Add to Friend circle	55
Figure 27 Search	56
Figure 28 Upload resume	56
Figure 29 User Registration	57
Figure 30 User Login	58
Figure 31 Change Password	59
Figure 32 Search User	60
Figure 33 Edit Permission	61
Figure 34 Logout	62
Figure 35 Block User	62
Figure 36 Post Job	63
Figure 37 View Internship	64
Figure 38 Upload Resume	65
Figure 39 Apply for Job	66
Figure 40 View CSP Job	67
Figure 41 View My Jobs	68
Figure 42 User Rating	69
Figure 43 Delete Job	70
Figure 44 Discussion	71

Figure 45 Add to Friend Circle	72
Figure 46 Search	73
Figure 47 External Model User	74
Figure 48 External Model UMS	75
Figure 49 Conceptual Model	76
Figure 50 Architecture Diagram	86
Figure 51 How to access JOIN	88
Figure 52 Home Screen (Without Login)	89
Figure 53 Login Button	89
Figure 54 Login form	90
Figure 55 Signup options	91
Figure 56 Signup form (Student)	92
Figure 57 Signup form (Organization)	93
Figure 58 Navigation Bar (Student)	94
Figure 59 Navigation Bar (Organization)	94
Figure 60 Navigation Bar (Student)	94
Figure 61 User Profile	95
Figure 62 Home Screen (After Login)	96
Figure 63 Find Job	97
Figure 64 Featured Jobs	98
Figure 65 Advertisements	98
Figure 66 My Jobs (Student)	99
Figure 67 In Progress Jobs (Student)	99
Figure 68 Applied Jobs	100
Figure 69 Completed Jobs	101

Figure 70 CSP Jobs	102
Figure 71 Internships	103
Figure 72 Friend Circle	104
Figure 73 View Profile Button	104
Figure 74 Remove Button	104
Figure 75 Remove Verification Pop-up	105
Figure 76 Friend Suggestions	105
Figure 77 Add to friend circle button	105
Figure 78 Search form	106
Figure 79 Advance search button	106
Figure 80 Advance search form	107
Figure 81 Job categories	107
Figure 82 Job Types	107
Figure 83 Find job results	108
Figure 84 Calendar	109
Figure 85 Message	110
Figure 86 Notifications	110
Figure 87 Student Profile	111
Figure 88 Profile options	112
Figure 89 Profile setting	113
Figure 90 Logout button	113
Figure 91 Home (for Organization)	114
Figure 92 My Jobs (Organization)	115
Figure 93 In Progress Jobs (Organization)	115
Figure 94 Job Requests (Organization)	116

Figure 95 Unassigned Jobs (Organization)	116
Figure 96 Post Job	117
Figure 97 Profile setting (organization)	119
Figure 98 Admin sidebar	120
Figure 99 Admin dashboard	121
Figure 100 User Groups	121
Figure 101 Add new user group button	122
Figure 102 Add new group form	122
Figure 103 View Users	123
Figure 104 CSP Jobs	124
Figure 105 Internships	125
Figure 106 User Job Status	125
Figure 107 Reports	126
Figure 108 Admin Logout	126

LIST OF SYMBOLS / ABBREVIATIONS

CSP	Community Support Program
PO SRC	Program Officer Student Resource Program
NGO	Non Commercial Organization

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
APPENDIX A:	User Story	129
APPENDIX B:	Requirement List	130

CHAPTER 1

INTRODUCTION

1.1 Background

Community Support work and internship are essential for students of Bahria university to complete their degree. Currently Bahria University students facing difficulties in completion of their CSP hours. NGOs contact student advisor or PO SRC for community support program then they announce the task in class rooms and display notification on notice board. So some students get opportunity to complete their essential CSP hours rest of the students will not be able to avail this opportunity. Same happens in the case of internship which is also mandatory for all students. Some students get internship opportunity by personal reference or university links.

Moreover, University management keep record of students which complete their internship and CSP hours.

1.2 Problem Statements

Bahria University does not have such independent platform yet which automate the CSP and internship process. Organizations and students are using different public platforms like Rozee.pk and Facebook pages for CSPs and internships. Moreover, maximum of the opportunities avails by personal reference in both cases.

1.3 Aims and Objectives

Objectives of JOIN would be as follows:

- i) To provide university students to explore suitable student activities in order to complete their essential CSP
- ii) To provide university students to get suitable internship for gaining experience
- iii) To provide NGOs with a portal to accelerate the Community Support work through allocate to suitable/matching student
- iv) To provide organization with a portal to accelerate the internships for students
- v) To rank different NGOs and organizations which help to measure their reputation
- vi) To rank different students on the basis of quality of work they provided

1.4 Scope of Project

Following points defines the scope of this project.

1. User management system

- Student management
Student management allows students to register using their details includes name, Institute name, enrollment no., email, contact number, expertise and experience.
- Organization management
Organization management allows organizations to register using their details includes name, email, contact number, address, work area.
- User Management
User management allows system admin to view all registered students and organizations. Moreover, system admin can check jobs status, edit and delete jobs.

2. Job Portal

- Community support work

Job portal module provides the facility to NGOs to post the required work.

- Internship

Job portal module provides the facility to Organizations to offer internship.

3. Job categorization

Posted jobs automatically group into relevant category and can be search easily.

4. User rating

Organizations and Students can rate their experience with each other that help to measure the user's reputation.

5. Job notifications

Student get job notification related to their interest after job posting.

6. Discussion

Discussion allow one to one and group discussion between organization and students.

7. Team up

If any NGO or organization need group of students for work, this module help to team up students for specific task

8. Friend Circle

Student can add people of same interest in circle.

9. Calendar

Student can schedule selected task on calendar.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2.1 User Classes and Characteristics

JOIN is consisting of three modules i.e. Internship, CSP Jobs and UMS. Following are the users of each module respectfully

2.1.1 Internship Module

- a) Organization that can post internship.
- b) Student that can search and apply for internship.

2.1.2 Community Support Program Jobs (CSP Jobs)

- a) NGO that can post CSP (Community Support Program) Job.
- b) Student of Bahria University that can search and join CSP job and rank the NGO.

2.1.3 User Management System (UMS)

UMS manages all the users of JOIN and used by System Admin that can check students internship and CSP hours status, generate reports, edit user permissions, block user and view details of users.

2.2 Operating Environment

This is a web-based platform and run accurately on following web browsers

1. Google Chrome version 35.1 or higher
2. Mozilla Firefox version 40.0 or higher
3. Opera version 51.0 or higher

2.3 Design and Implementation Constraints

JOIN is a web-based system therefore front-end of JOIN developed using HTML, CSS and bootstrap and back-end developed using PHP and MySQL database. It uses PHP framework cod-igniter includes its Ion-Auth library for user registration and authentication. XAMPP server used as host server. JOIN developed using FDD Agile methodology, so it uses a modular design where features of every module wrapped separately furthermore each feature divided into possible sub-features.

2.4 Assumptions and Dependencies

JOIN is a web-based system so it requires Google Chrome, Mozilla Firefox or Safari browser installed on user's system. JOIN requires 35.1 or higher for Google Chrome, 50.0 or higher for Safari and 40.0 higher for Mozilla Firefox to run efficiently. There is a 24/7 host server of all time access for users of JOIN.

2.5 System Requirement Chart

Table 1 System Requirement Chart

ID	Priority	Type	Source	Contained in Use-Case	Description
JOIN-01	High	Functional	Bahria University	UCJ-001	User must be registered to access JOIN
JOIN-02	High	Functional	Bahria University	UCJ-002	User login
JOIN-03	Medium	Functional	Bahria University	UCJ-003	Reset user password
JOIN-04	Medium	Functional	Bahria University	UCJ-004	Edit user permissions
JOIN-05	Medium	Functional	Bahria University	UCJ-005	View/Search users
JOIN-06	Medium	Functional	Bahria University	UCJ-006	Block a user
JOIN-07	High	Functional	Bahria University	UCJ-007	User logout
JOIN-08	High	Functional	Bahria University	UCJ-008	Post an internship job
JOIN-09	Medium	Functional	Bahria University	UCJ-009	View an internship job
JOIN-10	Medium	Functional	Bahria University	UCJ-010	Apply for an internship job
JOIN-11	High	Functional	Bahria University	UCJ-008	Post a CSP job
JOIN-12	Medium	Functional	Bahria University	UCJ-011	View a CSP Job
JOIN-13	Medium	Functional	Bahria University	UCJ-012	Pick/Join a CSP Job
JOIN-14	Medium	Functional	Bahria University	UCJ-013	Student should be able to view his/her competed or incomplete CSP and internships jobs.

JOIN-15	Medium	Functional	Bahria University	UCJ-013	Organization should be able to view its posted jobs.
JOIN-16	Low	Functional	Bahria University	UCJ-015	Discussion between student and organization
JOIN-17	Medium	Functional	Bahria University	UCJ-016	System admin should be able to delete any post.
JOIN-18	Medium	Functional	Bahria University	UCJ-017	Student can rate organization for whom he/she worked and vice versa.
JOIN-19	Medium	Functional	Bahria University	UCJ-018	Discussion between multiple students and organization
JOIN-20	Low	Functional	Bahria University	UCJ-019	Student can make a friend circle by adding students of same interest.
JOIN-21	Medium	Functional	Bahria University	UCJ-020	User should be able to search specific job or category
JOIN-22	Medium	Functional	Bahria University	N/A	System should be able to automatically schedule selected task on calendar.
JOIN-23	Medium	Functional	Bahria University	N/A	System automatically suggest more posts of that NGO or organization.
JOIN-24	Medium	Functional	Bahria University	N/A	Student get job notification related to their interest after job posting.

JOIN-25	Medium	Functional	Bahria University	N/A	Posted jobs group into relevant category and can be search easily.
JOIN-26	Medium	Functional	Bahria University	UCJ-021	User should be able to upload his/her resume.
JOIN-27	Medium	Functional	Bahria University	UCJ-008	If any NGO need group of students for work, they can set required number of students while posting job.
JOIN-28	Low	Functional	Bahria University	N/A	In Discussion, every user can see his/her text in black colour with white background along with their user name.
JOIN-29	Low	Functional	Bahria University	UCJ-014	Organization can edit their posted jobs.
JOIN-30	High	Non-Functional	Bahria University	N/A	GUI should compatible with both Chrome and Mozilla.
JOIN-31	Low	Non-Functional	Bahria University	N/A	Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
JOIN-32	High	Non-Functional	Bahria University	N/A	Final, integrated project code will test to ensure that complete project is integrated well

					and functioning properly.
JOIN-33	High	Non-Functional	Bahria University	N/A	Display all the content after loading of web page.
JOIN-34	High	Non-Functional	Bahria University	N/A	Server response should be keep in mind while designing data retrieval algorithms.
JOIN-35	High	Non-Functional	Bahria University	N/A	All the passwords should save in encrypted form
JOIN-36	High	Non-Functional	Bahria University	N/A	NGOs and organizations can access CSP portal and internship portal.
JOIN-37	Medium	Functional	Bahria University	UCJ-021	User should be able to search specific job.
JOIN-38	Medium	Functional	Bahria University	UCJ-023	User should be able to search other user.

CHAPTER 3

DESIGN AND METHODOLOGY

1. Use case diagram
2. Use case description
3. Domain Model
4. Design Class Diagram
5. Data Model

3.1 System Use Case Diagram

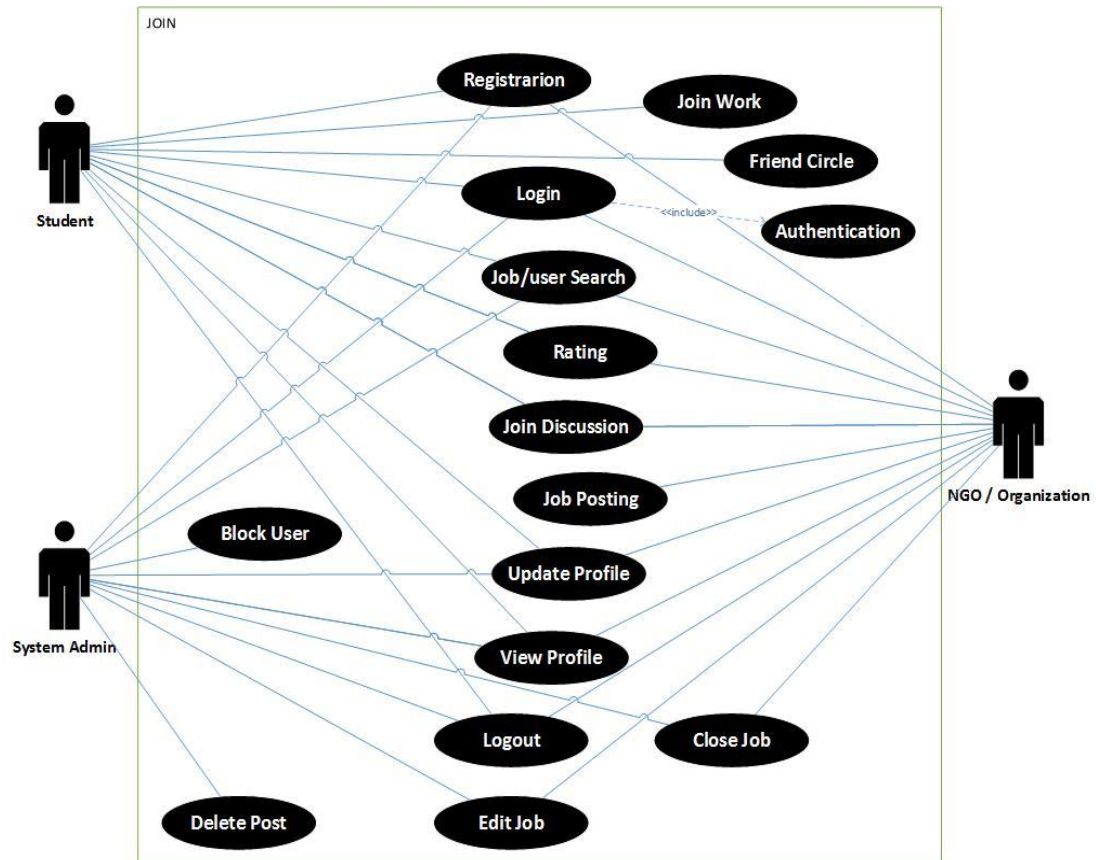


Figure 1 System use case diagram

3.2 Use Case Description

Following is the description of the use cases involved in the design of this project:

[1]

Table 2 Use Case User registration

Name	User registration
Use-Case ID	UCJ-001
Priority	High
Primary Actor(s)	Student Organization
Other participating Actor(s)	System admin
Description	This use case describes the event of a user registering at JOIN.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User should be regular student of Bahria or registered organization 2. User should not register yet
Trigger	This use case initiate when a new user going to register
Typical flow of events	<ol style="list-style-type: none"> 1. User access the registration page 2. User select registration type (student or organization) 3. User fill the registration form 4. A confirmation e-mail sent to user with verification link 5. User click the link with verification code in email 6. Email verified and user registered
Alternate flow of events	Alt-1 user gives wrong university identity and a popup error displays
Post condition	User is successfully registered
Alternate post condition	User isn't register

Table 3 Use Case User login

Name	User login
Use-Case ID	UCJ-002
Priority	High
Primary Actor(s)	Student Organization
Other participating Actor(s)	System admin
Description	This use case describes the event when a user going to login.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be registered at JOIN 2. User should not login
Trigger	This use case initiate when a user going to login
Typical Flow of events	<ol style="list-style-type: none"> 1. User gives resisted email and password 2. System verifies User Credentials
Alternate flow of events	Alt-1 User gives wrong credentials, system generate a popup error and take user to login page.
Post condition	User successfully logged in
Alternate post condition	Login failed

Table 4 Use Case Reset User Password

Name	Reset User Password
Use-Case ID	UCJ-003
Priority	Medium
Primary Actor(s)	Student Organization
Other participating Actor(s)	N/A
Description	This use case describes the event of a user when resting password.
Pre-condition(s)	1. User must joined JOIN earlier
Trigger	This use case initiate when a user going to reset its password
Typical flow of events	1. Click on Reset Password 2. Enter new password and enter confirm password 3. Click on update button
Alternate flow of events	Alt-2 The entered password not match system show error and ask for re enter password.
Post condition	Password successfully updated
Alternate post condition	Password not updated

Table 5 Use Case Edit Group Permissions

Name	Edit Group Permissions
Use-Case ID	UCJ-004
Priority	Medium
Primary Actor(s)	System Admin
Other participating Actor(s)	N/A
Description	This use case describes the event of a system admin editing any user's permissions.
Pre-condition(s)	<ol style="list-style-type: none"> 1. System admin must be logged in 2. Target User must joined JOIN earlier
Trigger	This use case initiate when a system admin going to edit user permissions
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Edit permissions 2. Edit target group's permissions from permission table 3. Click on save button
Alternate flow of events	N/A
Post condition	Permissions successfully updated
Alternate post condition	Permissions not updated

Table 6 Use Case Search user

Name	Search user
Use-Case ID	UCJ-005
Priority	Medium
Primary Actor(s)	Organization, Student
Other participating Actor(s)	System Admin
Description	This use case describes the event of a system admin search / view any user
Pre-condition(s)	<ol style="list-style-type: none"> 1. System User must be logged in 2. Target User must joined JOIN earlier
Trigger	This use case initiate when a user going to search / view a user
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Search Bar 2. Enter Username/email 3. Click on search button 4. View user
Alternate flow of events	Alt 4- Wrong user id entered and wrong result found
Post condition	User successfully searched
Alternate post condition	User not searched

Table 7 Use Case Block user

Name	Block user
Use-Case ID	UCJ-006
Priority	Medium
Primary Actor(s)	System Admin
Other participating Actor(s)	N/A
Description	This use case describes the event of a system admin block any user
Pre-condition(s)	<ol style="list-style-type: none"> 1. System Admin must be logged in 2. Target User must joined JOIN earlier
Trigger	This use case initiate when a system admin going to block a user
Typical flow of events	<ol style="list-style-type: none"> 1. Click on View/Block Users 2. Search User id 3. View user 4. Click on block button of target user
Alternate flow of events	Alt 4- Wrong user id entered and wrong result found
Post condition	User successfully Blocked
Alternate post condition	User not Blocked

Table 8 Use Case User logout

Name	User logout
Use-Case ID	UCJ-007
Priority	High
Primary Actor(s)	Student Organization
Other participating Actor(s)	System Admin
Description	This use case describes the event when a user going to logout.
Pre-condition	<ol style="list-style-type: none"> 1. User must be registered at JOIN 2. User must be login
Trigger	This use case initiate when a user going to logout
Typical Flow of events	<ol style="list-style-type: none"> 1. User click on profile picture 2. User click logout from dropdown
Alternate flow of events	N/A
Post condition	User successfully logout
Alternate post condition	Logout failed

Table 9 Use Case Job Posting

Name	Job Posting
Use-Case ID	UCJ-008
Priority	High
Primary Actor(s)	Organization
Other participating Actor(s)	N/A
Description	This use case describe the event when an organization going to post an internship or CSP job.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. User should be an organization
Trigger	This use case initiate when a user going to post an internship.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Post Job button 2. Fill the required fields 3. Click on Post
Alternate flow of events	Alt-2 User does not fill any mandatory field and click on Post button an error occurred and displays an error message.
Post condition	Internship successfully posted
Alternate post condition	Internship not posted

Table 10 Use Case View Internship

Name	View Internship
Use-Case ID	UCJ-009
Priority	Medium
Primary Actor(s)	Student
Other participating Actor(s)	Organization
Description	This use case describes the event of a student viewed an internship posted by organization.
Pre-condition(s)	<ol style="list-style-type: none"> 1. Student must be logged in 2. Internship must be posted earlier
Trigger	This use case initiate when a student going to view an internship posted by organization.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Internship from navigation bar 2. Internship page will open 3. Click on internship to view details
Alternate flow of events	N/A
Post condition	User view internship
Alternate post condition	User not view internship

Table 11 Use Case Apply for Internship

Name	Apply for Internship
Use-Case ID	UCJ-010
Priority	Medium
Primary Actor(s)	Student
Other participating Actor(s)	N/A
Description	This use case describes the event of a student applying for an internship.
Pre-condition(s)	<ol style="list-style-type: none"> 1. Student must be logged in 2. Internship must be posted earlier
Trigger	This use case initiate when a student going to apply for an internship.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Internship title/ details 2. Internship details page open 3. Attach CV 4. Click on Apply button
Alternate flow of events	N/A
Post condition	Student is successfully Apply for internship
Alternate post condition	Student is not successfully apply for internship

Table 12 Use Case View CSP Jobs

Name	View CSP Jobs
Use-Case ID	UCJ-011
Priority	Medium
Primary Actor(s)	Student
Other participating Actor(s)	Organization
Description	This use case describes the event of a student viewed a CSP job posted by organization.
Pre-condition(s)	<ol style="list-style-type: none"> 1. Student must be logged in 2. CSP Job must be posted earlier
Trigger	This use case initiate when a student going to view a CSP Job posted by organization.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on CSP Jobs from navigation bar 2. CSP Jobs page will open 3. Click on CSP Job to view details
Alternate flow of events	N/A
Post condition	User view CSP Job
Alternate post condition	User not view CSP Job

Table 13 Use Case Join CSP Job

Name	Join CSP Job
Use-Case ID	UCJ-012
Priority	Medium
Primary Actor(s)	Student
Other participating Actor(s)	N/A
Description	This use case describes the event of a student joining a CSP job.
Pre-condition(s)	<ol style="list-style-type: none"> 1. Student must be logged in 2. Job must be posted earlier
Trigger	This use case initiate when a student going to join a virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Job title/ details 2. Job details page open 3. Click on Join button 4. An acknowledgement message will be displayed
Alternate flow of events	N/A
Post condition	Student is successfully joined job
Alternate post condition	Student is not successfully joined job

Table 14 Use Case View My Jobs

Name	View My Jobs
Use-Case ID	UCJ-013
Priority	Medium
Primary Actor(s)	Student and Organization
Other participating Actor(s)	N/A
Description	This use case describes the event of a student viewed his CSP jobs and internships he/she selected. Both completed and uncompleted jobs and internships will be under My Jobs.
Pre-condition(s)	<ol style="list-style-type: none"> 1. Student must be logged in 2. Job must be posted earlier 3. Student complete or select jobs
Trigger	This use case initiate when a student going to view a CSP Job posted by organization.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on My Jobs from navigation bar 2. My Jobs page will open 3. Click on any Job to view details
Alternate flow of events	N/A
Post condition	User view My Job
Alternate post condition	There are no job under My Jobs

Table 15 Use Case Edit Job

Name	Edit Job
Use-Case ID	UCJ-014
Priority	Medium
Primary Actor	Organization
Other participating Actor(s)	N/A
Description	This use case describe the event when an organization wants to edit his posted job when it completed.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Job status must be unassigned
Trigger	This use case initiated when an organization going to edit his posted job.
Typical flow of events	<ol style="list-style-type: none"> 1. User (Organization) open his posted job by click on View job 2. Click on edit Job button 3. System display editable job options 4. User make editing 5. System display acknowledgement message
Alternate flow of events	N/A
Post condition	Job successfully edited
Alternate post condition	Job not edited

Table 16 Use Case Discussion

Name	Discussion
Use-Case ID	UCJ-015
Priority	Low
Primary Actor(s)	Organization Student
Other participating Actor(s)	N/A
Description	This use case describe the event when a user want to discuss about specific job
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Targeted job must be created
Trigger	This use case initiated when a user going to discuss about assigned or created job
Typical flow of events	<ol style="list-style-type: none"> 1. User open the job 2. A message box appears 3. Submit message
Alternate flow of events	N/A
Post condition	Discussion successfully created
Alternate post condition	Discussion not created

Table 17 Use Case Delete Post

Name	Delete Post
Use-Case ID	UCJ-016
Priority	Medium
Primary Actor(s)	System Admin
Other participating Actor(s)	N/A
Description	This use case describes the event of a system admin deleting any users post
Pre-condition(s)	<ol style="list-style-type: none"> 1. System Admin must be logged in 2. Target post must be posted earlier
Trigger	This use case initiate when a system admin going to delete a post
Typical flow of events	<ol style="list-style-type: none"> 1. Click on target post 2. Select Delete action from action 3. Confirm delete action
Alternate flow of events	Alt 1- Wrong post selected and wrong result found
Post condition	Post successfully deleted
Alternate post condition	Post not deleted

Table 18 Use Case User rating

Name	User rating
Use-Case ID	UCJ-017
Priority	Medium
Primary Actor(s)	Student Organization
Other participating Actor(s)	N/A
Description	This use case describes the event of a user rating any other user
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Both users must be part of one particular job
Trigger	This use case initiate when a user going to rate other user
Typical flow of events	<ol style="list-style-type: none"> 1. Open targeted users profile 2. Select rate user option 3. Fill the form 4. Click on submit
Alternate flow of events	<p>Alt 1- Wrong profile selected and wrong result found</p> <p>Alt 3- An error accrue if form not completely filled</p>
Post condition	User successfully rated
Alternate post condition	User not successfully rated

Table 19 Use Case Friend Circle

Name	Friend Circle
Use-Case ID	UCJ-019
Priority	Low
Primary Actor(s)	Student Organization
Other participating Actor(s)	N/A
Description	This use case describe the event when a user wants to add other users in his friend circle.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Targeted user must have same interest
Trigger	This use case initiated when any user going to add other user in his friend circle.
Typical flow of events	<ol style="list-style-type: none"> 1. User search the targeted user or open member list 2. User click on Add button
Alternate flow of events	Alt-1 User not found
Post condition	User added in Friend Circle
Alternate post condition	Member not added

Table 20 Use Case Job Search

Name	Job Search
Use-Case ID	UCJ-020
Priority	Medium
Primary Actor(s)	Student
Other participating Actor(s)	N/A
Description	This use case describe the event when a student wants to search job.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Job must be posted earlier
Trigger	This use case initiated when any user going to search job.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on search bar 2. User enter keywords 3. System show jobs related to searched keyword
Alternate flow of events	Alt-2 Wrong keyword show wrong result
Post condition	List of jobs show to user
Alternate post condition	Result not found

Table 21 Use Case Uploading resume

Name	Uploading resume
Use-Case ID	UCJ-021
Priority	Medium
Primary Actor(s)	Student
Other participating Actor	N/A
Description	This use case describe the event when a user going to upload a resume.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Resume should be not exceeding the size of uploading limit
Trigger	This use case initiate when a user going to upload a resume.
Typical flow of events	<ol style="list-style-type: none"> 1. Select an internship 2. Internship open 3. Click on apply button 4. Browse and select the file 5. Click on upload 6. Click on Send
Alternate flow of events	Alt-2 User doesn't select any file and click on ok button, an error occurred and displays an error message.
Post condition	File successfully uploaded
Alternate post condition	Not uploaded

Table 22 Use Case User Search

Name	User Search
Use-Case ID	UCJ-022
Priority	Medium
Primary Actor(s)	Organization Student
Other participating Actor(s)	N/A
Description	This use case describe the event when a user wants to search any other user.
Pre-condition(s)	<ol style="list-style-type: none"> 1. User must be logged in 2. Targeted user must be register member of JOIN
Trigger	This use case initiated when any user going to search user.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on search bar 2. Enter username to search 3. System show searched users profile
Alternate flow of events	Alt-2 Wrong keyword show wrong result
Post condition	System show user profile
Alternate post condition	User not found

Table 23 Use Case Advance Search

Name	Advance Search
Use-Case ID	UCJ-23
Priority	Medium
Primary Actor	Student
Other participating Actor(s)	
Description	This use case describe the event when a user wants to search jobs against different parameter.
Pre-condition	1. User must be logged in
Trigger	This use case initiated when any student going to search jobs.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on search icon 2. Select Advanced Search option 3. Select appropriate parameters 4. System show searched jobs
Alternate flow of events	Alt-2 If searched job not exist system show “No result found”
Post condition	System show related jobs
Alternate post condition	No job found related to searched parameters

Table 24 Use Case Close Job

Name	Close Job
Use-Case ID	UCJ-24
Priority	High
Primary Actor	Organization
Other participating Actor(s)	N/A
Description	This use case describe the event when an organization wants to close his posted job when it completed.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Job must be completed
Trigger	This use case initiated when an organization going to close his posted job.
Typical flow of events	<ol style="list-style-type: none"> 1. User (Organization) open his posted job by click on View job 2. Click on Close Job button 3. System display acknowledgement message
Alternate flow of events	N/A
Post condition	Job successfully closed
Alternate post condition	Jon not closed

Table 25 Use Case Update Profile

Name	Update Profile
Use-Case ID	UCJ-26
Priority	Low
Primary Actor	Organization Student
Other participating Actor(s)	N/A
Description	This use case describe the event when a user wants to update his profile details.
Pre-condition	1. User must be logged in
Trigger	This use case initiated when any user going to update his profile details.
Typical flow of events	<ol style="list-style-type: none"> 1. User open his profile 2. Click on setting option 3. System display editable profile options 4. User make editing and update 5. System display acknowledgement message
Alternate flow of events	N/A
Post condition	Profile successfully updated
Alternate post condition	Profile not updated successfully

Table 26 Use Case Update Profile Picture

Name	Update Profile Picture
Use-Case ID	UCJ-27
Priority	Low
Primary Actor	Organization Student
Other participating Actor(s)	----
Description	This use case describe the event when a user wants to update his profile picture.
Pre-condition	1. User must be logged in
Trigger	This use case initiated when any user going to update his profile details.
Typical flow of events	<ol style="list-style-type: none"> 1. User open his profile 2. Click on setting option 3. Select update profile picture option 4. User attach picture of allowed extension 5. System update profile picture
Alternate flow of events	Alt-4 If picture format not supported an error displayed
Post condition	Profile picture successfully updated
Alternate post condition	Profile picture not changed

