



03-134142-092 ZAHID IQBAL

03-134142-050 MUHAMMAD JAFFER

Bahria One Link

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Dr Muhammad Aasim Qureshi

Department of Computer Sciences
Bahria University, Lahore Campus

June 2018

Certificate



We accept the work contained in the report titled

“Bahria One Link”

written by

ZAHID IQBAL

MUHAMMAD JAFFER

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor:

Dr Muhammad Aasim Qureshi

(Signature)

June 4th, 2018

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134142-092	ZAHID IQBAL	
03-134142-050	MUHAMMAD JAFFER	

Date : _____

We dedicate this work to our parents and all our lecturers of the Department of Computer Sciences, at Bahria University Lahore Campus, for their tireless and selfless efforts they have spared in making us who we are today.

Zahid Iqbal

Muhammad Jaffer

ACKNOWLEDGEMENTS

We would like to acknowledge our deepest gratitude to our supervisor, Dr. Muhammad Aasim Qureshi, for his guidance, constant attention, valuable suggestion, enthusiastic support and personal concern during the project. Furthermore, our appreciation also goes to our lab engineers from the Department of Computer Sciences for their time, attention and guidance. Finally, we would like to thank the beginning teachers and mentor-teachers that assisted us with this project.

BAHRIA ONE LINK (BOL)

ABSTRACT

This project aims to give a deeper understanding of the latest technologies for web-application development, by utilising an appropriate web development framework to implement a system that includes learning management system, correspondence system and user management system for Bahria University. This report follows the design and development of a final year computer science project from inception to completion. The objective of this project is to provide a platform that connect students and teachers of Bahria. Allow teachers and students to share study related material. Moreover, enable the official correspondence electronically. As mentioned above this project is represented in three modules including Correspondence System that helps the management to communicate more effectively, Learning Management System that provides e-learning facilities and User management system that helps the system admin to manage the users.

The front-end of this application is developed using HTML, CSS and JavaScript. Backend is developed in php using codeIgnitor framework. Some algorithms also designed for the conformance of critical requirements. This application automate the learning in Bahria University and also minimize the load of managing physical files of minutes. Moreover, by using of this application man power also reduced in official communication.

TABLE OF CONTENTS

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
 2	 Software Requirement Specification	 5
	2.1 User Classes and Characteristics	5
	2.2 Operating Environment	7
	2.3 Design and Implementation Constraints	7
	2.4 Assumptions and Dependencies	7
	2.5 System Requirement Chart	8
 3	 DESIGN	 11
	3.1 Use Case Diagram (System Level)	12
	3.1.1 Correspondence System Use Case Diagram	13
	3.1.2 Learning Management System Use Case Diagram	14
	3.1.3 User Management System Use Case Diagram	15
	3.2 Use Case Description	16
	3.3 Sequence Diagram	30
	3.4 Collaboration Diagram	45
	3.5 Domain Model	73
	3.6 Design Class Diagram	75
	3.7 Data Model	78
	3.7.1 Conceptual Model	78
	3.7.2 Internal Model	79

4	IMPLEMENTATION AND METHODOLOGY	85
4.1	Methodology	85
4.1.1	Present Available methodologies	85
4.1.2	Choosing a Suitable Methodology	88
4.1.3	Start working with FDD	89
4.1.3.1	Develop Overall Model	89
4.1.3.2	Build the feature list	90
4.1.3.3	Plan by feature	90
4.1.3.4	Design by Feature	90
4.1.3.5	Build by Feature	91
4.2	Implementation	91
4.2.1	Algorithm Design	91
4.2.2	Technical Aspect of the BOL	92
4.2.3	Back-end Technologies	93
4.2.4	Front-end Technologies	94
	Technology	94
	Major functions on a web page	94
	HTML	94
	Outline the structure of a web, it serves as the building blocks	94
	CSS	94
	Set the visual style to provide a better user interface	94
	JavaScript/JQuery	94
	Provide interactive elements	94
4.2.5	Basic structure of user Interface	94
4.2.6	Understand the flow of developed system	95
4.2.6.1	Bahria One Link (Complete)	95
4.2.6.2	User Management System (Module 1)	96
4.2.6.3	Learning Management System (Module 2)	101
4.2.6.4	Correspondence System (module 3)	106

5	USER MANUAL	111
5.1	Getting started	111
	5.1.1 User Access, Roles and Privileges	111
	5.1.2 How to access BOL	113
	5.1.3 Login & Registration at BOL	113
	5.1.4 Role based authorized home screens	115
	5.1.5 Introduction to Navigation Bar	117
5.2	Learning Management System (LMS)	118
	5.2.1 MyClass	120
5.2.1.1	Create a new class	120
5.2.1.2	Search and join a class	122
5.2.1.3	Open a class	122
5.2.1.4	Upload and Download Resource from class	123
5.2.1.5	Submit assignment in class	125
5.2.1.6	View / Make Announcement in class	126
	5.2.2 Library	127
5.2.2.1	Upload and Download resource from Library	128
5.2.2.2	Advance search in Library	129
5.2.2.3	Active/Dismiss in library	130
	5.2.3 Alumni	131
	5.2.4 Profile	132
5.2.4.1	Update profile Information	132
5.2.4.2	Update Profile Picture	133
5.3	Correspondence System (CS)	134
	5.3.1 Create Minute	135
5.3.1.1	Send a Minute	136
5.3.1.2	Save a Minute	138
	5.3.2 Inbox	139
5.3.2.1	Open a received minute	139
5.3.2.2	Giving feedback of received minute	143

5.3.2.3	Giving Feedback to a specific person	144
5.3.2.4	Print a received minute and/or Performa	145
	5.3.3 Sent	146
5.3.3.1	Open a sent minute	146
5.3.3.2	Giving feedback of Sent minute	150
5.3.3.3	Print a Sent minute and/or Performa	151
5.3.3.4	Close a minute	152
	5.3.4 Draft	153
5.3.4.1	Edit a saved minute	153
5.4	User Management System (UMS)	154
	5.4.1 ViewDeleteUser	155
5.4.1.1	Search a User	155
5.4.1.2	Block/Active a User	156
	5.4.2 ResetPassword	157
5.4.2.1	Reset password of a user	158
1)	Click on 'ResetPassword' from sidebar	158
	5.4.3 EditPermissions	159
5.4.3.1	Edit Permission of a group	160
5.5	Contact	161
6	CONCLUSION AND RECOMMENDATIONS	163
	REFERENCES	165
	APPENDICES	167

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1:	System Requirement Chart	8
Table 2:	Use Case User Registration	16
Table 3:	Use Case User login	17
Table 4:	Use Case User logout	17
Table 5:	Use Case Reset User Password by admin	18
Table 6:	Use Case Reset User Password by user	18
Table 7:	Use Case Edit Group Permissions	19
Table 8:	Use Case View/Search User	19
Table 9:	Use Case Active/Block User	20
Table 10:	Use Case Create Virtual Class	20
Table 11:	Use Case Upload File in Virtual Class	21
Table 12:	Use Case Download File From Virtual Class	21
Table 13:	Use Case Open Virtual Class	22
Table 14:	Use Case Join Virtual Class	22
Table 15:	Use Case Make Announcement in Virtual Class	23
Table 16:	Use Case View Announcement From Virtual Class	23
Table 17:	Use Case Alumni Searching	24
Table 18:	Use Case Alumni Profile Open	24
Table 19:	Use Case Alumni Update Profile	25
Table 20:	Use Case Upload File in Library	25
Table 21:	Use Case Active/Dismiss New Library Content	25

Table 22: Use Case Download File From Library	26
Table 23: Use Case Create Minute	27
Table 24: Use Case Save Minute	27
Table 25: Use Case Feedback of Minute	28
Table 26: Use Case Throw back a minute	28
Table 27: Use Case Print Minute	29
Table 28: Use Case View Minute	29
Table 29: User Internal Modal	79
Table 30: Privileges Internal Modal	79
Table 31: Side Bar Internal Modal	80
Table 32: Groups Internal Modal	80
Table 33: User Notification Internal Modal	80
Table 34: Virtual Class Internal Modal	81
Table 35: User Class Mapping Internal Modal	81
Table 36: Virtual Class Resource Internal Modal	81
Table 37: Virtual Class Announcement Internal Modal	82
Table 38: Library Internal Modal	82
Table 39: Alumni Internal Modal	83
Table 40: Correspondence Internal Modal	83
Table 41: User Correspondence Mapping Internal Modal	84
Table 42: Correspondence Feedback Internal Modal	84
Table 43: Correspondence Trail Internal Model	84
Table 44: Front End Technologies	94
Table 45: Module Authorization and Privileges	111

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1:	BOL System Level Use Case Diagram	12
Figure 2:	Sequence Diagram Registration	30
Figure 3:	Sequence Diagram Login	31
Figure 4:	Sequence Diagram Reset User Password by Admin	31
Figure 5:	Sequence Diagram Reset User Password by User	32
Figure 6:	Sequence Diagram View User	32
Figure 7:	Sequence Diagram Active/Block User	33
Figure 8:	Sequence Diagram Edit Permissions	34
Figure 9:	Sequence Diagram View Profile By Alumni	34
Figure 10:	Sequence Diagram Create Virtual Class	35
Figure 11:	Sequence Diagram Upload Files in Class	35
Figure 12:	Sequence Diagram Download Resource from Class	36
Figure 13:	Sequence Diagram Open Virtual Class	36
Figure 14:	Sequence Diagram Join Class	37
Figure 15:	Sequence Diagram Make Announcement in class	37
Figure 16:	Sequence Diagram View Announcement from Class	38
Figure 17:	Sequence Diagram Alumni Search	38
Figure 18:	Sequence Diagram Update Alumni Profile	39
Figure 19:	Sequence Diagram Upload Resource in Library	39

Figure 20: Sequence Diagram Download Resource from Library	40
Figure 21: Sequence Diagram Active/Dismiss New Content From Library	41
Figure 22: Create a Minute	41
Figure 23: Sequence Diagram Feedback a Minute	42
Figure 24: Sequence Diagram Throw back a Minute	42
Figure 25: Sequence Diagram Save a Minute	43
Figure 26: Sequence Diagram Print a Minute	43
Figure 27: Sequence Diagram View a Minute	44
Figure 28: Sequence Diagram Logout	44
Figure 29: Collaboration Diagram Registration	46
Figure 30: Collaboration Diagram Login	47
Figure 31: Collaboration Diagram Reset User Password by Admin	48
Figure 32: Collaboration Diagram Reset Password by User	49
Figure 33: Collaboration Diagram View User	50
Figure 34: Collaboration Diagram Active/Block User	51
Figure 35: Collaboration Diagram Edit Permissions	52
Figure 36: Collaboration Diagram Create Virtual Class	53
Figure 37: Collaboration Diagram Upload Files in Virtual Class	54
Figure 38: Collaboration Diagram Download Resource from Class	55
Figure 39: Collaboration Diagram Open Virtual Class	56
Figure 40: Collaboration Diagram Join Virtual Class	57
Figure 41: Collaboration Diagram Make Announcement in Class	58
Figure 42: Collaboration Diagram View Announcement from Class	59

Figure 43: Collaboration Diagram Alumni Search	60
Figure 44: Collaboration Diagram View Profile by Alumni	61
Figure 45: Collaboration Diagram Update Alumni Profile	62
Figure 46: Collaboration Diagram Resource in Library	63
Figure 47: Collaboration Diagram Active/Dismiss New Content from Library	64
Figure 48: Collaboration Diagram Download Resource from Library	65
Figure 49: Collaboration Diagram Create a Minute	66
Figure 50: Collaboration Diagram Feedback on a Minute	67
Figure 51: Collaboration Diagram Throw Back a Minute	68
Figure 52: Collaboration Diagram Save a Minute	69
Figure 53: Collaboration Diagram Print a Minute	70
Figure 54: Collaboration Diagram View a Minute	71
Figure 55: Collaboration Diagram Logout	72
Figure 56: Domain Model - 1	73
Figure 57: Domain Model - 2	74
Figure 58: Class Diagram - 1	76
Figure 59: Class Diagram - 2	77
Figure 60: Conceptual Model	78
Figure 61: Waterfall Model	86
Figure 62: Agile Model	87
Figure 63: RAD Model	87
Figure 64: FDD Model	88
Figure 65: Waterfall vs. Agile	89
Figure 66: Multi Upload Algorithm - 1	91

Figure 67: Multi Upload Algorithm - 2	92
Figure 68: Application Architecture	93
Figure 69: User Interface Structure	94
Figure 70: Flow Chart BOL	95
Figure 71: Flow Chart User Management System	96
Figure 72: Flow Chart View Delete User UMS	97
Figure 73: Flow Chart Edit Group Permissions UMS	98
Figure 74: Flow Chart Reset User Password UMS	99
Figure 75: Flow Chart Reset Password	100
Figure 76: Flow Chart Learning Management System	101
Figure 77: Flow Chart My Class LMS	102
Figure 78: Flow Chart Library LMS	103
Figure 79: Flow Chart Alumni Search LMS	104
Figure 80: Flow Chart Profile LMS	105
Figure 81: Flow Chart Correspondence System	106
Figure 82: Flow Chart Create Minute CS	107
Figure 83: Flow Chart Draft CS	108
Figure 84: Flow Chart Inbox CS	109
Figure 85: Flow Chart Sent CS	110
Figure 86: How to Access BOL	113
Figure 87: Login Screen	113
Figure 88: SignUp Screen	114
Figure 89: Home Screen for Faculty and HODs	115
Figure 90: Home Scree for Admin	116
Figure 91: Introduction to Navigation Bar	117

Figure 92: Reset Password by User	118
Figure 93: LMS Home Screen	118
Figure 94: Shrink Side bar	119
Figure 95: MyClass View	120
Figure 96: Create New Class Button	120
Figure 97: Create New Class Screen	121
Figure 98:Class Create Message	121
Figure 99: Class Access Code Screen	122
Figure 100: Search Class Result Screen	122
Figure 101: Registered Classes Screen	123
Figure 102: Registered Class View after Open	123
Figure 103: Upload Resource Button	123
Figure 104: Upload Resource in Class	124
Figure 105: Virtual Class Resources View	124
Figure 106: Submit Assignment button	125
Figure 107: Assignment Submit Form	125
Figure 108: Assignment Submitted Section	125
Figure 109: Make Announcement Screen	126
Figure 110: Announcement View	126
Figure 111: Library Home Screen	127
Figure 112: Upload Resource Button in Library	128
Figure 113: Upload Resource in Library	128
Figure 114: Advance Search Button	129
Figure 115: Advance Search Screen	129
Figure 116: Search Result Screen	130

Figure 117: New Uploaded Content Approve/Disapprove in Library	130
Figure 118: Download Link in Library	130
Figure 119: Library Content Active Button	130
Figure 120: Library Content Dismiss Button	131
Figure 121: Alumni Advance Search Screen	131
Figure 122: Alumni Search Result Screen	131
Figure 123: Alumni Profile View	132
Figure 124: Edit User Profile Tab	132
Figure 125: Update User Information Screen	133
Figure 126: Update User Profile Button	133
Figure 127: Edit User Profile Tab	133
Figure 128: Update User Profile Picture	134
Figure 129: Correspondence Home Screen	134
Figure 130: Minute List	135
Figure 131: Create Minute Screen	135
Figure 132: Performa Attached with Minute	136
Figure 133: Recipient List	136
Figure 134: Recipient List	137
Figure 135: Minute Input Area	137
Figure 136: Minute Flags without Attachments	137
Figure 137: Minute Flags with Attachments	138
Figure 138: Sent Minute Button	138
Figure 139: Minute Input Area	138
Figure 140: Save Minute Button	138
Figure 141: Inbox View	139

Figure 142: Received Minute View	140
Figure 143: Back to Inbox Button	140
Figure 144: Print Minute Button	140
Figure 145: Print Performa Button	141
Figure 146: Received Minute Text	141
Figure 147: Sender Detail in Received Minute	141
Figure 148: Final Recipient in Received Minute	141
Figure 149: Through Recipient of Received Minute	141
Figure 150: Feedback List in Received Minute	142
Figure 151: Flags in Received Minute	142
Figure 152: Attachments in flag in Received Minute	142
Figure 153: Feedback Input Area	142
Figure 154: Throw Back Button	143
Figure 155: Post Feedback Button	143
Figure 156: Add Flags Button	143
Figure 157: Filled Feedback Input Area	143
Figure 158: Post Feedback Button	144
Figure 159: Feedback View	144
Figure 160: Post To button	144
Figure 161: Feedback Dialog box	144
Figure 162: Print Button in Received Minute	145
Figure 163: Print Performa Button in Received Minute	145
Figure 164: Print Minute Screen in Received Minute	145
Figure 165: Print Button	146
Figure 166: Sent Minute View	146

Figure 167: Sent Minute View	147
Figure 168: Back to Sent Button	147
Figure 169: Print Minute Button in Sent Minute	147
Figure 170: Print Performa Button in Sent Minute	148
Figure 171: Sent Minute Text	148
Figure 172: Sender Detail in Sent Minute	148
Figure 173: Final Recipient in Sent Minute	148
Figure 174: Through Recipient in Sent Minute	148
Figure 175: Feedback in Sent Minute	149
Figure 176: Flags in Sent Minute	149
Figure 177: Attachments in flag in Sent Minute	149
Figure 178: Feedback Input Area in Sent Minute	149
Figure 179: Post Feedback Button in Sent Minute	150
Figure 180: Add Flags Button in Sent Minute	150
Figure 181: Filled Feedback Input area in Sent Minute	150
Figure 182: Post Feedback Button in Sent Minute	150
Figure 183: Feedback List in Sent Minute	151
Figure 184: Print Minute Button in Sent Minute	151
Figure 185: Print Performa Button in Sent Minute	151
Figure 186: Print Minute View in Sent Minute	152
Figure 187: Print Button	152
Figure 188: Close Correspondence Button	152
Figure 189: Correspondence Close Confirmation Dialog	153
Figure 190: Draft View	153
Figure 191: Filled Minute Input Area	154

Figure 192: Admin Home Screen	154
Figure 193: View Delete User Screen	155
Figure 194: Search User Field	156
Figure 195: Search User Result	156
Figure 196: Search User Field	156
Figure 197: Registered Users	157
Figure 198: Active (Unblock) User Button	157
Figure 199: Block User Button	157
Figure 200: Reset User Password View	158
Figure 201: Check User Existence Filed	159
Figure 202: New and Conform Password Fields	159
Figure 203: Reset Password Button by Admin	159
Figure 204: Edit Permission View	160
Figure 205: User Group Detail	160
Figure 206: User Group Permission Detail	160
Figure 207: Save Button	161

LIST OF ABBREVIATIONS

BOL	Bahria One Link
UMS	User Management System
LMS	Learning Management System
CS	Correspondence System
FDD	Feature Driven Development
HOD	Head of Department
PO SRC	Program Officer Student Resource Centre

CHAPTER 1

INTRODUCTION

1.1 Background

Universities with multiple departments/campuses that are in different locations even in this age of email, cell phones and web conferencing, have communication gaps. Most of the universities in Pakistan now shifting towards e-learning that minimizes the communication gap between students and teachers. Moreover, organizations uses traditional ways of communication (i.e. circulation of piece of papers called memos or minutes) between or inside departments.

Bahria University uses different learning platforms like Piazza, Moodle for e-learning purposes but not all faculty members strictly follows these platforms. Some faculty also uses social media platform like Facebook, Google Groups. Due to these multiple learning methodologies makes confusion or difficulties for students. Moreover, management of Bahria University uses minutes for inter department communication in which a person is responsible for physically circulate the minute to related people.

1.2 Problem Statements

Students and teachers even in this modern era of technology follows the traditional way of learning. Course related material cannot provide to all students efficiently even not available at the time of need. Moreover, most of the students did not find the course book and course outline.

Official correspondence between management, staff and faculty, also between departments/units, due to reliance on hardcopy documents, physical signatures and manual processes, are inconsistent, complicated, resource intensive and time-consuming.

1.3 Aims and Objectives

The objectives of this project are shown as following:

- i) To provide a platform that connect students and teachers of Bahria University
- ii) Allow teachers and students to share study related material
- iii) Enable the official correspondence electronically

1.4 Scope of Project

Following points defines the scope of this project.

Correspondence System (CS)

- User can create different type of minutes as below
 - i. Trip
 - ii. General
 - iii. Procurement
 - iv. Event along with Performa
 - v. TY Duty along with Performa

- User can send and receive official correspondence in official format
- System is able to store correspondence (i.e. send and receive) separately
- System is able to save the correspondence
- User can view and print any correspondence when require
- User can give Feedback on a minute
- User can Throw back a minute
- Sender/initiator can close the minute

Learning Management System (LMS)

- Teacher can upload study related material like lectures, hand outs, books assignments and reference links
- Students can submit assignments
- Virtual class facility (like Piazza)
- Batch classification for alumni searching
- E-Library

User Management System (UMS)

- Authentication of university personnel
- Authorization of users
- Edit permissions of users
- Block/Active users
- Reset user password

CHAPTER 2

Software Requirement Specification

2.1 User Classes and Characteristics

Bahria One Link (BOL) is divided into three modules i.e. LMS, CS and UMS. Following are the users of each module respectfully.