Table 27 Use Case Change Password

Name	Change Password
Use-Case ID	UCJ-28
Priority	High
Primary Actor	Organization
Other participating Actor(s)	N/A
Description	This use case describe the event when any user wants to change his password.
Pre-condition	1. User must be logged in
Trigger	This use case initiated when a user going to change his password.
Typical flow of events	<ol style="list-style-type: none"> 1. User open his profile 2. Click on setting option 3. Select update profile picture option 4. User provide old password 5. User provide new password 6. User re-type new password 7. System match old password 8. System update password 9. System display acknowledgement message
Alternate flow of events	N/A
Post condition	Password successfully changed
Alternate post condition	Password not changed successfully

3.3 Domain Model

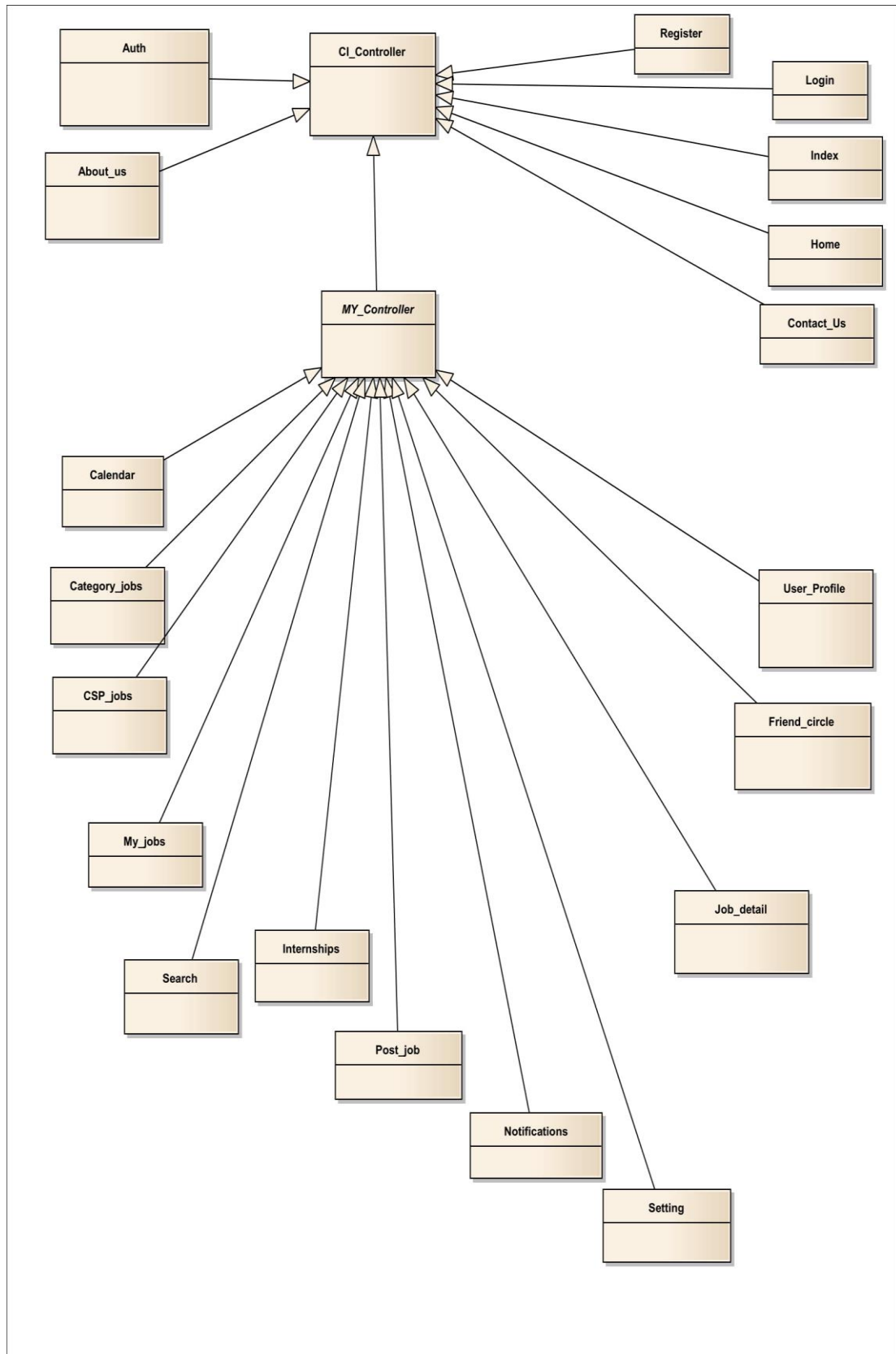


Figure 2 Domain Model User Side 1

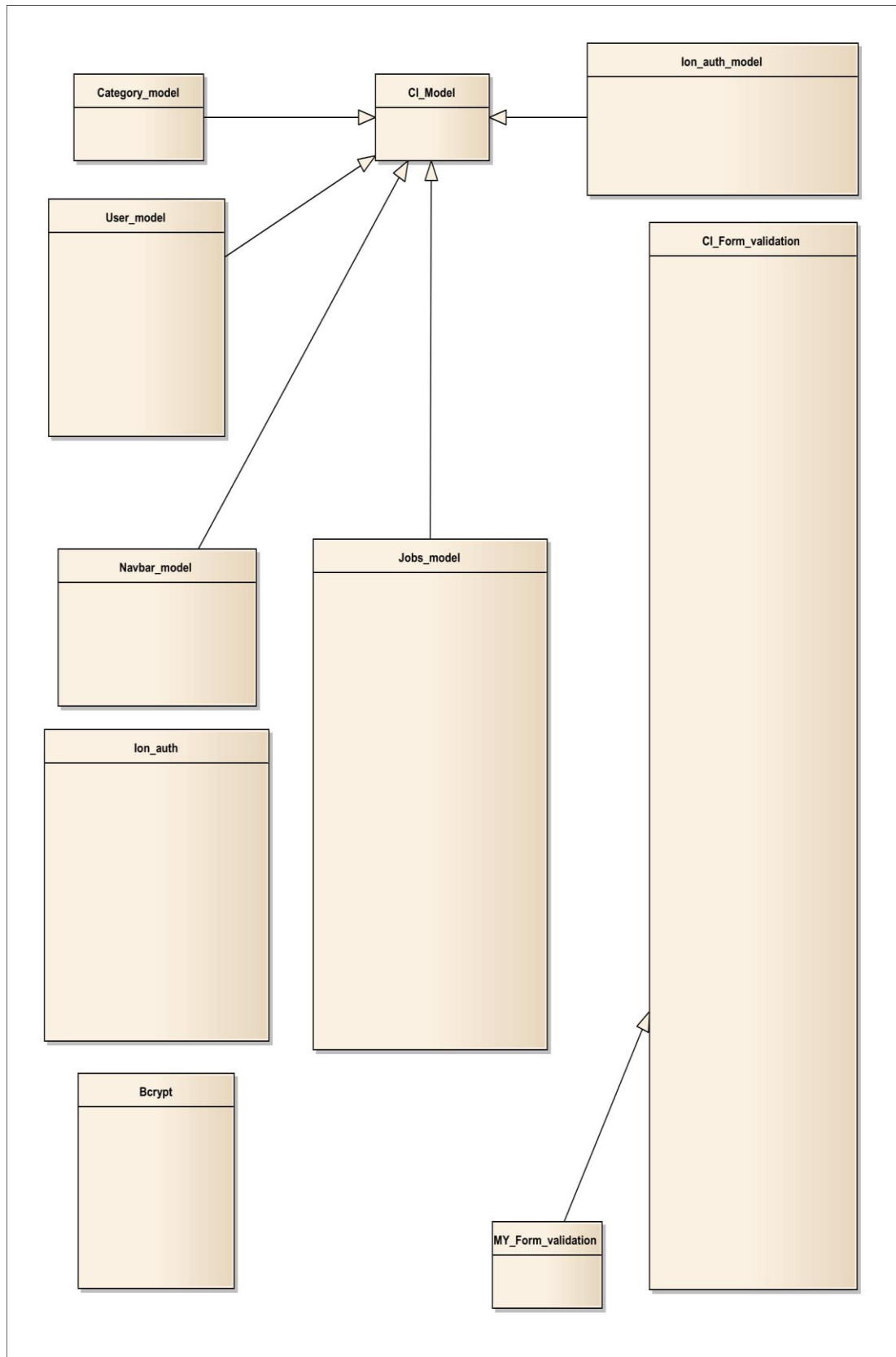


Figure 3 Domain Model User Side 2

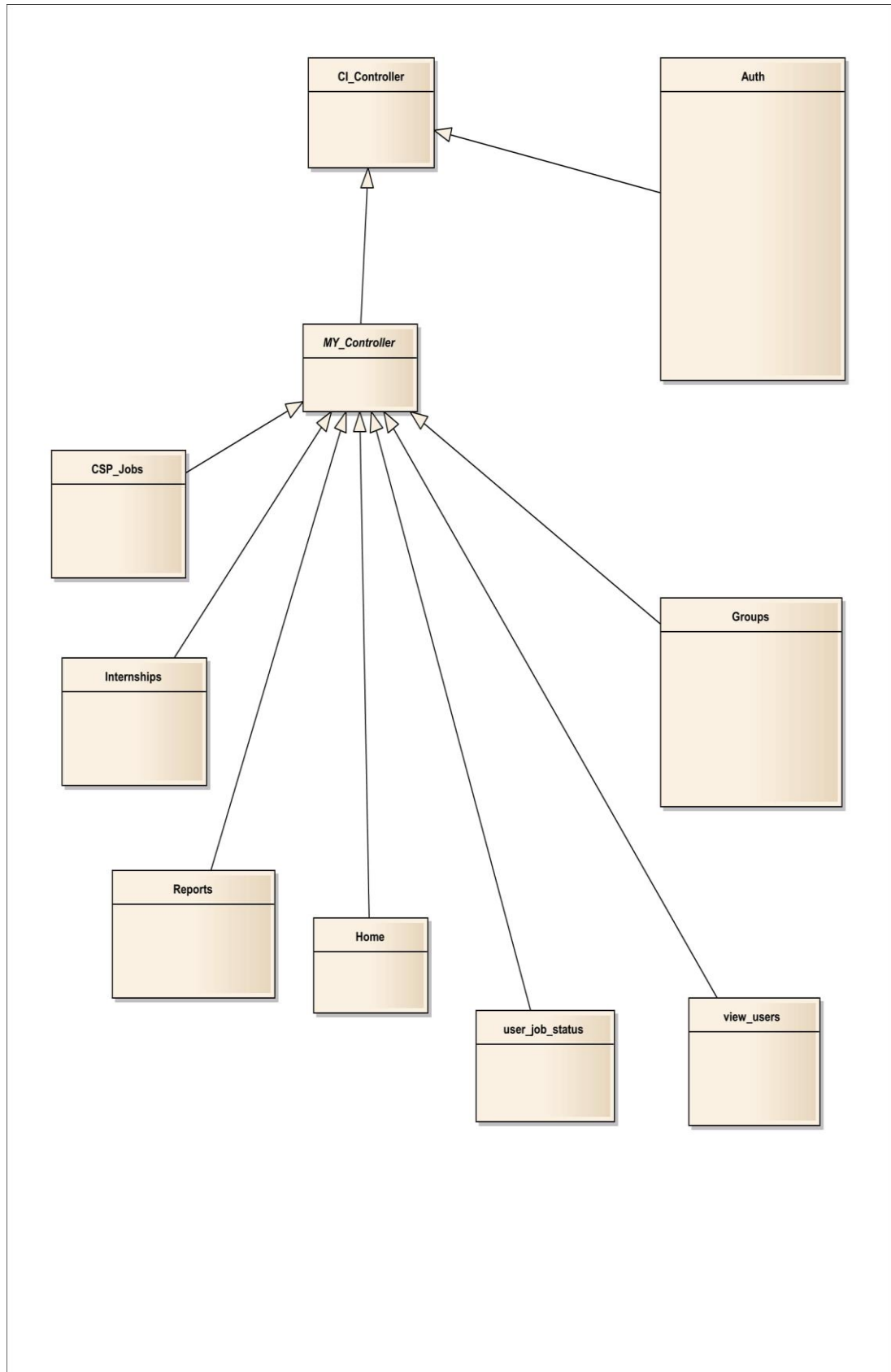


Figure 4 Domain Model Admin Side 1

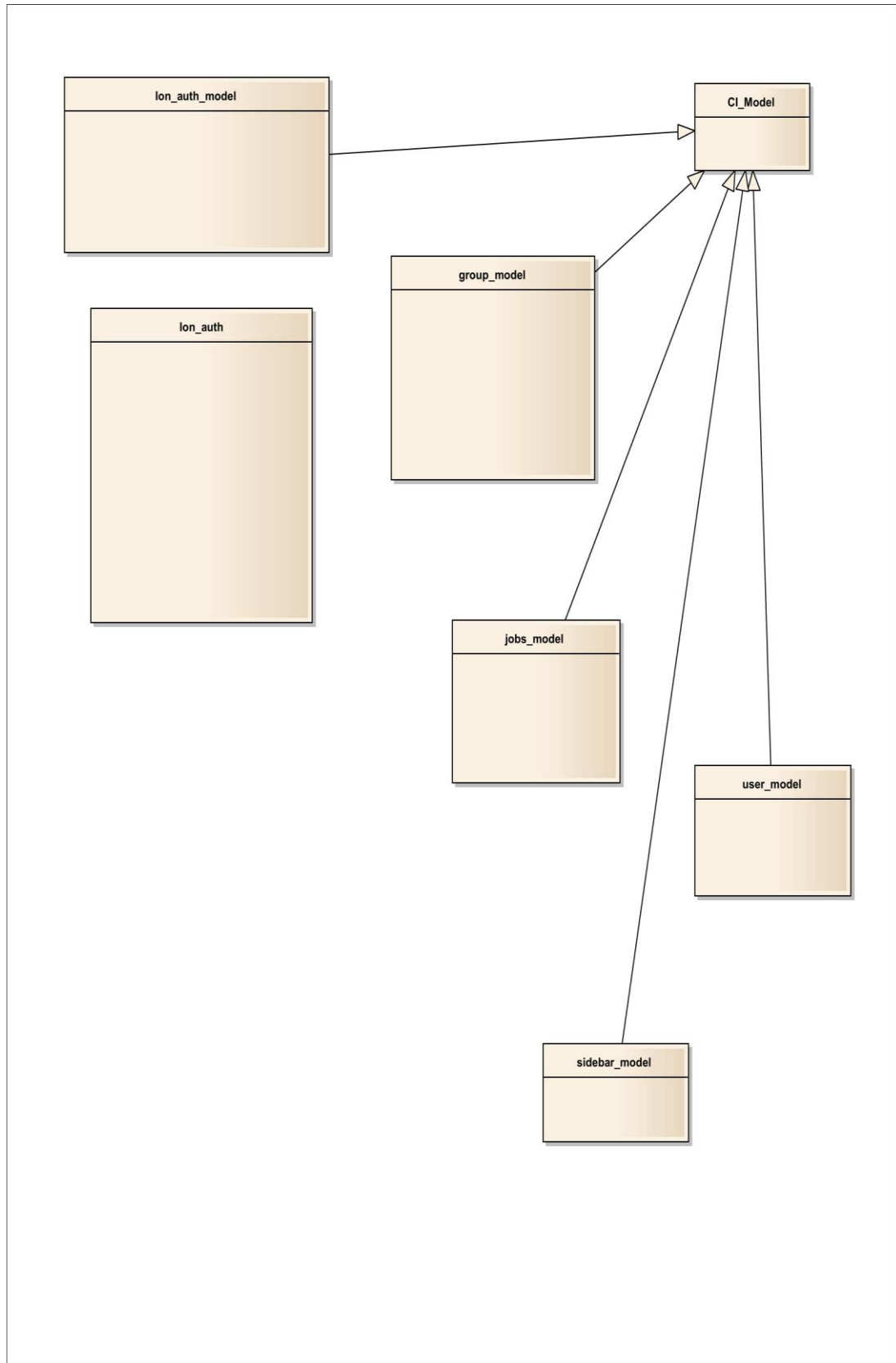


Figure 5 Domain Model Admin Side 2

3.4 Class Diagrams

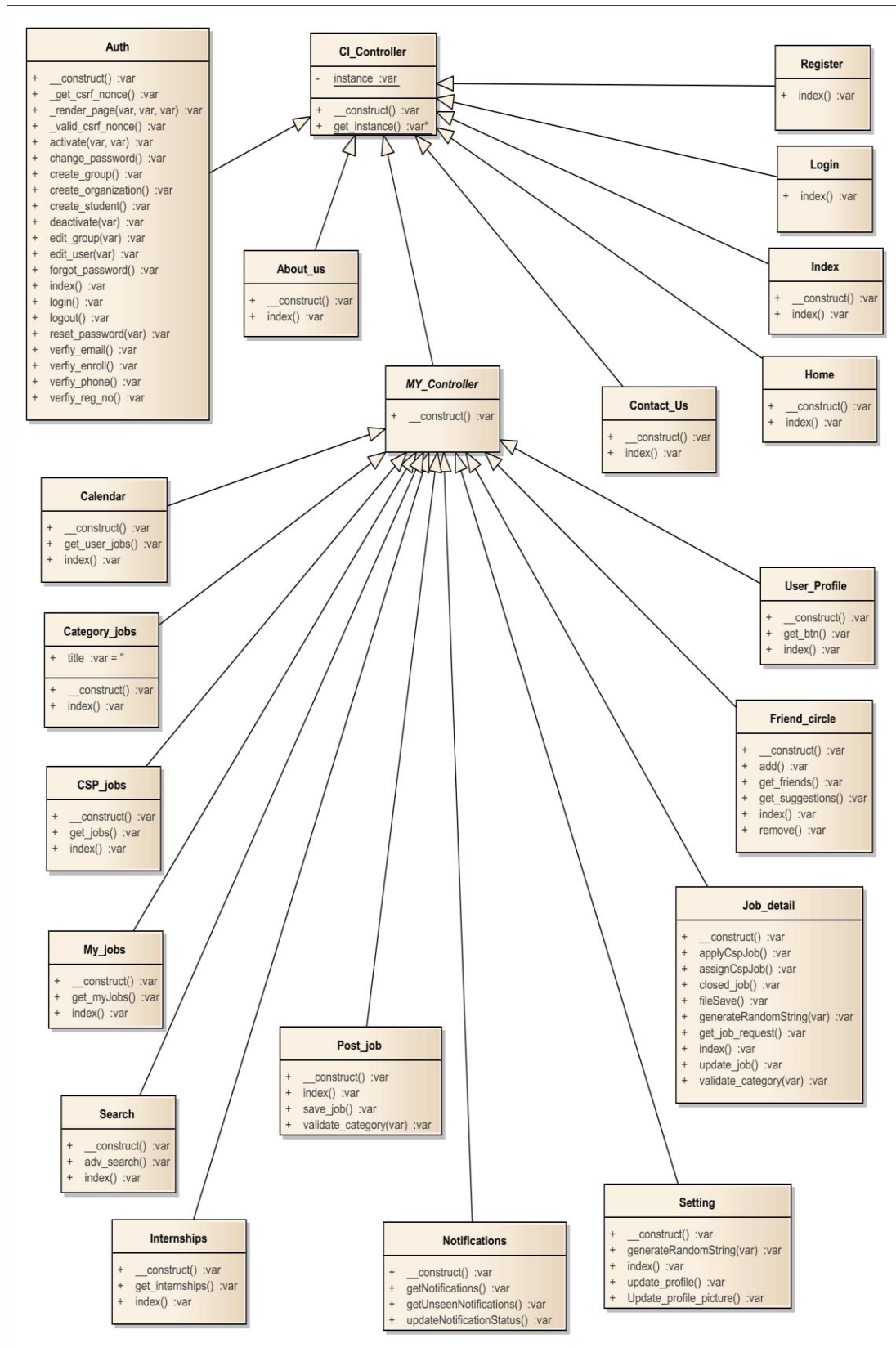


Figure 6 User Side Controller Class Diagram

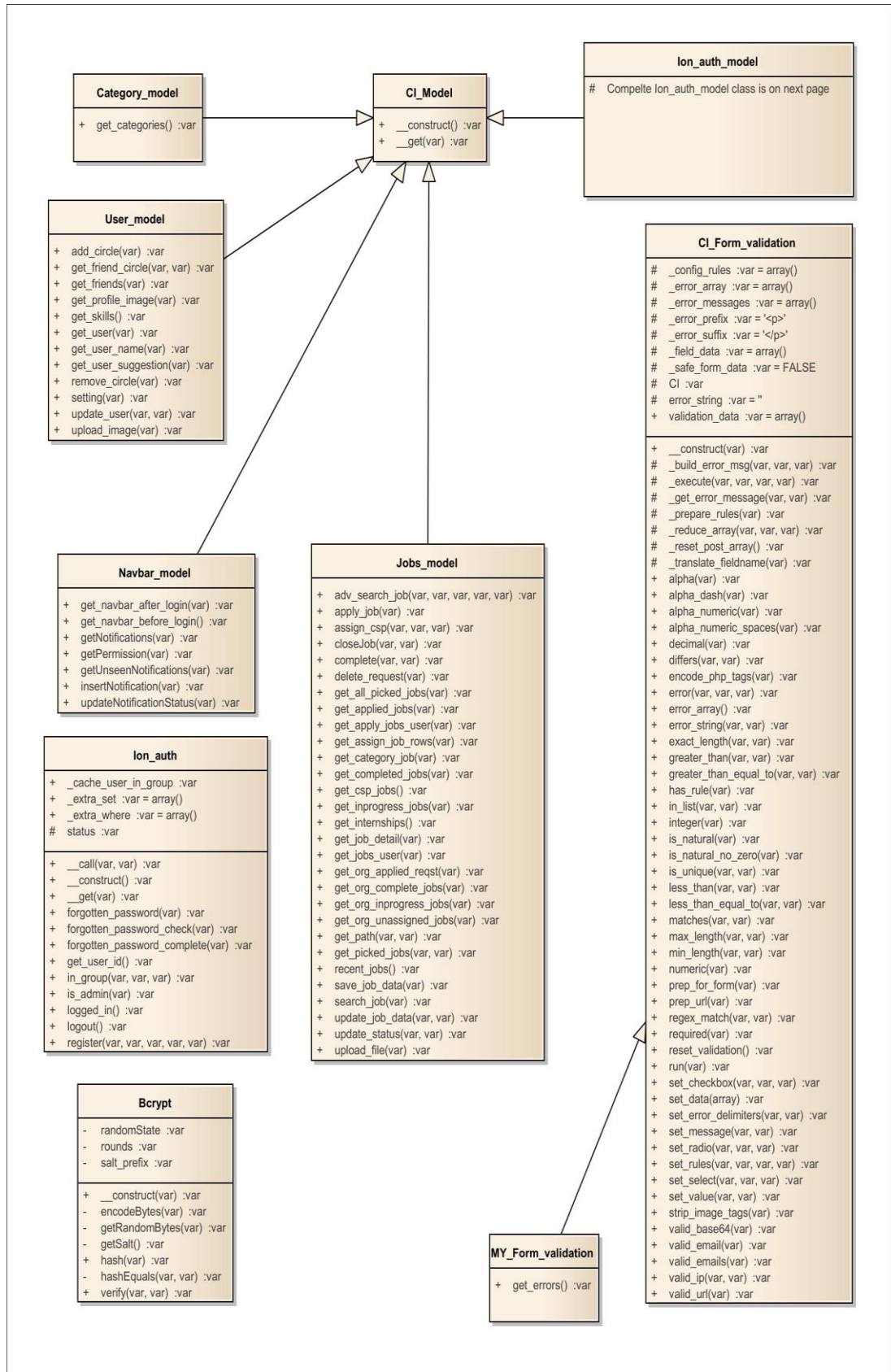


Figure 7 User Side Model Class Diagram

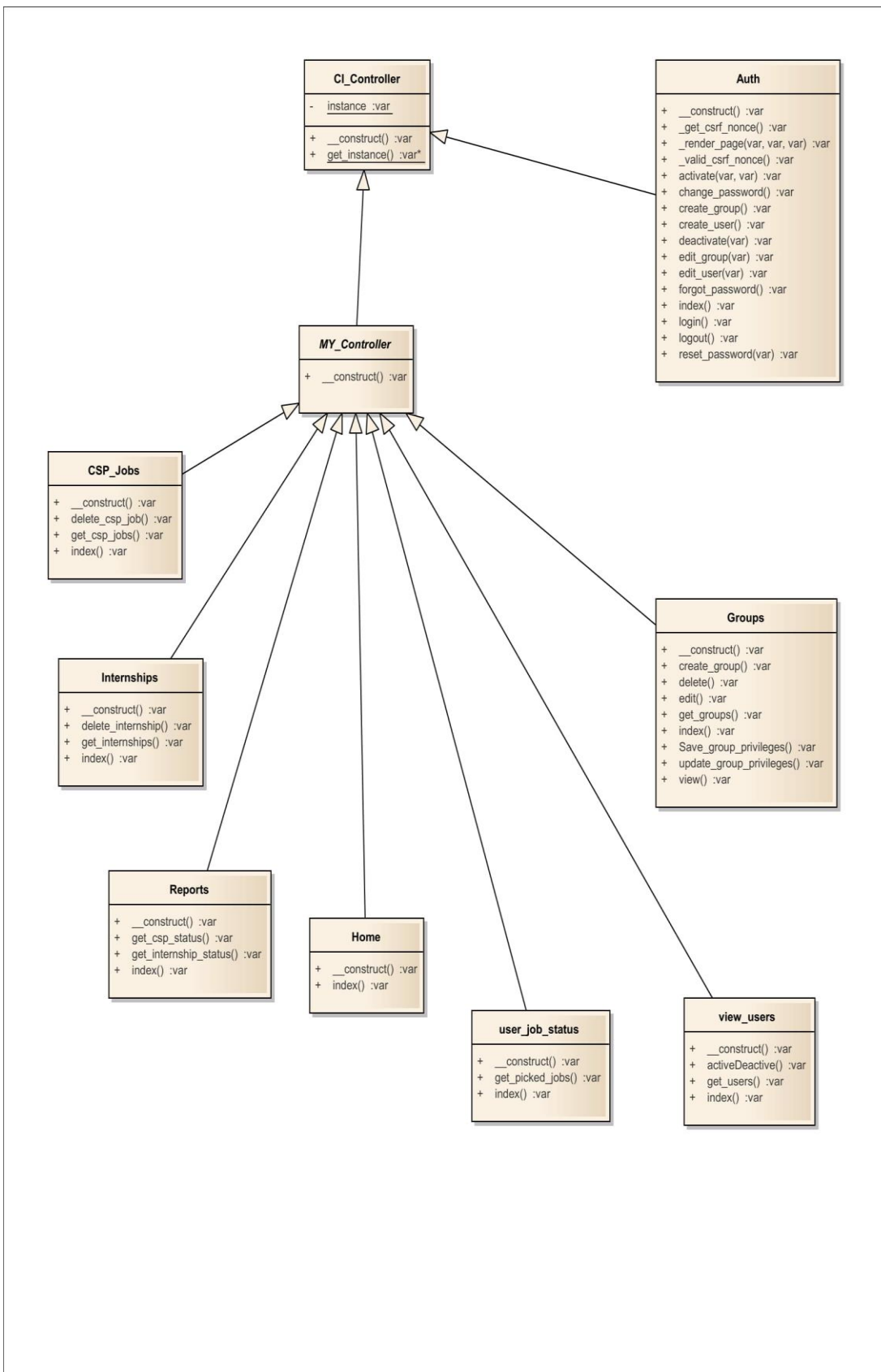


Figure 8 Admin Side Controller Class Diagram

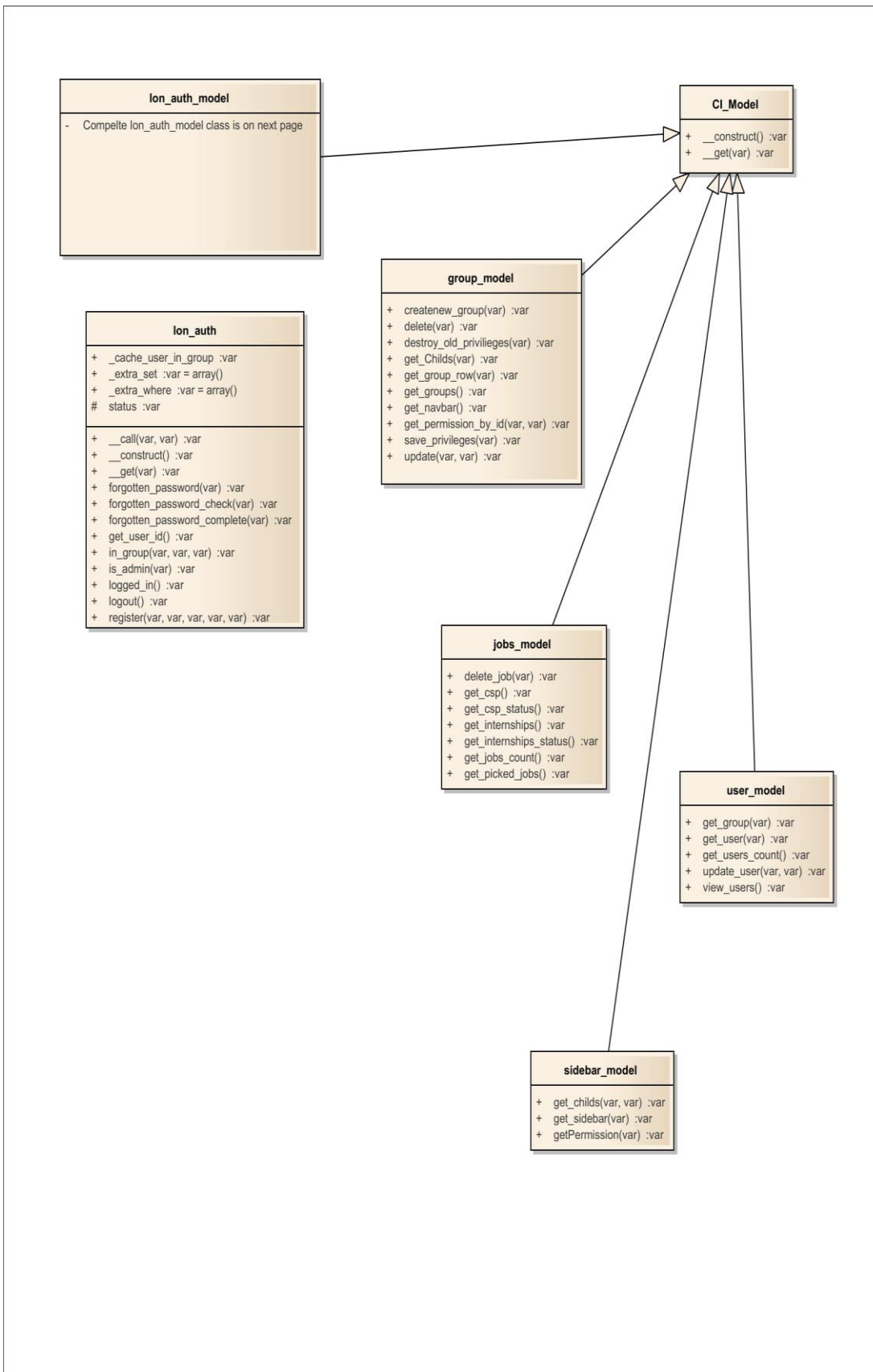


Figure 9 Admin Side Model Class Diagram

Ion_auth_model	
<pre> # _cache_groups :var = array() + _cache_user_in_group :var = array() # _ion_hooks :var + _ion_like :var = array() + _ion_limit :var = NULL + _ion_offset :var = NULL + _ion_order :var = NULL + _ion_order_by :var = NULL + _ion_select :var = array() + _ion_where :var = array() + activation_code :var # error_end_delimiter :var # error_start_delimiter :var # errors :var + forgotten_password_code :var + identity :var # messages :var + new_password :var # response :var = NULL + tables :var = array() </pre>	<pre> + hash_code(var) :var + hash_password(var, var, var) :var + hash_password_db(var, var, var) :var + identity_check(var) :var + increase_login_attempts(var) :var + is_max_login_attempts_exceeded(var, var) :var + is_time_locked_out(var, var) :var + is_unique_id(var) :var + like(var, var, var) :var + limit(var) :var + login(var, var, var) :var + login_remembered_user() :var + messages() :var + messages_array(var) :var + num_rows() :var + offset(var) :var + order_by(var, var) :var + recheck_session() :var + register(var, var, var, var, var) :var + remember_user(var) :var + remove_from_group(var, var) :var + remove_hook(var, var) :var + remove_hooks(var) :var + reset_password(var, var) :var + result() :var + result_array() :var + row() :var + row_array() :var </pre>
<pre> + __construct() :var # _call_hook(var, var) :var # _filter_data(var, var) :var # _prepare_ip(var) :var + activate(var, var) :var + add_to_group(var, var) :var + add_user(var) :var + change_password(var, var, var) :var + clear_errors() :var + clear_forgotten_password_code(var) :var + clear_login_attempts(var, var, var) :var + clear_messages() :var + create_group(var, var, var) :var + deactivate(var) :var + delete_group(var) :var + delete_user(var) :var + email_check(var) :var + errors() :var + errors_array(var) :var + forgotten_password(var) :var + forgotten_password_complete(var, var) :var + get_attempts_num(var, var) :var + get_last_attempt_ip(var) :var + get_last_attempt_time(var, var) :var + get_users_groups(var) :var + group(var) :var + groups() :var </pre>	<pre> + salt() :var + select(var) :var + set_error(var) :var + set_error_delimiters(var, var) :var + set_hook(var, var, var, var, var) :var + set_lang(var) :var + set_message(var) :var + set_message_delimiters(var, var) :var + set_session(var) :var + set_user_group(var, var) :var + trigger_events(var) :var + update(var, array) :var + update_group(var, var, var) :var + update_last_login(var) :var + user(var) :var + username_check(var) :var + users(var) :var + verify_phone(var) :var + verify_email(var) :var + verify_id(var) :var + where(var, var) :var </pre>