Learning Management System (LMS)

LMS automates the learning and used by:

- a) Faculty, that can create virtual class, make announcement in virtual class, upload/download study related material to/from both e-library and virtual class. Teacher also can search for Alumni.
- b) Student, that can join virtual class, upload/download study related material to/from both virtual class and e-library. Student also can search for Alumni.
- c) Librarian, that have right to active or dismiss newly uploaded content in library. Librarian also can upload/download content to/from library.

Correspondence System (CS)

Correspondence system automates the official correspondence. It used by all employees of Bahria University that can send, save, receive, view and print correspondence according to their privileges. Employees includes:

- Director
- Deputy Director
- HODs
- Faculty
- Student Advisors
- Account officer
- Examination Department Head
- Finance Officer
- Procurement Officer
- Admission officer
- PO SRC
- Network Engineer
- System engineer

User Management System (UMS)

UMS manages all the users of Bahria One Link (BOL). It is used by System Engineer who is administrator of Bahria One Link that can reset passwords on users' requests, edit group permissions and view all details of users. Moreover, active/block users.

2.2 Operating Environment

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

- Google Chrome version 35.1 or higher

2.3 Design and Implementation Constraints

BOL is a web-based system therefore front-end of BOL developed using HTML and CSS 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. BOL 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

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

2.5 System Requirement Chart

Table 1: System Requirement Chart

ID	Priority	Type	Source	Contained in Use-Case	Description
UMS-R1	High	Functional	Bahria University	BOL-UMS-1.0.0	User must be registered to access BOL
UMS-R2	High	Functional	Bahria University	BOL-UMS-2.0.0	User login
UMS-R3	Medium	Functional	Bahria University	BOL-UMS-3.0.0	Reset user password by Admin
UMS-R4	Medium	Functional	Bahria University	BOL-UMS-4.0.0	Edit Group Permissions
UMS-R5	Medium	Functional	Bahria University	BOL-UMS-5.0.0	View/Search Users
UMS-R6	Medium	Functional	Bahria University	BOL-UMS-6.0.0	Block/Active a User
UMS-R7	Medium	Functional	Bahria University	BOL-UMS-7.0.0	User Logout
UMS-R8	Medium	Functional	Bahria University	BOL-UMS-8.0.0	Reset user password by User
LMS-R1	Medium	Functional	Bahria University	BOL-LMS-1.0.0	Create a Virtual Class
LMS-R2	Medium	Functional	Bahria University	BOL-LMS-1.1.0	Upload resources in virtual class
LMS-R3	Medium	Functional	Bahria University	BOL-LMS-1.2.0	Download resource from virtual class
LMS-R4	Medium	Functional	Bahria University	BOL-LMS-1.3.0	Open a Virtual Class
LMS-R5	Medium	Functional	Bahria University	BOL-LMS-1.4.0	Join a virtual class
LMS-R6	Medium	Functional	Bahria University	BOL-LMS-1.5.0	Make an Announcement in Virtual Class
LMS-R7	Medium	Functional	Bahria University	BOL-LMS-1.6.0	View Announcements in virtual class
LMS-R8	Low	Functional	Bahria University	BOL-LMS-2.0.0	Search for Alumni
LMS-R9	Medium	Functional	Bahria University	BOL-LMS-4.0.0	Upload study resource in library
LMS-R10	Medium	Functional	Bahria University	BOL-LMS-4.1.0	Active/Dismiss New Library Content
LMS-R11	Medium	Functional	Bahria University	BOL-LMS-5.0.0	Download resource from Library
LMS-R12	Medium	Functional	Bahria University	BOL-LMS-6.0.0	Alumni Profile open
LMS-R13	Medium	Functional	Bahria University	BOL-LMS-6.1.0	Alumni Profile Update

CS-R1	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-1.0.0	Creating a new Minute
CS-R2	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-1.0.0	Add flags having attachments in Minute
CS-R3	Medium	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-1.1.0	Save Minute
CS-R4	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-1.2.0	Give Feedback of a Minute
CS-R5	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-1.3.0	Throw back a Minute
CS-R6	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-2.0.0	Printing a Minute
CS-R7	High	Functional	Mr. Farhan Sherazi (HoD CS&IT)	BOL-CS-3.0.0	Viewing a Minute
LMS-R14	Medium	Functional	Bahria University	N/A	There should be a popup window for uploading.
LMS-R15	Low	Functional	Bahria University	N/A	There should be type of users like faculty, student, alumni etc.
LMS-R16	Medium	Functional	Bahria University	N/A	Students who are graduated will be promoted automatically to the type of alumni.
CS-R8	Low	Functional	Bahria University	N/A	Correspondence system should be able to note the feedback time of receiver and use it at the time of printing.
BOL-R1	High	Non-Functional	Bahria University	N/A	GUI should compatible with Chrome and Mozilla both.
BOL-R2	High	Non-Functional	Bahria University	N/A	Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
BOL-R3	Low	Non-Functional	Bahria University	N/A	Final, integrated project code will test to ensure that complete project

					is integrated well and functioning properly.
BOL-R4	High	Non-Functional	Bahria University	N/A	Display all the content after loading of web page.
BOL-R5	High	Non-Functional	Bahria University	N/A	Server response should be keep in mind while designing data retrieval algorithms.
UMS-R9	High	Non-Functional	Bahria University	N/A	All the passwords should save in encrypted form
CS-R9	High	Non-Functional	Bahria University	N/A	In correspondence system only related people can access certain communication
CS-R10	High	Non-Functional	Bahria University	N/A	In correspondence system privileges of sending correspondence should be strict to internal correspondence policy of Bahria University
CS-R11	High	Non-Functional	Bahria University	N/A	In correspondence system minute should save in Draft

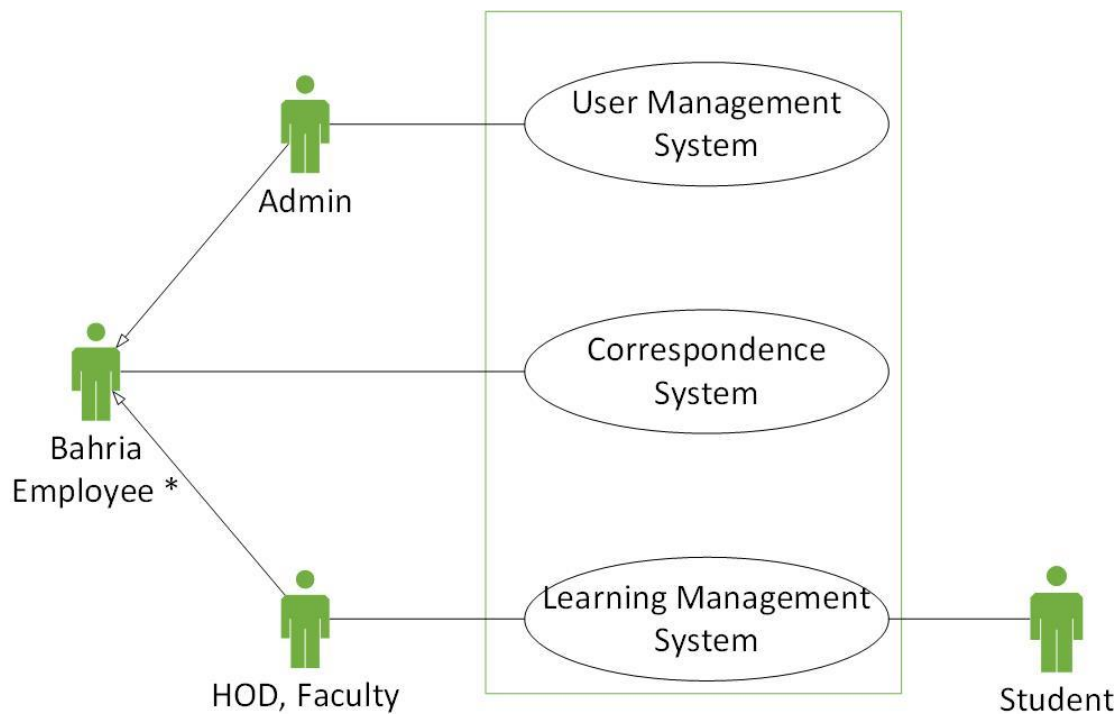
CHAPTER 3

DESIGN

Following artefacts included in this Chapter

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

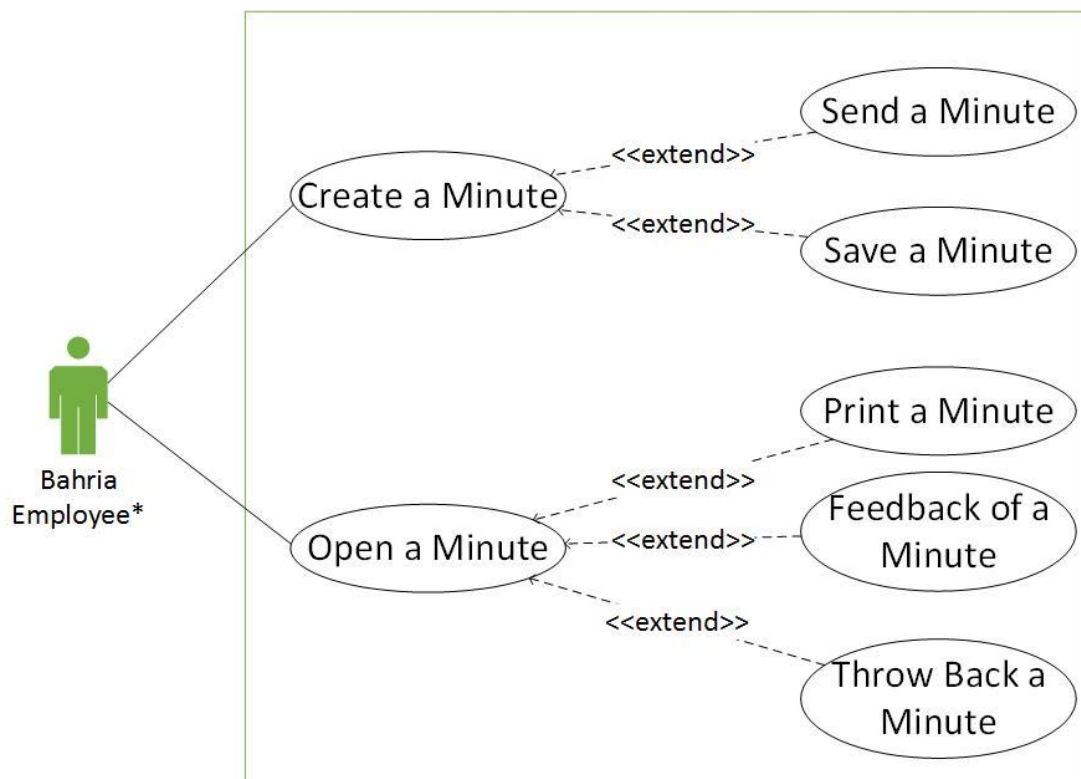
3.1 Use Case Diagram (System Level)



*Director, DY. Director, HODs, Student Advisors, Account officer, Examination, Department Head, Librarian, Finance Officer, Procurement Officer, Admission officer, PO SRC and Network Engineer

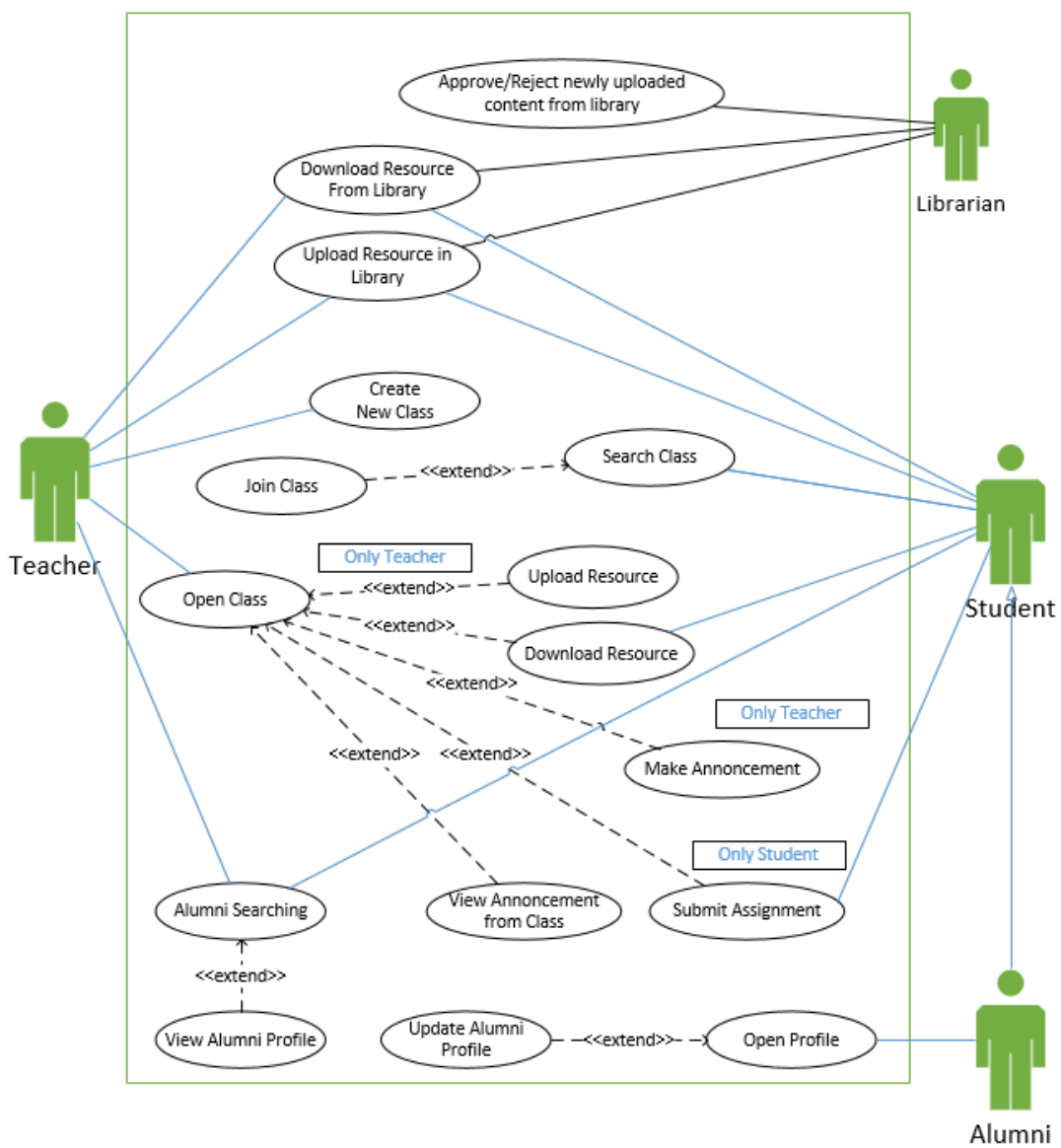
Figure 1: BOL System Level Use Case Diagram

3.1.1 Correspondence System Use Case Diagram

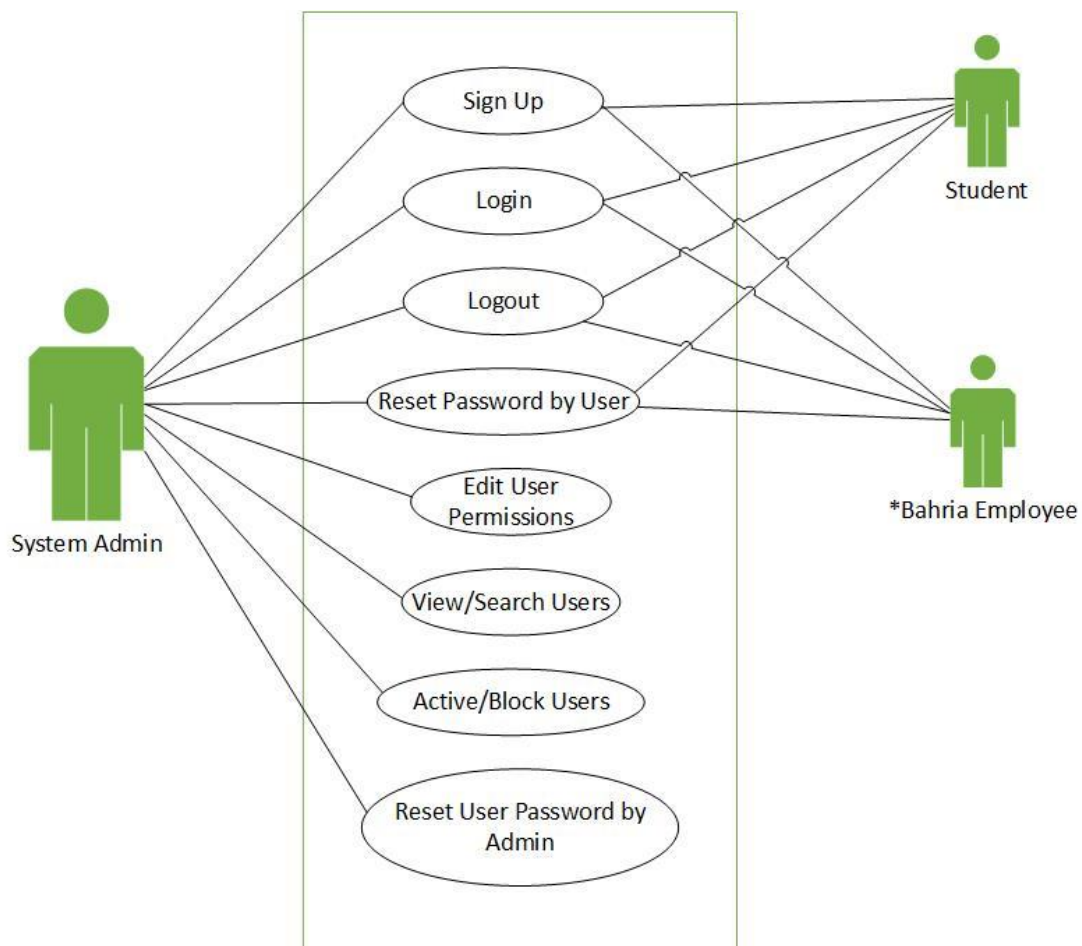


*Director, DY. Director, HODs, Student Advisors, Account officer, Examination, Department Head, Librarian, Finance Officer, Procurement Officer, Admission officer, PO SRC, System Engineer and Network Engineer

3.1.2 Learning Management System Use Case Diagram



3.1.3 User Management System Use Case Diagram



* Director, DY. Director, HODs, Student Advisors, Account officer, Examination, Department Head, Librarian, Finance Officer, Procurement Officer, Admission officer, PO SRC and Network Engineer

3.2 Use Case Description

Following are the narrative parts of every bubble in above use case diagrams [1]

Table 2: Use Case User Registration

Name	User Registration
Use-Case ID	BOL-UMS-1.00
Priority	High
Primary Actor	University personnel
Other participating Actor(s)	System
Description	This use case describes the event of a university personnel registering at BOL. User gives university ID and that ID validates from Bahria University database. If ID validated successfully user get registered at Bahria One Link.
Pre-condition	<ol style="list-style-type: none"> 1. User should be Bahria personnel 2. User should not register yet
Trigger	This use case initiate when a new personnel going to register
Typical flow of events	<ol style="list-style-type: none"> 1. User gives university identity and a new password 2. A confirmation e-mail sent to user with verification link 3. User click on verification link, it will active the user account and redirect user to the login screen
Alternate flow of event	Alt-1 user gives wrong university identity and an 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	BOL-UMS-2.00
Priority	High
Primary Actor	University personnel
Other participating Actor	System
Description	This use case describes the event when a user going to login at BOL. User fill out credential and given credentials validate from BOL database. If validation successful, then user logged in at Bahria One Link.
Pre-condition	<ol style="list-style-type: none"> 1. User must be registered at Bahria One Link 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 university identity and password 2. System verifies User Credentials
Alternate flow of events	Alt-1 User gives wrong credentials, system generate an error and take user to login page.
Post condition	User successfully logged in
Alternate post condition	Login failed

Table 4: Use Case User logout

Name	User logout
Use-Case ID	BOL-UMS-7.0.0
Priority	High
Primary Actor	University personnel
Other participating Actor	System
Description	This use case describes the event when a user going to logout. User select logout option to logout from Bahria One Link.
Pre-condition	<ol style="list-style-type: none"> 1. User must be registered at Bahria One Link 2. User should be logged in
Trigger	This use case initiate when a user going to logout.
Typical Flow of events	<ol style="list-style-type: none"> 1. User clicks on its profile from navigation bar 2. Click on logout from drop down menu.
Alternate flow of events	N/A
Post condition	User successfully logged out.

Alternate post condition	Still Login
--------------------------	-------------

Table 5: Use Case Reset User Password by admin

Name	Reset User Password by admin
Use-Case ID	BOL-UMS-3.0.0
Priority	Medium
Primary Actor	System Engineer
Other participating Actor(s)	-----
Description	This use case describes the event of a system engineer resetting any user's password. System engineer gives target user ID along with new password and confirm new password. New password will save in database.
Pre-condition	<ol style="list-style-type: none"> 1. System Engineer must be logged in 2. Target User must joined BOL earlier
Trigger	This use case initiate when a system engineer going to reset user password
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Reset Password 2. Enter target user id 3. Enter new password and enter confirm password 4. Click on update button
Alternate flow of event	N/A
Post condition	Password successfully updated
Alternate post condition	Password not updated

Table 6: Use Case Reset User Password by user

Name	Reset User Password by user
Use-Case ID	BOL-UMS-8.0.0
Priority	Medium
Primary Actor	User
Other participating Actor(s)	-----
Description	This use case describes the event of a user reset password. User gives old password along with new password and confirm new password. New password will save in database.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in
Trigger	This use case initiate when a user going to reset its password
Typical flow of events	<ol style="list-style-type: none"> 1. User clicks on its profile from navigation bar 2. Click on reset password from drop down menu. 3. Enter old password 4. Enter new password and enter confirm password 5. Click on reset password button
Alternate flow of event	N/A
Post condition	Password successfully updated

Alternate post condition	Password not updated
--------------------------	----------------------

Table 7: Use Case Edit Group Permissions

Name	Edit Group Permissions
Use-Case ID	BOL-UMS-4.0.0
Priority	Medium
Primary Actor	System Engineer
Other participating Actor(s)	-----
Description	This use case describes the event of a system engineer editing any group permissions. System admin edit permissions of a specific group from table and save new permissions.
Pre-condition	1. System Engineer must be logged in
Trigger	This use case initiate when a system engineer going to edit group permissions
Typical flow of events	<ol style="list-style-type: none"> 1. Click on Edit permissions 2. Edit target group permissions by checking/unchecking from permission table 3. Click on save button
Alternate flow of event	N/A
Post condition	Permissions successfully updated
Alternate post condition	Permissions not updated

Table 8: Use Case View/Search User

Name	View / Search user
Use-Case ID	BOL-UMS-5.0.0
Priority	Medium
Primary Actor	System Engineer
Other participating Actor(s)	-----
Description	This use case describes the event of a system engineer search / view any user. By giving target user ID, System admin will able to search user.
Pre-condition	<ol style="list-style-type: none"> 1. System Engineer must be logged in 2. Target User must joined BOL earlier
Trigger	This use case initiate when a system engineer going to search / view a user
Typical flow of events	<ol style="list-style-type: none"> 1. Click on View/Delete Users 2. Enter User id 3. View user
Alternate flow of event	Alt 4- Wrong user id entered and wrong/no result found
Post condition	User successfully searched
Alternate post condition	User not searched

Table 9: Use Case Active/Block User

Name	Block/ Active User
Use-Case ID	BOL-UMS-6.0.0
Priority	Medium
Primary Actor	System Engineer
Other participating Actor(s)	-----
Description	This use case describes the event of a system engineer block/active any user. System admin will able to block/active any user by giving it's ID.
Pre-condition	<ol style="list-style-type: none"> 1. System Engineer must be logged in 2. Target User must joined BOL earlier
Trigger	This use case initiate when a system engineer going to block/active a user
Typical flow of events	<ol style="list-style-type: none"> 1. Click on View/Delete Users 2. Enter User id 3. Click on block/active button
Alternate flow of event	Alt 4- Wrong user id entered and wrong result found
Post condition	User successfully deleted
Alternate post condition	User not deleted

Table 10: Use Case Create Virtual Class

Name	Create a virtual class
Use-Case ID	BOL-LMS-1.00
Priority	Medium
Primary Actor	Teacher
Other participating Actor	----
Description	This use case describes the event when a teacher is going to create a new class. Teacher will create a new virtual class by giving course code, class name and class description.
Pre-condition	<ol style="list-style-type: none"> 1. Teacher must be logged in
Trigger	This use case initiated when a teacher is going to create a new class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on create new class button 2. Enter class details 3. Click on save class button
Alternate flow of events	Alt-2 Teacher doesn't fill required detail and system generates an error message
Post condition	Class successfully created
Alternate post condition	Class not created

Table 11: Use Case Upload File in Virtual Class

Name	Uploading files in virtual class
Use-Case ID	BOL-LMS-1.10
Priority	Medium
Primary Actor	Teacher/Student
Other participating Actor	----
Description	This use case describes the event when a user going to upload a file in virtual class. user gives upload resource type, select a target file and click on upload button.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. File should be not exceeding the size of uploading limit
Trigger	This use case initiate when a user going to upload a file in virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Go to My Class and click on target class 2. Go to Resources 3. Click on upload resource button 4. Browse and select the file 5. Select type of document (e.g. book, lectures etc.) 6. Click on upload
Alternate flow of events	Alt-2 User doesn't select any file and click on upload button, an error occurred and displays an error message.
Post condition	File successfully uploaded
Alternate post condition	Not uploaded

Table 12: Use Case Download File From Virtual Class

Name	Download / View resource from Virtual Class
Use-Case ID	BOL-LMS-1.2.0
Priority	Medium
Primary Actor	Student/Teacher
Other participating Actor(s)	-----
Description	This use case describes the event of a user downloading/viewing any resource from virtual class. User select target file from virtual class resource and download it.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Target class should be created earlier 3. Target class should be joined earlier 4. Target Resource must uploaded earlier
Trigger	This use case initiate when a user going to download/view any resource from a virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on My Class 2. Click on target class 3. Go to Resources 4. Click on target resource
Alternate flow of event	Alt 4- User click on wrong Resource
Post condition	Student successfully Downloaded/viewed class resource

Alternate post condition	Student does not successfully Downloaded/viewed class resource
--------------------------	--

Table 13: Use Case Open Virtual Class

Name	Open Virtual Class
Use-Case ID	BOL-LMS-1.3.0
Priority	Medium
Primary Actor	Student/Teacher
Other participating Actor(s)	-----
Description	This use case describes the event of a user opening a virtual class. User select a target class and open it.
Pre-condition	<ol style="list-style-type: none"> 1. user must be logged in 2. Target class should be created earlier 3. Target class should be joined earlier
Trigger	This use case initiate when a user going to open a virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on My Class 2. Click on target class
Alternate flow of event	N/A
Post condition	Student is successfully opened class
Alternate post condition	Student is not successfully opened class

Table 14: Use Case Join Virtual Class

Name	Join Virtual Class
Use-Case ID	BOL-LMS-1.4.0
Priority	Medium
Primary Actor	Student
Other participating Actor(s)	-----
Description	This use case describes the event of a student joining a virtual class. Student gives access code and search the class and join the target class.
Pre-condition	<ol style="list-style-type: none"> 1. Student must be logged in 2. Target class should not be joined 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 My Class 2. Type the access code in search field and press search button 3. Click on join button
Alternate flow of event	Alt-3 Student gives wrong access code and 'No Result Found ' message displays or wrong class displays
Post condition	Student is successfully joined class
Alternate post condition	Student is not successfully joined class

Table 15: Use Case Make Announcement in Virtual Class

Name	Make an Announcement in Virtual Class
Use-Case ID	BOL-LMS-1.5.0
Priority	Medium
Primary Actor	Teacher
Other participating Actor(s)	-----
Description	This use case describes the event of a Teacher uploading any announcement in virtual class.
Pre-condition	<ol style="list-style-type: none"> 1. Teacher must be logged in 2. Target class should be created earlier
Trigger	This use case initiate when a user going to view any announcement from a virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on My Class 2. Click on target class 3. Go to Announcement 4. Click on 'make announcement' 5. Type Announcement and press upload button
Alternate flow of event	Alt 5- Teacher leaves announcement field blank and system generate error message
Post condition	Teacher successfully upload announcement
Alternate post condition	Teacher does not successfully upload announcement

Table 16: Use Case View Announcement From Virtual Class

Name	View Announcement from Virtual Class
Use-Case ID	BOL-LMS-1.6.0
Priority	Medium
Primary Actor	Student
Other participating Actor(s)	-----
Description	This use case describes the event of a user viewing any announcement from virtual class. User select announcement from virtual class.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Target class should be created earlier 3. Target class should be joined earlier 4. Target announcement must uploaded earlier
Trigger	This use case initiate when a user going to view any announcement from a virtual class.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on My Class 2. Click on target class 3. Go to Announcement 4. View announcement
Alternate flow of event	N/A
Post condition	Student successfully viewed announcement

Alternate post condition	Student does not successfully viewed announcement
--------------------------	---

Table 17: Use Case Alumni Searching

Name	Alumni searching
Use-Case ID	BOL-LMS-2.00
Priority	Low
Primary Actor	University personnel
Other participating Actor	----
Description	This use case describes the event when a university personnel going to search any alumni. User search the target alumni by name, enrollment, department, degree level and batch year.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Target student must be graduated/alumni
Trigger	This use case initiated when any user going to search for alumni.
Typical flow of events	<ol style="list-style-type: none"> 1. User click on alumni from side bar 2. Enter alumni detail 3. Click on search button 4. Record will be displayed 5. Click on view profile link to view the alumni profile
Alternate flow of events	Alt-4 User doesn't give complete details and system will show a popup error message.
Post condition	Searching complete successfully and record found
Alternate post condition	Record not found

Table 18: Use Case Alumni Profile Open

Name	Alumni Profile Open
Use-Case ID	BOL-LMS-6.0.0
Priority	Low
Primary Actor	alumni
Other participating Actor	----
Description	This use case describes the event when an alumni is going to open his/her profile.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. User must be graduated/alumni
Trigger	This use case initiated when an alumni going to open his/her profile.
Typical flow of events	<ol style="list-style-type: none"> 1. User click on profile from side bar 2. Alumni profile will be displayed
Alternate flow of events	
Post condition	Successfully open profile.

Alternate post condition	Not open
--------------------------	----------

Table 19: Use Case Alumni Update Profile

Name	Alumni Update Profile
Use-Case ID	BOL-LMS-6.1.0
Priority	Low
Primary Actor	alumni
Other participating Actor	----
Description	This use case describes the event when an alumni is going to update its profile information or profile picture.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. User must be graduated/alumni
Trigger	This use case initiated when an alumni going to edit his/her profile.
Typical flow of events	<ol style="list-style-type: none"> 1. User click on profile from side bar 2. Alumni profile will be displayed 3. Go to update profile tab 4. In personal info tab under update profile tab update the editable input fields and click on update button at bottom 5. In profile picture tab under update profile tab choose the updated profile picture and click on update picture
Alternate flow of events	Alt-4 wrong information entered by user Alt-5 Invalid picture format or picture size exceeds from limited size
Post condition	Successfully open profile.
Alternate post condition	Not open

Table 20: Use Case Upload File in Library

Table 21: Use Case Active/Dismiss New Library Content

Name	Active/Dismiss New Library Content
Use-Case ID	BOL-LMS-4.1.0
Priority	Medium
Primary Actor	Librarian
Other participating Actor(s)	-----
Description	This use case describes the event of a librarian approve or disapprove a new uploaded content in library
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. Target Resource must uploaded earlier
Trigger	This use case initiate when a librarian going to approve/disapprove any resource from a library.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on library 2. Under the section new content click on download link to view the new uploaded resource 3. Click on active button to approve the resource or click on dismiss button to disapprove the resource
Alternate flow of event	

Name	Uploading files in library
Use-Case ID	BOL-LMS-4.00
Priority	Medium
Primary Actor	Teacher/Student
Other participating Actor	----
Description	This use case describe the event when a user going to upload a file in Library.
Pre-condition	<ol style="list-style-type: none"> 1. User must be logged in 2. File should be not exceeding the size of uploading limit
Trigger	This use case initiate when a user going to upload a file in library.
Typical flow of events	<ol style="list-style-type: none"> 1. Go to Library 2. Click on upload resource button 3. Enter File details 4. Browse and select the file 5. Click on upload
Alternate flow of events	Alt-2 User doesn't select any file or wrong format and click on upload button, an error occurred and displays an error message.
Post condition	File successfully uploaded
Alternate post condition	Not uploaded
Post condition	librarian successfully approve/disapprove resource
Alternate post condition	librarian does not successfully approve/disapprove resource

Table 22: Use Case Download File From Library

Name	Download Resource from Library
Use-Case ID	BOL-LMS-5.0.0
Priority	Medium
Primary Actor	Student/Teacher
Other participating Actor(s)	-----
Description	This use case describes the event of a user downloading any resource from library.
Pre-condition	<ol style="list-style-type: none"> 3. User must be logged in 4. Target Resource must uploaded earlier
Trigger	This use case initiate when a user going to download any resource from a library.
Typical flow of events	<ol style="list-style-type: none"> 4. Click on library 5. Select the resource type tab (e.g. books, lectures) under the library content section 6. Click on download link
Alternate flow of event	Alt 5- User click on wrong Resource
Post condition	User successfully Downloaded/viewed resource
Alternate post condition	User does not successfully Downloaded/viewed class resource

Table 23: Use Case Create Minute

Name	Create new minute
Use-Case ID	BOL-CS-1.00
Priority	High
Primary Actor	University personnel (except student) – who initiate the minute
Other participating Actor(s)	University personnel (except student) – to whom the minute is forwarded
Description	This use case describes the event of a university personnel creating a new minute. After filling all the particular minute send to its recipient(s).
Pre-condition	1. User should be logged in
Trigger	This use case initiated when a new minute is created.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on “Create Minute” 2. Choose minute type 3. Add targeted recipient(s) 4. Write minute in the editor 5. Add flags if required 6. Fill the Performa if given 7. Click send button
Alternate flow of events	Alt-4. User doesn’t fill all required fields and error displays
Post condition	Minute successfully send
Alternate post condition	Minute is not send

Table 24: Use Case Save Minute

Name	Save Minute
Use-Case ID	BOL-CS-1.1.0
Priority	High
Primary Actor	University personnel (except student)
Other participating Actor(s)	University personnel (except student)
Description	This use case describes the event of a university personnel going to save a minute.
Pre-condition	1. User should be logged in
Trigger	This use case initiated when any minute going to save.
Typical flow of events	<ol style="list-style-type: none"> 1. Click on “Create Minute” 2. Choose minute type 3. Write minute in the editor 4. Click save button
Alternate flow of events	Alt-4. User doesn’t write minute in editor fields and error displays
Post condition	Minute successfully save
Alternate post condition	Minute is not save

Table 25: Use Case Feedback of Minute

Name	Feedback of a minute
Use-Case ID	BOL-CS-1.2.0
Priority	High
Primary Actor	University personnel (except student)
Other participating Actor(s)	University personnel (except student) related to that minute
Description	This use case describes the event of a university personnel giving feedback on a minute.
Pre-condition	1. Must receive or sent any minute
Trigger	This use case triggers when user gives feedback.
Typical flow of events	<ol style="list-style-type: none"> 1. Open received or sent minute 2. Write feedback in given space 3. Add flags if required 4. Click on post button
Alternate flow of events	N/A
Post condition	Feedback Posted
Alternate post condition	Minute is not posted

Table 26: Use Case Throw back a minute

Name	Throw back a minute
Use-Case ID	BOL-CS-1.3.0
Priority	High
Primary Actor	University personnel (except student)
Other participating Actor(s)	University personnel (except student) related to that minute
Description	This use case describes the event of a university personnel going to throw back a minute.
Pre-condition	1. Must receive any minute
Trigger	This use case triggers when user throw back a minute
Typical flow of events	<ol style="list-style-type: none"> 1. Open received minute 2. Click on throw back button
Alternate flow of events	N/A
Post condition	Minute deleted from inbox
Alternate post condition	Minute not deleted from inbox due to database error

Table 27: Use Case Print Minute

Name	Print minute
Use-Case ID	BOL-CS-2.0.0
Priority	Medium
Primary Actor	University personnel (except student)
Other participating Actor(s)	N/A
Description	This use case describes the event of a university personnel printing a stored minute. User select target mint, open it and print it.
Pre-condition	<ol style="list-style-type: none"> 1. User should be logged in 2. Required minute should be available
Trigger	This use case initiated when a minute is going to print.
Typical flow of events	<ol style="list-style-type: none"> 1. Open required minute 2. Click on print minute button if Performa is attached with minute also click on print Performa button
Alternate flow of events	N/A
Post condition	Minute successfully printed.
Alternate post condition	Minute is not printed.

Table 28: Use Case View Minute

Name	View minute
Use-Case ID	BOL-CS-3.0.0
Priority	high
Primary Actor	University personnel (except student)
Other participating Actor(s)	N/A
Description	This use case describes the event of a university personnel viewing a minute.
Pre-condition	<ol style="list-style-type: none"> 1. User should be logged in 2. Required minute should be available
Trigger	This use case initiated when a minute is open.
Typical flow of events	<ol style="list-style-type: none"> 1. Go to inbox or sent 2. Click on required minute
Alternate flow of events	N/A
Post condition	Minute successfully open
Alternate post condition	Minute is not open

3.3 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system. There exists sequence diagram against every use case. Following are the sequence diagrams of Bahria One Link (BOL).

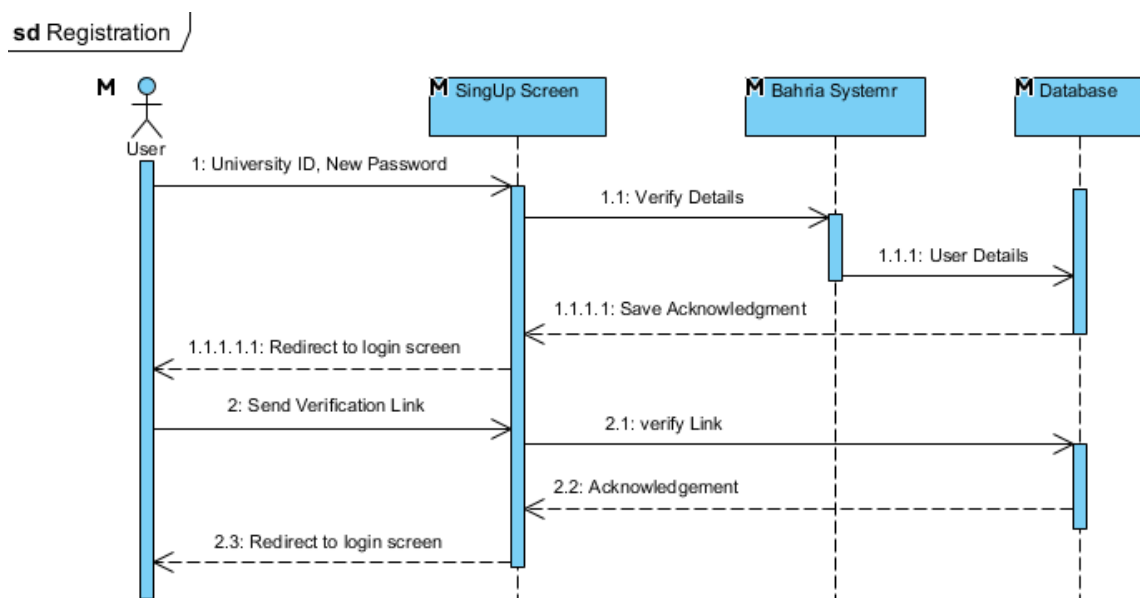


Figure 2: Sequence Diagram Registration

Above diagram describe the sequence of a user registration process in which user enter university id and password to signup screen. Entered university id validate from Bahria university database and verification link sent to user email. At end user clicks on verification link and redirect to the login screen.

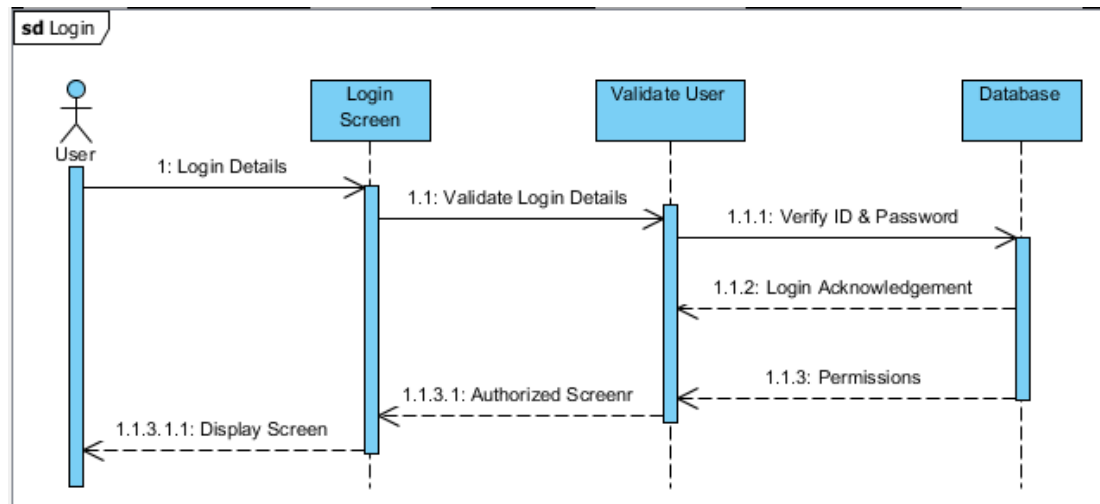


Figure 3: Sequence Diagram Login

Above diagram describes the sequence of a user login process in which user enters the login credentials on login screen. Those details will validate from the database and an acknowledgement sent from the database. At end, user redirect to the authorized page.