3.5 Sequence Diagrams

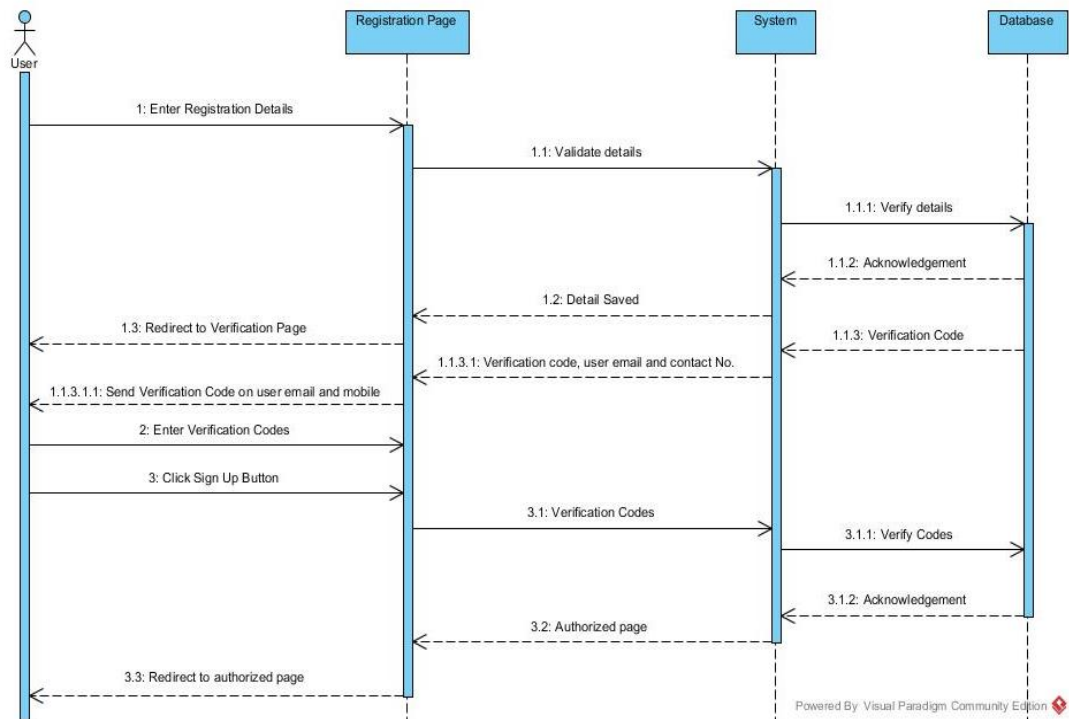


Figure 10 User Registration

Figure 10, describe the sequence of user registration process in which user enter registration details and then system validates those details from database. After validation, system sends verification code to user's email for email verification. At end user, enter that verification code to complete registration process.

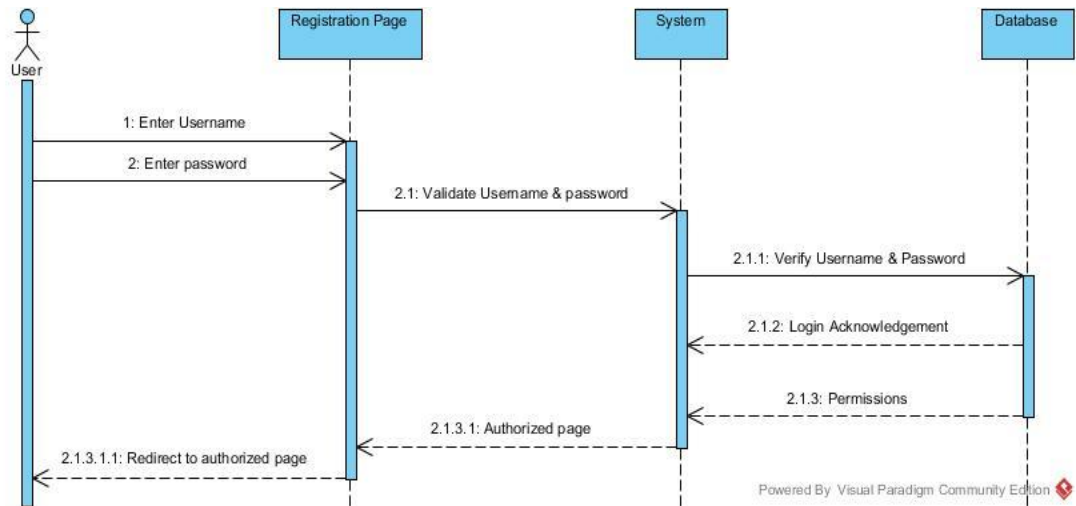


Figure 11 User login

Figure 11, describe the sequence of user login process in which user enter login credential and system verify users credential from database. If credentials validate from database system redirect user to authorized page.

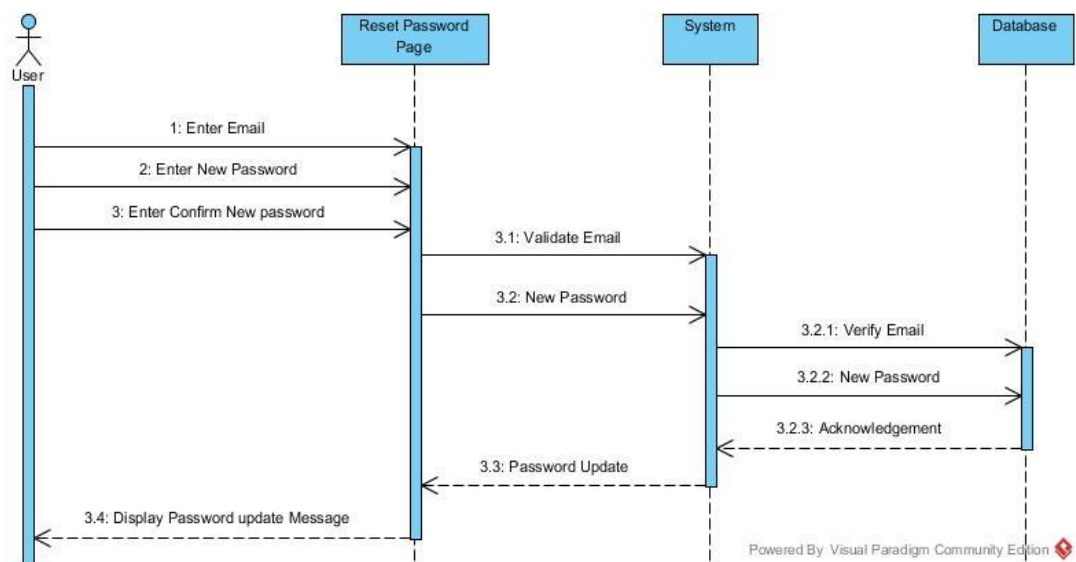


Figure 12 Reset password

Figure 12, describe the sequence of reset password process in which user enter email id and new password then system validate email id and update new password in database and display successful message on user view.

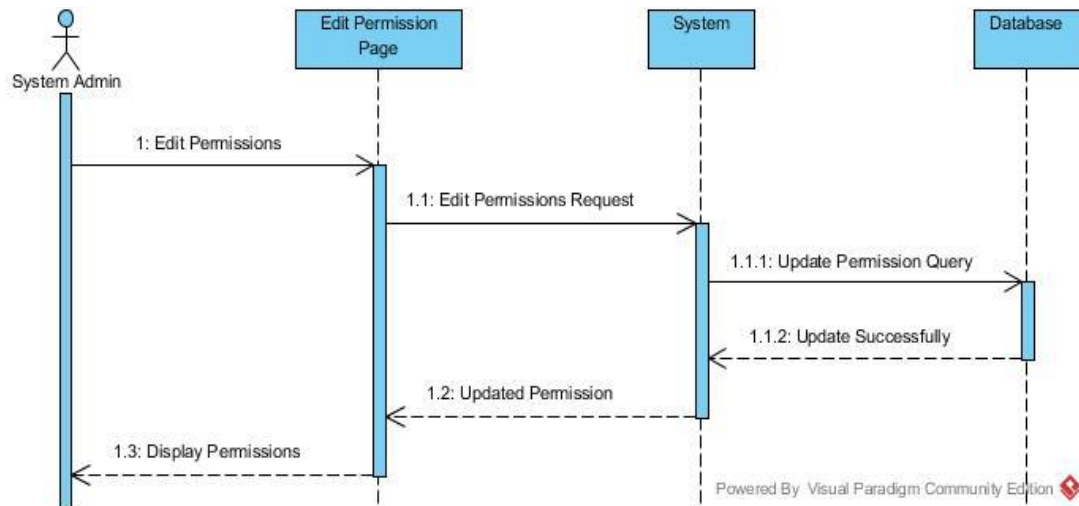


Figure 13 Edit User Permissions

Figure 13, describe the sequence of Edit User Permissions process in which system admin edit user's permission and system update in database.

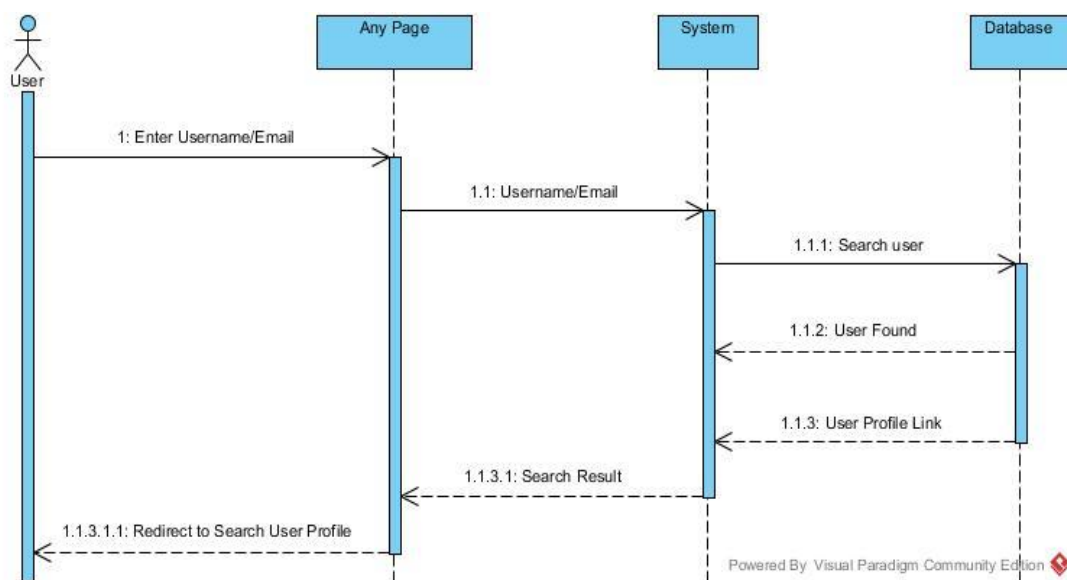


Figure 14 User search

Figure 14, describe the sequence of user search process in which user enter the target user's username or email id then system check that id or username in database and display the result.

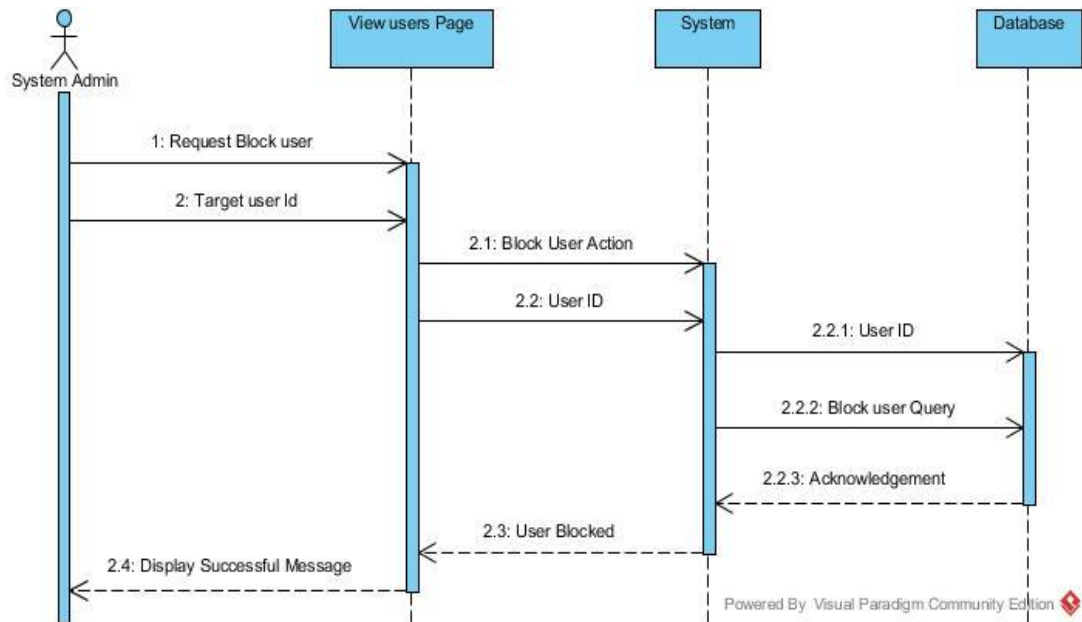


Figure 15 Block user

Figure 15, describe the sequence of block user process in which system admin enter the target user's id then system find it from database and update its status to blocked.

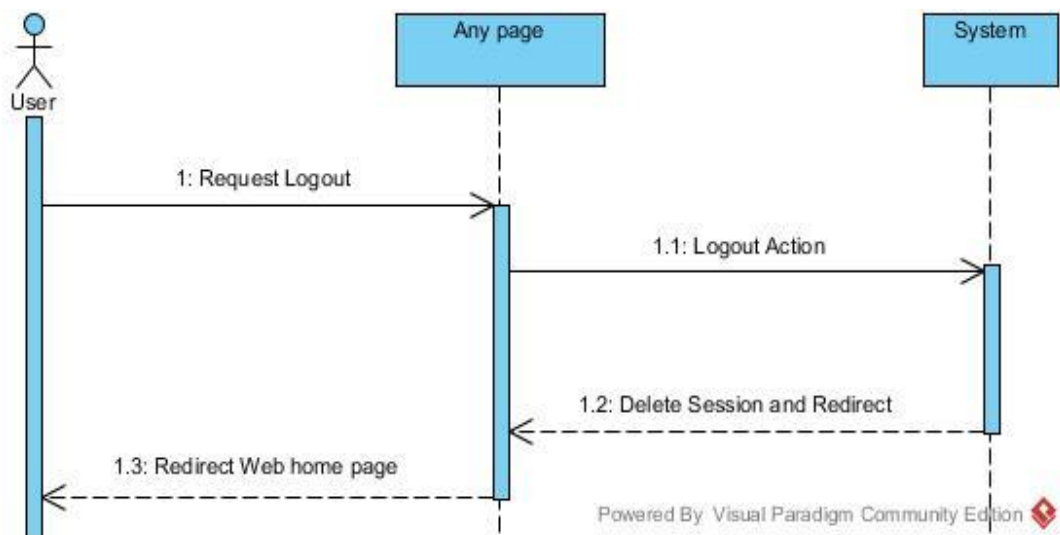


Figure 16 Logout

Figure 16, describe the sequence of logout process in which user click on logout button and system end the session and redirect user to home page.

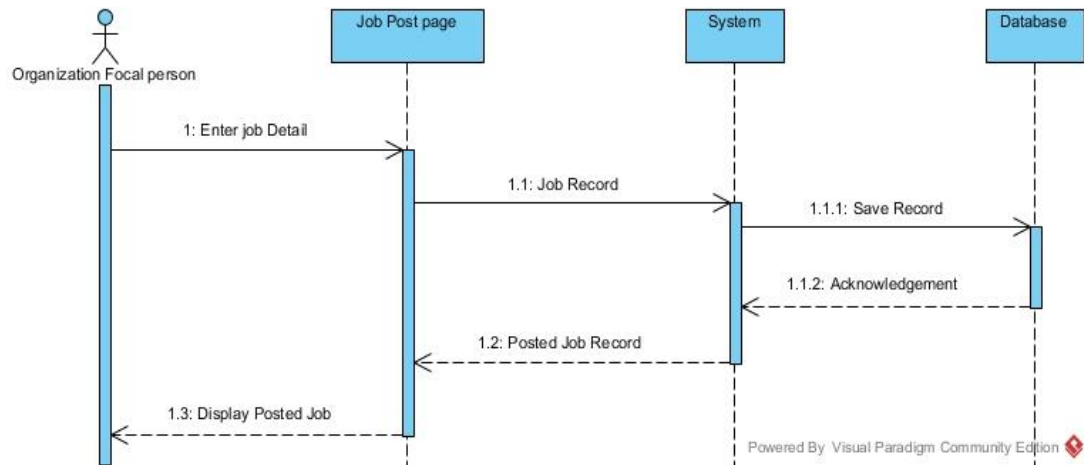


Figure 17 Job posting

Figure 17, describe the sequence of job posting process in which user enter the internship details then system save record in database and send acknowledge message to user.

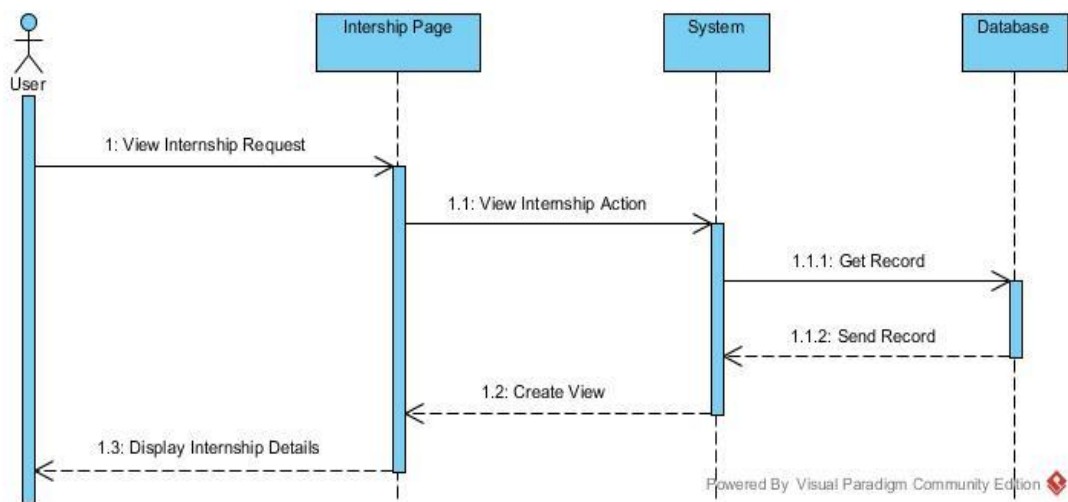


Figure 18 View internship

Figure 18, describe the sequence of view internship process in which user click on the specific job and then system load the details from database on view internship page.

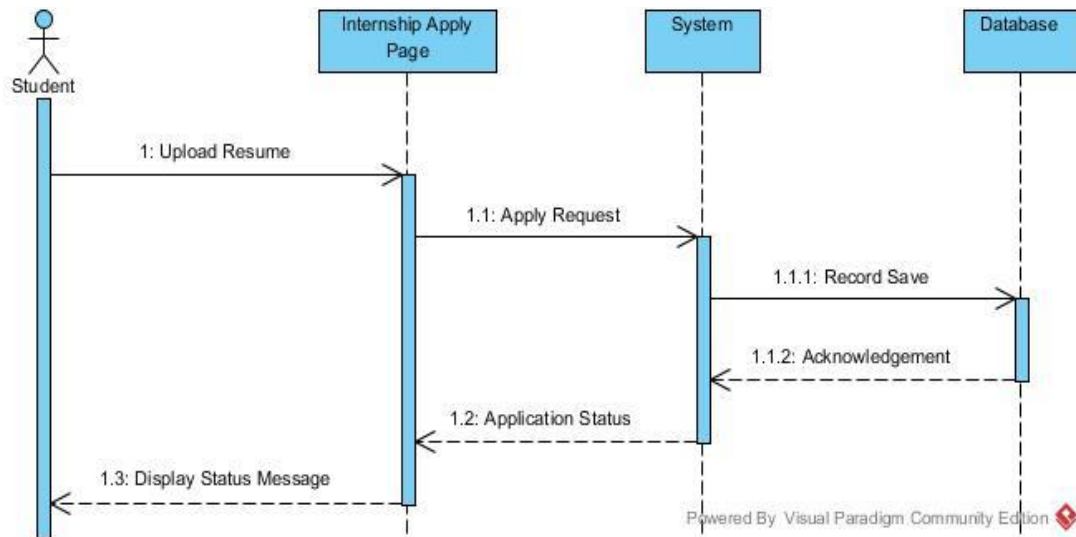


Figure 19 Apply for internship

Figure 19, describe the sequence of apply for internship process in which user select the specific job and enter the required information to apply and submit. System update information in database and display status message.

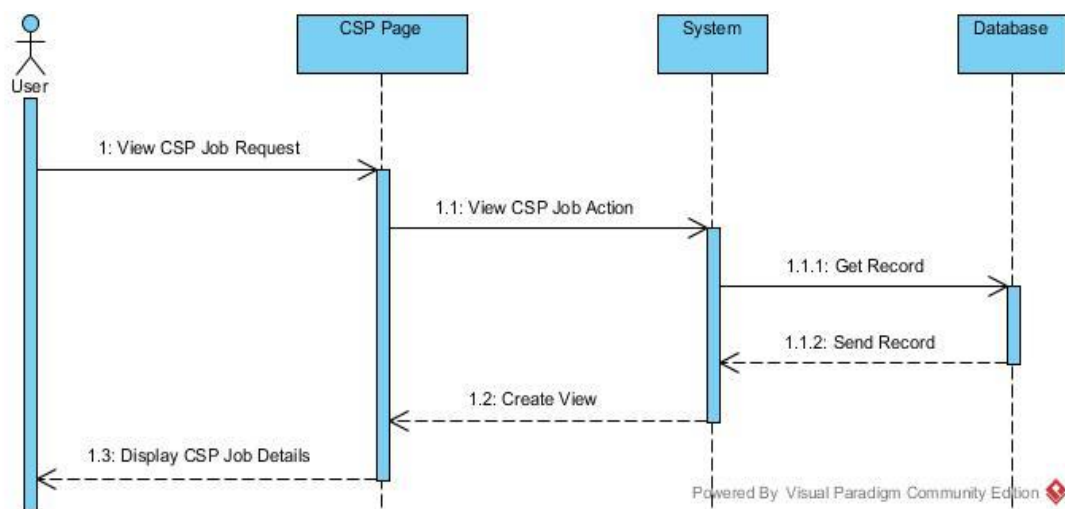


Figure 20 View CSP job

Figure 20, describe the sequence of view CSP job process in which user click on the specific job and then system load the details from database on view CSP Jobs page.

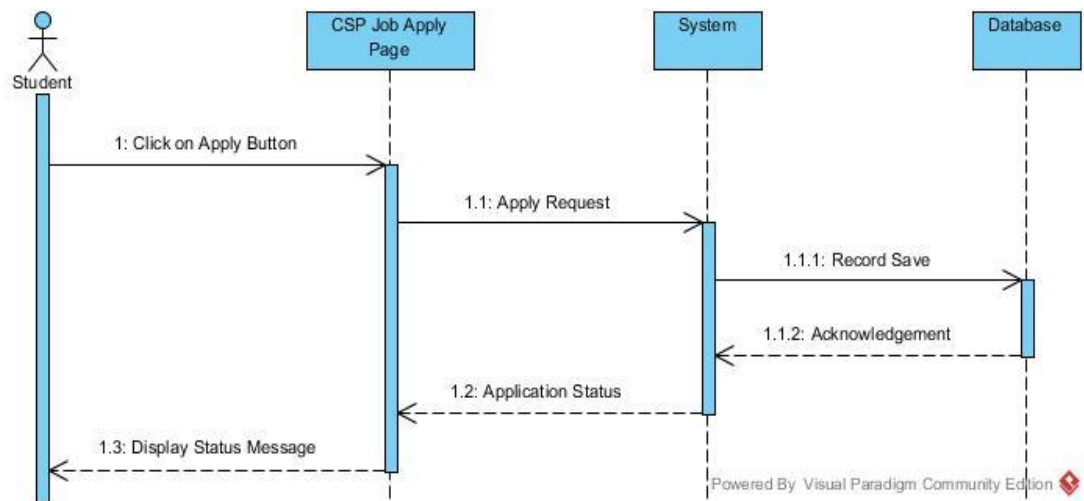


Figure 21 Apply for CSP job

Figure 21, describe the sequence of apply for CSP job process in which user select the specific job and click on apply button. System assign targeted job to user and update in database and display status message.

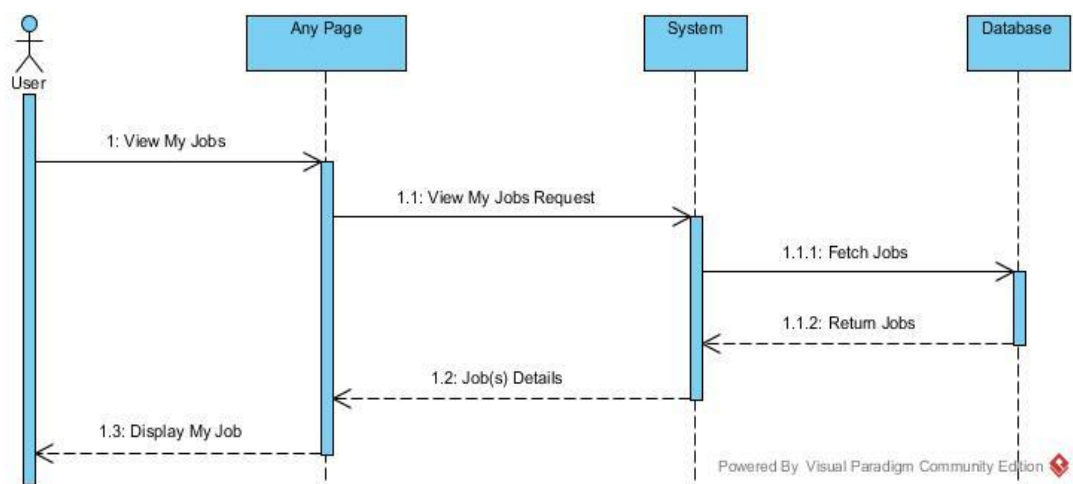


Figure 22 View my jobs

Figure 22, describe the sequence of view my jobs process in which user click on View My Jobs button and then system load all completed and selected jobs from database.

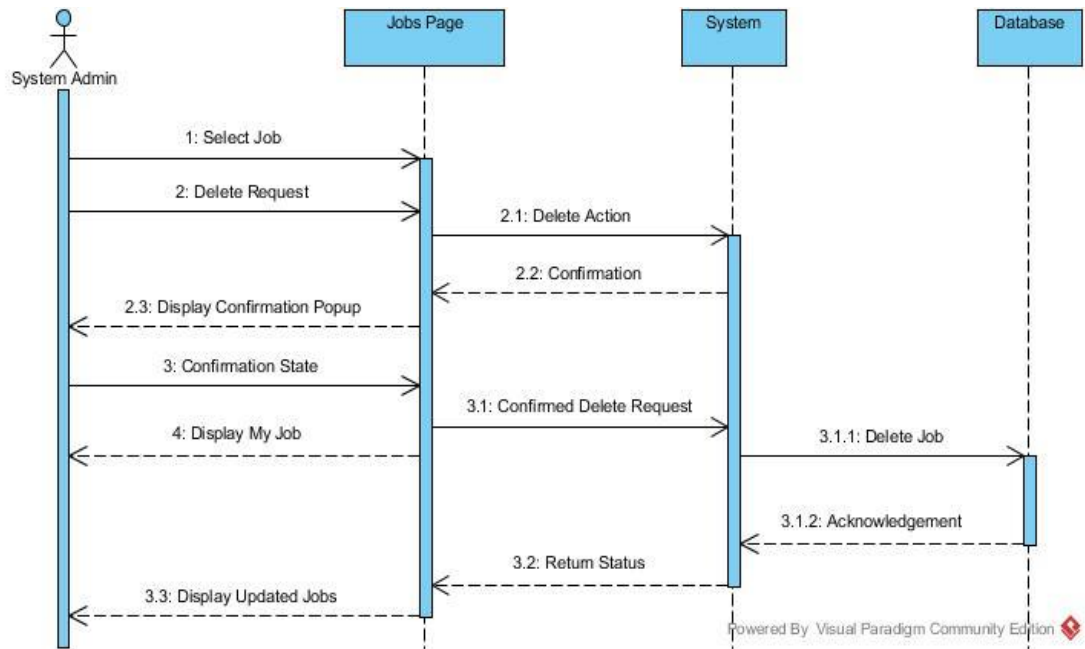


Figure 23 Delete Job

Figure 23, describe the sequence of Delete Job process in which system admin select a job and click on delete button then system locate that job in database and remove the record from database.

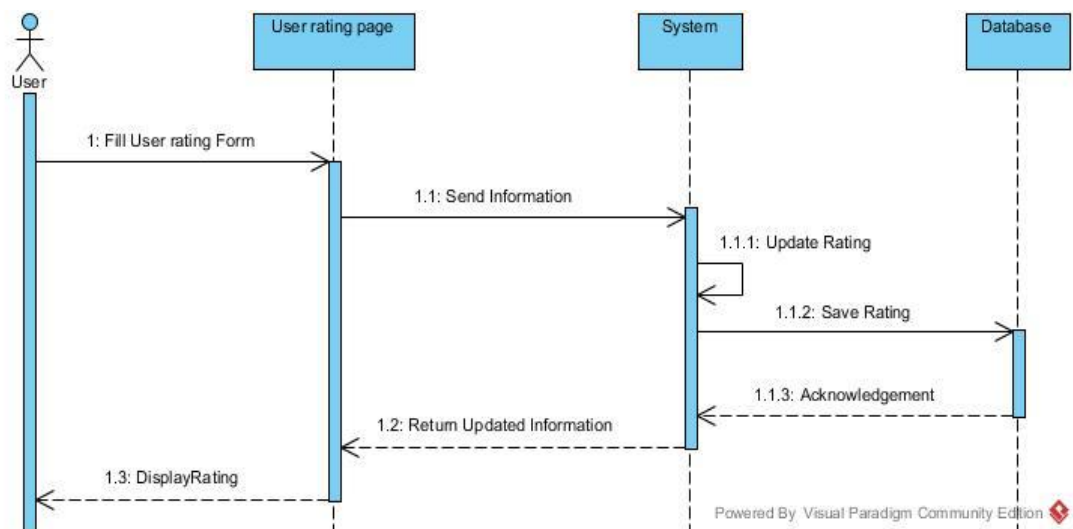


Figure 24 User rating

Figure 24, describe the sequence of user rating process in which user fill the rating form and submit then system calculate user rating on various parameters and update in database.

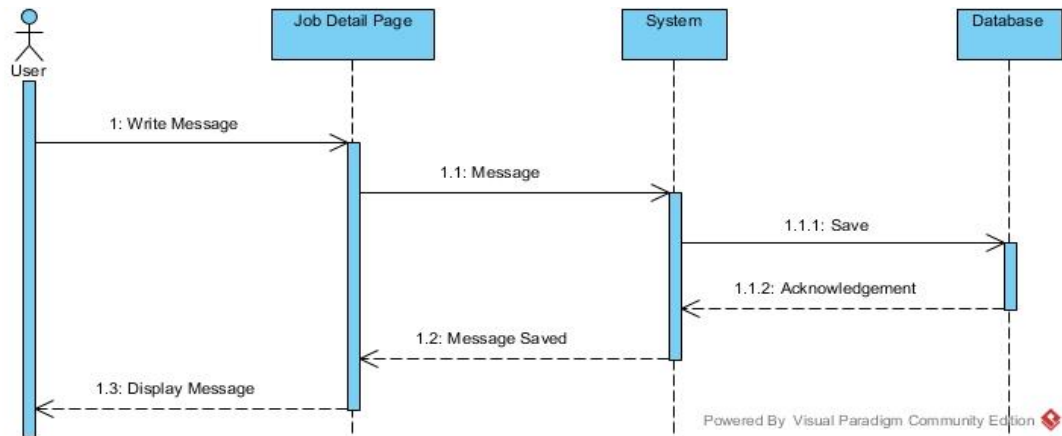


Figure 25 Discussion

Figure 25, describe the sequence of Discussion process in which user can discuss about their posted job with other users in text box type message and post it then system send notification to target user(s).

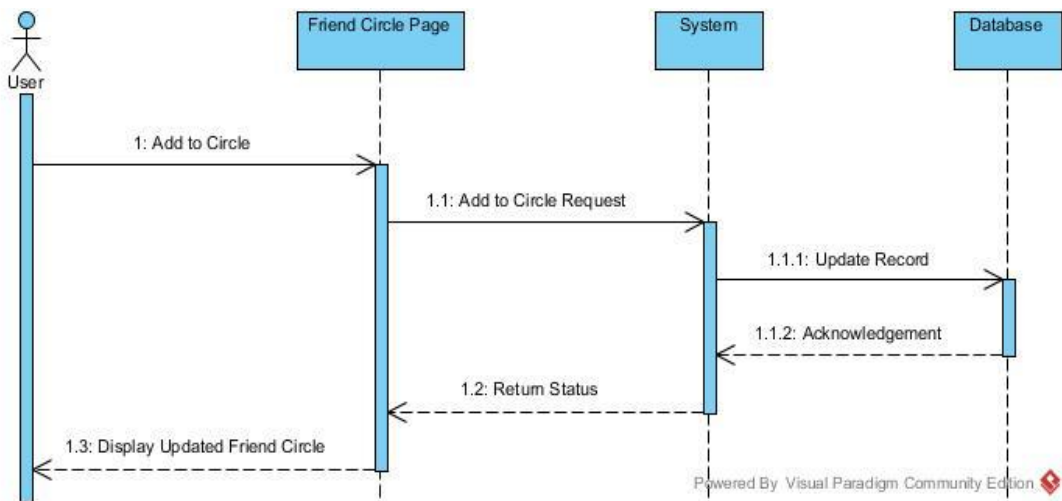


Figure 26 Add to Friend circle

Figure 26, describe the sequence of add user in Friend Circle process in which user click on add to circle button and then system update record in database and display target user's id in user's friend circle.

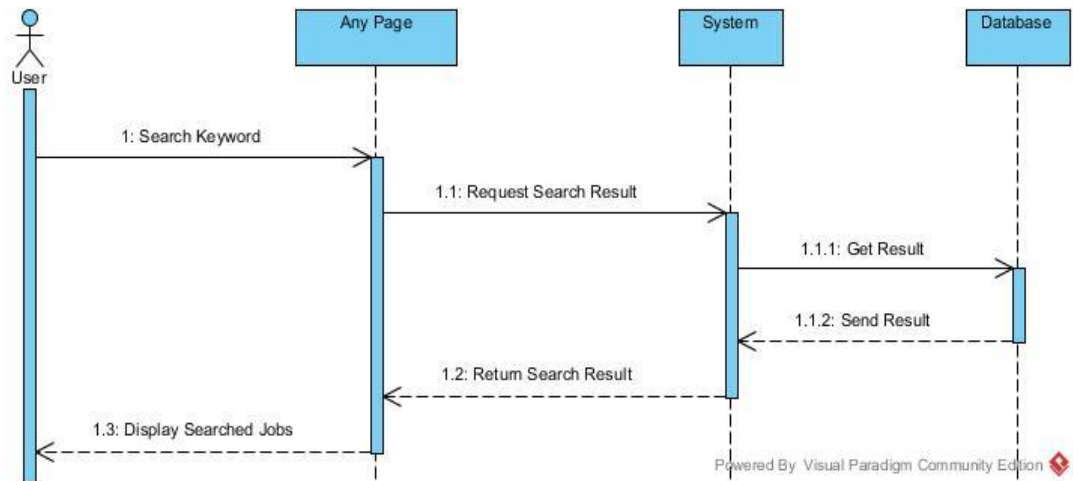


Figure 27 Search

Figure 27, describe the sequence of search process in which user enter the keyword to search then system check records against that keyword in database and display the result.

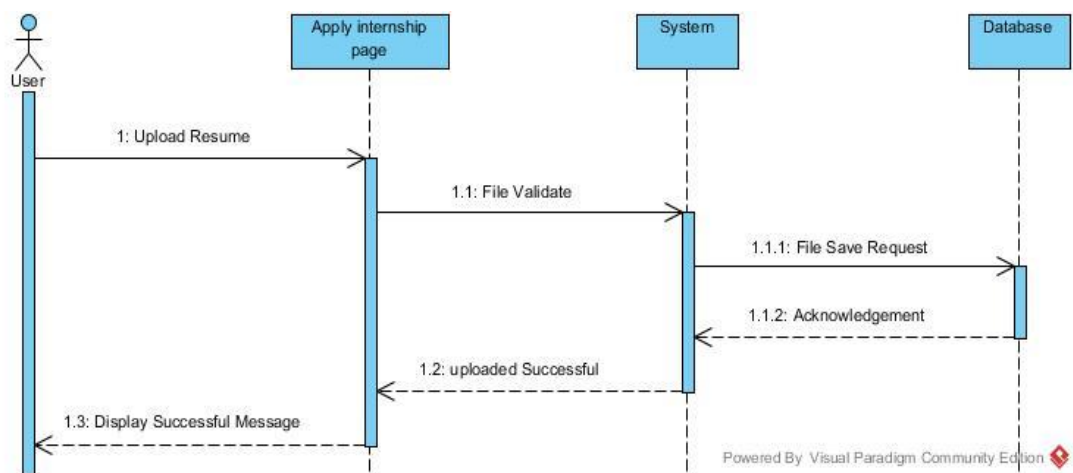


Figure 28 Upload resume

Figure 28, describe the sequence of uploading resume for internship process in which user upload resume then system validate the format and save file on server and update file location in database.

3.6 Collaboration Diagrams

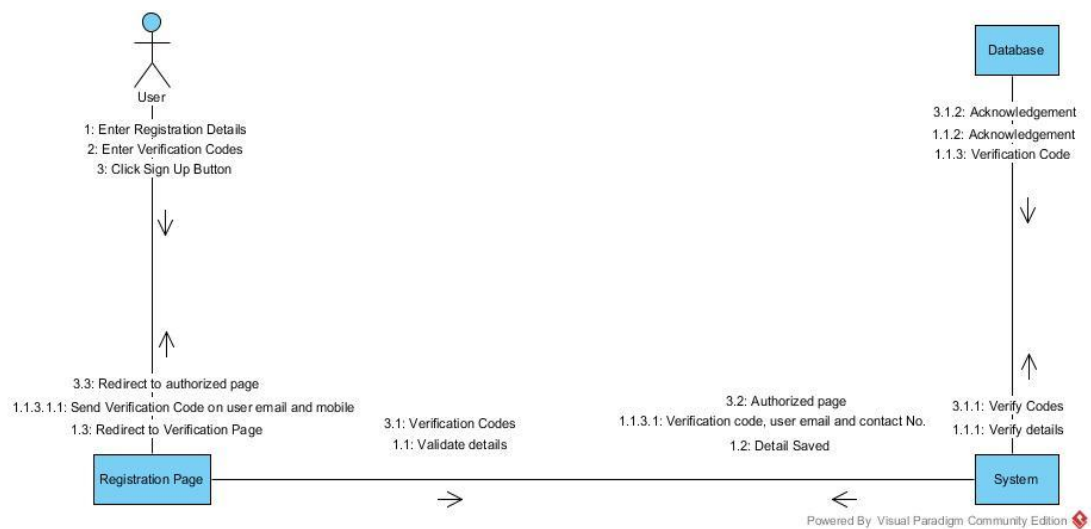


Figure 29 User Registration

- User class instance was created
- Bahria University/Organization database connection instance was created
- JOIN database connection instance was created
- User instance matched user entered ID with Bahria/Organization database instance
- JOIN database instance was saved required user detail from Bahria/Organization database instance and password entered by user

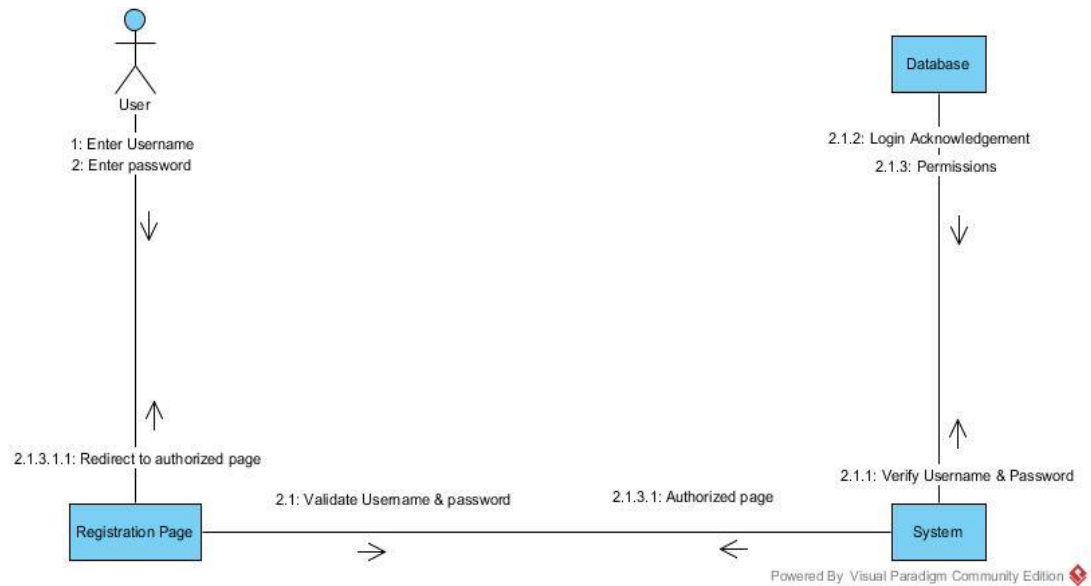


Figure 30 User Login

- User class instance was created
- JOIN database connection instance was created
- User class instance matched user entered id and password with JOIN database instance's each id and password respectively
- User session was created

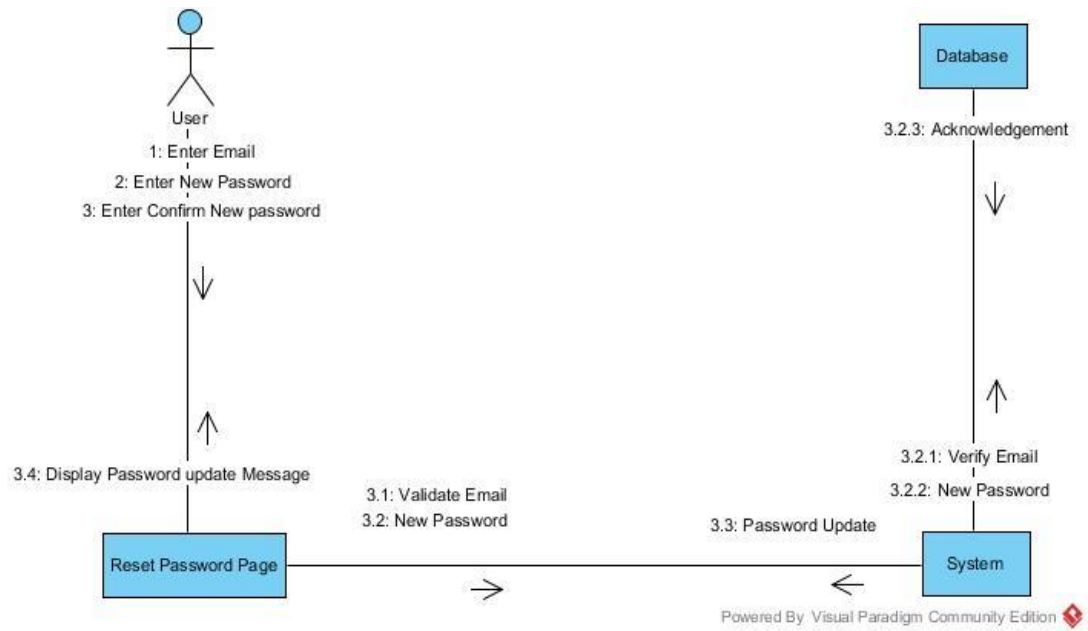


Figure 31 Change Password

- User class instance was created
- JOIN database connection instance was created
- User instance matched target user id with JOIN database instance
- JOIN database instance update the target user password

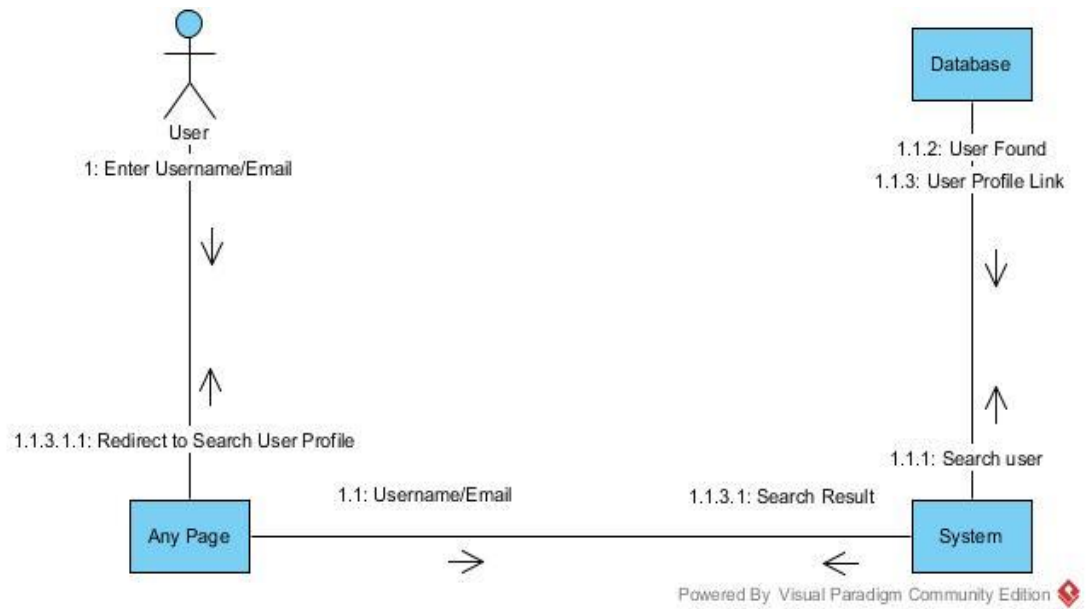


Figure 32 Search User

- User class instance was created
- JOIN database connection instance was created
- JOIN database instance matched each userId with user instance userId
- JOIN database returned user details to user instance

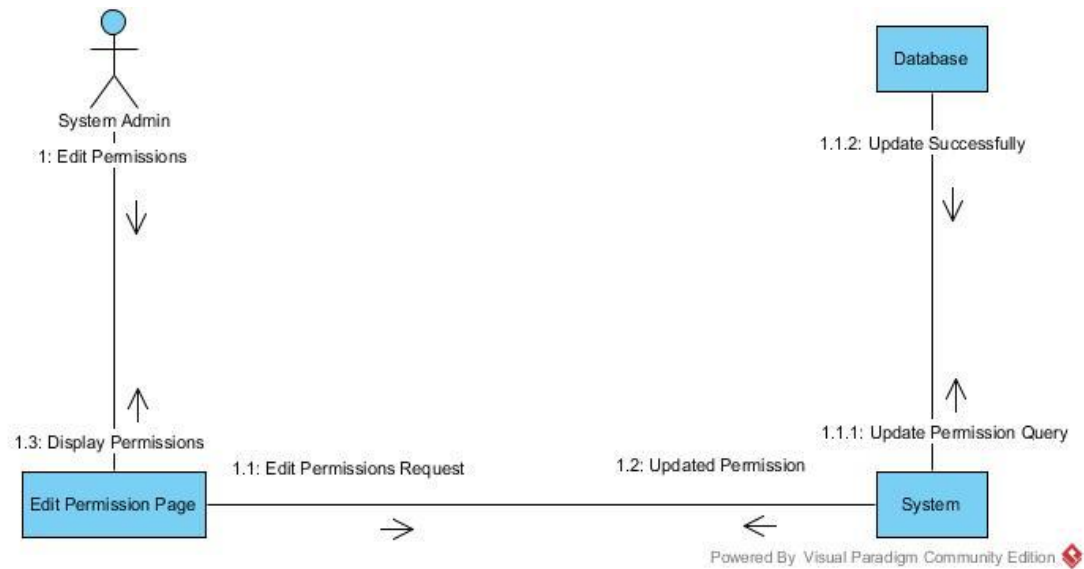


Figure 33 Edit Permission

- User class instance was created
- Group class instance was created
- NavBar class instance was created
- PermissionSet class instance was created
- JOIN database connection instance was created
- JOIN database instance gave groupId, NavBarId and PermissionSet matched with group instance id, sidebar instance id and put into permission set instance respectively given by user instance
- JOIN database instance updated permissionSet given by permissionSet instance

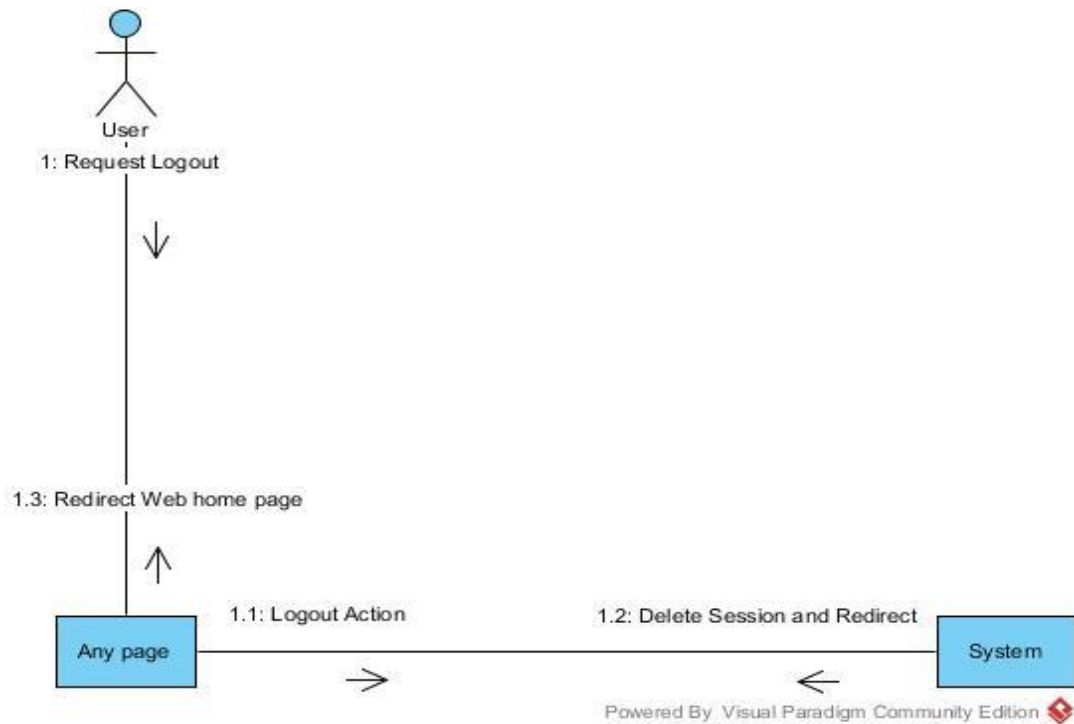


Figure 34 Logout

- User created instance and session was deleted
- JOIN database connection was disconnecting

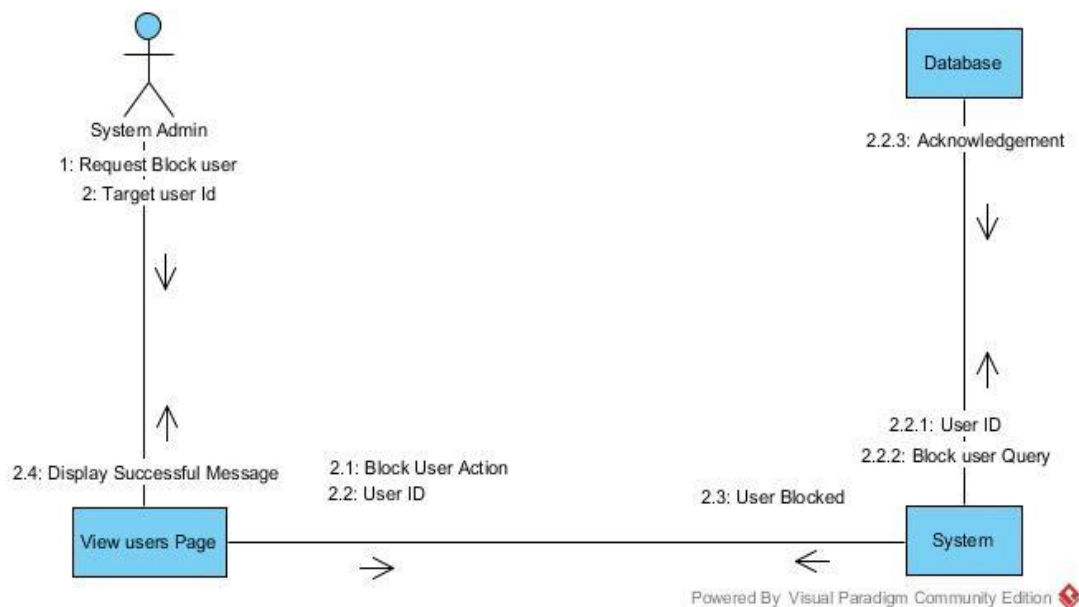


Figure 35 Block User

- User class instance was created
- JOIN database connection instance was created
- JOIN database instance matched each user Id with user instance user Id
- JOIN database returned user details to user instance
- Target user record was deleted by JOIN database instance

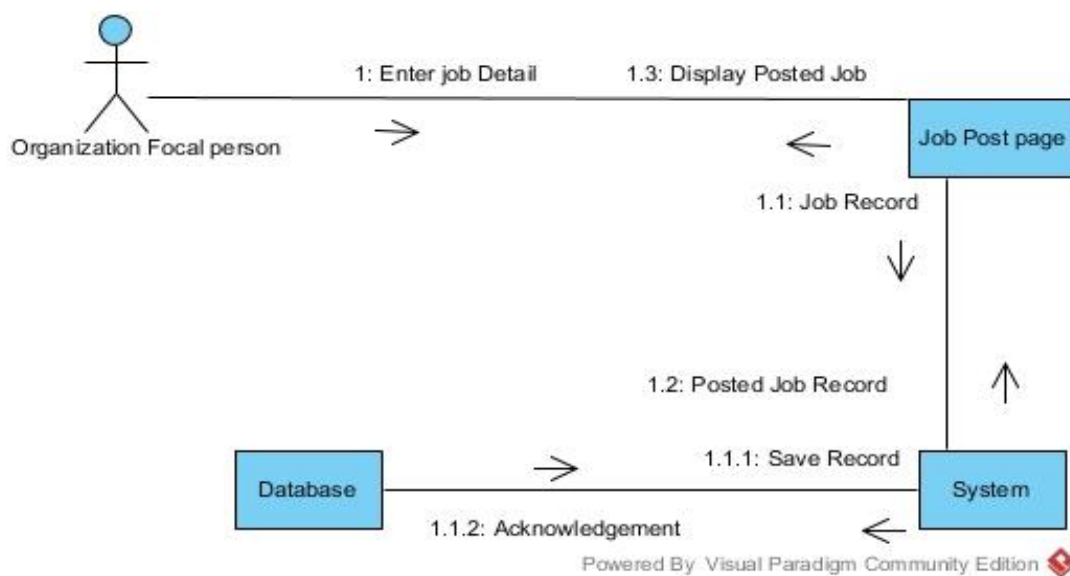


Figure 36 Post Job

- User class instance was created
- Internship class instance was created
- JOIN database connection instance was created
- User instance put internship details to internship instance
- Internship instance put details to JOIN database instance