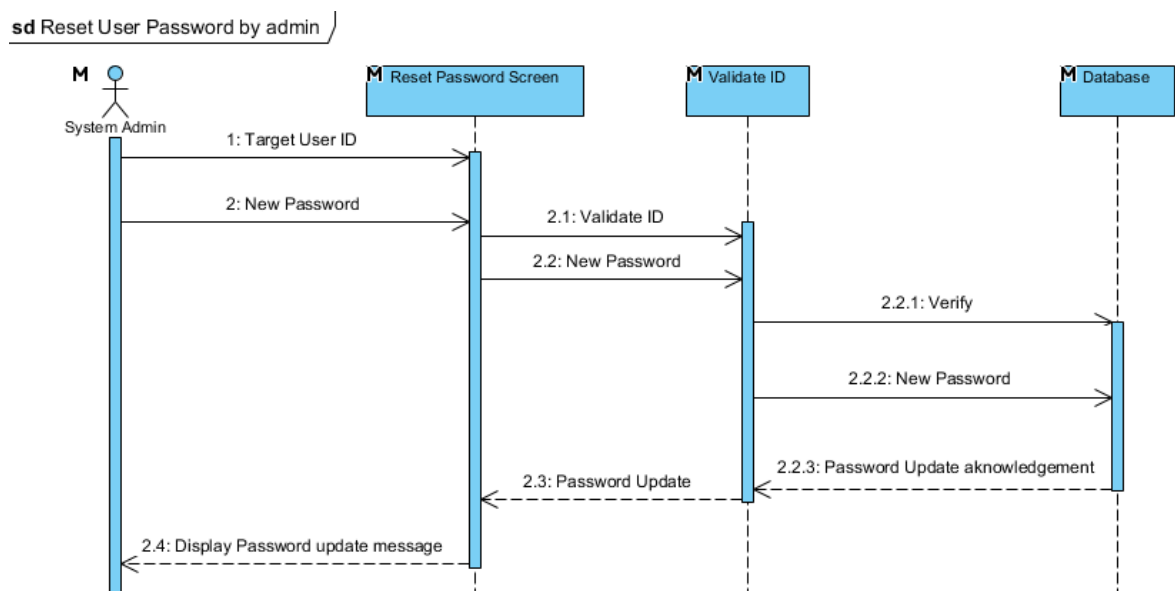


Figure 4: Sequence Diagram Reset User Password by Admin

Above diagram describe the sequence of resetting user password by admin in which admin enters user id and a new password. Entered user id verified from the database and an acknowledgement sent which displays password reset message.

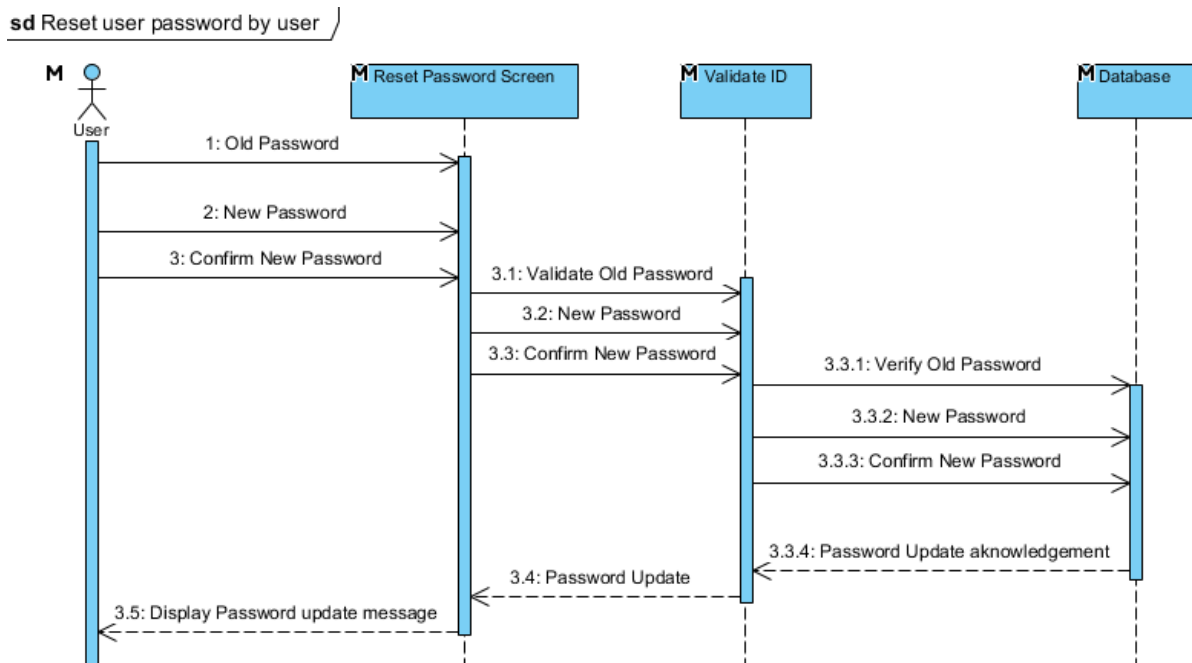


Figure 5: Sequence Diagram Reset User Password by User

Above diagram describe the sequence of resetting user password by user in which user enters old password and a new password. Entered old password verified from the database and an acknowledgement sent which displays password reset message.

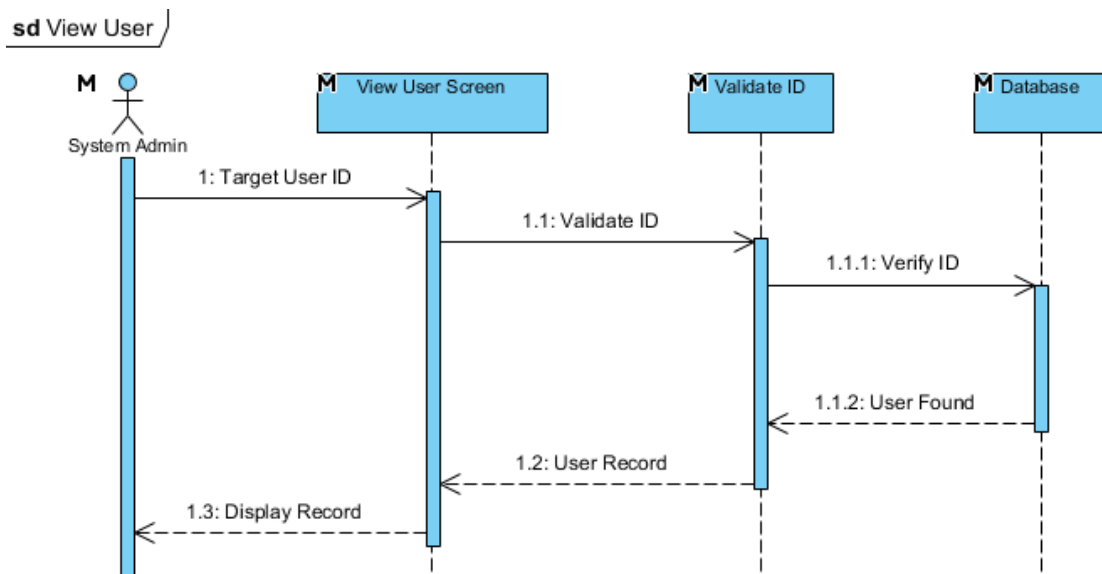


Figure 6: Sequence Diagram View User

Above diagram describe the sequence of view user by admin in which admin enters user id. Entered user id verified from the database and an acknowledgement sent which displays user record.

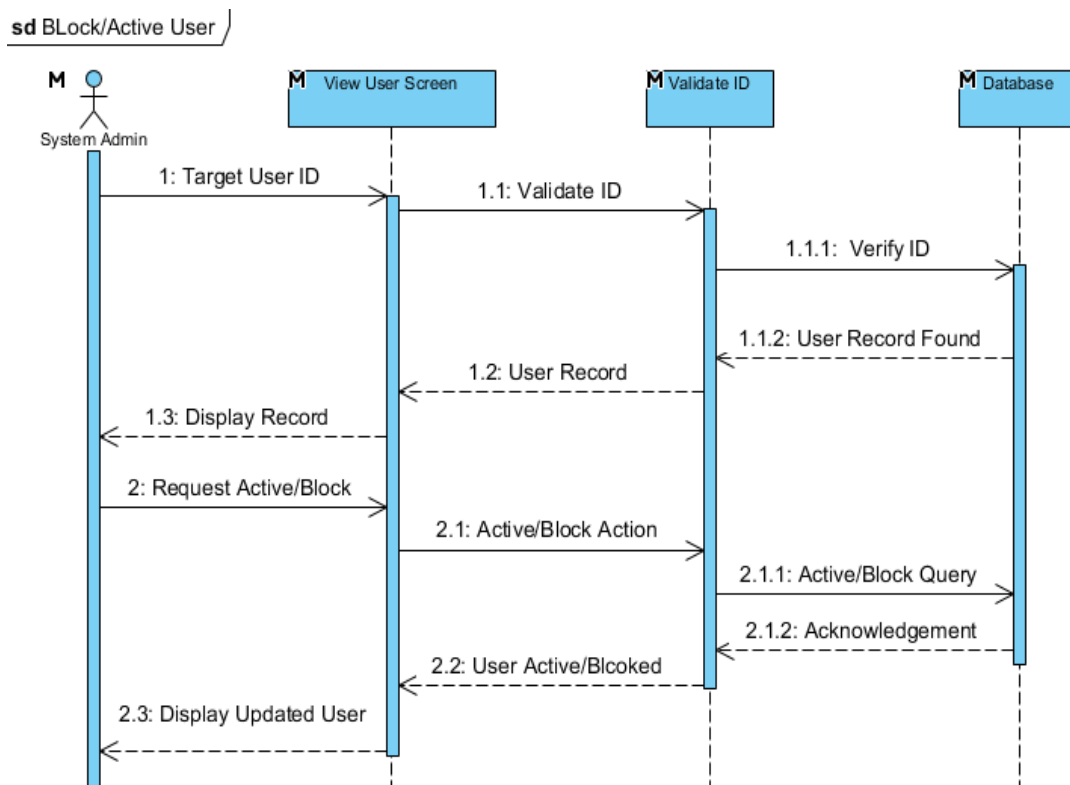


Figure 7: Sequence Diagram Active/Block User

Above diagram describe the sequence of Active/Block user by admin in which admin enters user id. Entered user id verified from the database and an acknowledgement sent which displays user record and admin click on Active/Block button. At end database sent acknowledgement which displays user Active/Block message.

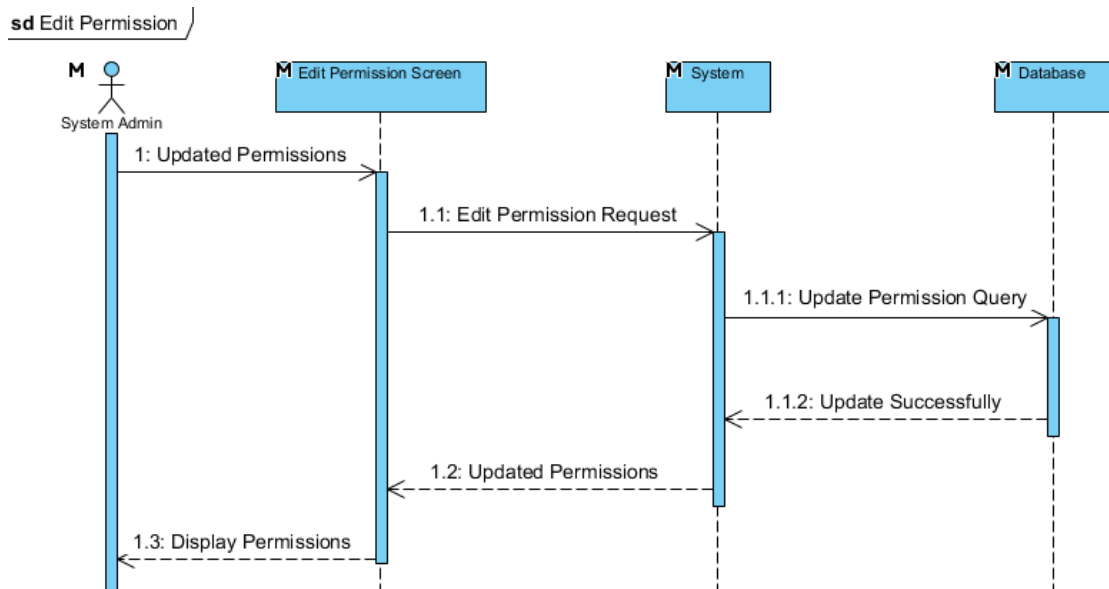


Figure 8: Sequence Diagram Edit Permissions

Above diagram describe the sequence of edit group permission by admin in which admin Check/Uncheck update permissions. At end database sent acknowledgement which displays the updated permissions.

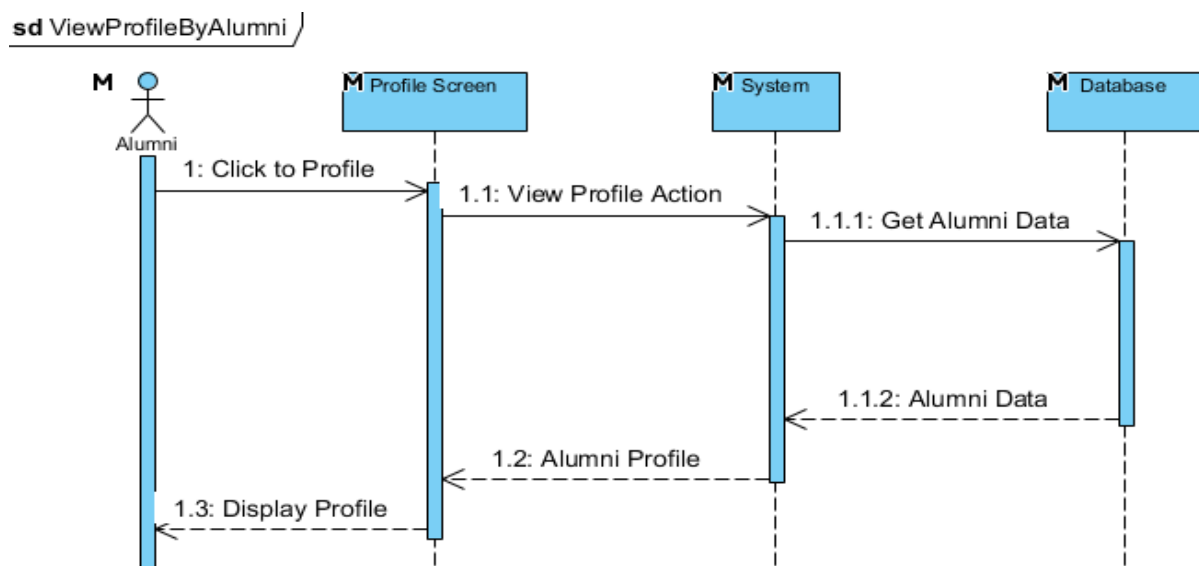


Figure 9: Sequence Diagram View Profile By Alumni

Above diagram describe the sequence of view a profile by alumni in which alumni click on profile and profile displayed to the alumni.

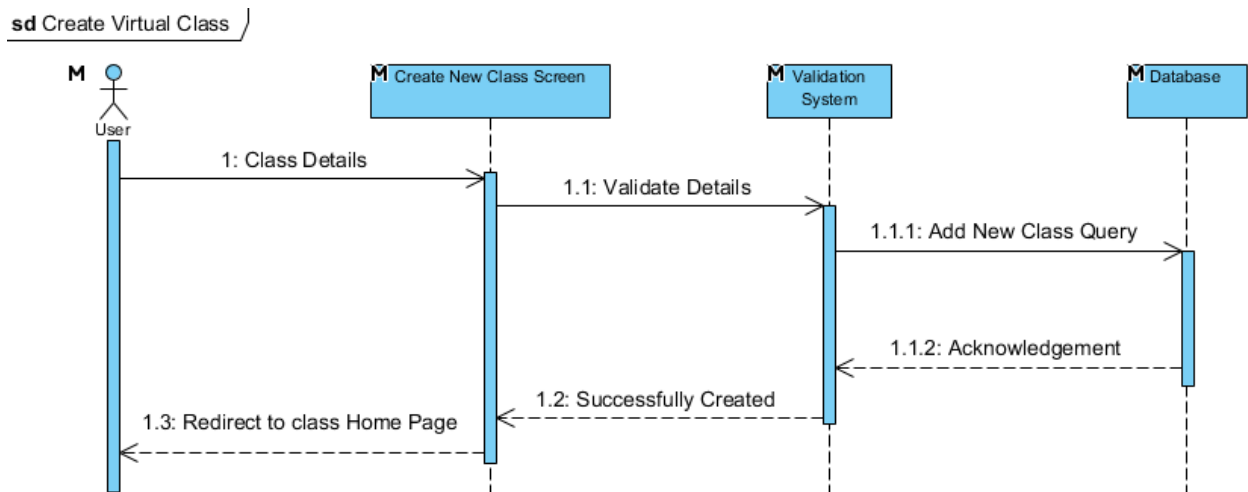


Figure 10: Sequence Diagram Create Virtual Class

Above diagram describe the sequence of create virtual class in which user enters class details. Entered details validates according to the format and virtual class details save in database. At end database sent acknowledgement which displays the class creation message and redirect to new created class.