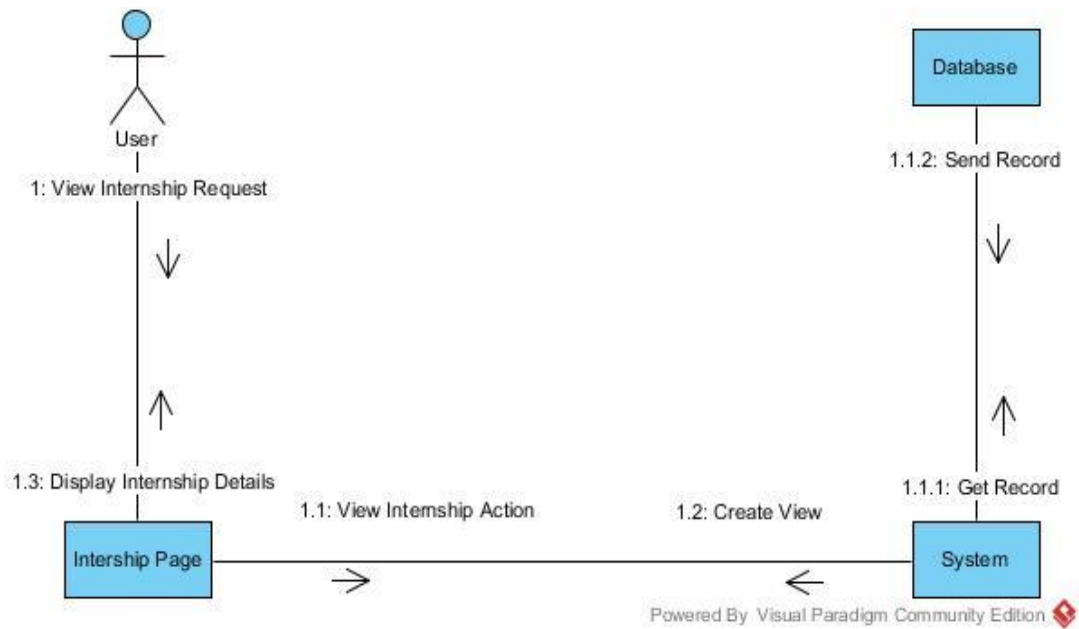


Figure 37 View Internship

- User instance was created
- Internship instance was created
- JOIN database instance was created
- User instance put internship id to internship instance
- Internship instance put internship id to JOIN database instance
- JOIN database instance put internship details to internship instance

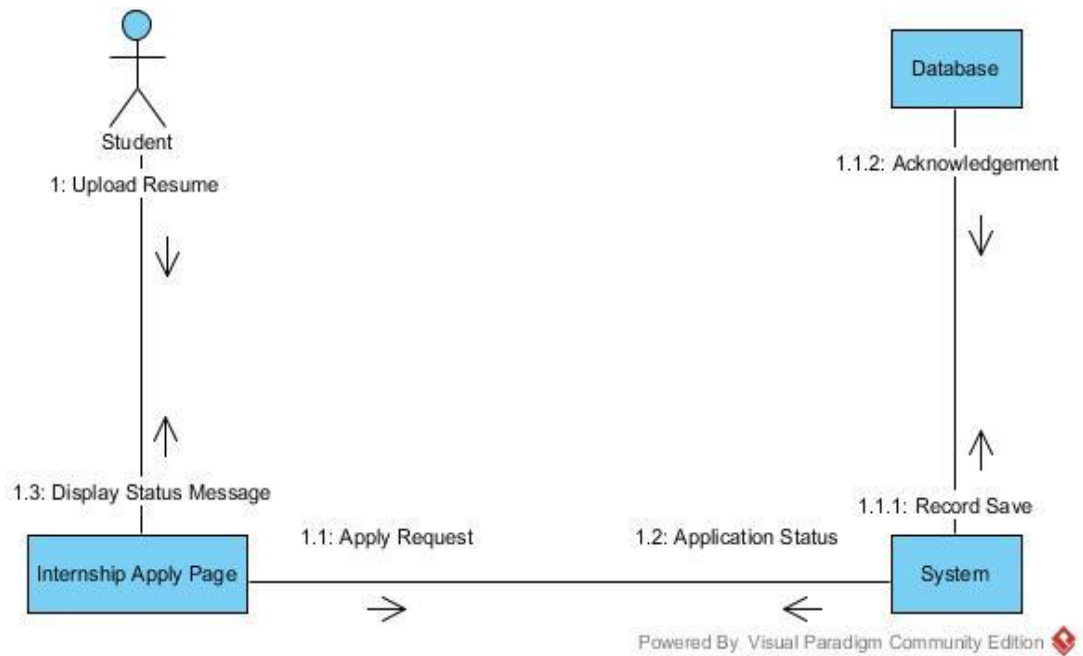


Figure 38 Upload Resume

- User class instance was created
- Attachment instance was created
- JOIN database connection instance was created
- User class instance put file details to Attachment instance
- Attachment instance put file details to JOIN database instance

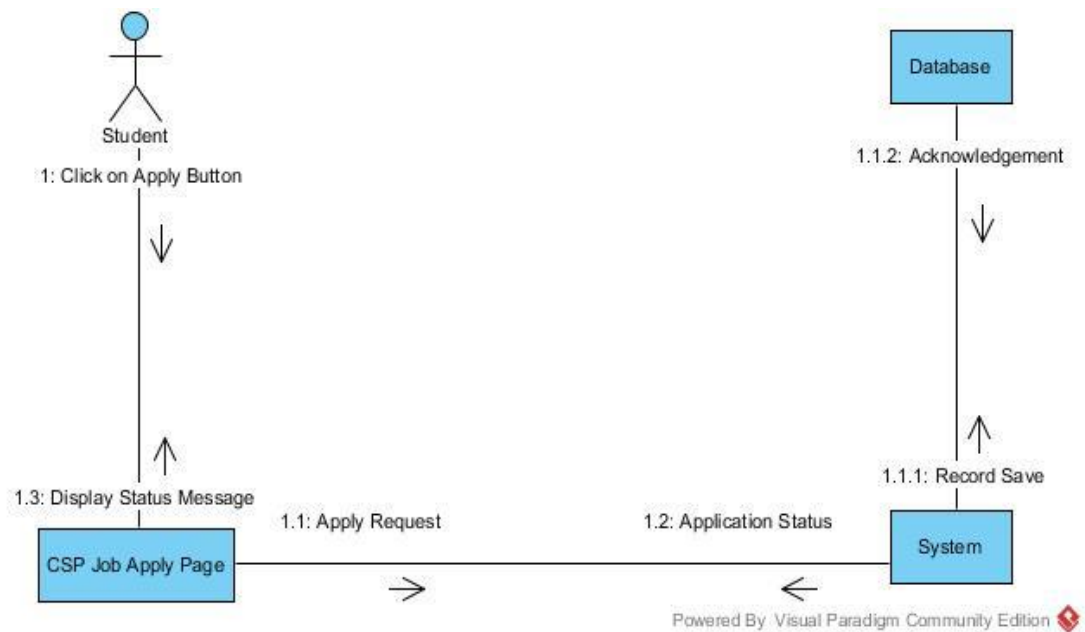


Figure 39 Apply for Job

- User class instance was created
- Job Post instance was created
- Picked jobs instance was created
- JOIN database connection instance was created
- User instance put user id and job post instance put job details to picked jobs instance
- Picked jobs instance put details in JOIN database instance

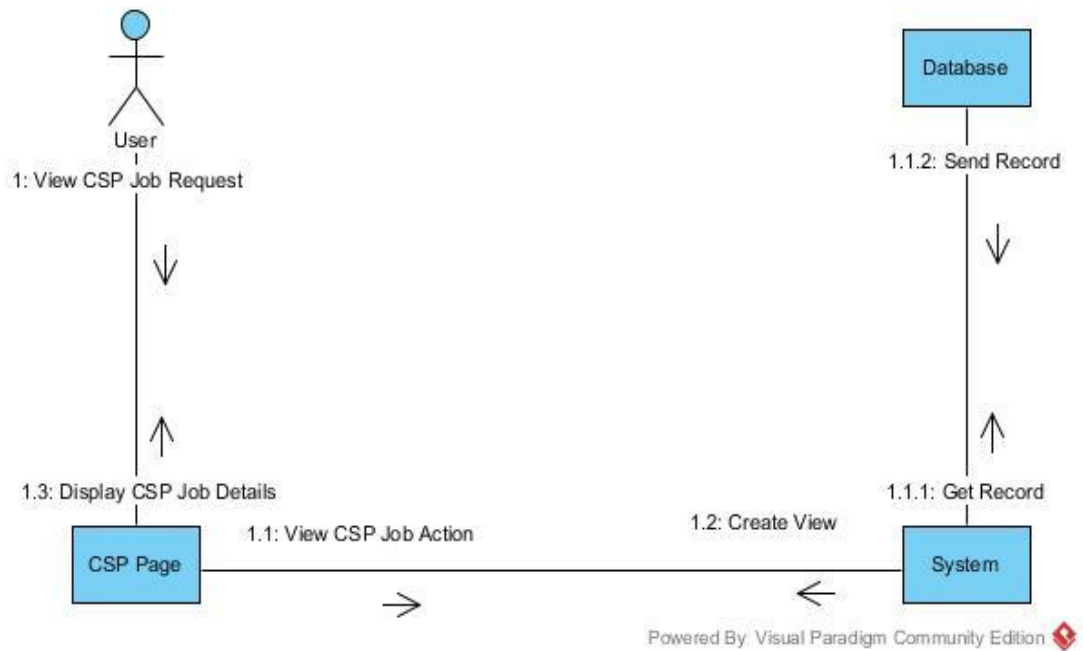


Figure 40 View CSP Job

- User instance was created
- CSP Job instance was created
- JOIN database instance was created
- User instance put CSP Job id to CSP Job instance
- CSP Job instance put CSP Job id to JOIN database instance
- JOIN database instance put CSP Job details to CSP Job instance

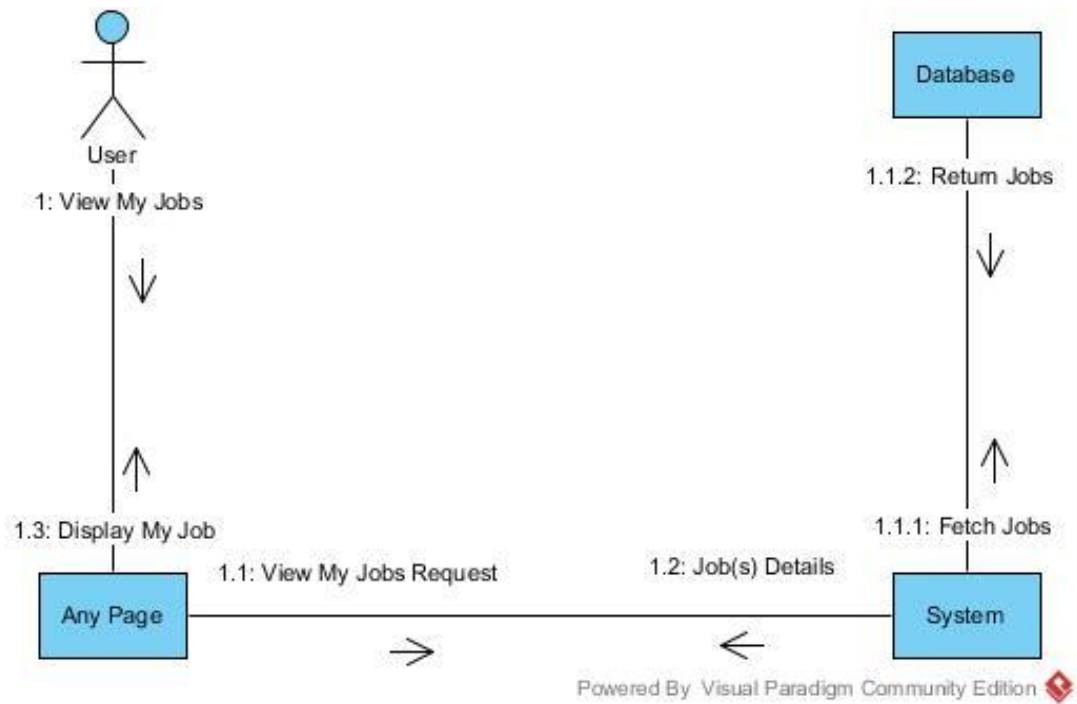


Figure 41 View My Jobs

- User instance was created
- Picked Jobs instance was created
- JOIN database instance was created
- User instance put user id to picked jobs instance
- Picked jobs instance put user id to JOIN database instance
- JOIN database instance returned Jobs against given user id to Picked jobs instance

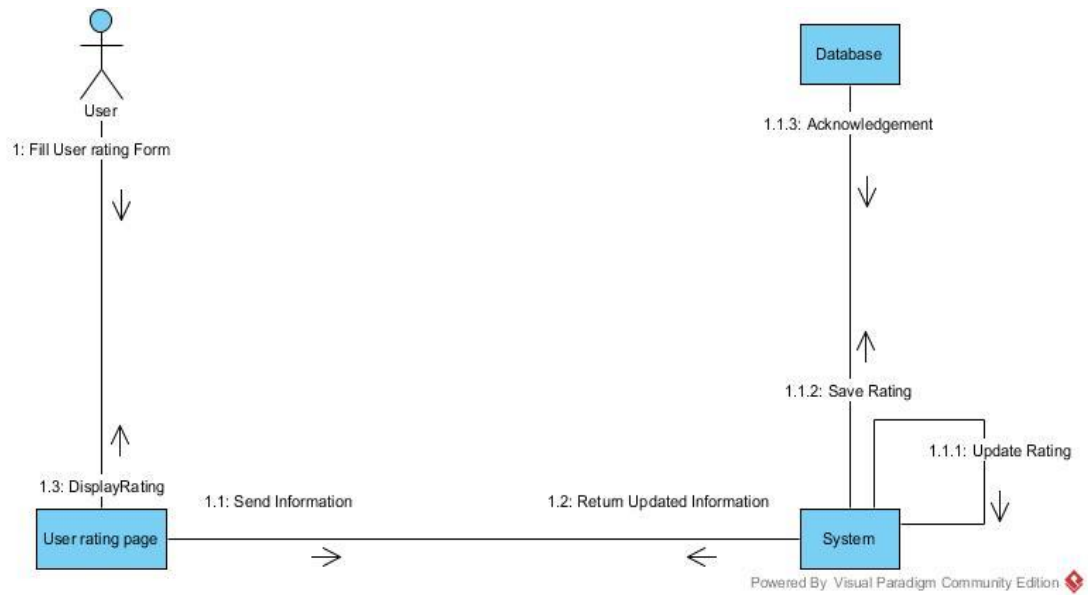


Figure 42 User Rating

- User instance was created
- UserRating instance was created
- JOIN database instance was created
- User instance put User rating details to UserRating instance
- UserRating instance put rating detail to JOIN database instance

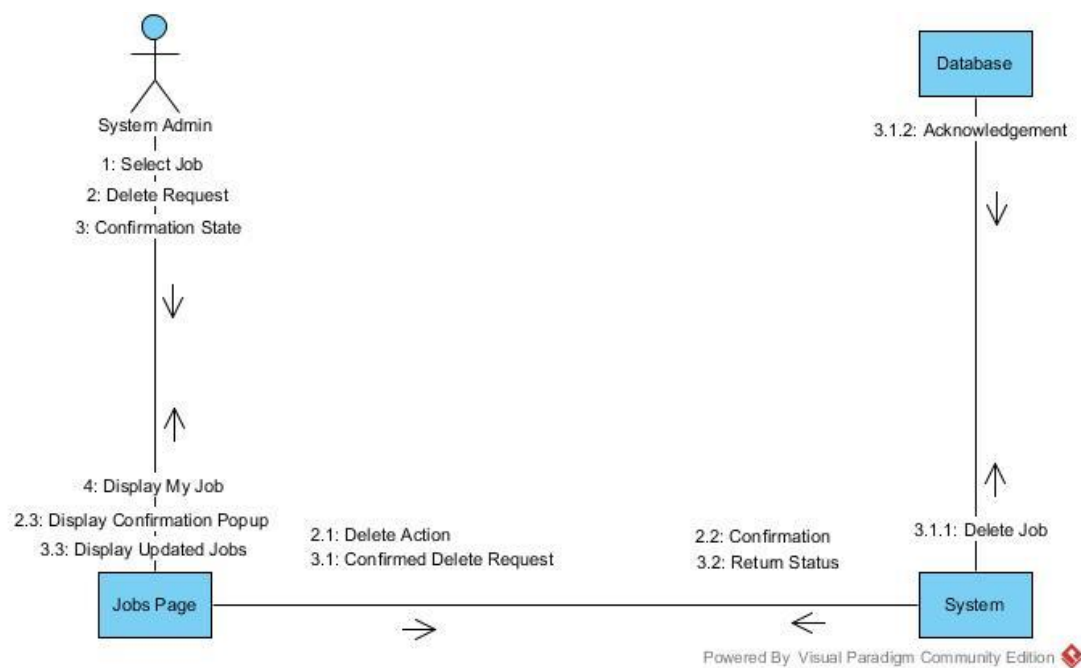


Figure 43 Delete Job

- User class instance was created
- JOIN database connection instance was created
- JOIN database instance matched each Job Id with user instance Job Id
- JOIN database returned user details to user instance
- Target user record was deleted by JOIN database instance

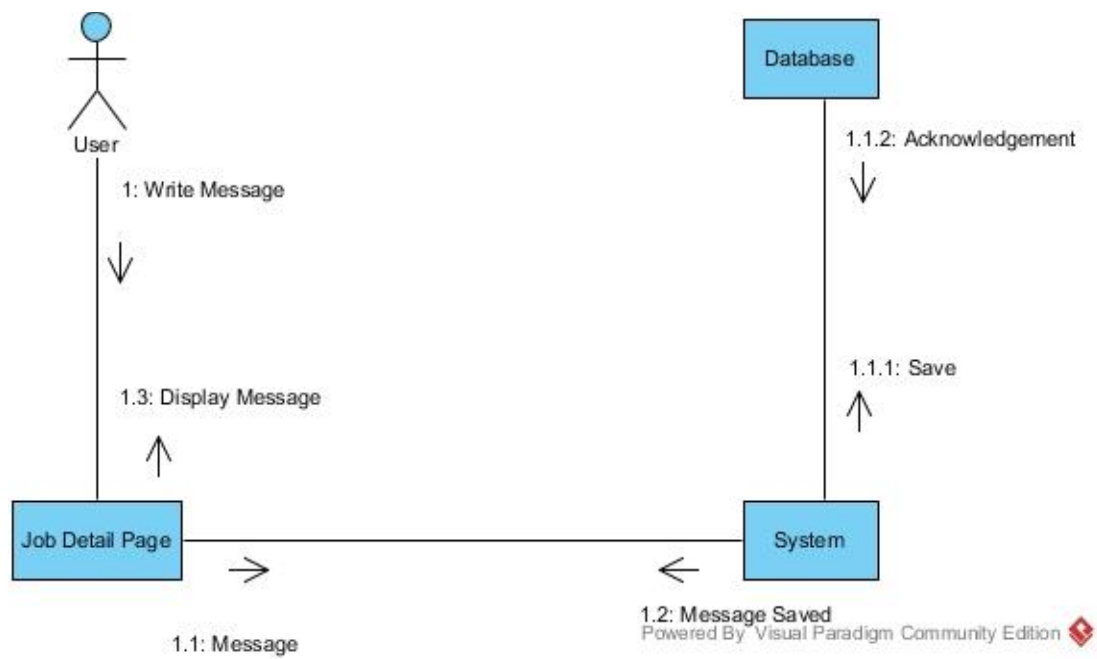


Figure 44 Discussion

- User instance was created
- Job instance was created
- discussion instance was created
- JOIN database instance was created
- User instance put chat details to discussion instance
- Discussion instance put chat detail to JOIN database instance

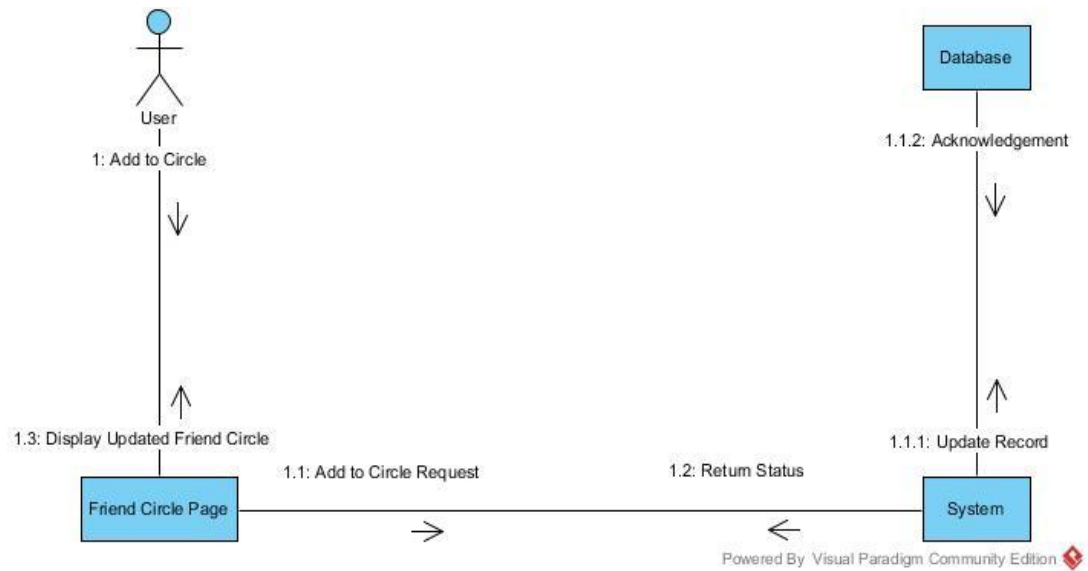


Figure 45 Add to Friend Circle

- User instance was created
- FriendCircle class instance was created
- JOIN database instance was created
- User instance put targeted user id to FriendCircle instance
- FriendCircle instance put resource id to JOIN database instance
- JOIN database instance returned targeted users detail to FriendCircle instance
- FriendCircle instance put current user id and targeted user id in JOIN database instance

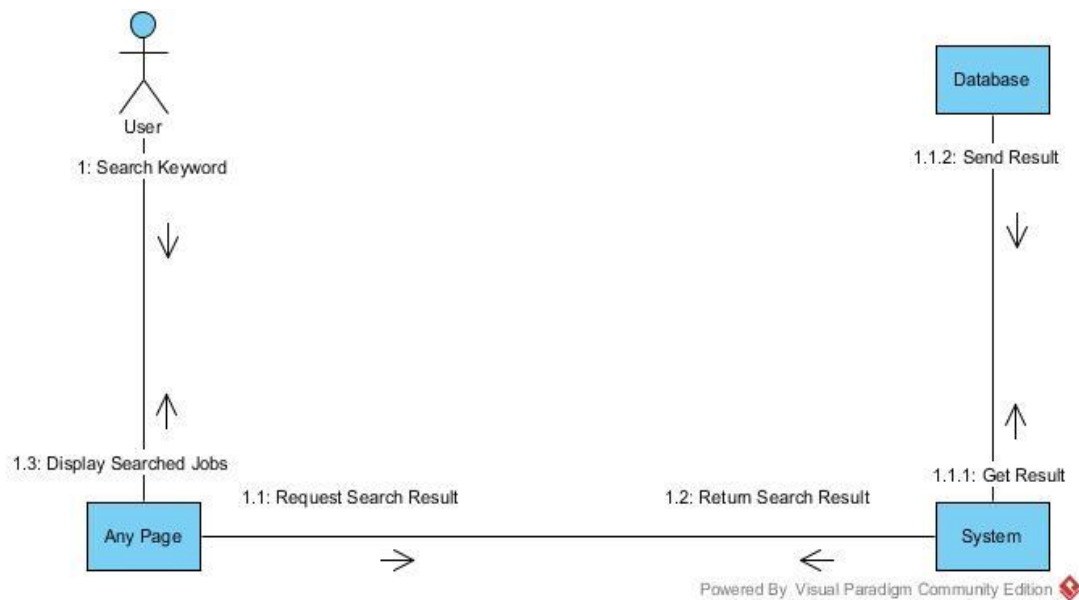


Figure 46 Search

- User instance was created
- jobSearch instance was created
- JOIN database instance was created
- User instance put keyword to jobSearch instance
- jobSearch instance put job ID to JOIN database instance
- JOIN database instance returned jobs detail to jobSearch instance

3.7 Data Model

The data model is a subset of the implementation model, which describes the logical and physical representation of persistent data in the system.