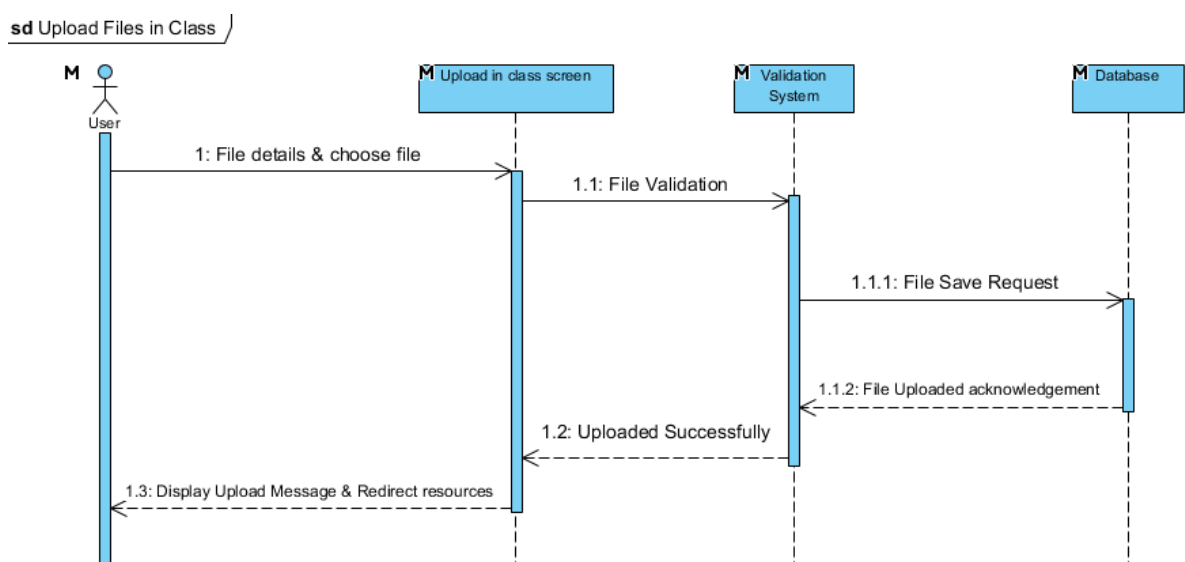


Figure 11: Sequence Diagram Upload Files in Class

Above diagram describe the sequence of upload file in virtual class in which user selects a file. Selected file validates w.r.t it's file format. At end database sent acknowledgement which displays the upload file message and redirect to class resource.

sd Download Resource from Class

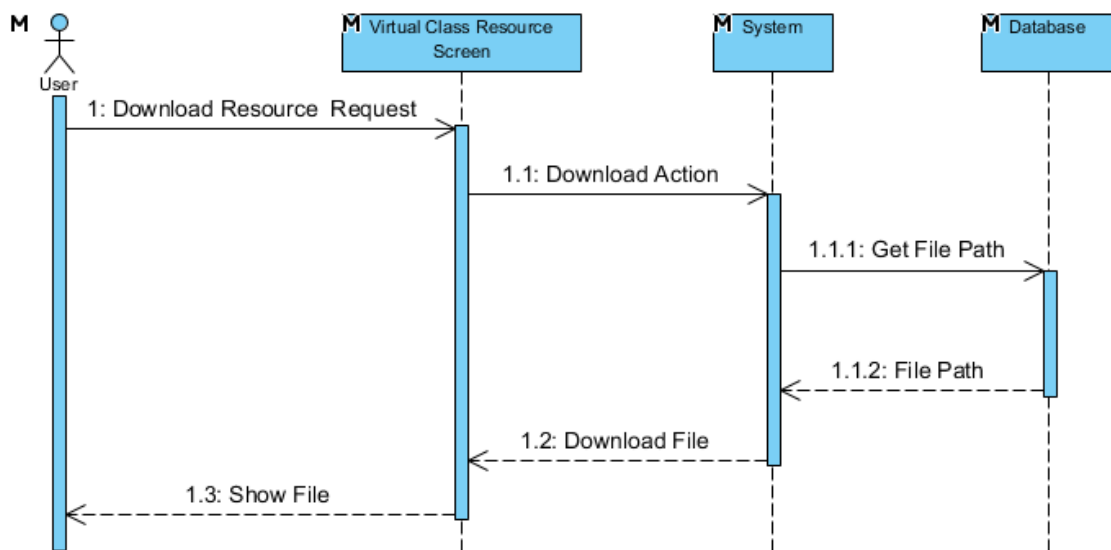


Figure 12: Sequence Diagram Download Resource from Class

Above diagram describe the sequence of download resource from virtual class in which user selects file from virtual class resource and click on download link. At end database provides download path to system and system download the file.

sd Open Virtual Class

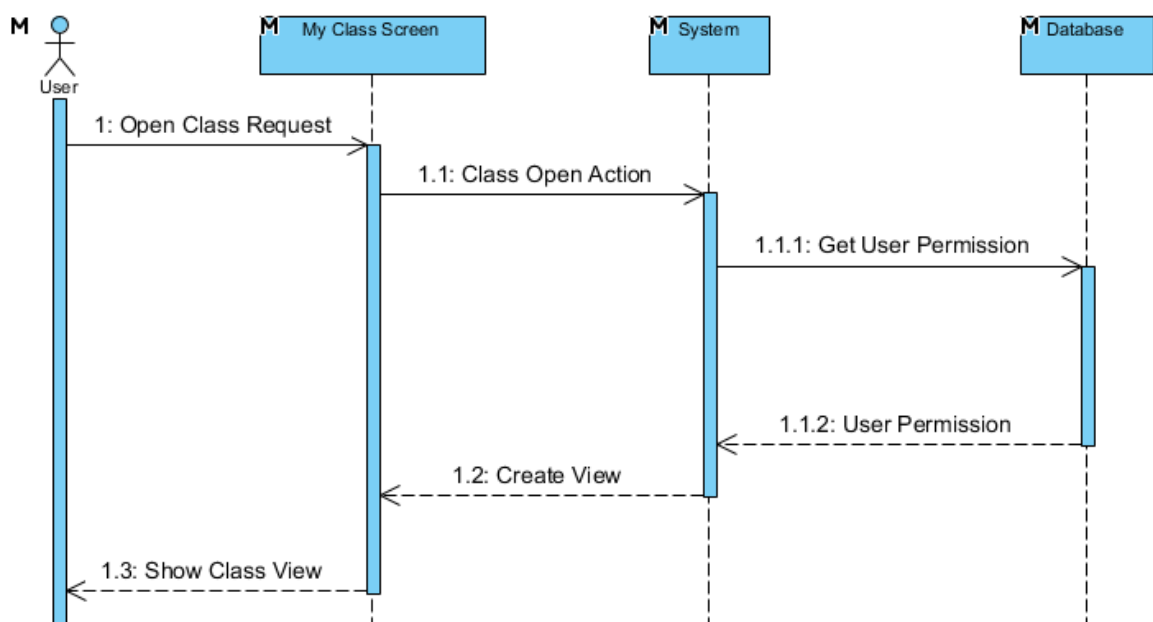


Figure 13: Sequence Diagram Open Virtual Class

Above diagram describe the sequence of open virtual class in which user select class. System get data from database. At end acknowledgement sent to the user and displays profile page.

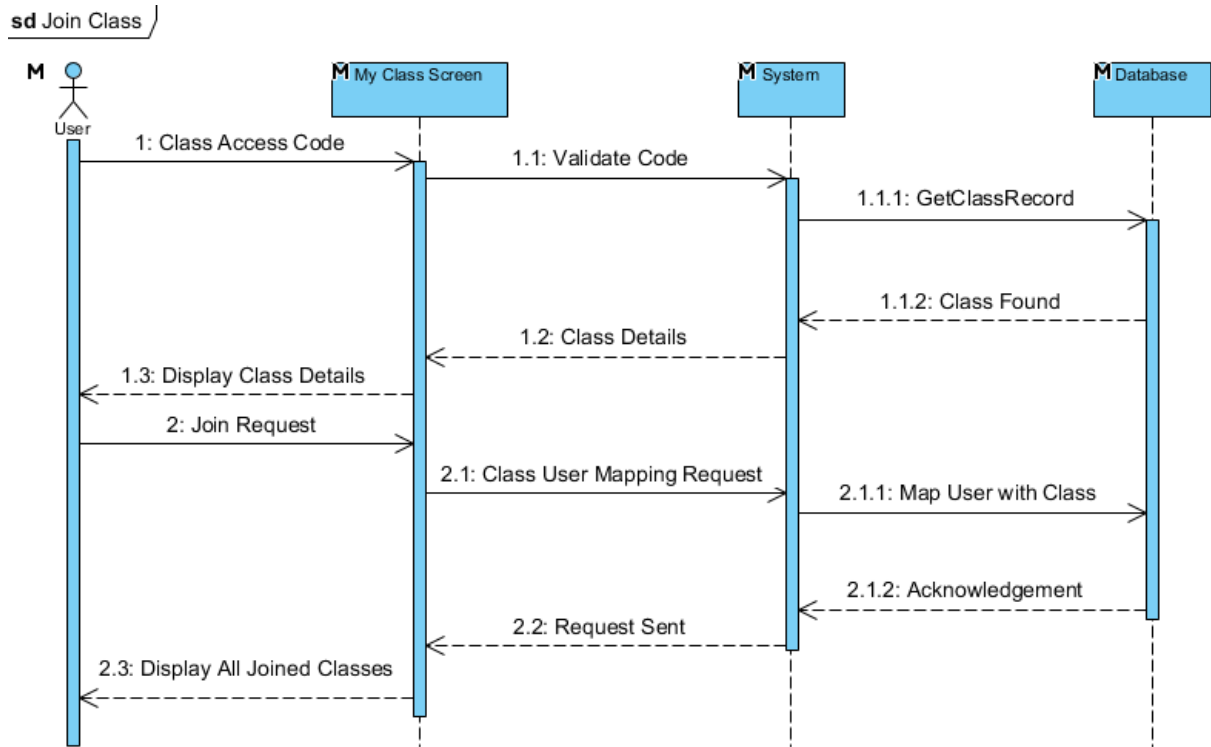


Figure 14: Sequence Diagram Join Class

Above diagram describe the sequence of join virtual class in which user enters course code which validates from database and class name sent to the user. User click on join class link and user maps with the class in database.

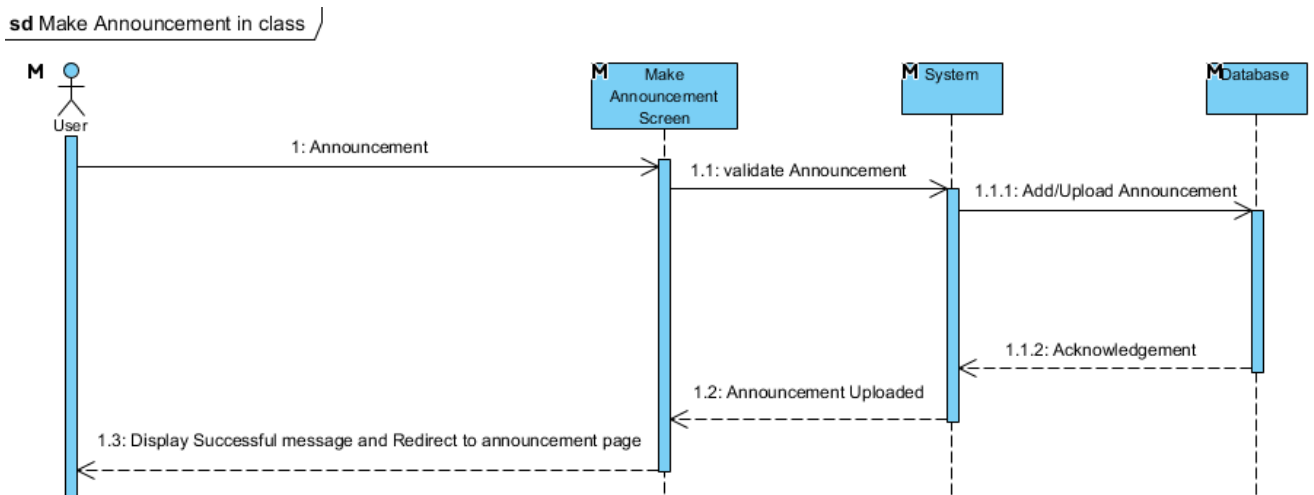


Figure 15: Sequence Diagram Make Announcement in class

Above diagram describe the sequence of make announcement in virtual class in which user make new announcement and write the announcement. At end announcement save in database and acknowledgement sent to the user and redirect to the announcement page.

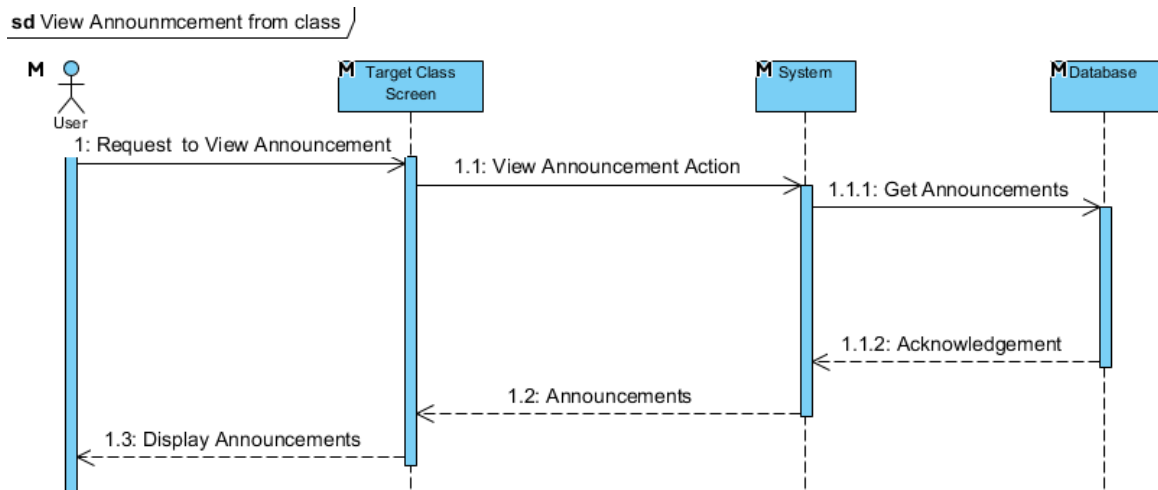


Figure 16: Sequence Diagram View Announcement from Class

Above diagram describe the sequence of view announcement in virtual class in which user click on. At end acknowledgement sent to the user and displays announcement page.

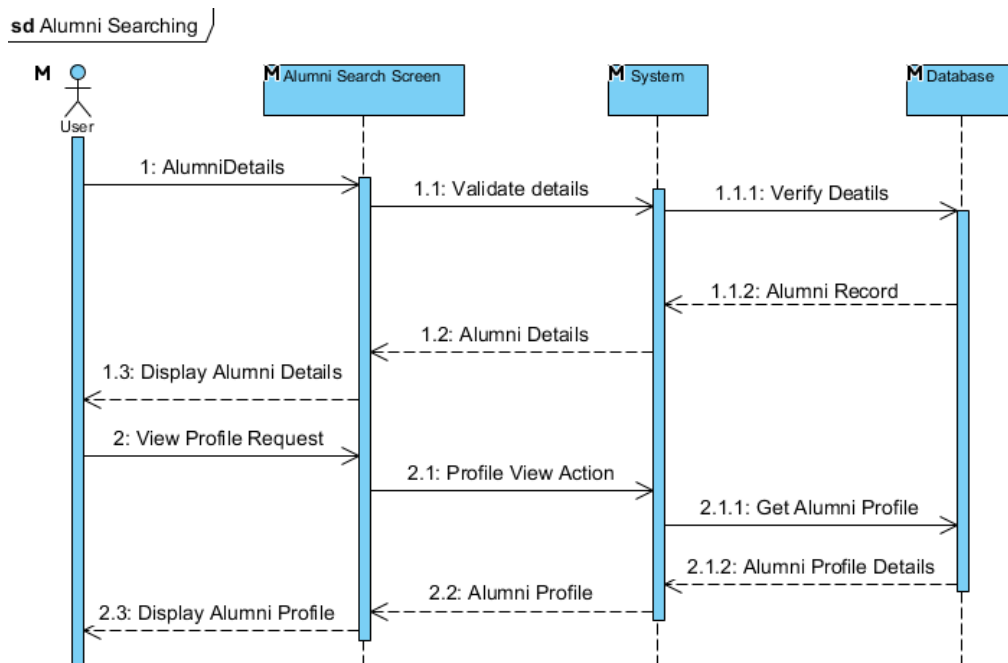


Figure 17: Sequence Diagram Alumni Search

Above diagram describe the sequence of alumni searching in which user enters alumni details which validates from database. At end target alumni profile displays to the user.

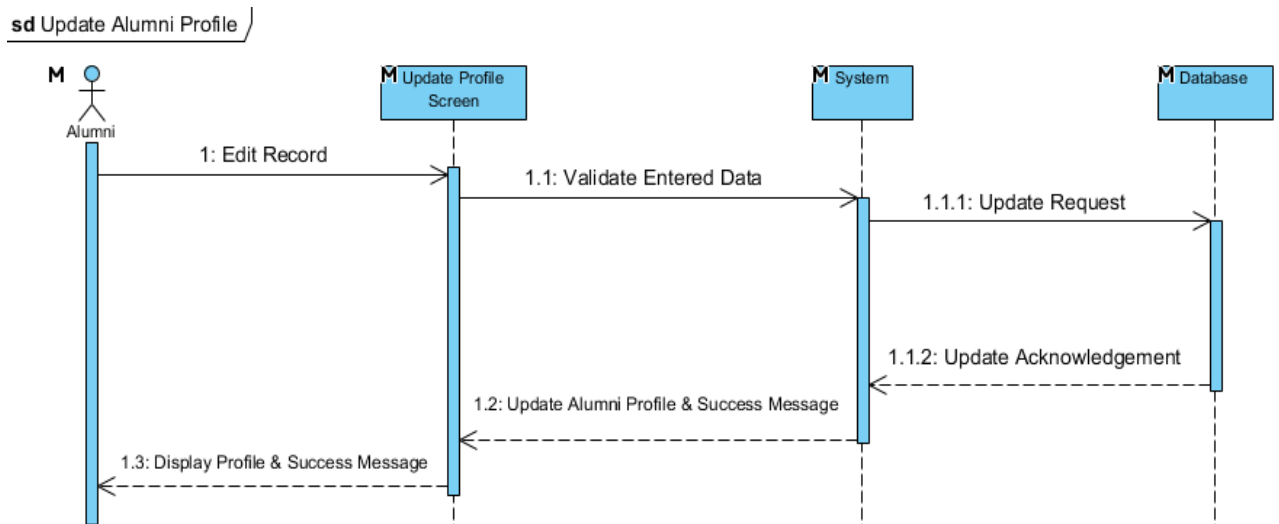


Figure 18: Sequence Diagram Update Alumni Profile

Above diagram describe the sequence of update alumni profile in which user enters updated information. At end updated information save in database and acknowledgement sent to the user and redirect to profile page.

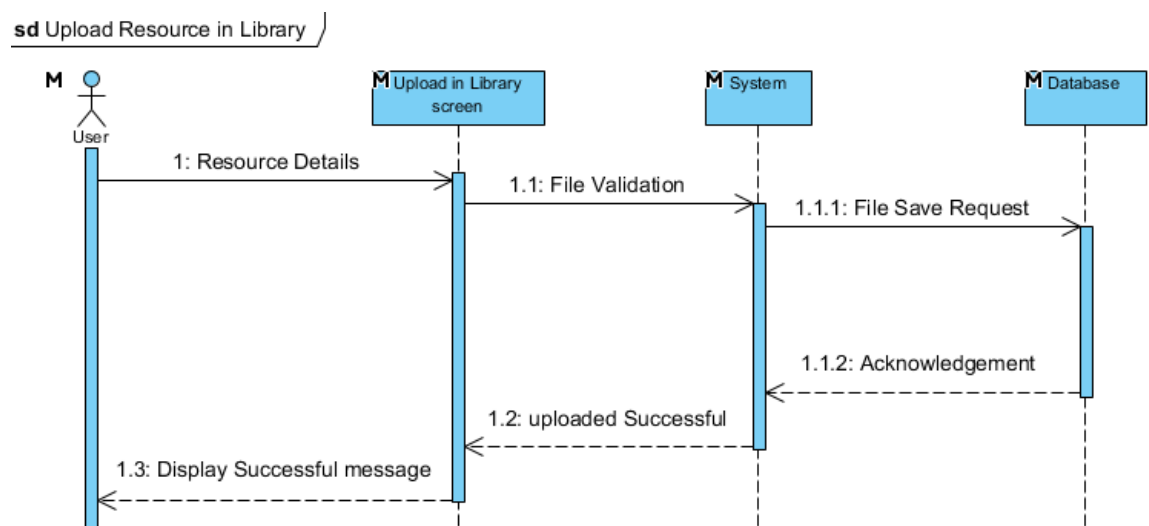


Figure 19: Sequence Diagram Upload Resource in Library

Above diagram describe the sequence of upload file in library in which user selects a file. Selected file validates w.r.t it's file format. At end database sent acknowledgement which displays the upload file message and redirect to library.

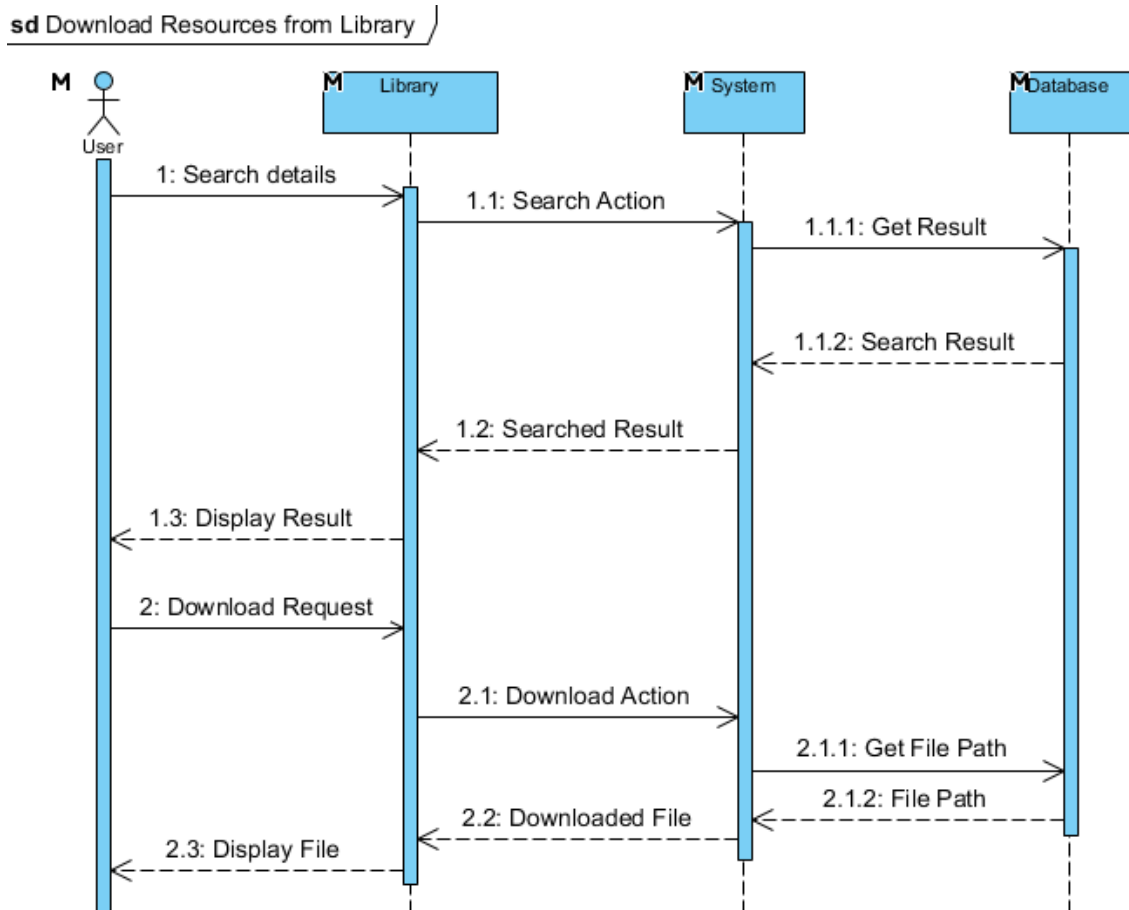


Figure 20: Sequence Diagram Download Resource from Library

Above diagram describe the sequence of download resource from library in which user selects file from library resource and click on download link. At end database provides download path to system and system download the file.