3.7.1 External Model

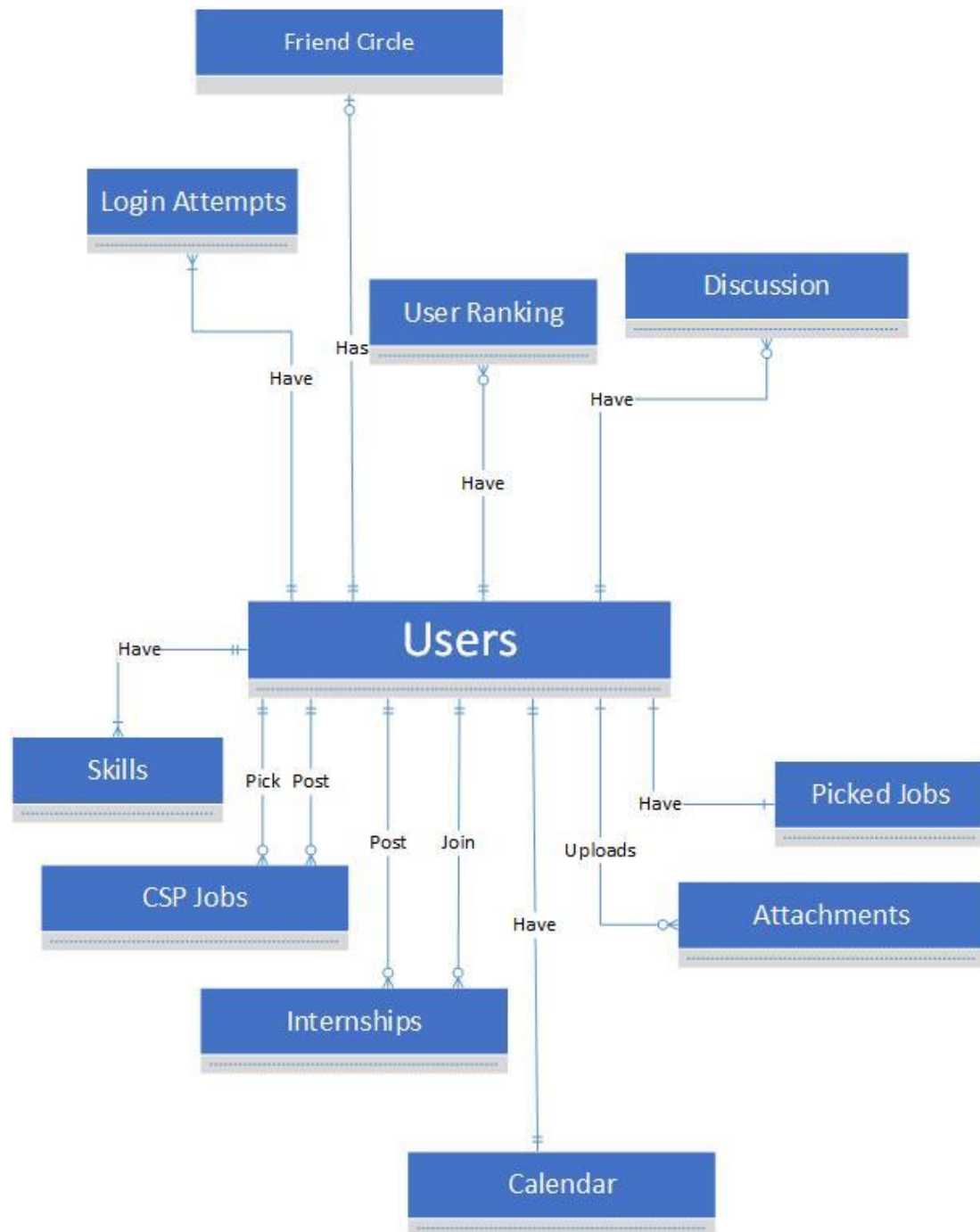


Figure 47 External Model User

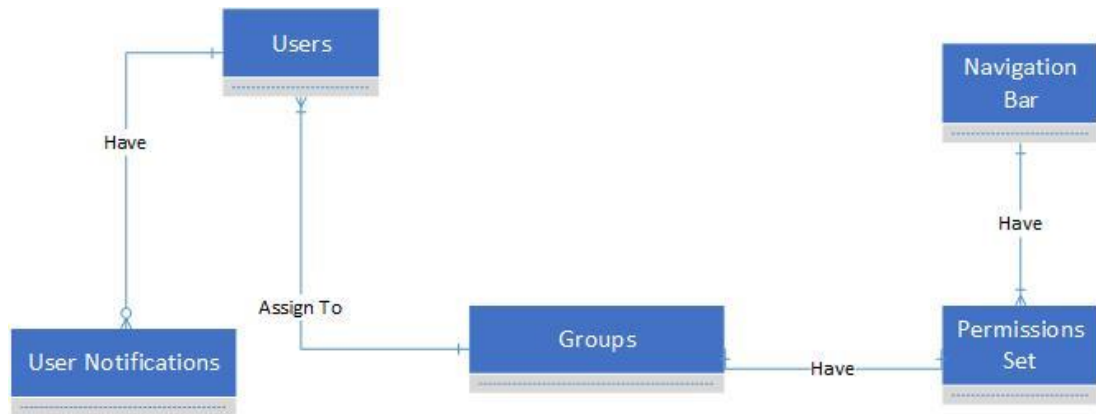


Figure 48 External Model UMS

3.7.2 Conceptual Model

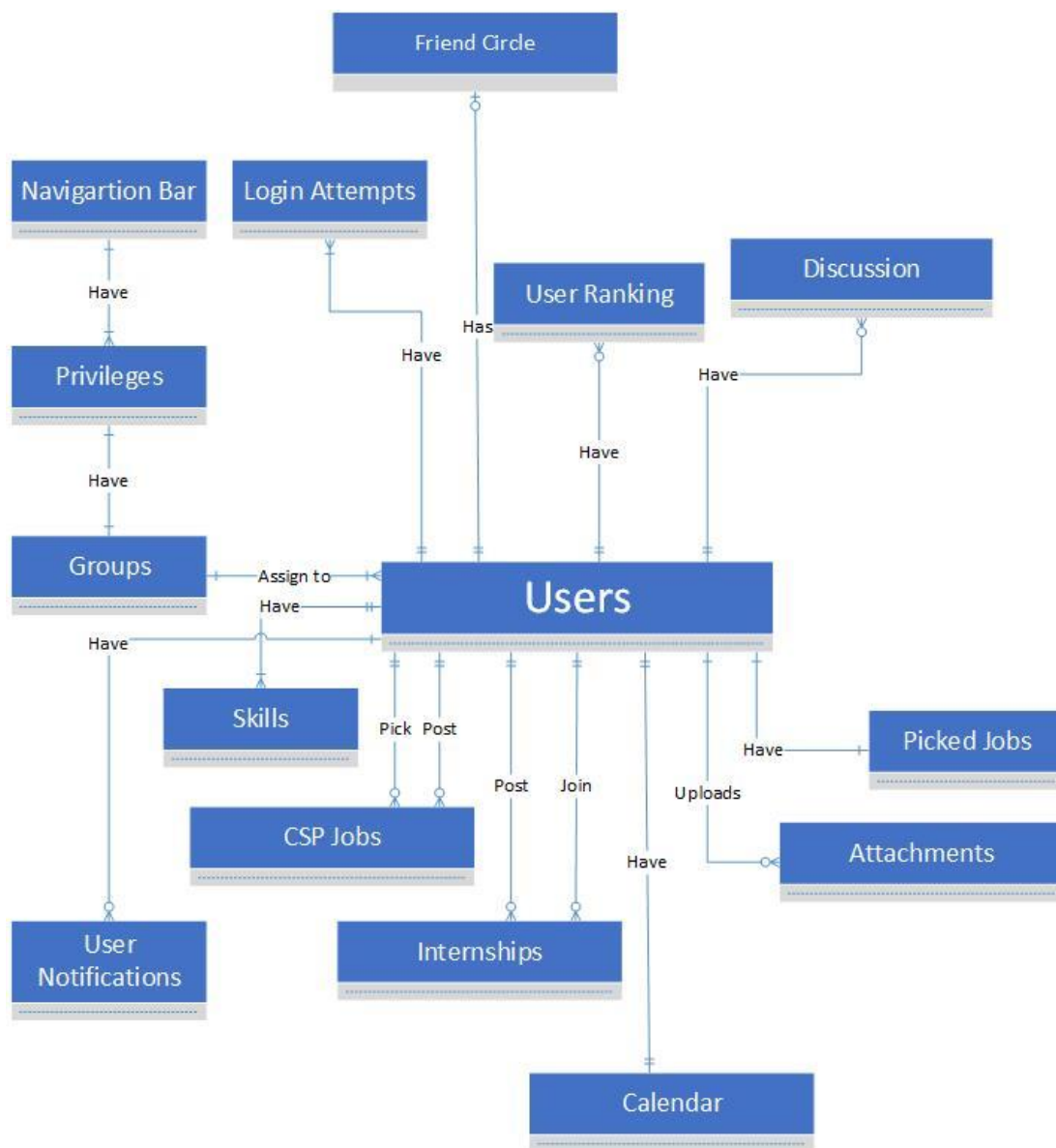


Figure 49 Conceptual Model

3.7.3 Internal Model

attachments

Attribute	Type
id	int(3)
user_id	int(3)
job_id	int(3)
type	varchar(20)
file_path	varchar(100)

categories

Attribute	Type
cat_id	int(3)
cat_title	varchar(30)
controller	varchar(30)

friend_circle

Attribute	Type
fc_id	int(3)
user_id	int(3)
friend_id	int(3)

groups

Attribute	Type
id	mediumint(8)
name	varchar(20)
description	varchar(100)

discussion

Attribute	Type
chat_id	int(3)
job_id	int(3)
user_id	int(3)
datetime	datetime
message	text

jobs

Attribute	Type
job_id	int(3)
user_id	int(3)
cat_id	int(3)
job_title	varchar(120)
description	varchar(1000)
required_no_of_resources	int(3)
location	varchar(100)
duration	int(3)
stipend	int(4)
type	varchar(50)
posted_on	datetime
status	varchar(20)

navbar

Attribute	Type
navbar_id	int(3)
navbar_name	varchar(20)
navbar_order	int(10)
parent_id	int(3)
navbar_icon	varchar(100)
navbar_view	varchar(50)
controller	varchar(50)

subcontroller	varchar(50)
category	varchar(10)

privileges

Attribute	Type
p_id	int(3)
navbar_id	int(3)
group_id	int(3)
permission	varchar(10)

users

Attribute	Type
id	int(11)
name	varchar(50)
password	varchar(255)
email	varchar(100)
enrollment	varchar(20)
registration_no	varchar(20)
activation_code	varchar(40)
forgotten_password_code	varchar(40)
forgotten_password_time	int(11)
remember_code	varchar(40)
created_on	int(11)
last_login	int(11)
active	tinyint(1)
address	varchar(100)
phone	varchar(20)
skills	varchar(100)
about	varchar(1200)
education	varchar(100)
certification	varchar(100)
salt	varchar(255)

user_groups

Attribute	Type
Id	int(11)
user_id	int(11)
group_id	mediumint(8)

user_notification

Attribute	Type
un_id	int(3)
user_id	int(3)
type	varchar(20)
date_time	datetime
url	varchar(50)
description	text
status	tinyint(1)

user_picked_jobs

Attribute	Type
id	int(3)
user_id	int(3)
job_id	int(3)
status	varchar(20)
assign_time	datetime

user_rating

Attribute	Type
ur_id	int(3)
job_id	int(3)
rated_by	int(3)
rated_user	int(3)
rating	varchar(50)

login_attempts

Attribute	Type
id	int(11)
ip_address	varchar(15)
login	varchar(100)
time	int(11)

skills

Attribute	Type
id	int(3)
cat_id	int(3)
skills	varchar(50)

CHAPTER 4

METHODOLOGY & IMPLEMENTATION

4.1 Methodology

Many methodologies now exist for developing applications on the web, so to choose an appropriate style for this project, research into the various options is necessary. A relevant summary of this research will be presented below, followed by a justified choice, and a description of how the methodology will be applied in this project.

4.1.1 Present Available methodologies

In the first methodologies created are described as being based on the waterfall model, which instructed developers to create systems in a linear step-by-step manner. The model states that before any work can begin on the actual software solution, a detailed feasibility requirements analysis must be conducted, and converted into a specification for the system. The software and all its features should then be thoroughly designed before implementation can occur. The finished product should then be tested, and any problems rectified, before being deployed in the appropriate environment.

Waterfall model was used successfully for many years, however eventually the limitations of the model become clear. The requirements and specifications for a project aren't always fully known initially, so it can become apparent later during the implementation phase that the system design is inconsistent, which can often lead to

project failure. These problems were addressed in later years, when agile methodologies were devised to develop systems faster, and with greater flexibility.

In agile methodologies are described as those where the system is developed using a prototype and refined through user feedback of the system in use and changes in the application itself. The main advantage to this is the constant improvement of the system to fulfil the requirements of the users. Source however describes how prototypes can often be inefficient and difficult to maintain, and may not scale well to large systems.

Incremental methodologies were first applied in iterative rapid application development (RAD) models, whereby the waterfall model was repeatedly applied. The primary objectives of RAD include fast development of high quality solutions at a low cost. This methodology is now in widespread use, and continues to be a highly regarded approach. In addition, Feature Driven Development (FDD) is a production process, which highly oriented on resulting out small blocks of client-valued functionality. This drives developers to come up with working features once every two weeks typically and it can track down the project progress with precision. FDD, which is one of a number of agile development processes, is an iterative and incremental software development process having the main purpose of delivering tangible working software repeatedly in a timely manner.

4.1.2 Choosing a Suitable Methodology

After conducting the research described above into methodologies, and understanding the requirements from the university, it was decided to follow an agile methodology called feature driven development (FDD), to complete the project. Class diagram and sequence diagram will be used in the system analysis, and feedback on the initial design will be acquired from real users of the system. This methodology was chosen since the web development frameworks which will be used are relatively unfamiliar to the developer, and so an iterative approach will allow for comprehension of the framework features between iterations, and project risk will be controlled by implementing features on a priority basis. By creating screen prototypes at each stage,

user feedback can be incorporated, and human-computer interaction issues can be addressed. [2]

4.1.3 Start working with FDD

As FDD is a feature driven methodology and for building the list of features we should have all the requirements in a manner that they can understand accurately. Therefore, we started with the first phase FDD develop overall model.

4.1.3.1 Develop Overall Model

In first phase, we collect all the requirements from the university and identify all the stakeholders of the system. After that, we analyse the requirements, identify user classes and characteristics and make a requirement chart with difference of functional and non-functional requirements. After making the requirement chart, we made a meeting with PO SRC and Internship coordinator for the conformance of the requirements. This phase ends by making domain model and class diagram. [3]

4.1.3.2 Build the feature list

In this phase, we categorise the features with respect to each of our system module and divide the features that can be divided into sub-features and also we break the feature into a sub-feature that take time more than two weeks. Moreover, priorities assigned to the features and after doing all these things we have a list of features as a result.

4.1.3.3 Plan by feature

In this phase, we plan every feature by identifying their backlogs and according to their priorities. Features are also planned according to their dependencies between features in terms of classes involved, balancing load across class owners, and complexity of the features to be implemented. [4]

4.1.3.4 Design by Feature

Now from this phase the actual working starts that includes the thoroughly design of every feature and then build it. This is an iterative phase that iterates along with the 'Build by feature' phase that is next to it. In this phase, we made use-case, sequence diagram and collaboration diagram of each feature and then review it with the class diagram if any change require in class diagram then do otherwise continue to build the feature.

4.1.3.5 Build by Feature

In this phase, the design made in previous feature implements here and the coding begins for each feature. This is also an iterative phase and execute after every feature design. We strictly follow the feature list and planning we made in the 2nd and 3rd phase in these iterative phases. Moreover, unit testing also done in this phase by each developer. The outcome of this phase is a working feature of system. [5]

4.2 Implementation

The implementation phase of the project is the development of the designs produced during the design phase.

4.2.1 Algorithm Design

As this is a web-based project and by following a framework, the best practices are available so there is no really need to design such an algorithm but sometimes to full-fill the business need it may require.

4.2.2 Technical Aspect of the JOIN

Since the JOIN is a web application, both web server and database management system are necessary. A high-level system architecture diagram is illustrated below. The system consists of front-end and back-end and is categorised into three layers.

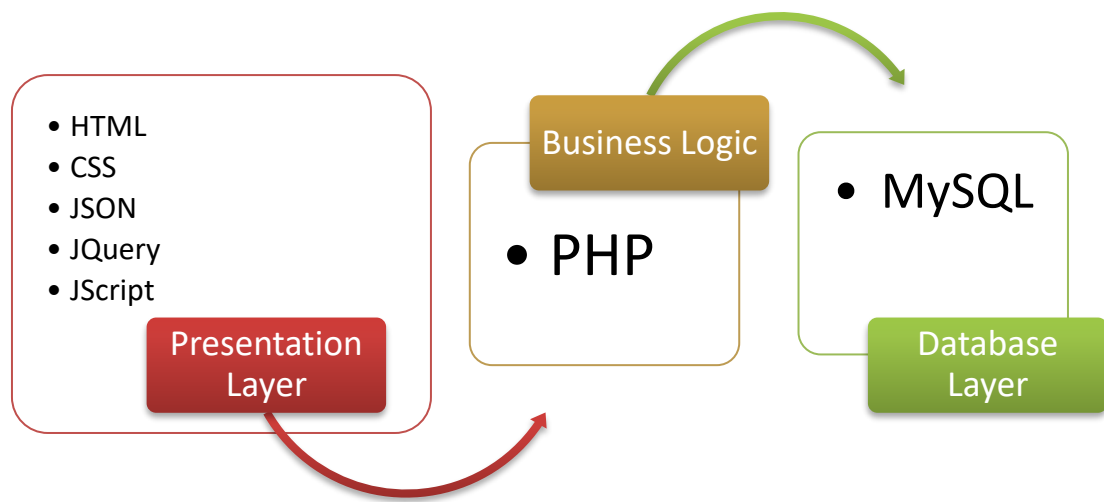


Figure 50 Architecture Diagram

CHAPTER 5

USER MANUAL

5.1 Getting started

5.1.1 User Access, Roles and Privileges

The users of the application are the students and administration of Bahria University and Organizations, each user assigned to a specific group, and each group has defined privileges according to their role. [6]

The table below explains the different levels of the access rights and corresponding explanation for each role.

User Group	Authorization
Student	Pick CSP job Add other students in friend circle Apply for Internship Discussion with Organization personal Remove a student from friend circle
Organization	Post, Edit, Delete and Close Jobs Discussion with students Rate student

System Admin	<ul style="list-style-type: none"> View all users Change access Manage privileges View and Edit Groups View posted internships and CSP jobs Block and Unblock users Delete Jobs Generate reports
--------------	--

5.1.2 How to access JOIN

Open internet browser (i.e. Google Chrome, Mozilla Firefox and Opera) and type in the following URL <http://joinforbahria.tk> on the browser's address bar and press 'Enter key' (As shown in below figure)

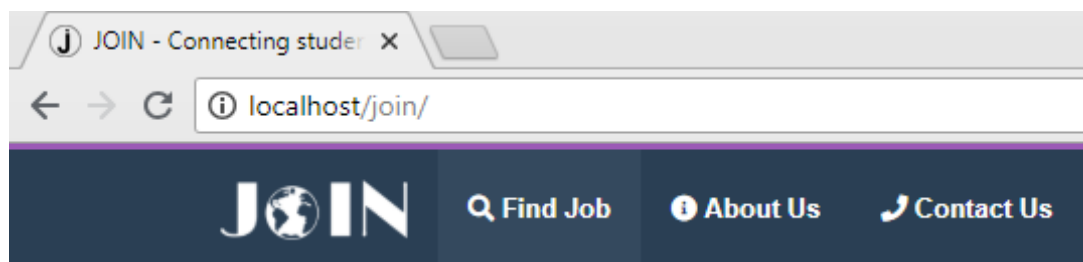


Figure 51 How to access JOIN

5.1.3 Home Screen for non-register members

After accessing JOIN, you will see the following home screen

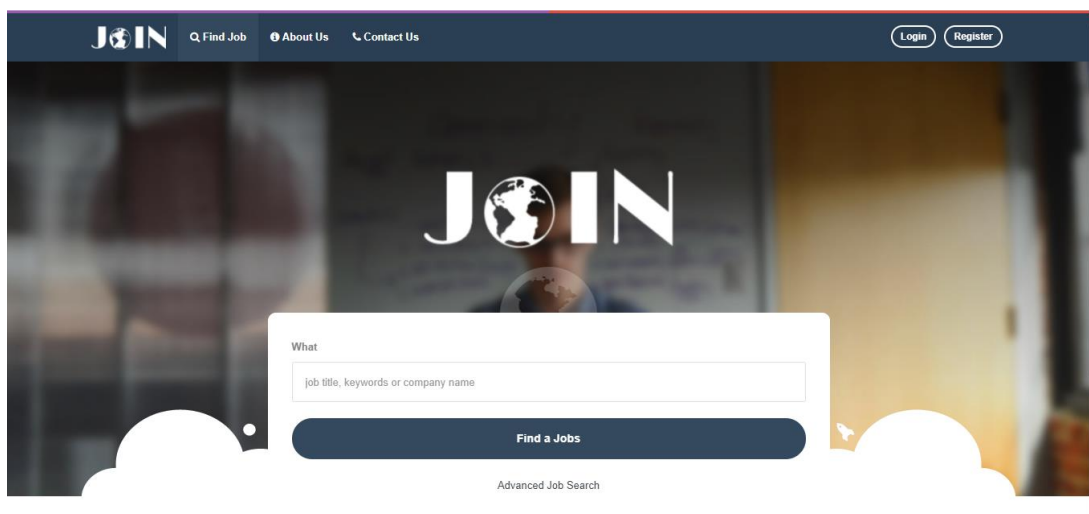


Figure 52 Home Screen (Without Login)

- 1) Here you can find the posted jobs on JOIN
- 2) By clicking on About us you can know more about JOIN
- 3) If you want to contact the Admin of JOIN, click on Contact us
- 4) If you are already registered on JOIN you can login by clicking on login button
- 5) Non-registered visitor can register on JOIN by providing required details

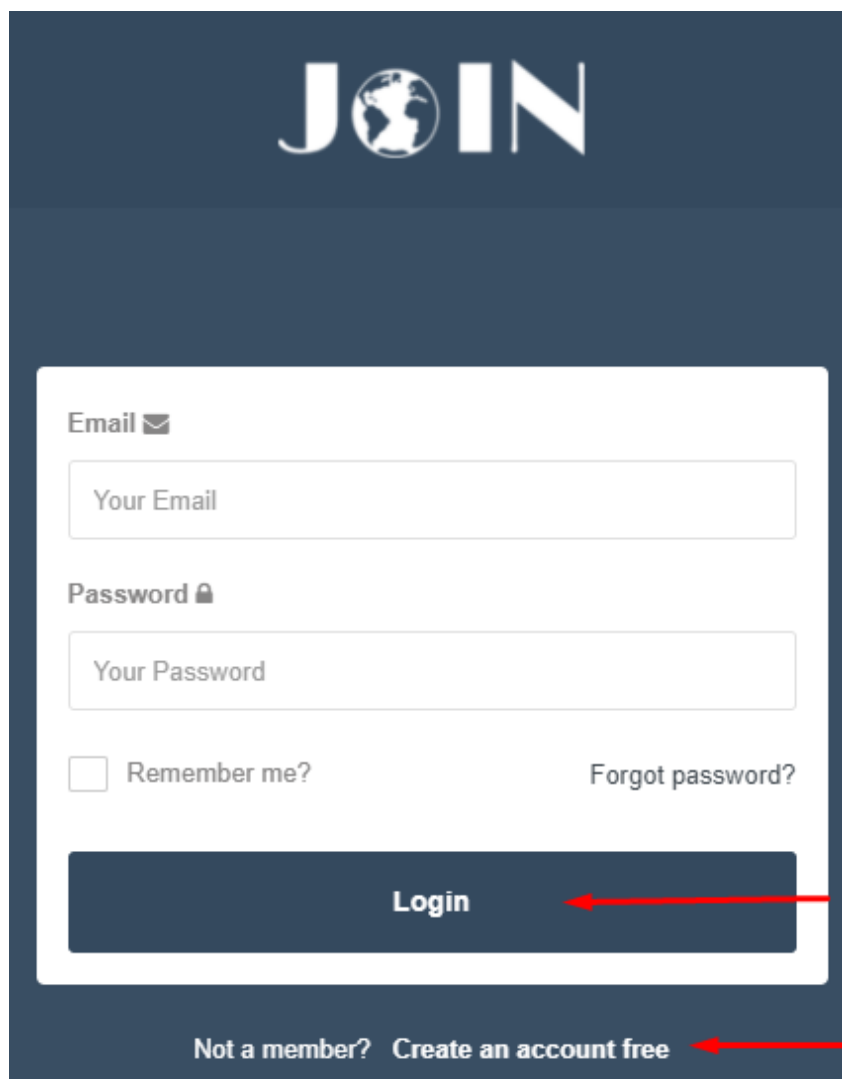
5.1.4 Login & Registration at JOIN

- 1) Click on login button to logged in your account as shown below (if you are already registered at join)



Figure 53 Login Button

2) After click on login button flowing screen will show up



The image shows a login form on a dark blue background. At the top, the word "JOIN" is displayed in white, with a globe icon integrated into the letter "O". Below this, there is a white rectangular form containing the following elements:

- Email** with an envelope icon, followed by a text input field containing the placeholder text "Your Email".
- Password** with a lock icon, followed by a text input field containing the placeholder text "Your Password".
- A checkbox labeled "Remember me?".
- A link labeled "Forgot password?".
- A dark blue button with the text "Login" in white.

Below the form, there is a link that says "Not a member? Create an account free". Two red arrows point to the "Login" button and the "Create an account free" link.

Figure 54 Login form

1) Enter login credentials and click on Login button and check Remember me check box if you want browser remember your email and password.

- 2) Click on 'Create an account' to registered at JOIN. After click on 'Create an account' following screen will show

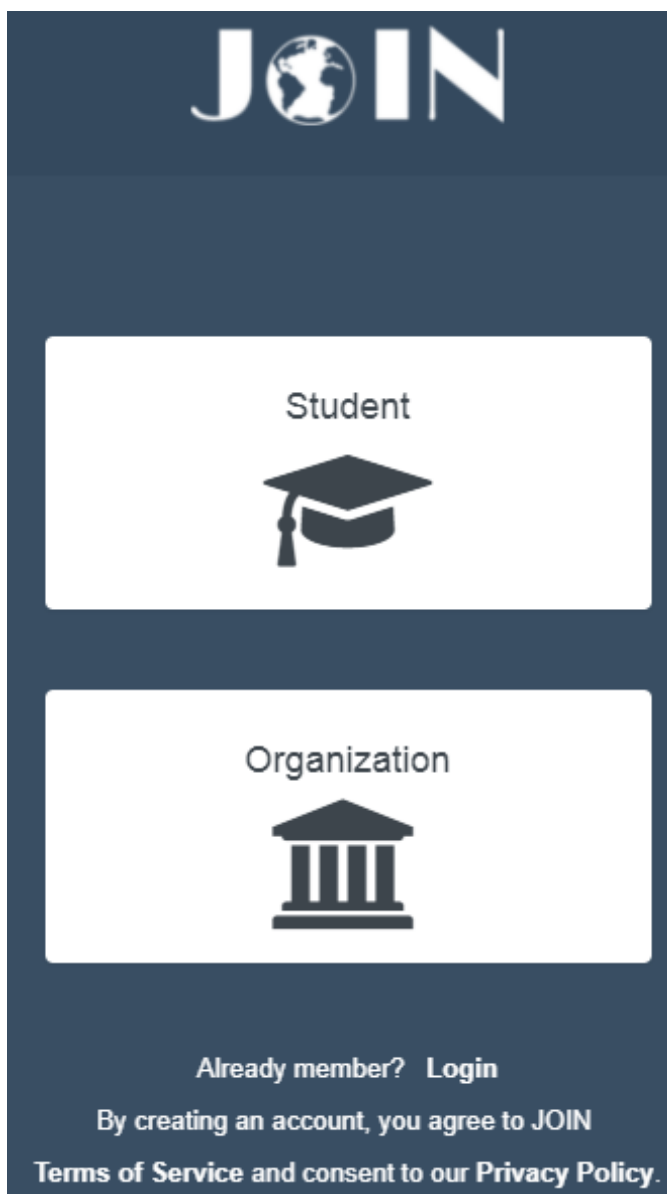


Figure 55 Signup options

Choose your registration type

- a) Click on student icon to register as student and after click on student icon following form will be show up

Name

Enrollment No.

Email

Contact No.

Password

Re-type Password

Register

Figure 56 Signup form (Student)

- b) Click on Organization icon to register as student and after click on student icon following form will be show up

Name

Registratio No.

Email

Contact No.

Address

Password

Re-type Password

Register

Figure 57 Signup form (Organization)

- 1) Fill the registration form by providing required information and click on 'Create User' button it will redirect you to Login and send an account activation link to your email. By clicking on activation link, your account will activate and you can now login from login screen.

5.1.5 Role based authorized Navigation bars

As discussed in section 5.1.1 there are roles of every user. Following are navigation bars of each user type

- 1) Student will access only their authorized navigation bar after successful login that will be discussed thoroughly in section 5.2.1

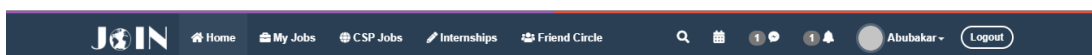


Figure 58 Navigation Bar (Student)

- 2) Organization will access only their authorized navigation bar after successful login that will be discussed thoroughly in section 5.2.2

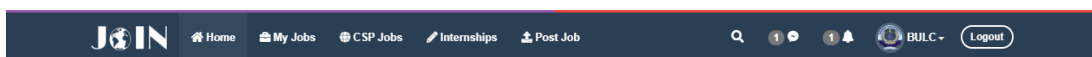


Figure 59 Navigation Bar (Organization)

- 3) System Admin view will be discussed later thoroughly in section 5.2.3

5.2 JOIN for Student

5.2.1 Navigation bar for students

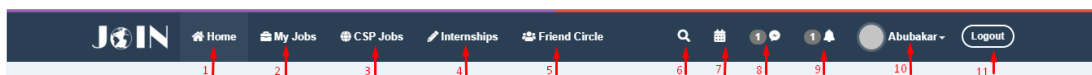


Figure 60 Navigation Bar (Student)

- 1) Home option will be on left side of navigation bar after JOIN logo you can go on home screen from any page by clicking on Home
- 2) By clicking on My Jobs option you can see all jobs and internship that you are applied for
- 3) By clicking on CSP jobs option you can see all CSP jobs posted by organizations
- 4) By clicking on Internships option you can see all Internships posted by organizations

- 5) By clicking on Friend Circle you can see list of all friends you have added in your friend circle
- 6) By clicking on Search icon you can search different job/user
- 7) By clicking on Calendar icon you can see your jobs on calendar
- 8) Message icon will show messages you received from organizations
- 9) Bell icon will display notifications
- 10) By clicking on user profile picture following dropdown will appear

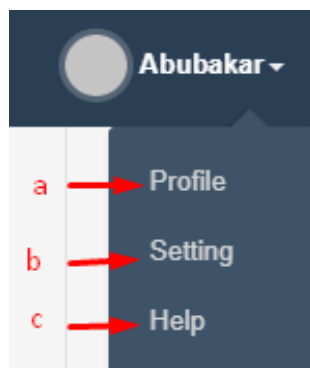


Figure 61 User Profile

- a) Profile will redirect to your profile where you can see your information
 - b) Setting option will redirect you to profile setting where you can update your information
 - c) Help option will redirect you to help page where you can read FAQs
-
- 11) By click on 'Logout' user will redirect to 'Login' page and required to re-logged in to access the JOIN

5.2.2 Home Page for students

By clicking on Home following page will be show

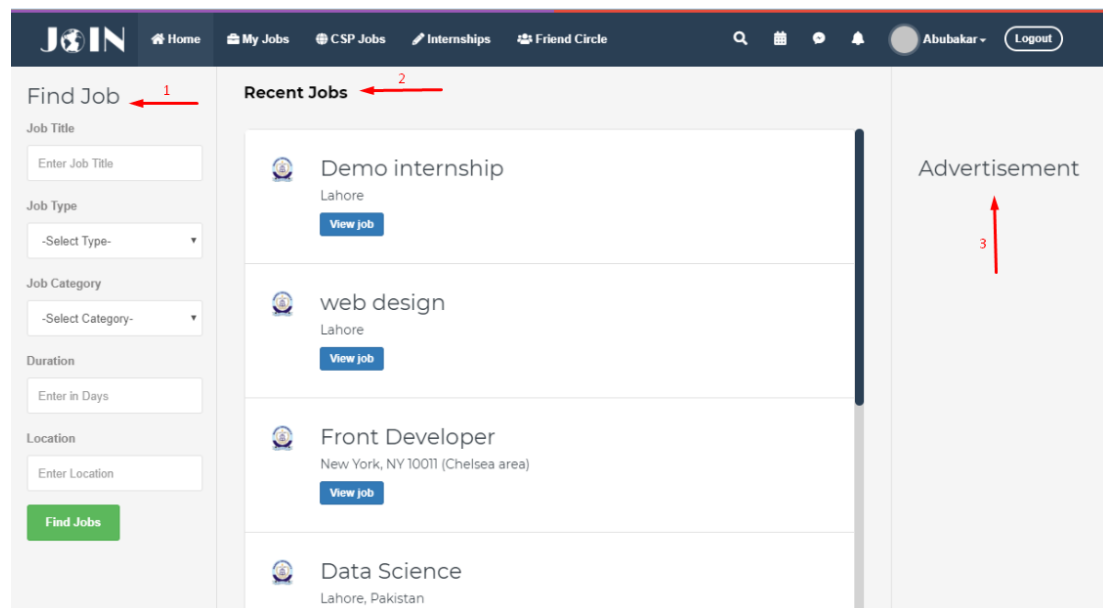
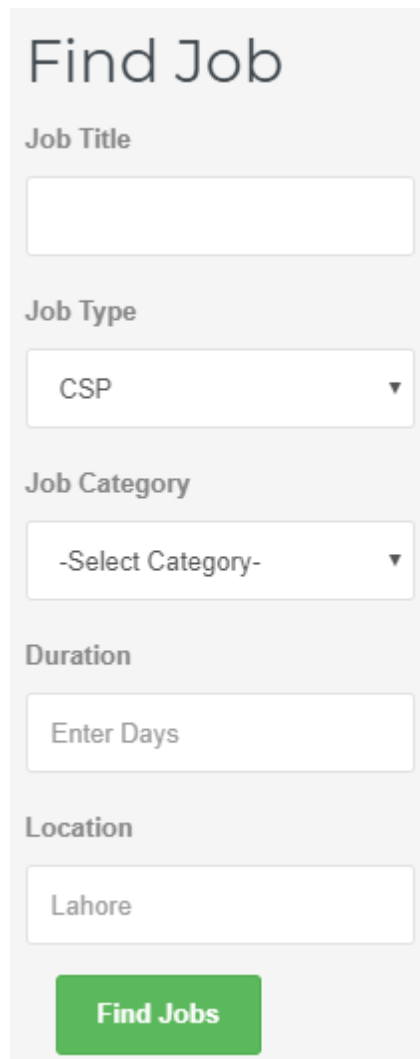


Figure 62 Home Screen (After Login)

- 1) You can find relevant jobs by setting appropriate filter
- 2) Recent jobs can be seen on home page
- 3) Advertise from Bahria University can be seen in advertisement section

5.2.2.1 Find Job

You can find relevant jobs by searching through different criteria (as shown in figure)



The image shows a search form titled "Find Job". It contains five input fields and a button:

- Job Title:** An empty text input field.
- Job Type:** A dropdown menu with "CSP" selected.
- Job Category:** A dropdown menu with "-Select Category-" selected.
- Duration:** A text input field with the placeholder text "Enter Days".
- Location:** A text input field with "Lahore" entered.
- Find Jobs:** A green button with white text.