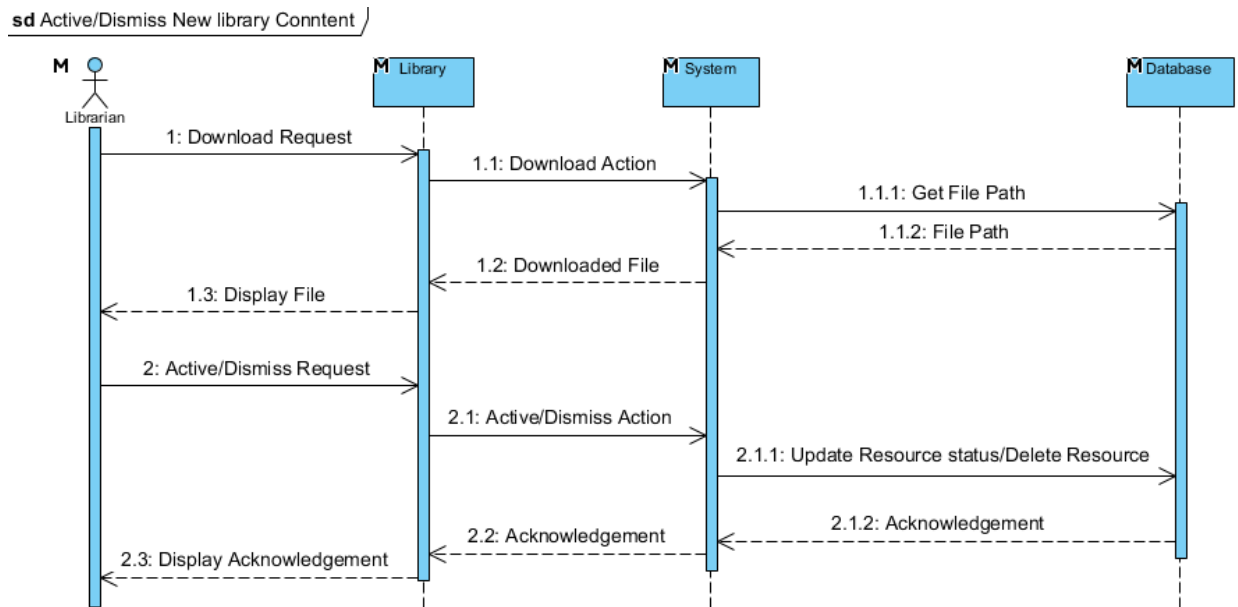


Figure 21: Sequence Diagram Active/Dismiss New Content From Library

Above diagram describe the sequence of Active/Dismiss new content from library in which librarian click on Active/Dismiss button from New Library Content. At end file save/delete in or from and acknowledgement sent to the user.

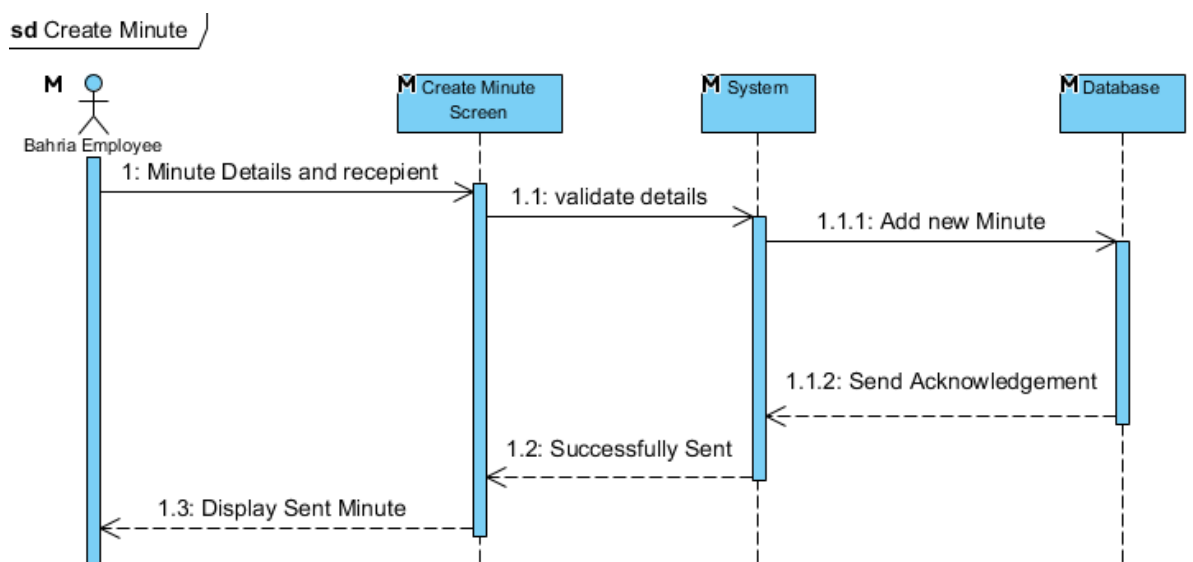


Figure 22: Create a Minute

Above diagram describe the sequence of creating a minute in which user enters minute details and recipients. Entered details validates. At the end minute save in database and acknowledgement sent to user.

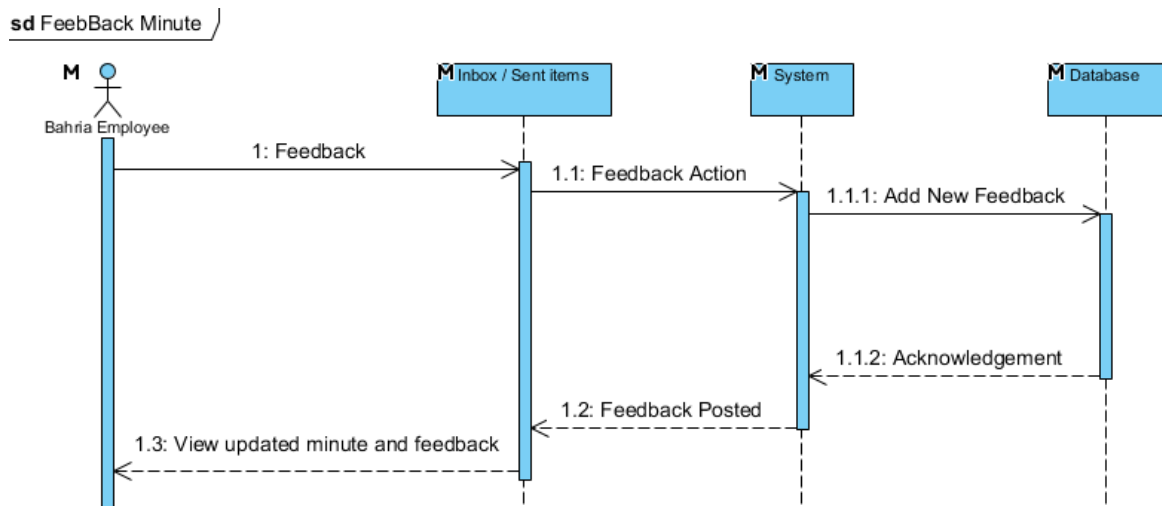


Figure 23: Sequence Diagram Feedback a Minute

Above diagram describe the sequence of feedback of a minute in which user enters minute feedback. At the end minute updated in database and acknowledgement sent to user.

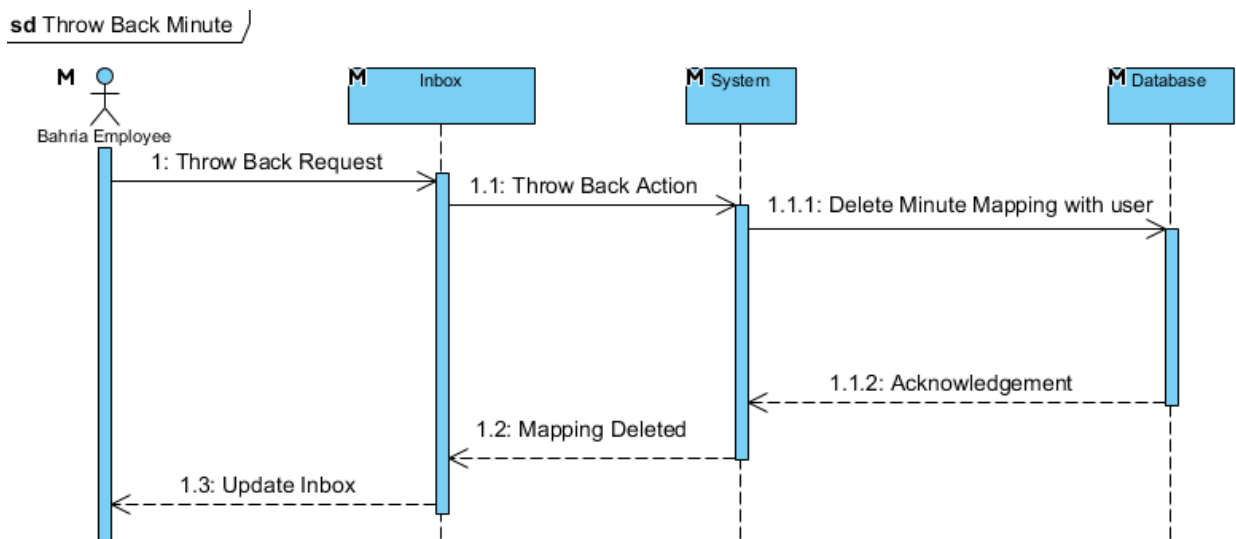


Figure 24: Sequence Diagram Throw back a Minute

Above diagram describe the sequence of throwback a minute in which user clicks on throwback button. At the end minute sent back to the initiator and delete from database for current user and acknowledgement sent to user.

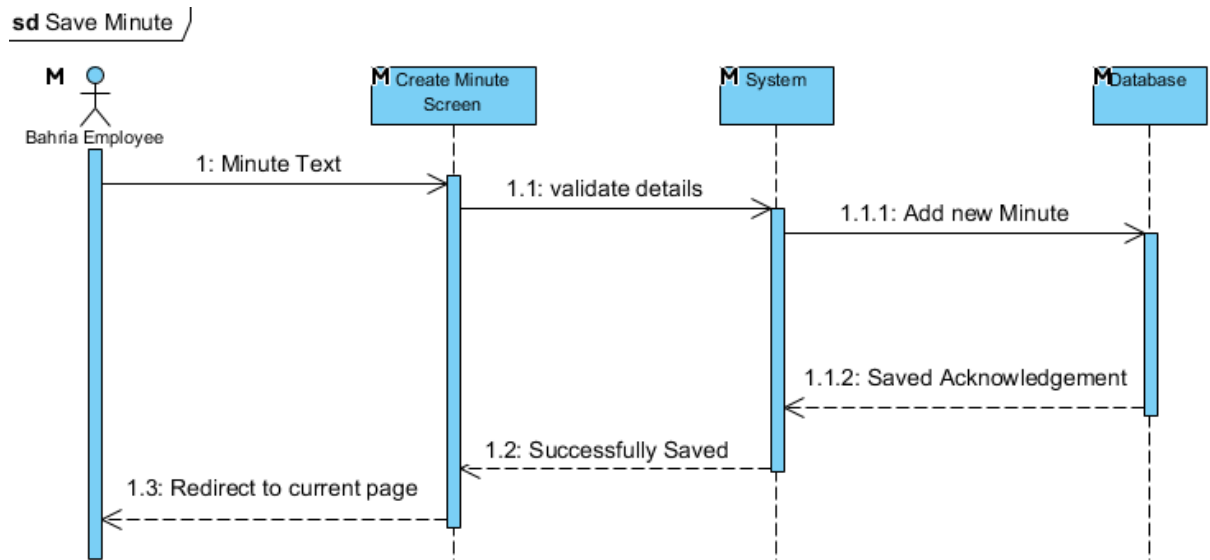


Figure 25: Sequence Diagram Save a Minute

Above diagram describe the sequence of save a minute in which user enters minute and clicks on save button. At the end minute save in database and acknowledgement sent to user.

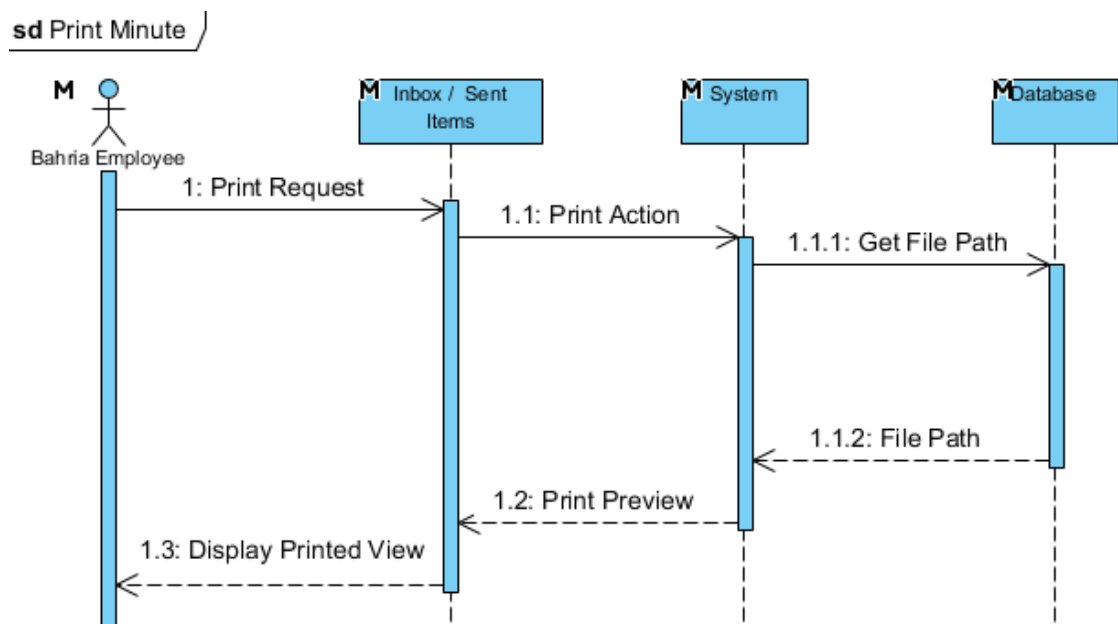


Figure 26: Sequence Diagram Print a Minute

Above diagram describe the sequence of print a minute in which user open a minute and give print command. System get the minute from database and print it

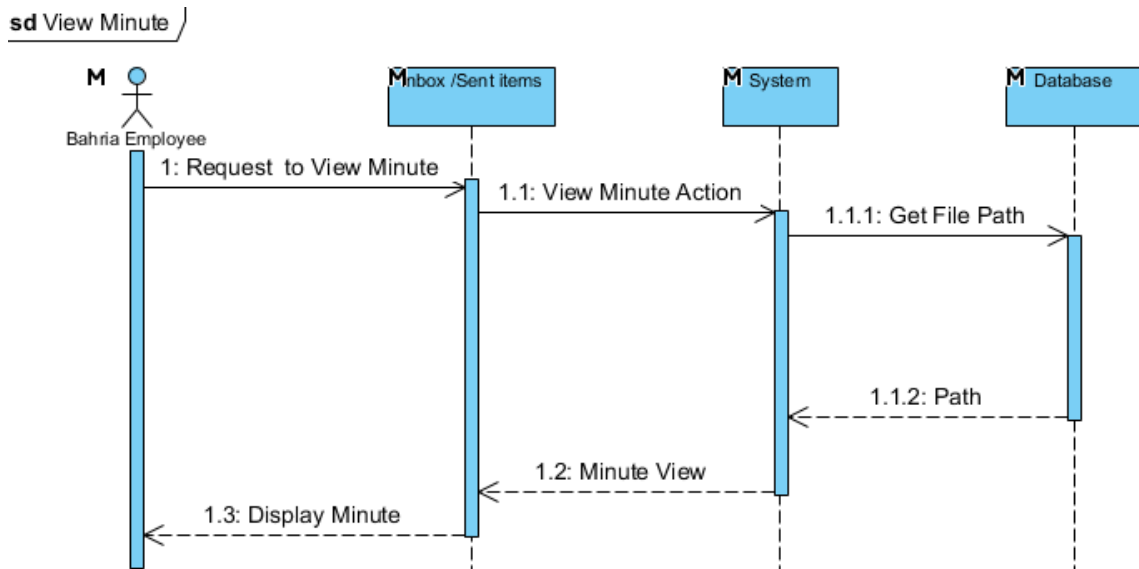


Figure 27: Sequence Diagram View a Minute

Above diagram describe the sequence of view a minute in which user click on a minute. System get the minute from database and view it.

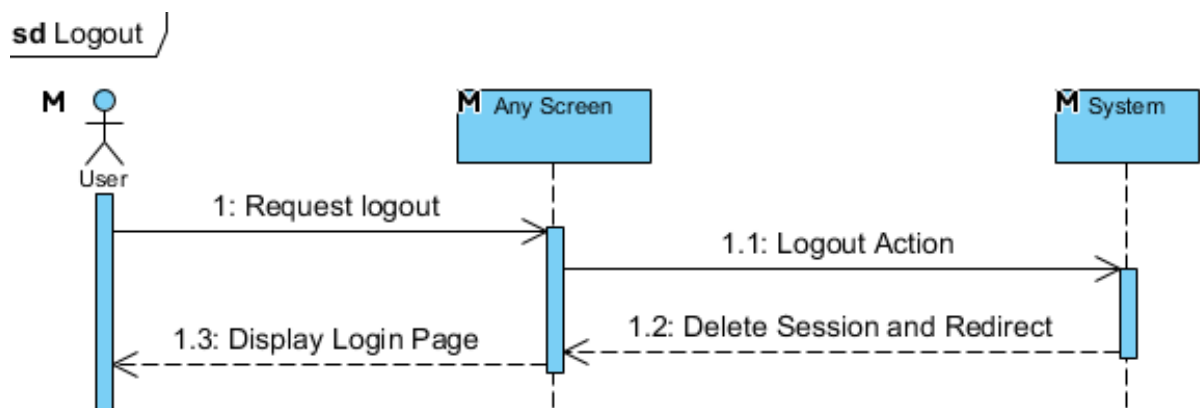


Figure 28: Sequence Diagram Logout

Above diagram describe the sequence of logout of a user in which user clicks on logout option. System delete user session and redirect to login page.

3.4 Collaboration Diagram

UML collaboration/communication diagrams like UML sequence diagrams, are used to explore the dynamic nature of your software. Collaboration diagrams show the message flow between objects in an OO application, and also imply the basic associations (relationships) between classes. The UML Collaboration diagram is used to model how objects involved in a scenario interact, with each object instantiating a particular class in the system. Objects are connected by links, each link representing an instance of an association between the respective classes involved. There exists collaboration diagram against every sequence diagram.

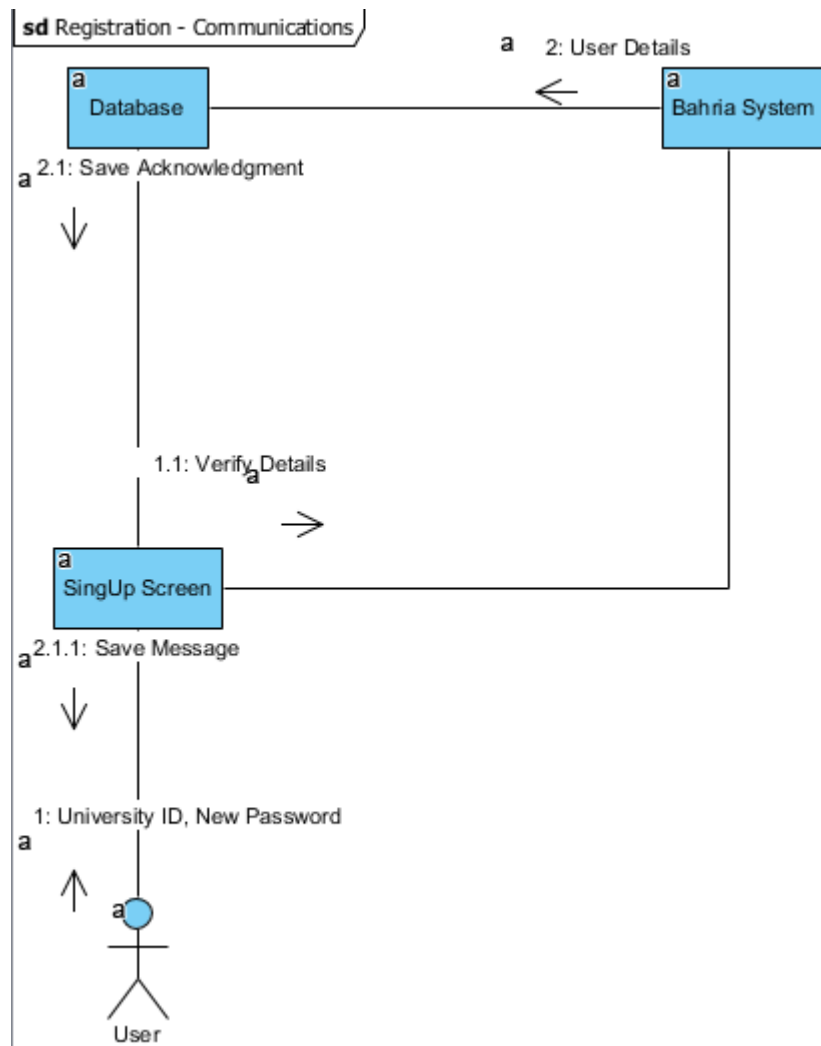


Figure 29: Collaboration Diagram Registration

Above diagram describes the communication between objects in registration process as follows

- User class instance was created
- Bahria University database connection instance was created
- BOL database connection instance was created
- User instance matched user entered id with Bahria database instance
- BOL database instance was saved required user detail from Bahria database instance and password entered by user

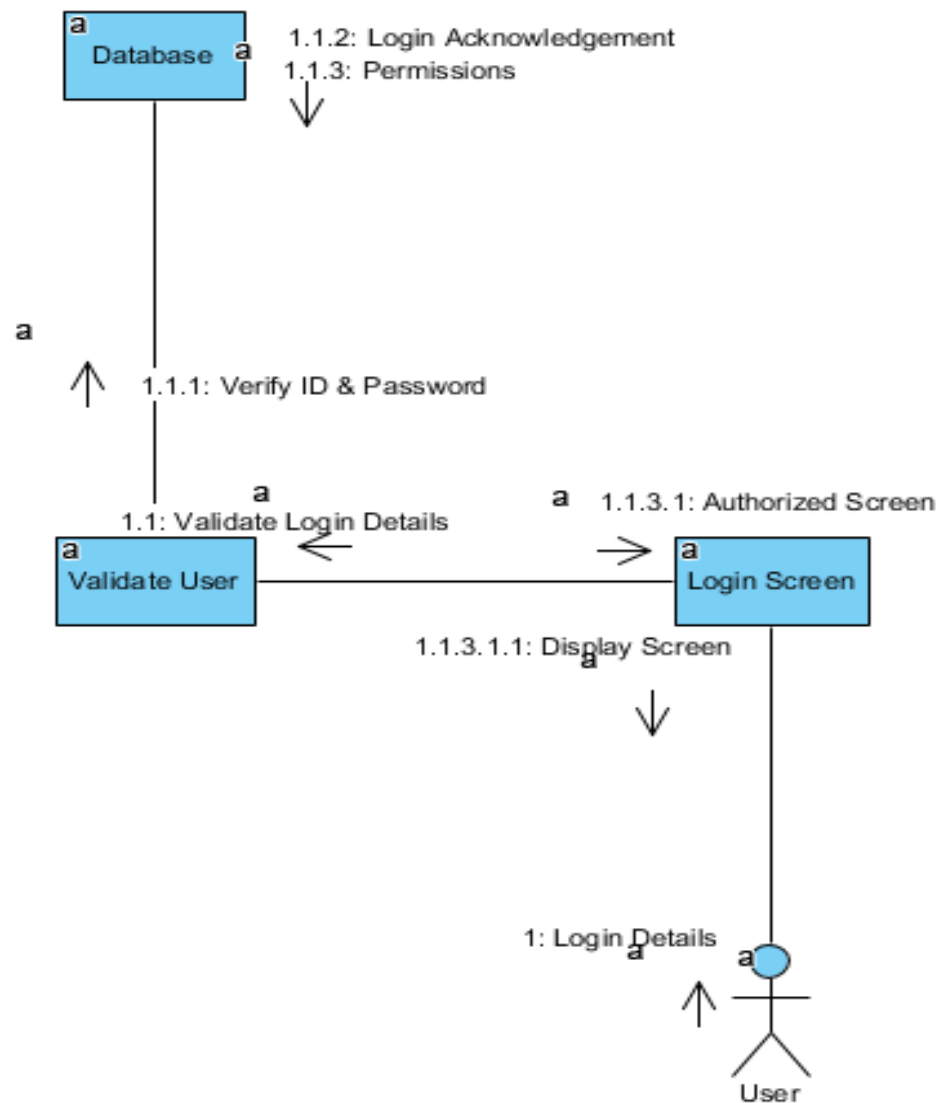


Figure 30: Collaboration Diagram Login

Above diagram describes the communication between objects in login process as follows

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

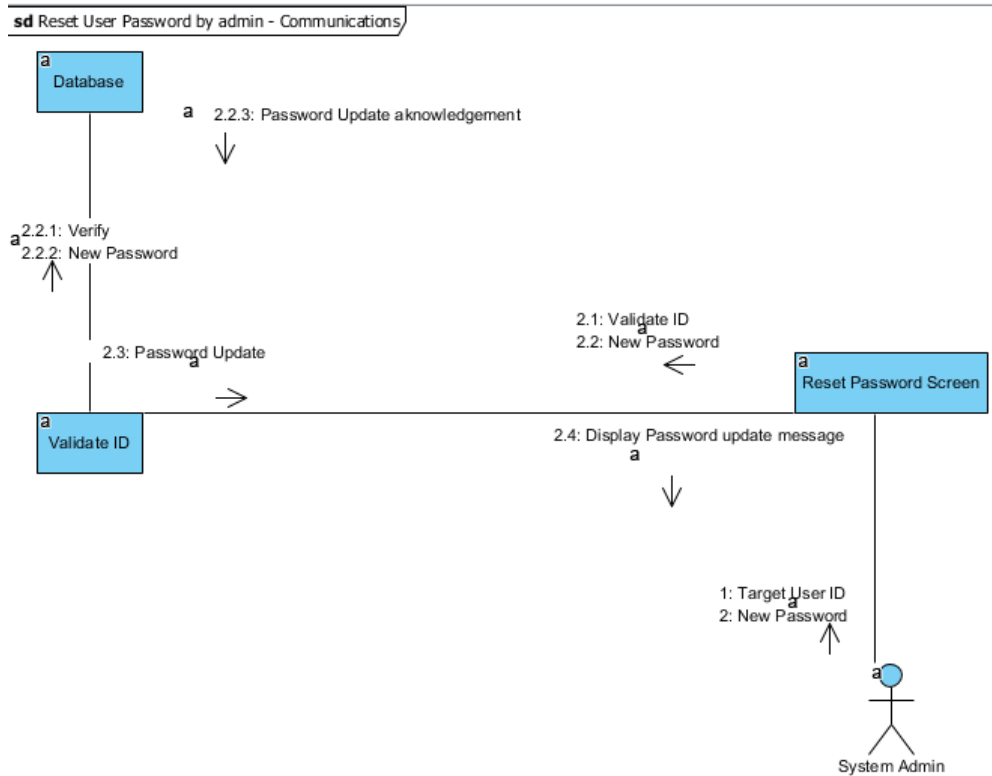


Figure 31: Collaboration Diagram Reset User Password by Admin

Above diagram describes the communication between objects in reset user password process as follows

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

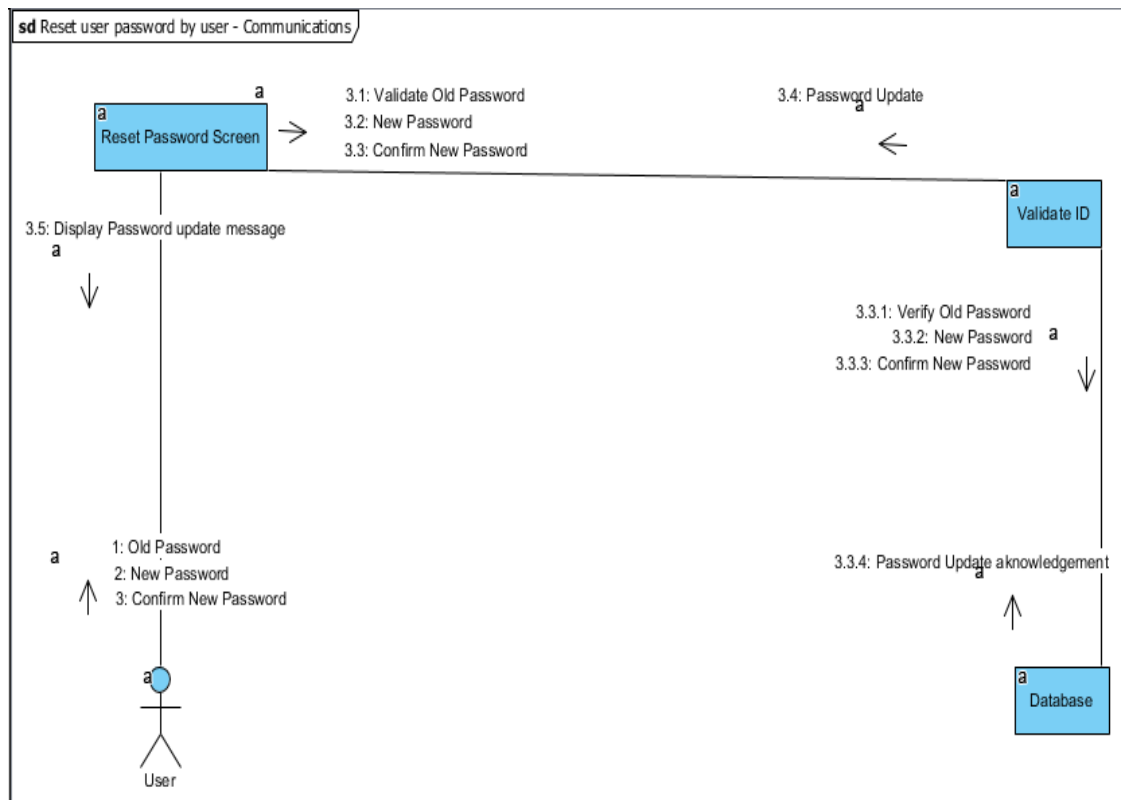


Figure 32: Collaboration Diagram Reset Password by User

Above diagram describes the communication between objects in reset user password by user process as follows

- User class instance was created
- BOL database connection instance was created
- User instance matched target user old password with BOL database instance
- BOL database instance update the target user password

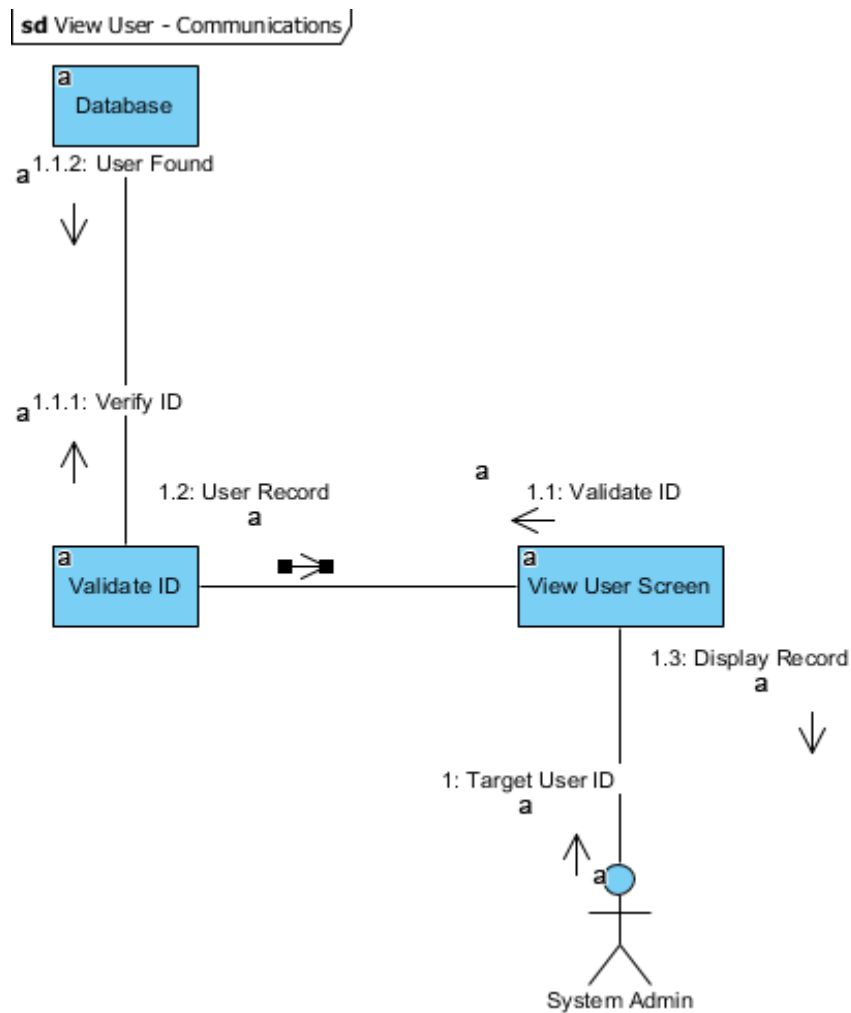


Figure 33: Collaboration Diagram View User

Above diagram describes the communication between objects in view user process as follows

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

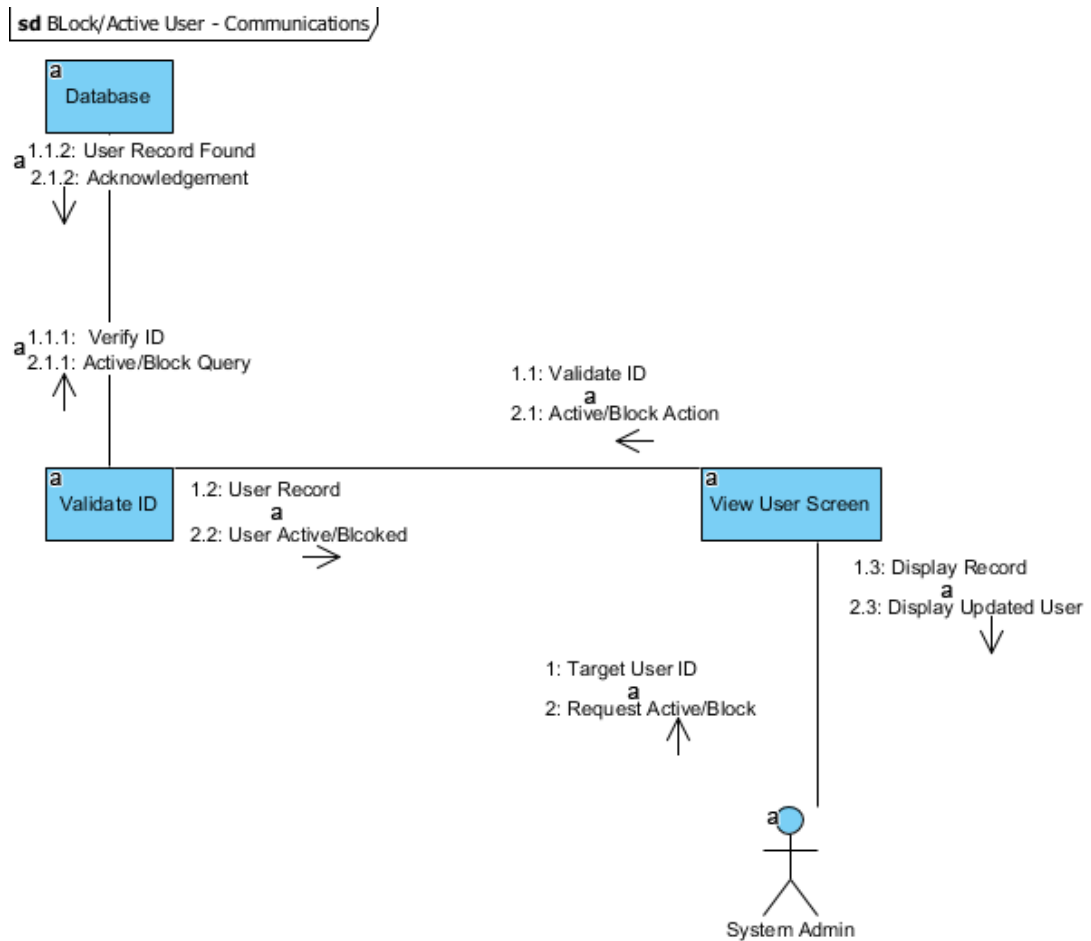


Figure 34: Collaboration Diagram Active/Block User

Above diagram describes the communication between objects in Active/Block user process as follows

- User class instance was created
- BOL database connection instance was created
- BOL database instance matched each userId with user instance userId
- BOL database returned user details to user instance
- Target user record was Active/Block by BOL database instance

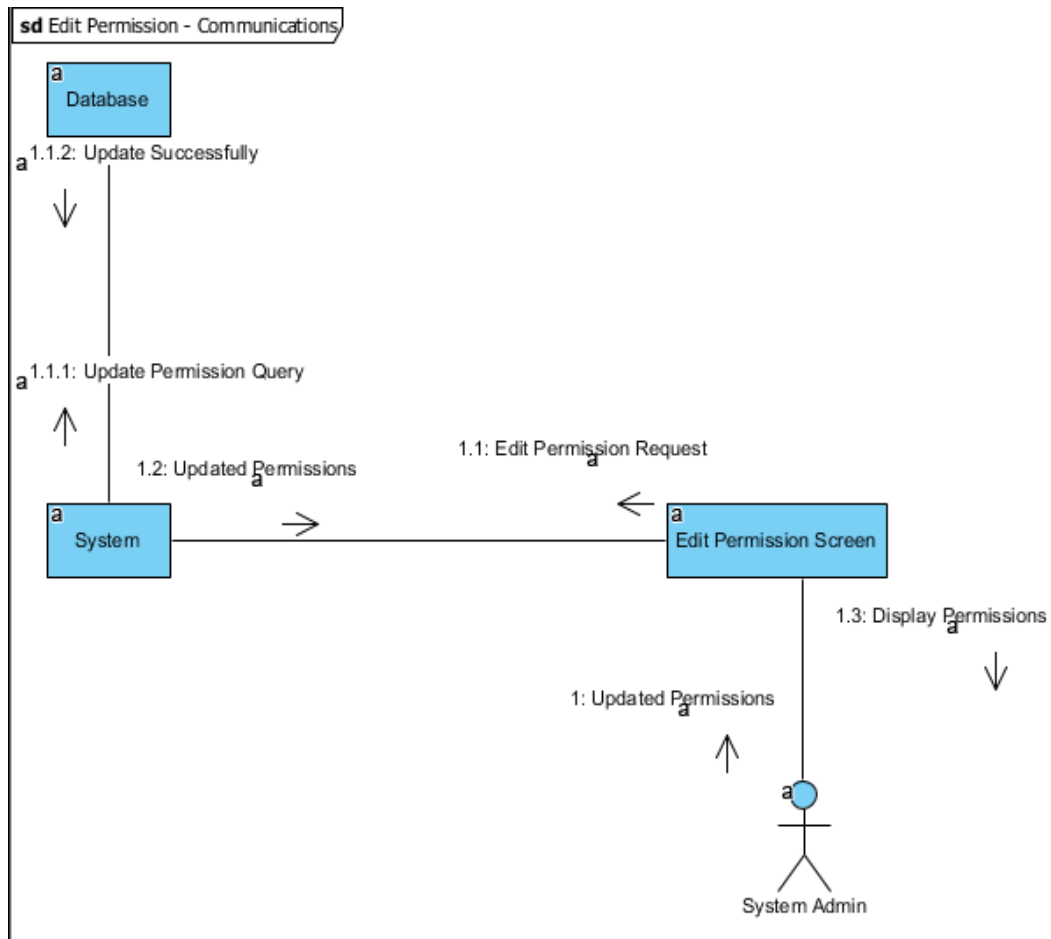


Figure 35: Collaboration Diagram Edit Permissions

Above diagram describes the communication between objects in edit group permissions process as follows

- User class instance was created
- Group class instance was created
- SideBar class instance was created
- PermissionSet class instance was created
- BOL database connection instance was created
- BOL database instance gave groupId, SideBarId and PermissionSet matched with group instance id ,sidebar instance id and put into permission set instance respectively given by user instance