Figure 63 Find Job

- 1) You can search by job title, type, category, duration or location
- 2) Set the search filter and hit Find Jobs button

5.2.2.2 Recent Jobs

In this section you can find recent posted jobs and apply for relevant jobs (as shown in figure).

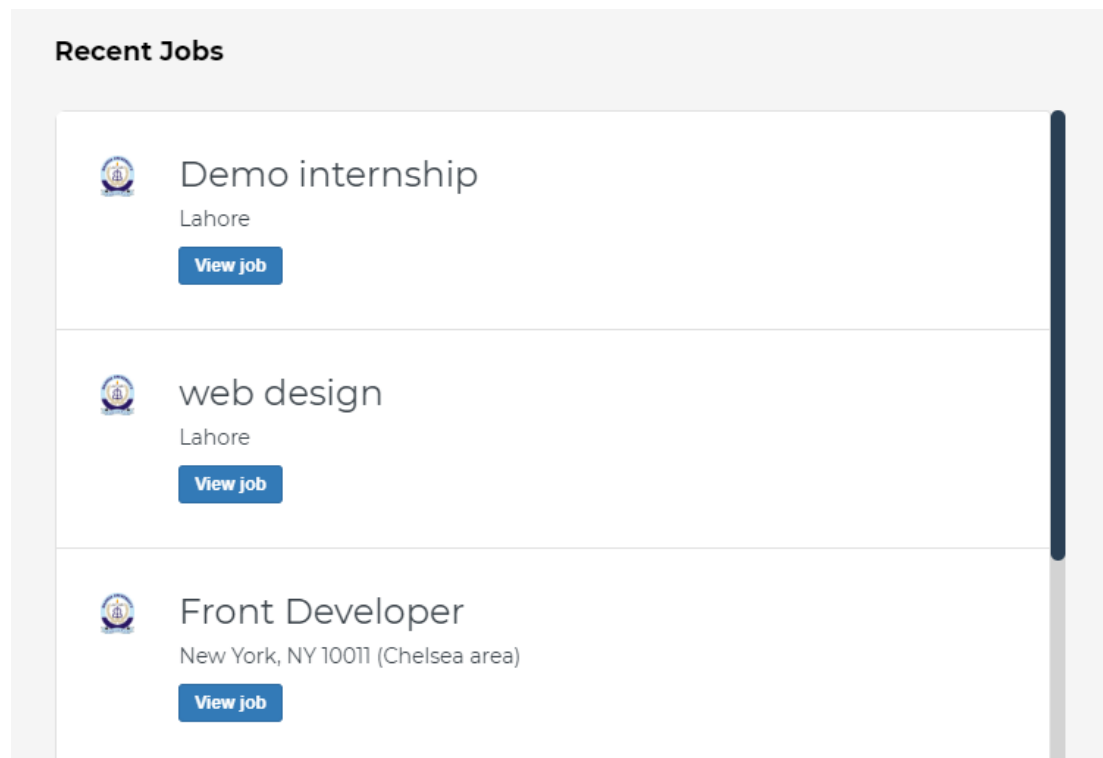


Figure 64 Featured Jobs

5.2.2.3 Advertisement

In this section advertisement of Bahria University will be displayed (as shown in figure).

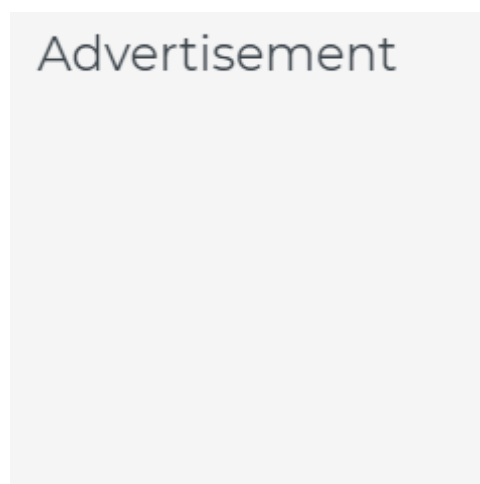


Figure 65 Advertisements

5.2.3 My Jobs for Student

You can find your “In progress”, “Applied” and “Completed” jobs under my jobs (as shown in figure).

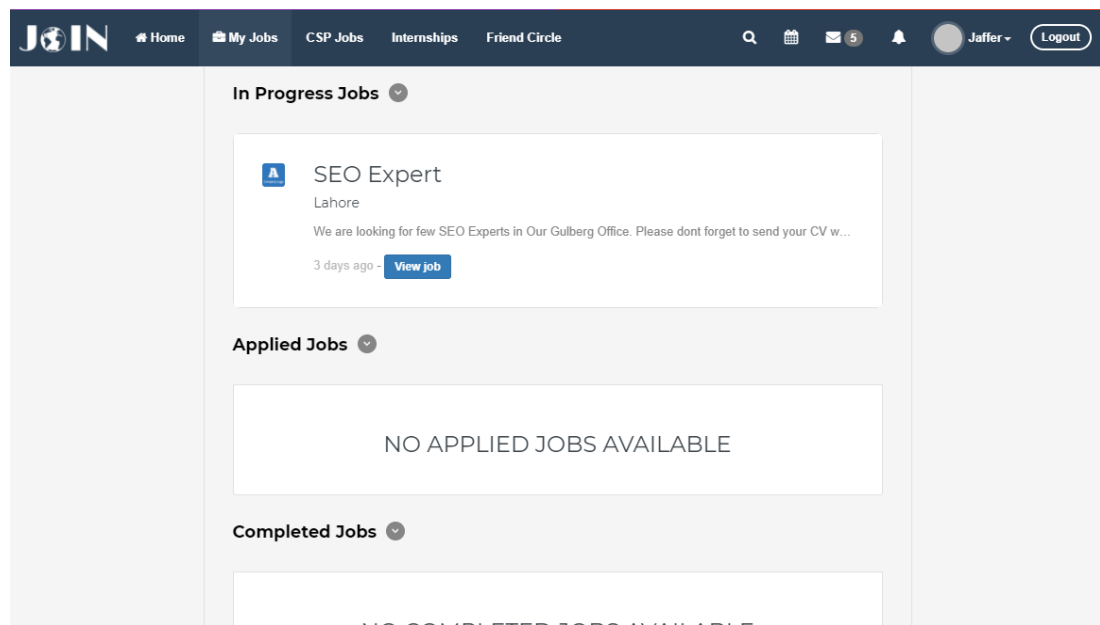


Figure 66 My Jobs (Student)

5.2.3.1 In progress Jobs

You can find your “In progress jobs” in this section.

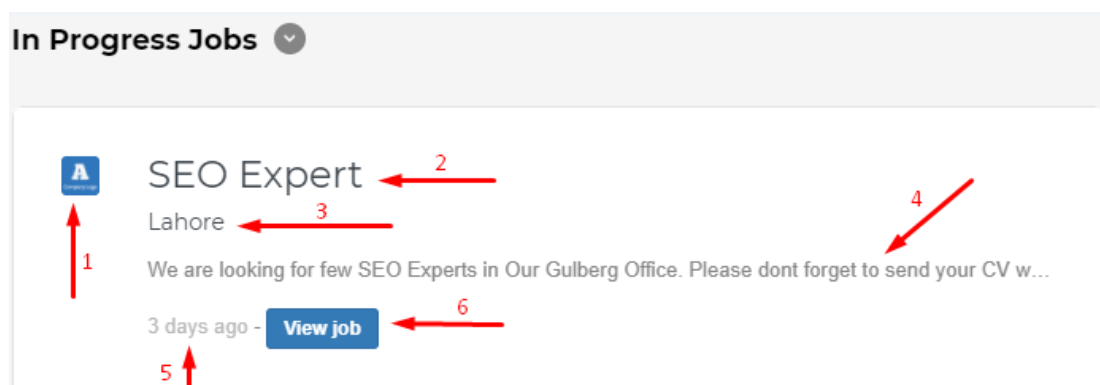


Figure 67 In Progress Jobs (Student)

- 1) Logo of organization that posted job will be seen before title
- 2) You can view job detail by clicking on job title
- 3) Job location will be shown under job title
- 4) Short description can be seen here
- 5) Job age can be seen
- 6) Job details can also be seen by clicking on View job button

5.2.3.1.1 Job details

You can find complete job details by clicking on view job



BAHRIA UNIVERSITY
PAKISTAN

Unity 3D Developer

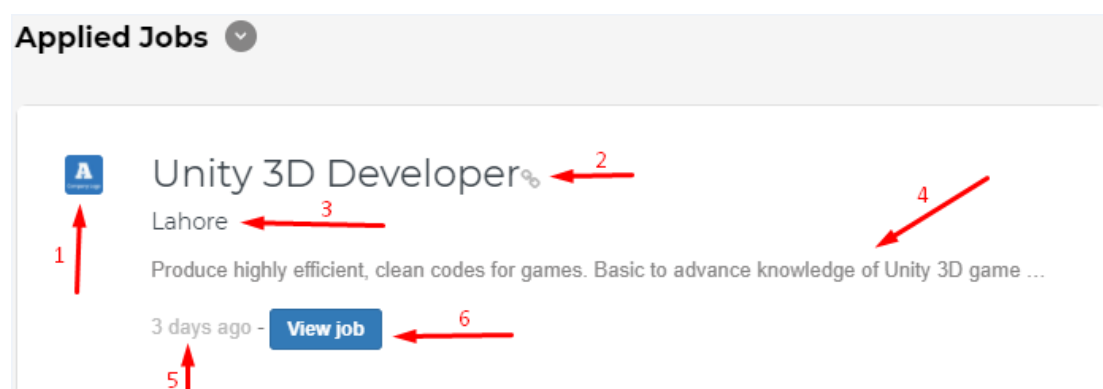
📅 10 Days 📍 Lahore 💰 0 Rs.

Produce highly efficient, clean codes for games. Basic to advance knowledge of Unity 3D game development engine. knowledge of gameplay programming. Experience with Physics programming. Experience with AI programming.

[Apply](#)

5.2.3.2 Applied Jobs

You can find your applied jobs in this section.



Applied Jobs ▼

A Unity 3D Developer  ← 2

Lahore ← 3

Produce highly efficient, clean codes for games. Basic to advance knowledge of Unity 3D game ... ← 4

3 days ago - [View job](#) ← 6

↑ 1 ↑ 5

Figure 68 Applied Jobs

- 1) Logo of organization that posted job will be seen before title
- 2) You can view job detail by clicking on job title
- 3) Job location will be shown under job title
- 4) Short description can be seen here
- 5) Job age can be seen
- 6) Job details can also be seen by clicking on View job button

5.2.3.3 Completed Jobs

You can find your completed jobs in this section.



Figure 69 Completed Jobs

- 1) Logo of organization that posted job will be seen before title
- 2) You can view job detail by clicking on job title
- 3) Job location will be shown under job title
- 4) Short description can be seen here
- 5) Job age can be seen
- 6) Job details can also be seen by clicking on View job button

5.2.4 CSP Jobs

All CSP jobs can be find in CSP Jobs section

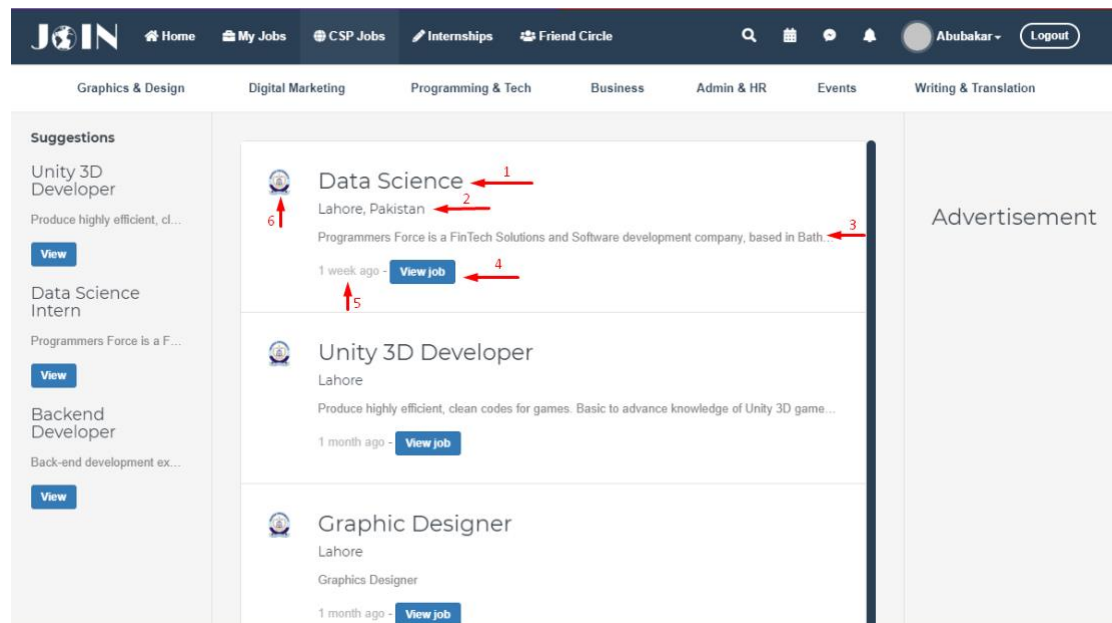


Figure 70 CSP Jobs

- 1) There is navigation bar under main navigation bar from where user can select Job categories
- 2) CSP jobs listed here
 - 1) You can view job detail by clicking on job title
 - 2) Job location will be shown under job title
 - 3) Short description can be seen here
 - 4) Job details can also be seen by clicking on View job button
 - 5) Job age can be seen
 - 6) Logo of organization that posted job will be seen before title

5.2.5 Internships

All internships can be find in CSP Jobs section

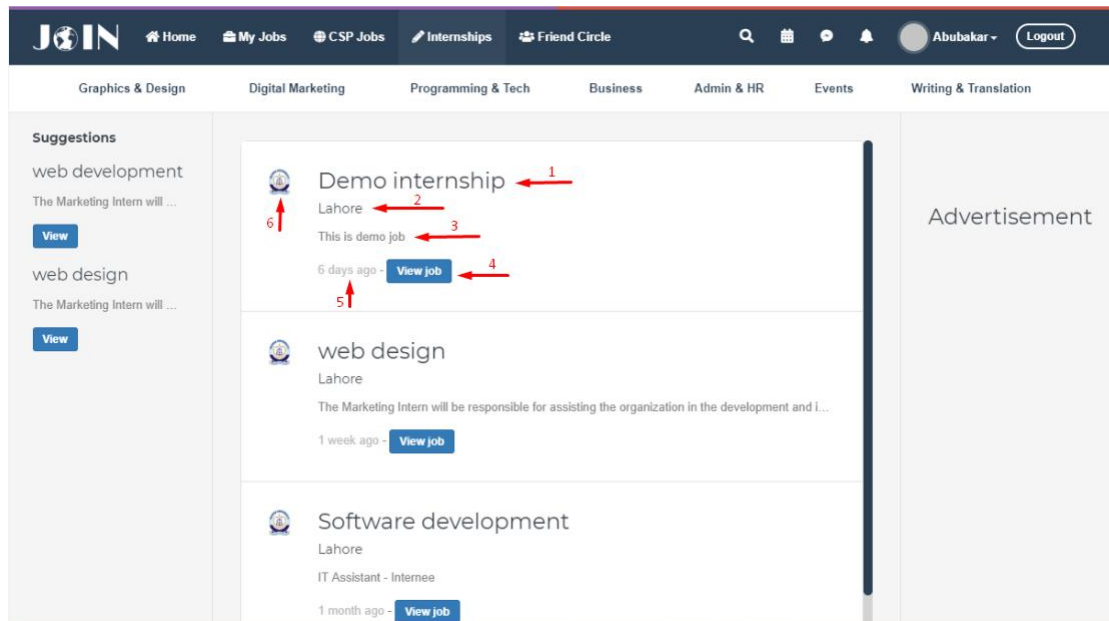


Figure 71 Internships

- 1) There is navigation bar under main navigation bar from where user can select Job categories
- 2) This section contains all the internship listing for the category selected by student
 - 1) You can view job detail by clicking on job title
 - 2) Job location will be shown under job title
 - 3) Short description can be seen here
 - 4) Job details can also be seen by clicking on View job button
 - 5) Job age can be seen
 - 6) Logo of organization that posted job will be seen before title

5.2.6 Friend Circle

Users with mutual interests can add each other on friend circle (as shown in figure below)

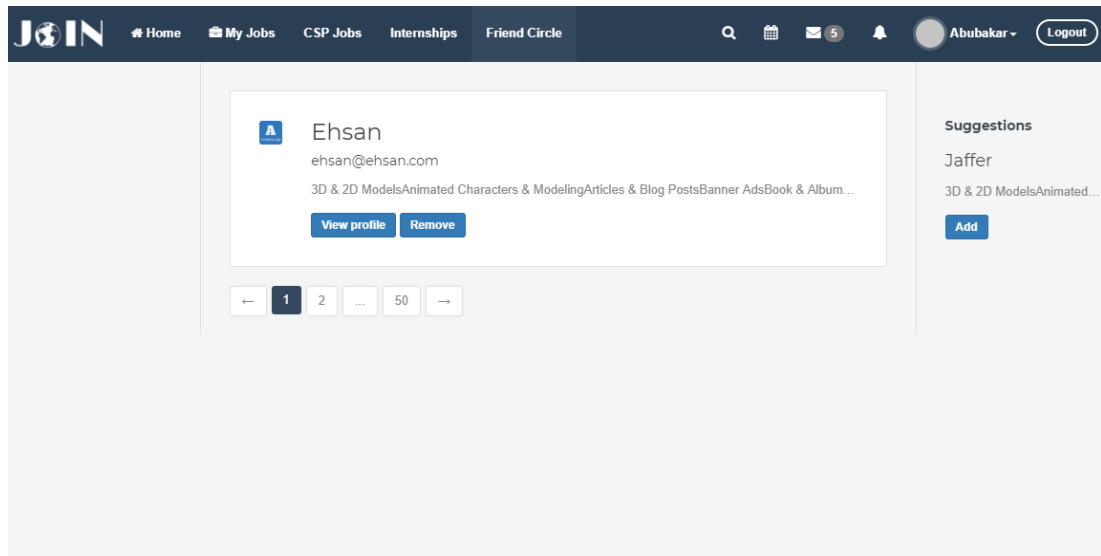


Figure 72 Friend Circle

- 1) Details (i.e. Profile picture, Name, email and skills) are displayed in this section for each user in friend circle
- 2) You can view student profile
- 3) You can remove a student from your friend circle
- 4) List of students with mutual interest are shown in suggestion section

5.2.6.1 View profile

View profile

Figure 73 View Profile Button

By clicking on View profile you can see that user profile

5.2.6.2 Remove

Remove

Figure 74 Remove Button

Remove button is used for removing a student from your Friend Circle

When you click on remove button a pop-up appear to verify that you want to remove this user from your friend circle

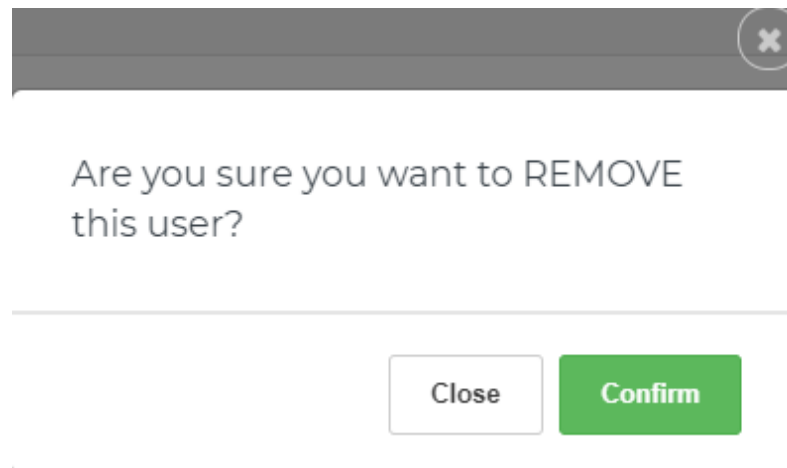


Figure 75 Remove Verification Pop-up

Which contains to buttons

- 1) Confirm to remove that user from your friend circle
- 2) Close for closing pop-up and user will remain in your friend circle

5.2.6.3 Suggestions

List of students with mutual interest are shown in suggestion section (as shown in figure below)

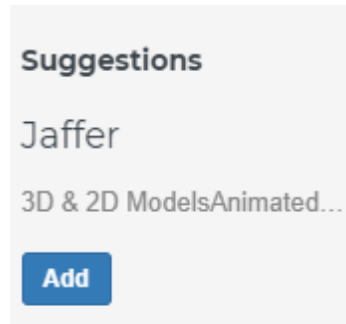


Figure 76 Friend Suggestions

5.2.6.4 Add

You can add only student with mutual interests/skills



Figure 77 Add to friend circle button

By clicking on add button targeted student will be added in your friend circle

5.2.7 Search

Search option is used for search jobs and users on JOIN



Figure 78 Search form

- 1) This field is used to input search keyword
- 2) By pressing the Search button, you will get search result
- 3) Close button is used for closing search modal
- 4) Advanced Search option is used for searching jobs with specific criteria

5.2.7.1 Advanced Search

Advanced Search option is used for searching jobs with specific criteria



Figure 79 Advance search button

Advanced search can be access by clicking on Advanced Search option at search screen (following page will appear see next page).

Advanced Job Search

Find Jobs

Job Title

Job Type

Job Category

Duration

Location

Close

Find Jobs

Figure 80 Advance search form

Job Category

- Select Category-
- Graphics & Design
- Digital Marketing
- Programming & Tech
- Business
- Admin & HR
- Events
- Writing & Translation

Figure 81 Job categories

Job Type

Figure 82 Job Types

- 1) Set search criteria by input or choosing options from drop-down
- 2) Click on Find Jobs button to search
- 3) You can close advanced search modal by clicking on Close button

5.2.7.2 Find Jobs

By clicking on Find Jobs result of your search will be displayed

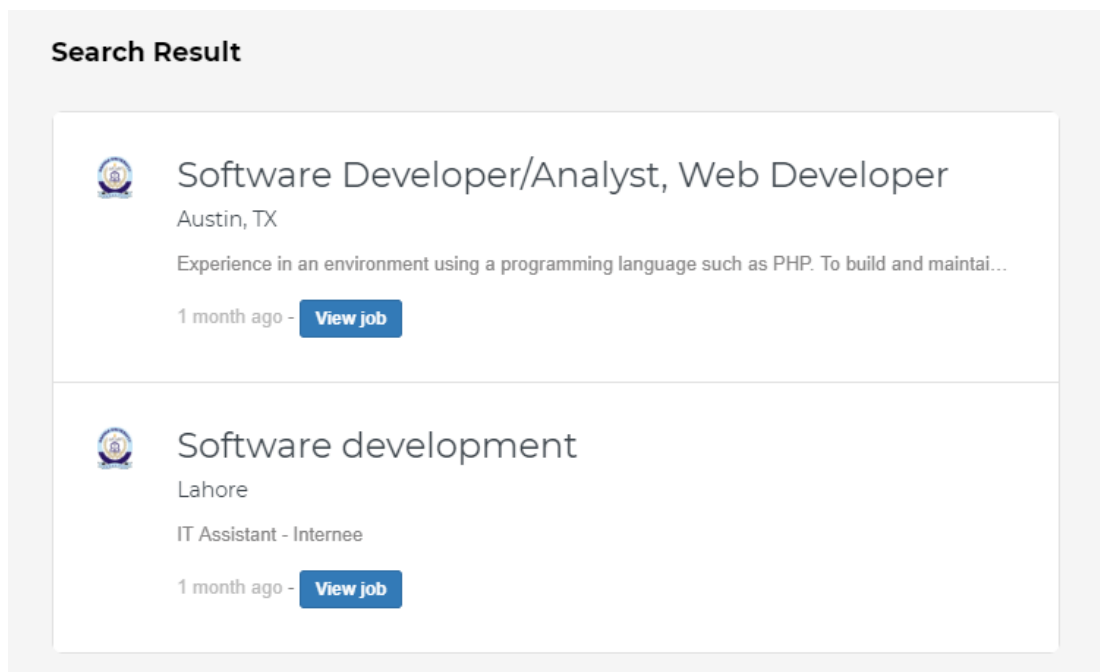


Figure 83 Find job results

5.2.8 Calendar

By clicking on Calendar icon you can view your jobs on calendar

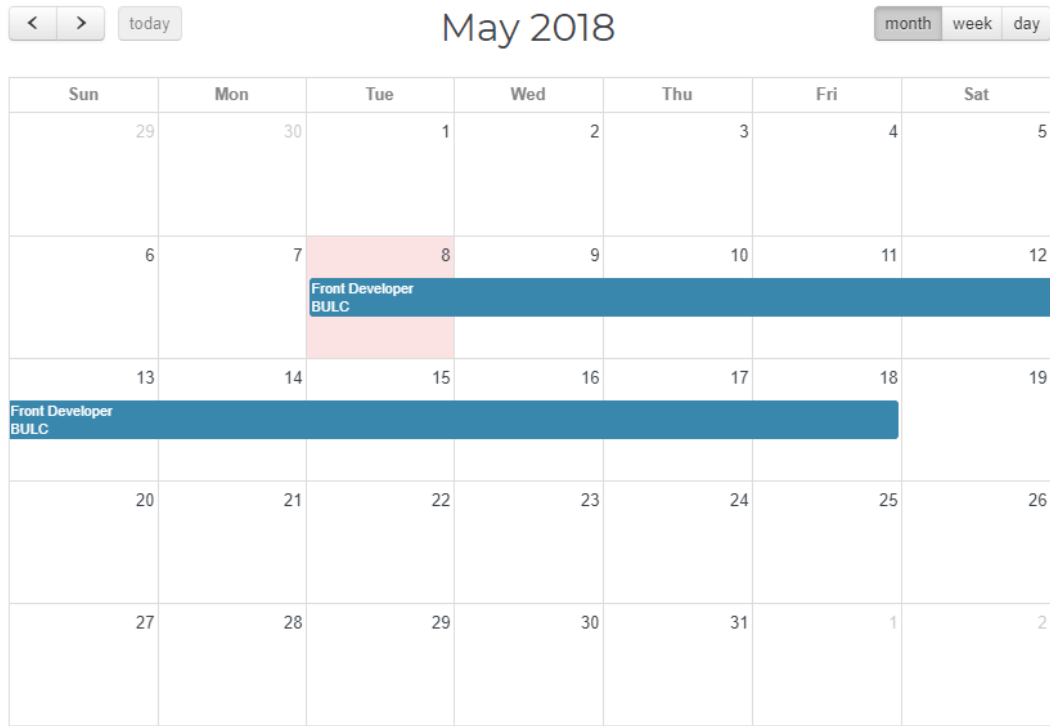


Figure 84 Calendar

5.2.9 Discussion

Discussion option contains conversation between students and organizations (view will be look like figure below)

Discussion

BULC 13 mins ago
what about the progress?

Ehsan 12 mins ago
80% completed. still working.

Leave a Message

Message

Send Message

Figure 85 Message

5.2.10 Notifications

Notification of newly posted jobs and assigned jobs can be seen by clicking bell icon. By clicking on notification you will be redirected to job associated to that notification.

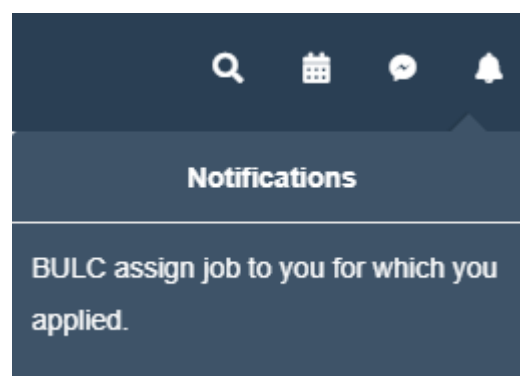


Figure 86 Notifications

5.2.11 Student Profile

By clicking on user name you can view, update profile and get help.

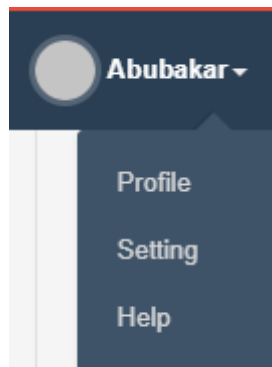


Figure 87 Student Profile

5.2.11.1 Profile

You can see your profile by clicking on profile

The image shows a user profile page for 'Abubakar'. The profile includes a placeholder for a profile picture, a name 'Abubakar', and a rating of 4.5 stars. The main content area is divided into sections: Contact (Email: cs.abubakar@gmail.com, Mobile: 0331-4314141), About (Student of computer Science), Program (BSCS), Skills (Web Programming), and Jobs (1 job: Data Science in Lahore, Pakistan). On the right side, there is a vertical navigation menu with an 'Edit Profile' button at the top, followed by links for 'Contact', 'About', 'program', 'Skills', and 'Jobs'.

Edit Profile
Contact
About
program
Skills
Jobs

Figure 88 Profile options

5.2.11.2 Setting

The screenshot shows a 'Profile Setting' page. On the left sidebar, there is a profile picture placeholder (6), the name 'Abubakar' (7), and two menu items: 'Change Password' (8) and 'Update profile Picture' (9). The main content area is titled 'Profile Setting' and contains several sections: 'About' with a text area containing 'Student of computer Science.' (1); 'Contact No.' with a text input field containing '0331-4314141' (2); 'Skills' with a dropdown menu showing 'Web Programming' (3); 'Program' with a dropdown menu showing '--Select Program--' (4); and 'Current Program: BSCS'. At the bottom of the main content area is a dark blue 'Update' button (5).

Figure 89 Profile setting

5.2.11.3 Help

This option contains FAQ about JOIN.

5.2.12 Logout



Figure 90 Logout button

5.3 JOIN for Organization

This section contains the user guide for organization users.

5.3.1 Home Page for Organizations

By clicking on Home following page will be show

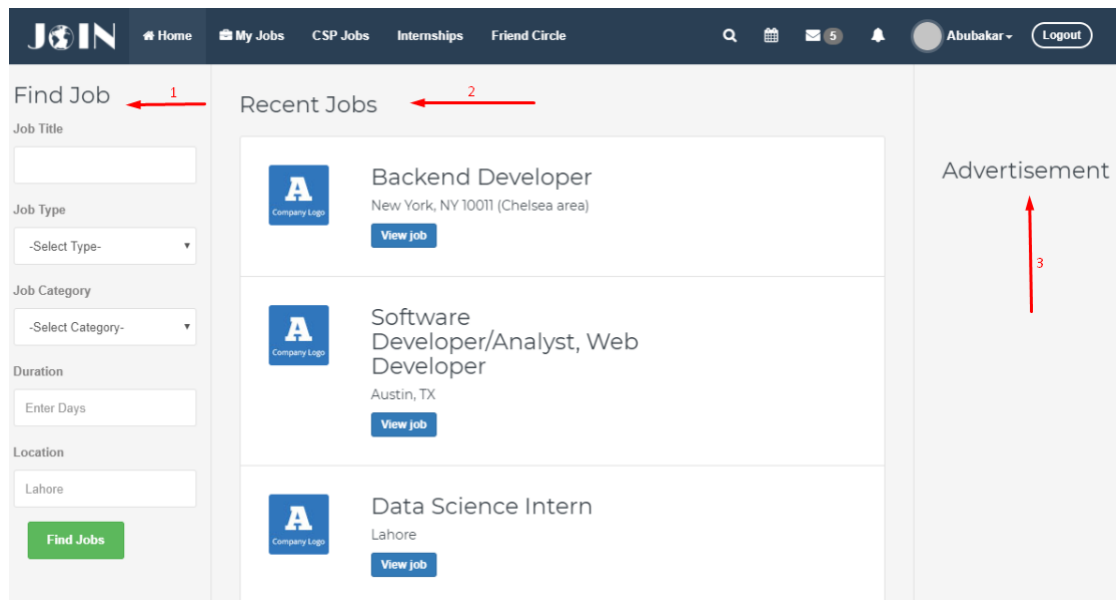


Figure 91 Home (for Organization)

- 4) You can find relevant jobs by setting appropriate filter
- 5) Recent jobs can be seen on home page
- 6) Advertise from Bahria University can be seen in advertisement section

5.3.1.1 Find Job

See section 5.2.1.1

5.3.1.2 Recent Jobs

See section 5.2.1.2

5.3.1.3 Advertisement

See section 5.2.1.3

5.3.2 My Jobs for Organization

You can find your “In progress”, “Jobs requests” and “unassigned jobs” under my jobs (as shown in figure on next page).

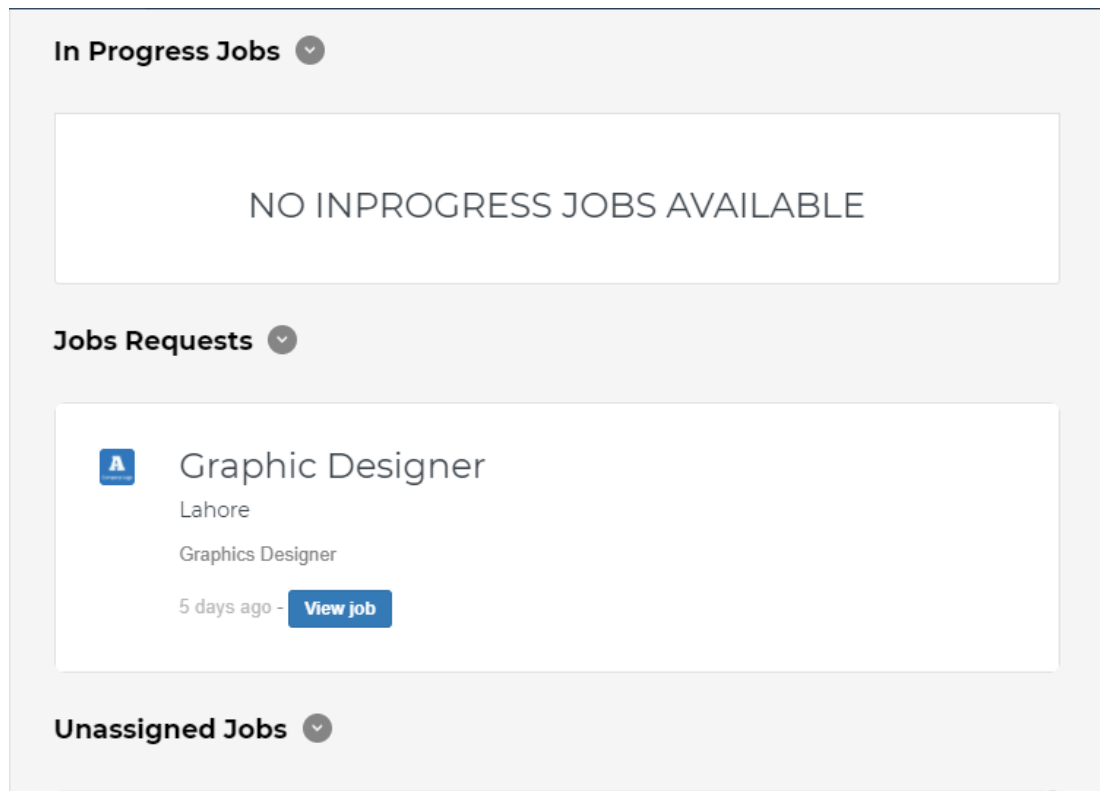


Figure 92 My Jobs (Organization)

5.3.2.1 In progress Jobs

You can view your “In progress jobs” in this section.

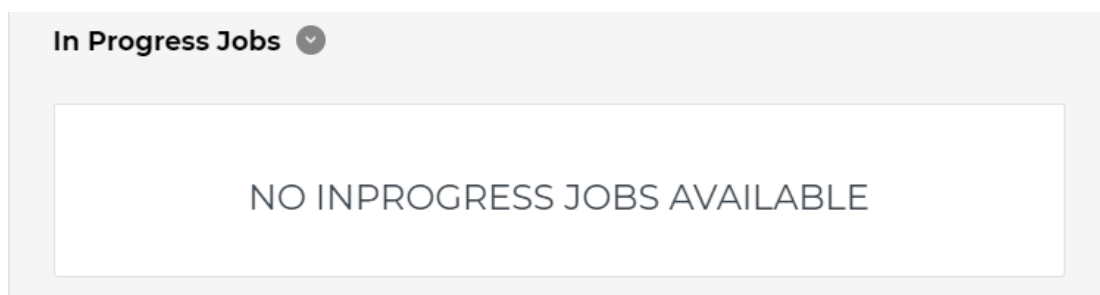


Figure 93 In Progress Jobs (Organization)

5.3.2.2 Job Requests

You can view job request received from students under “Jobs requests” section.

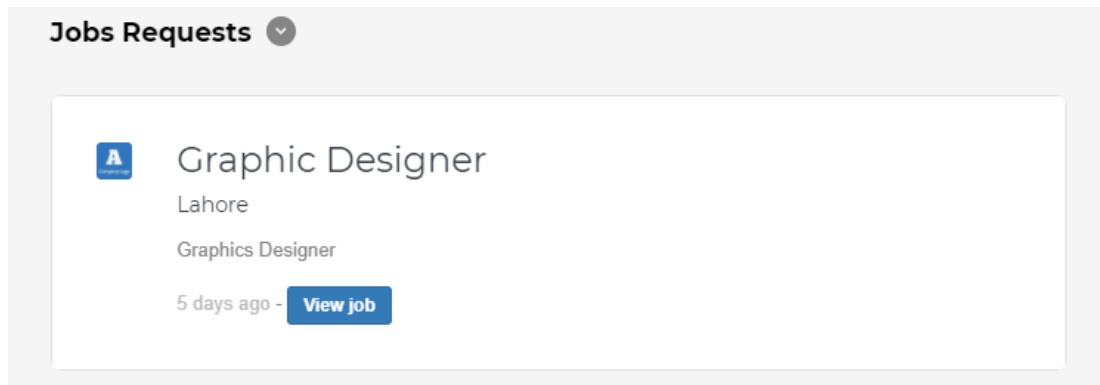


Figure 94 Job Requests (Organization)

5.3.2.3 Unassigned Jobs

You can view your unassigned jobs under this section.

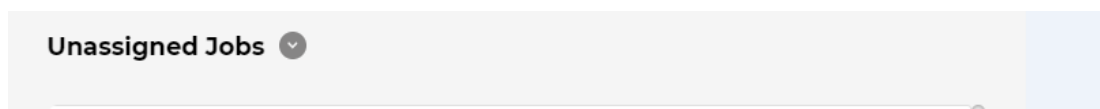


Figure 95 Unassigned Jobs (Organization)

5.3.3 CSP Jobs

See section 5.2.3

5.3.4 Internships

See section 5.3.4

5.3.7 Calendar

See section 5.2.7

5.3.8 Message

See section 5.2.8

5.3.9 Notifications

See section 5.2.9

5.3.10 Organization Profile

See section 5.2.10

5.3.10.1 Profile

See section 5.2.10.1

5.3.10.2 Setting

The screenshot displays the 'Profile Setting' interface for an organization. On the left sidebar, there is a profile picture placeholder (5), the organization name 'Bahria University' (6), and two menu items: 'Change Password' (7) and 'Update profile Picture' (8). The main content area is titled 'Profile Setting' and contains three sections: 'About' with a text area (1) for 'Tell about your self.', 'Contact No.' with a text input field (2) containing 'xxxx-xxxxxxx', and 'Address' with a text area (3) for 'Address'. At the bottom of the main area is a dark blue 'Update' button (4).

Figure 97 Profile setting (organization)

5.3.10.3 Help

See section 5.2.10.3

5.3.11 Logout

See section 5.2.11

5.4 Join for Admin

This section contains user guide for admin user.

5.4.1 Admin Sidebar

Admin side bar have following options

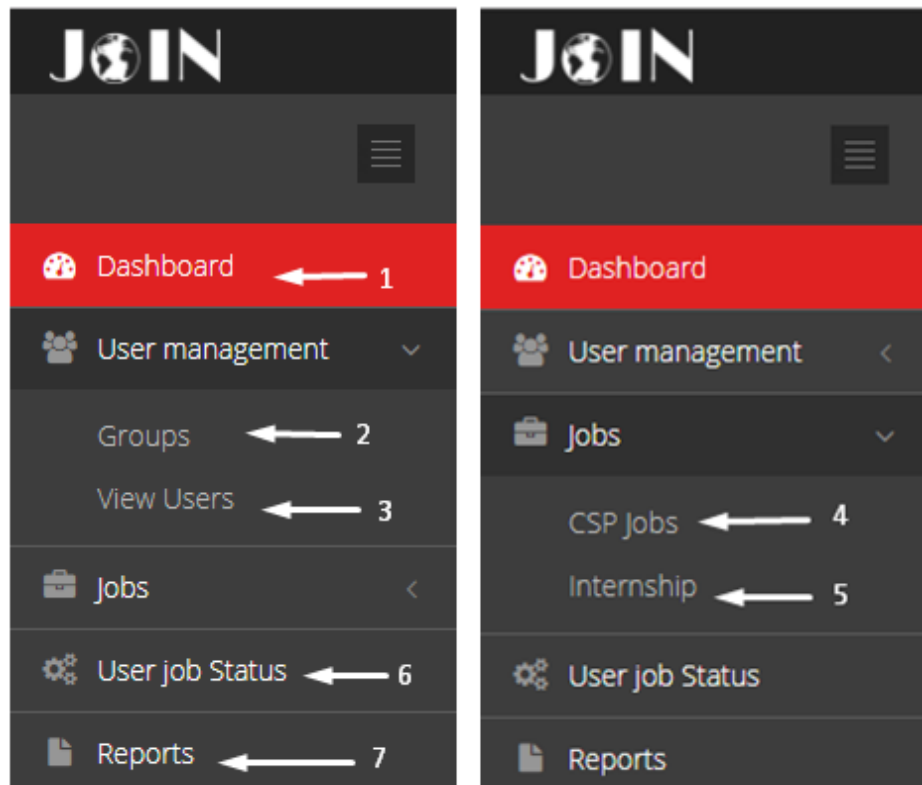


Figure 98 Admin sidebar

- 1) By clicking on Dashboard admin can view statistics of system
- 2) By clicking on Groups admin can view all created groups and create new groups
- 3) By clicking on View users admin can view all registered users
- 4) By clicking on CSP Jobs admin can view all posted CSP jobs
- 5) By clicking on Internships admin can view all posted internships
- 6) By clicking on User Jobs Status admin can view status of all jobs
- 7) By clicking on Reports admin can generate various type of reports

5.4.2 Dashboard

Admin dashboard contains statistics of system

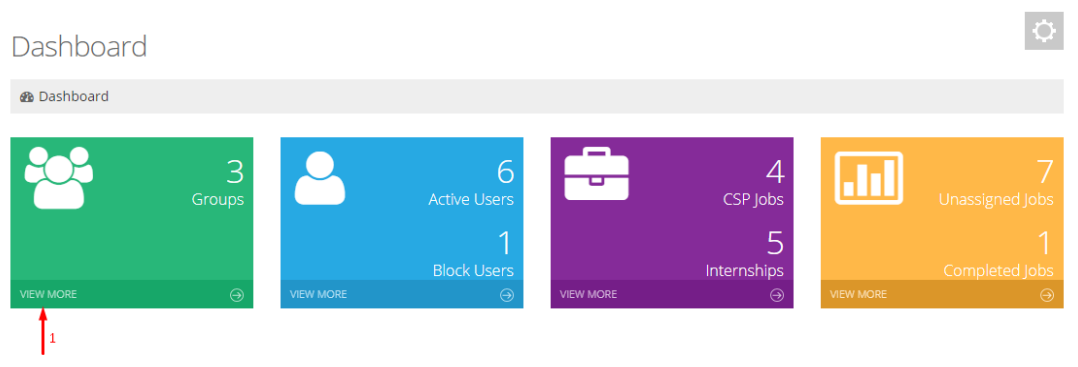


Figure 99 Admin dashboard

- 1) By clicking on View More admin can view full list of that category

5.4.3 User Management

User management option contains user groups and all users list

5.4.3.1 Groups

“Groups” option contains the all users groups created on JOIN by admin

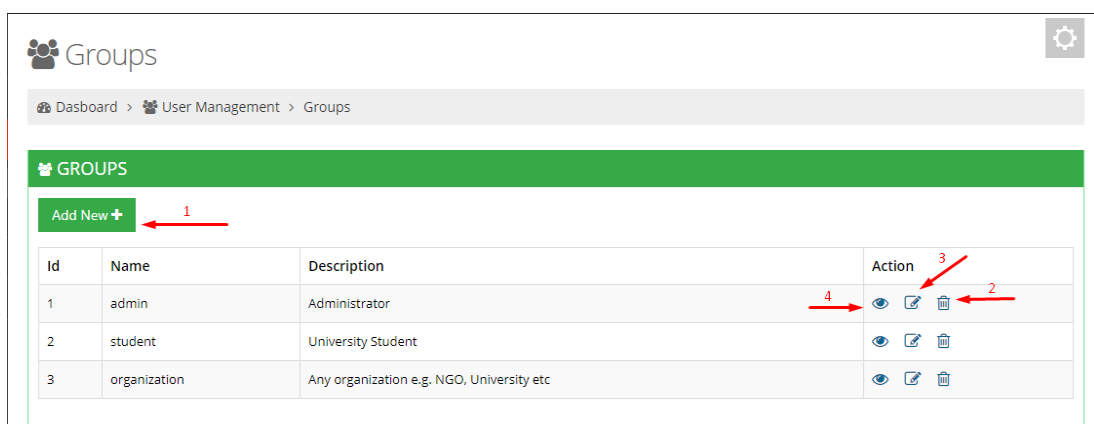


Figure 100 User Groups

- 1) By clicking on Add New button admin can add new user group and set its permissions
- 2) By clicking admin can delete a group

- 3) By clicking on edit button admin can edit group permissions
- 4) By clicking on view button admin can view group

5.4.3.1.1 Add New

Admin can add new group by clicking on add new button



Figure 101 Add new user group button

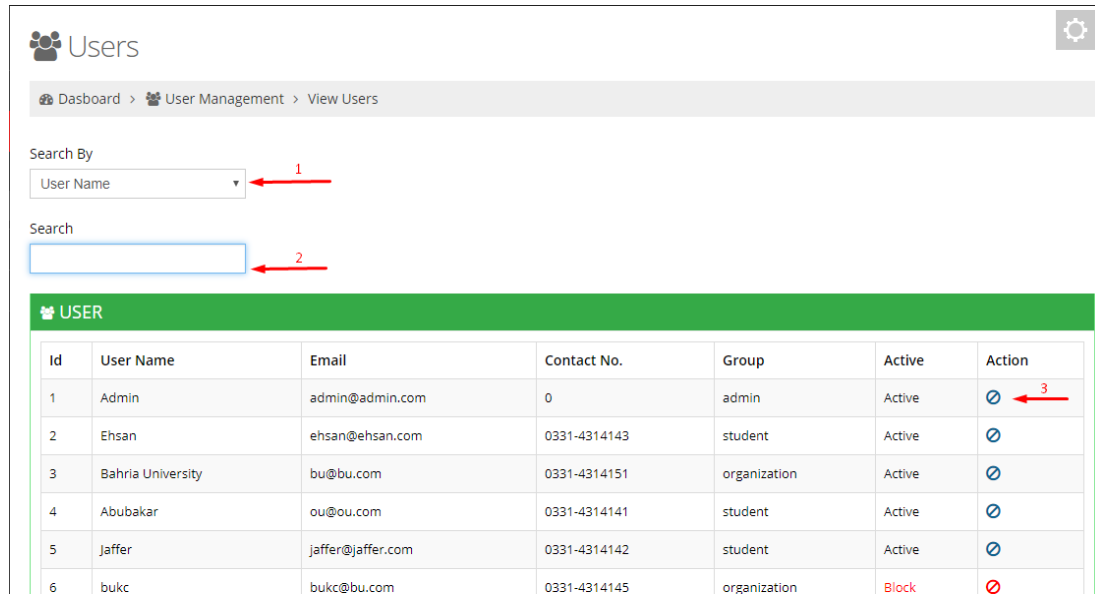
Admin enter group details, assign permissions to that group and click on “Save” button

A screenshot of a web form titled "+ Add New Group". The form has a blue header bar with the title. Below the header, there are two input fields: "Name" with a placeholder "Enter Group Name" and "Description" with a placeholder "Enter description". Below these fields, there are several sections, each starting with a checkbox and a label, followed by four sub-checkboxes: "Create", "Read", "Update", and "Delete". The sections are: "Home", "My Jobs", "CSP Jobs", "Internships", "Friend Circle", "Post Job", and "Find Job". Each section's sub-checkboxes are currently unchecked.

Figure 102 Add new group form

5.4.3.2 View Users

View users option contains list of all users where admin can search a specific user and block or activate user



The screenshot shows the 'View Users' page with a search form and a table of users. The search form includes a 'Search By' dropdown menu with 'User Name' selected (indicated by arrow 1) and a 'Search' input box (indicated by arrow 2). The table below has a green header 'USER' and columns for Id, User Name, Email, Contact No., Group, Active, and Action. The 'Action' column contains icons for activating (blue) or blocking (red) users. Arrow 3 points to the blue activate icon for the 'Admin' user.

Id	User Name	Email	Contact No.	Group	Active	Action
1	Admin	admin@admin.com	0	admin	Active	ⓘ
2	Ehsan	ehsan@ehsan.com	0331-4314143	student	Active	ⓘ
3	Bahria University	bu@bu.com	0331-4314151	organization	Active	ⓘ
4	Abubakar	ou@ou.com	0331-4314141	student	Active	ⓘ
5	Jaffer	jaffer@jaffer.com	0331-4314142	student	Active	ⓘ
6	bukc	bukc@bu.com	0331-4314145	organization	Block	⊘

Figure 103 View Users

- 1) Select the search criteria from “Search by” dropdown
- 2) Input search keyword in “Search” box
- 3) Click on action icon to active or block user the block user will be red and active users will be blue

5.4.4 Jobs

Jobs section contains all CSP jobs and Internships admin can perform operation on any job

5.4.4.1 CSP Jobs

Admin can search, view all posted CSP jobs and perform operation on them

CSP Jobs

Dashboard > Jobs > CSP Jobs

Search By
-Select- 1

Search
 2

Id	Job Title	Organization Name	Duration	Location	No. of Resources	Status	Action
1	Backend Developer	Bahria University	3	New York, NY 10011 (Chelsea area)	1 5	unassigned	4 3
2	Graphic Designer	Bahria University	10	Lahore	2	unassigned	5
3	Unity 3D Developer	Bahria University	10	Lahore	1	unassigned	5
4	SEO Expert	Bahria University	2	Lahore	1	inprogress	5

Figure 104 CSP Jobs

- 1) Select the search criteria from “Search by” dropdown
- 2) Input search keyword in “Search” box
- 3) Click on delete icon to delete a job
- 4) Click on view icon to view job details

5.4.4.2 Internships

Admin can search, view all posted Internships and perform operation on them

Internships

Dashboard > Jobs > Internships

Search By
-Select-

Search

Id	Job Title	Organization Name	Duration	Location	No. of Resources	Stipend	Status	Action
1	Software Developer/Analyst, Web Developer	Bahria University	2	Austin, TX	1	0	unassigned	
2	Data Science Intern	Bahria University	40	Lahore	1	1000	unassigned	

Figure 105 Internships

- 1) Select the search criteria from “Search by” dropdown
- 2) Input search keyword in “Search” box
- 3) Click on delete icon to delete an internship
- 4) Click on view icon to view internship details

5.4.5 User Job Status

Admin can search and view status of any job picked or applied by student

User job Status

Dashboard > User job Status

Search By
-Select-

Search

Sr. No.	User Name	Enrollment	Job Title	Organization	Status
1	Jaffer	03-134142-050	SEO Expert	Bahria University	inprogress
2	Jaffer	03-134142-050	Unity 3D Developer	Bahria University	applied

Figure 106 User Job Status

5.4.6 Reports

Admin can generate various type of reports

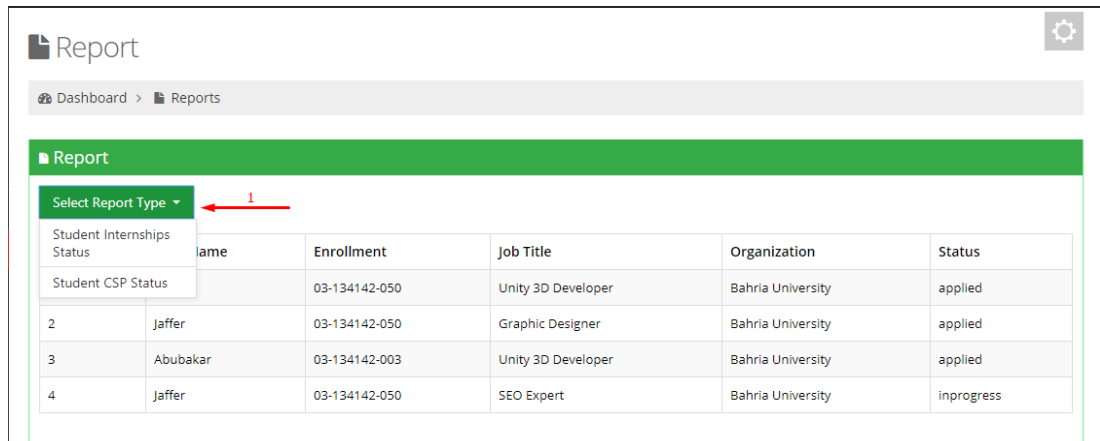


Figure 107 Reports

- 1) Select the report type and system will generate report of selected type

5.4.7 Admin logout

Admin can logout by clicking on logout option under his picture on upper right corner

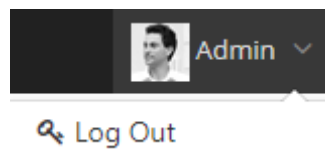


Figure 108 Admin Logout

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Objective of this project is to provide a platform called JOIN to students of Bahria University and organizations that offers internships or community support work for students. The web-based nature of this application allows great flexibility across devices. This application with number of different organizations will allow students of different disciplines to complete their CSP and Internship credit hours which are mandatory for degree completion. Unlike current process of offering internships or community jobs this applications automates whole process from inception to completion for each job. JOIN provided separate functionalities to students and organizations.

Moreover, there is a separate sub-system for admin to control all activities and users at JOIN. The development focused primarily on the usability of such an application, under strict usability guidelines, and the functionality needs of the user. The roll out of this application onto the World Wide Web shows the project was successful.

REFERENCES

- [1] Jeffrey L. Whitten, *System Analysis and Design Methods*, 7th Edition, McGraw-Hill, New York, 2007
- [2] P, C. (2007). *Software Engineering with Web Development Frameworks*. minerva.
- [3] Wikidot, “*Agile-Feature Driven Development*”, 2011, Available at: sliitmscfdd.wikidot.com/introduction-to-fdd
- [4] Martin, “*FDD & Web Development*”, 2003, Available at: www.featuredrivendevelopment.com/node/550
- [5] Palmer, S. (2009, 11 20). *An Introduction to Feature-Driven Development*. Retrieved from dzone.com: <https://dzone.com/articles/introduction-feature-driven>
- [6] United Nations, “*NAIIS Web Application User Manual*”, 2014, Available at: https://unfccc.int/files/national_reports/non-annex_i_national_communications/non-annex_i_inventory_software/application/pdf/naiis-user-manual.pdf

APPENDICES

APPENDIX A: User Story

- As a guest, I want to register to be a registered User.
- As a registered User I want to login to be an authenticated User.
- As a NGO user, I want to post the required work so that students can complete their CSP hours and NGOs can complete their work.
- As a company user, I want to post the internship for students so that student can get experience and companies can boost their work force.
- As a user, I want discussion facility so that user can communication with each other.
- As a student user, I want skill circle so that students with similar skill can get help from each other.
- As a user, I want auto job categorization facility so that all the jobs can be found in specific category.
- As a student user, I want to rate my experience with NGO or Organization so that they can be ranked.
- As an organization user, I want to rate my experience with student so that it can be ranked.
- As a student user, I want job notification so that I get notification when job related to my interest posted.
- As a user, I want team up facility so that NGOs can get team of students for specific task.
- As a user, I want selected tasks automatically scheduled on calendar so that user can check its pending tasks.

APPENDIX B: Requirement List

1. User must be registered to access JOIN
2. User login
3. Reset user password
4. Edit User Permissions
5. View/Search Users
6. Block a User
7. User Logout
8. Post an internship job
9. View an internship job
10. Apply for an internship job
11. Post a CSP job
12. View a CSP Job
13. Pick a CSP Job
14. Student should be able to view his/her competed or incomplete CSP and internships jobs.
15. Organization should be able to view its posted jobs.
16. Discussion between student and organization
17. System admin should be able to delete any post.
18. Student can rate organization for whom he/she worked and vice versa.
19. Discussion between multiple students and organization
20. Student can make a friend circle by adding students of same interest.
21. User should be able to search specific job or category
22. System should be able to automatically schedule selected task on calendar.
23. System automatically suggest similar type posts or more posts of that NGO or organization.
24. Student get job notification related to their interest after job posting.
25. Posted jobs automatically group into relevant category and can be search easily.
26. User should be able to upload his/her resume.
27. There should be rank of users like student, organization and university.

28. If any NGO need group of students for work, they can set required number of students while posting job.
29. If any Organization need group of students for work, they can set required number of students while posting job.
30. GUI should compatible with both Chrome and Mozilla.
31. Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
32. Final, integrated project code will test to ensure that complete project is integrated well and functioning properly.
33. Display all the content after loading of web page.
34. Server response should be keep in mind while designing data retrieval algorithms.
35. All the passwords should save in encrypted form
36. NGOs and organizations can access CSP portal and internship portal respectively
37. User should be able to search specific job
38. User should be able to search other users