- BOL database instance updated permissionSet given by permissionSet instance

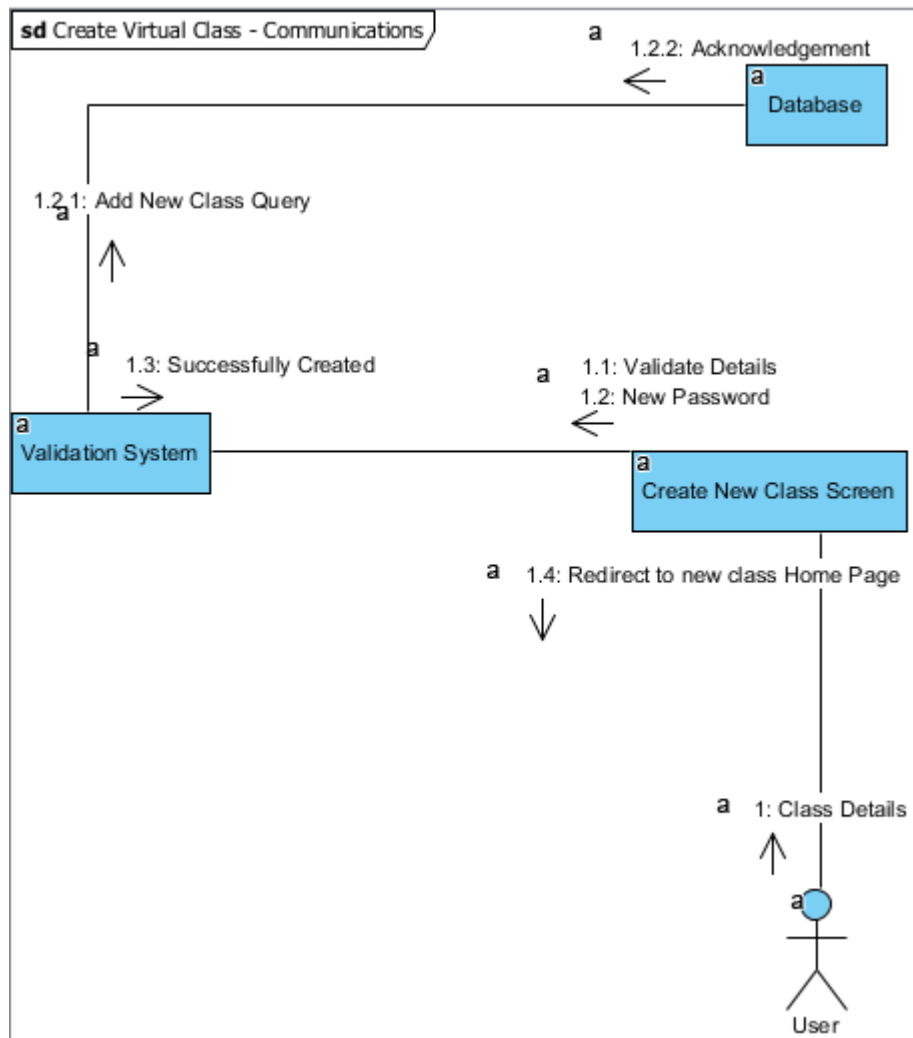


Figure 36: Collaboration Diagram Create Virtual Class

Above diagram describes the communication between objects in create virtual class process as follows

- User class instance was created
- Virtual class instance was created
- BOL database connection instance was created
- User instance put class details to virtual class instance
- Virtual class instance put details to BOL database instance

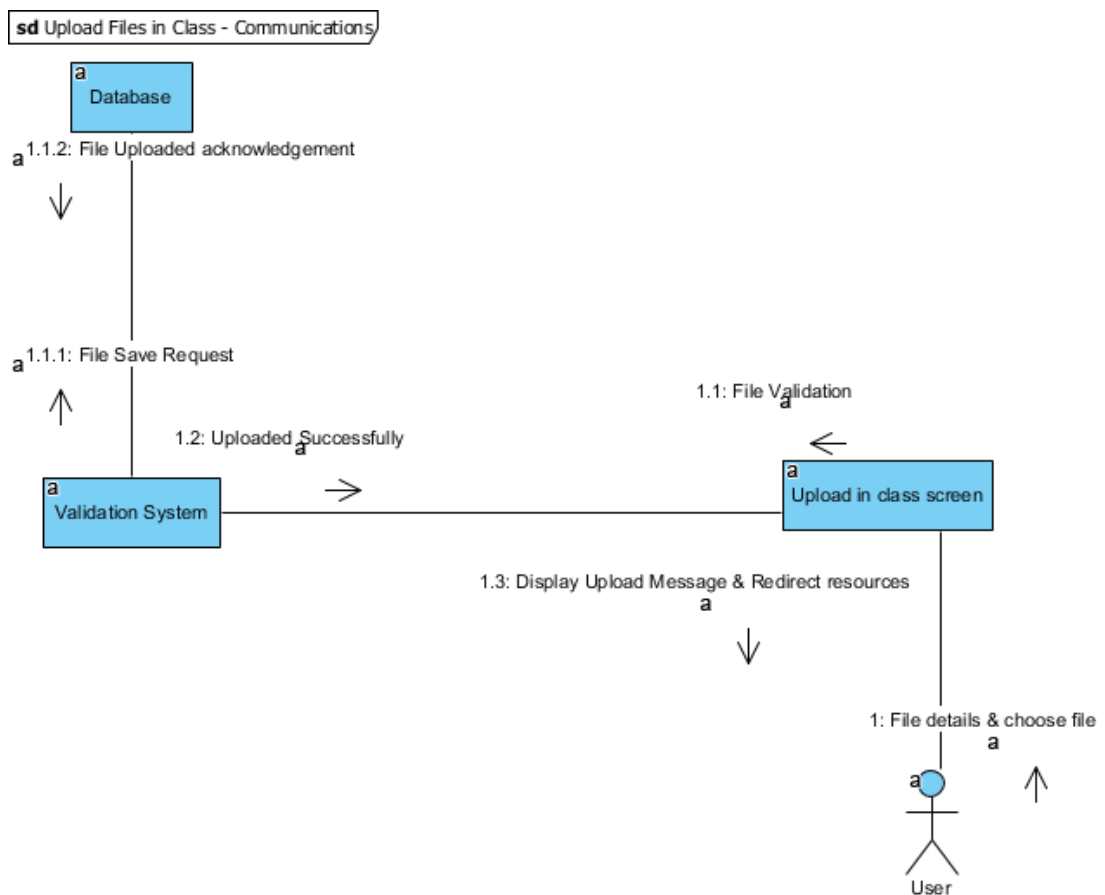


Figure 37: Collaboration Diagram Upload Files in Virtual Class

Above diagram describes the communication between objects in upload files in virtual class process as follows

- User class instance was created
- Virtual class instance was created
- Virtual class resource instance was created
- BOL database connection instance was created
- User class instance put file details to virtual class resource instance and virtual class instance put virtual class id to virtual class resource instance
- Virtual class resource instance put file details to BOL database instance

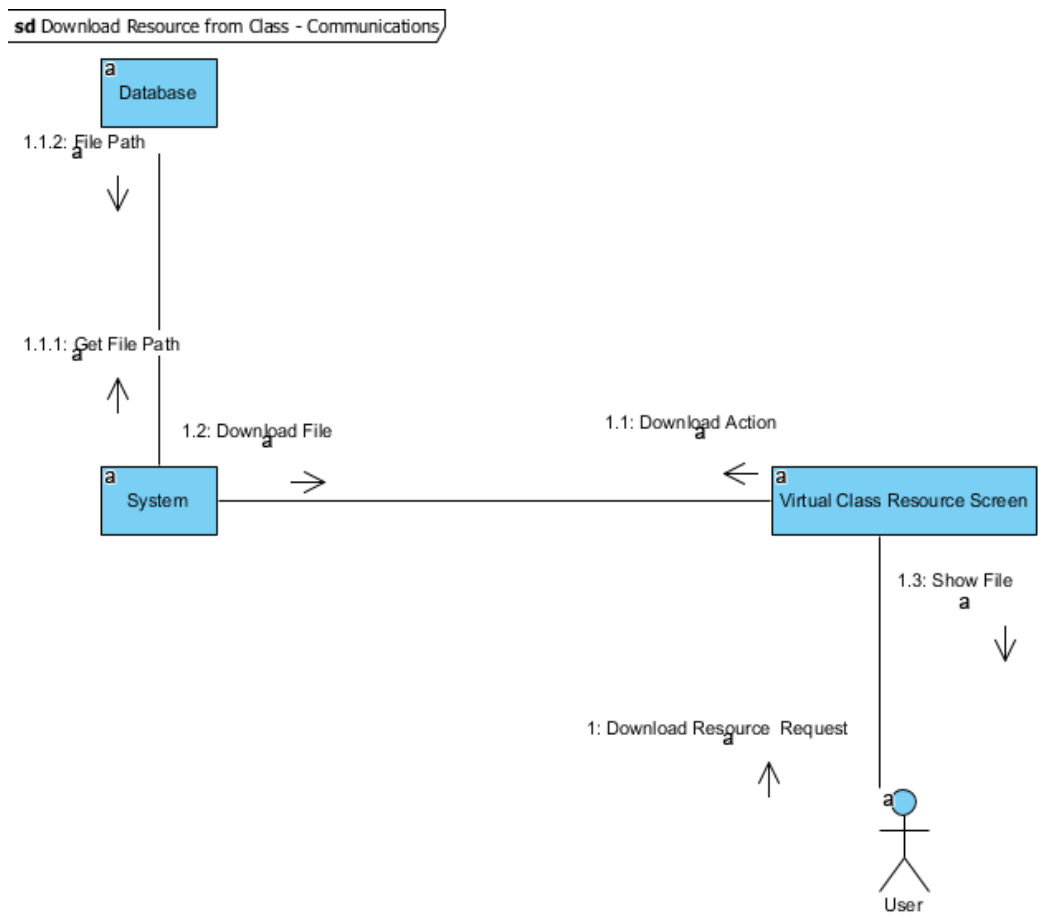


Figure 38: Collaboration Diagram Download Resource from Class

Above diagram describes the communication between objects in download files from virtual class process as follows

- User class instance was created
- Virtual class instance was created
- Virtual class resource instance was created
- BOL database connection instance was created
- User class instance put file details to virtual class resource instance and virtual class instance put virtual class id to virtual class resource instance
- Virtual class resource instance put file details to BOL database instance
- BOL database instance put file path to virtual class resource instance

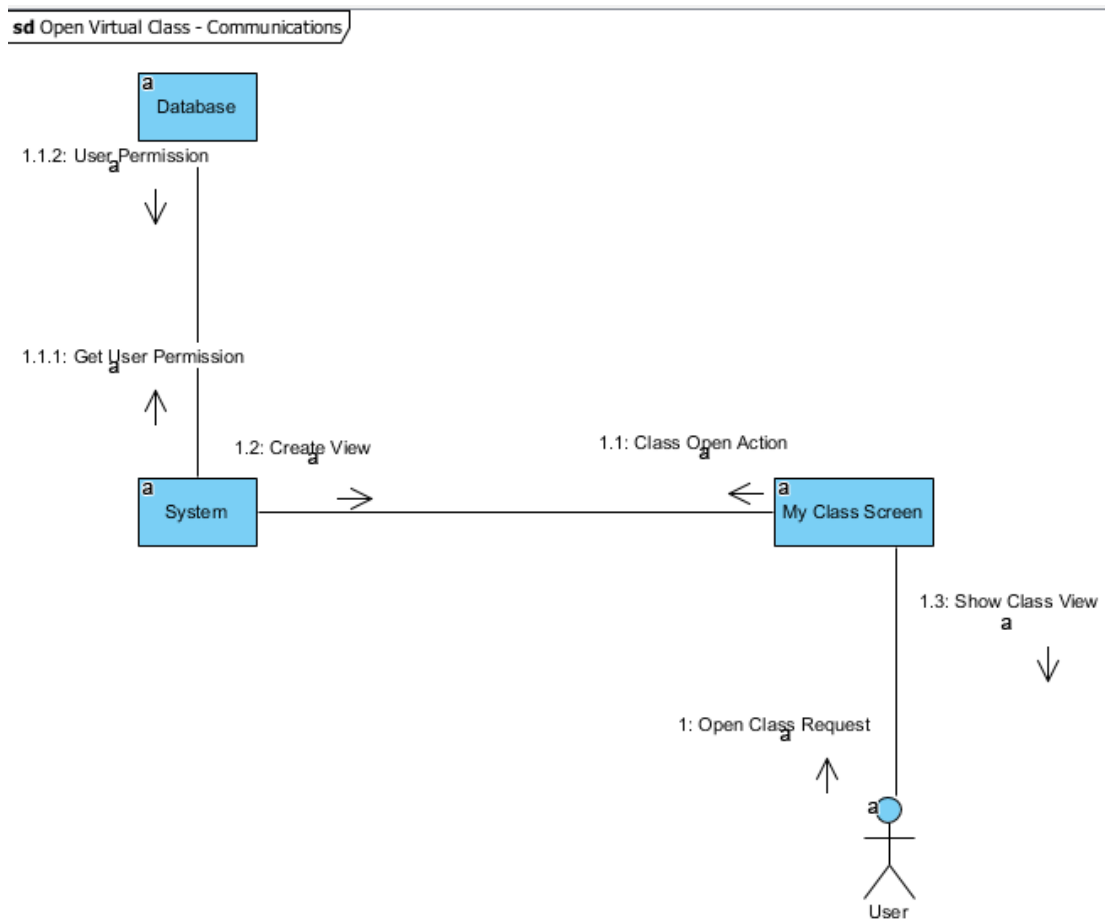


Figure 39: Collaboration Diagram Open Virtual Class

Above diagram describes the communication between objects in open virtual class process as follows

- User class instance was created
- Virtual class instance was created
- BOL database connection instance was created
- User instance was put class id to BOL database instance
- BOL database instance returned target class details to virtual class instance

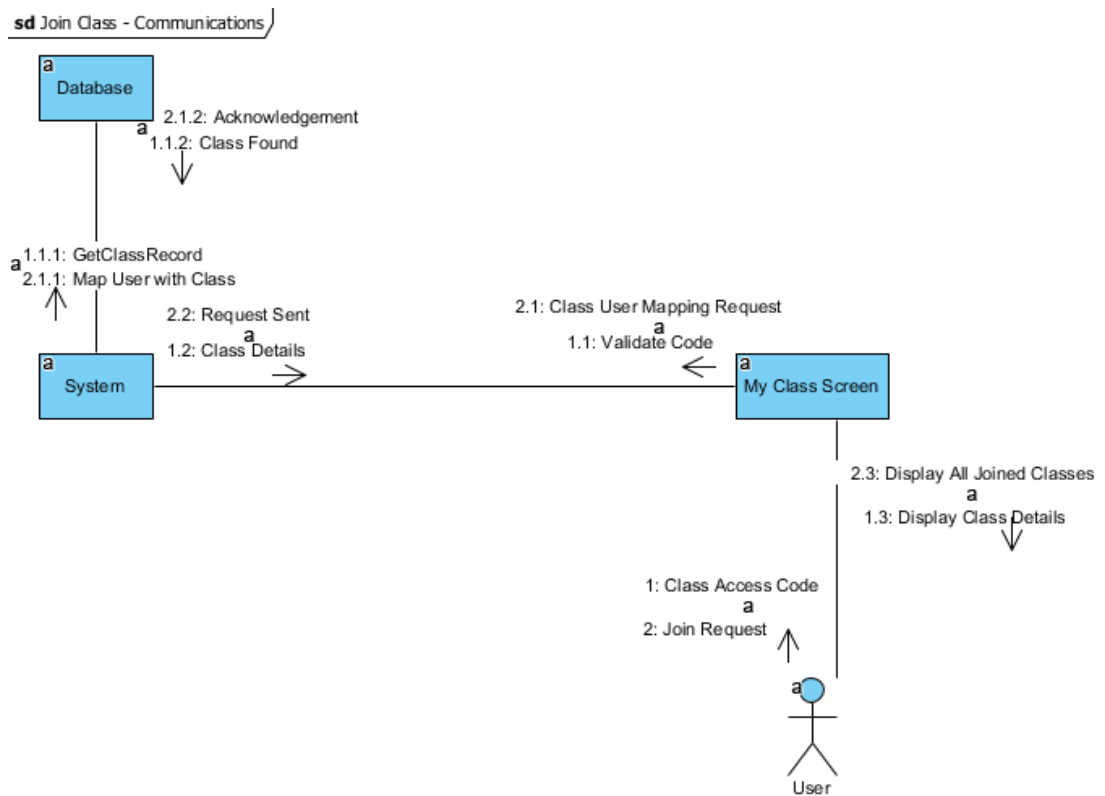


Figure 40: Collaboration Diagram Join Virtual Class

Above diagram describes the communication between objects in join virtual class process as follows

- User class instance was created
- Virtual class instance was created
- BOL database connection instance was created
- User instance put course code to BOL database instance
- BOL database return class details to virtual class instance
- User class instance was put user detail to virtual class instance

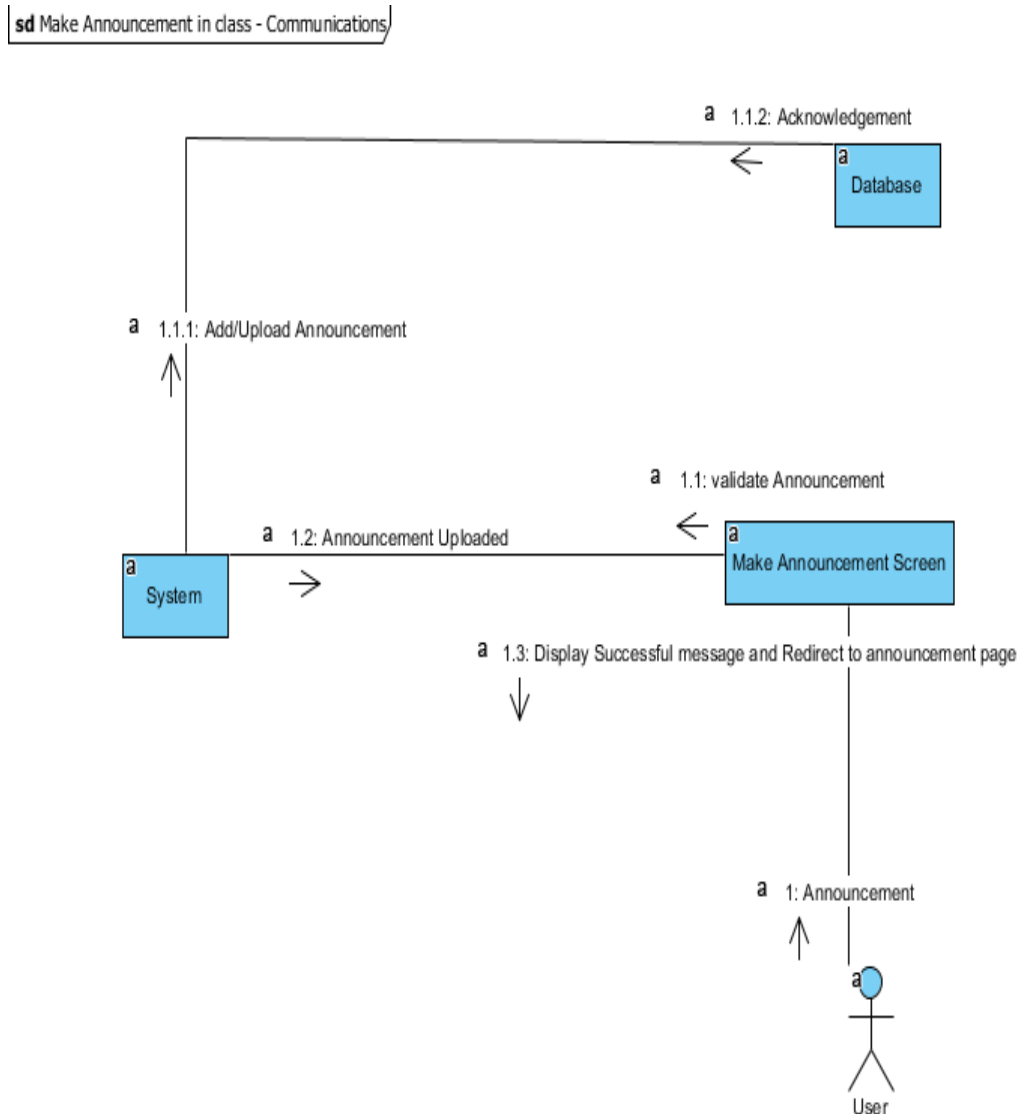


Figure 41: Collaboration Diagram Make Announcement in Class

Above diagram describes the communication between objects in make announcements in virtual class process as follows

- User class instance was created
- Virtual class instance was created
- Virtual class announcement instance was created
- User instance put announcement and virtual class instance put class id in virtual class announcement instance respectively
- virtual class announcement instance put announcement details in BOL database instance

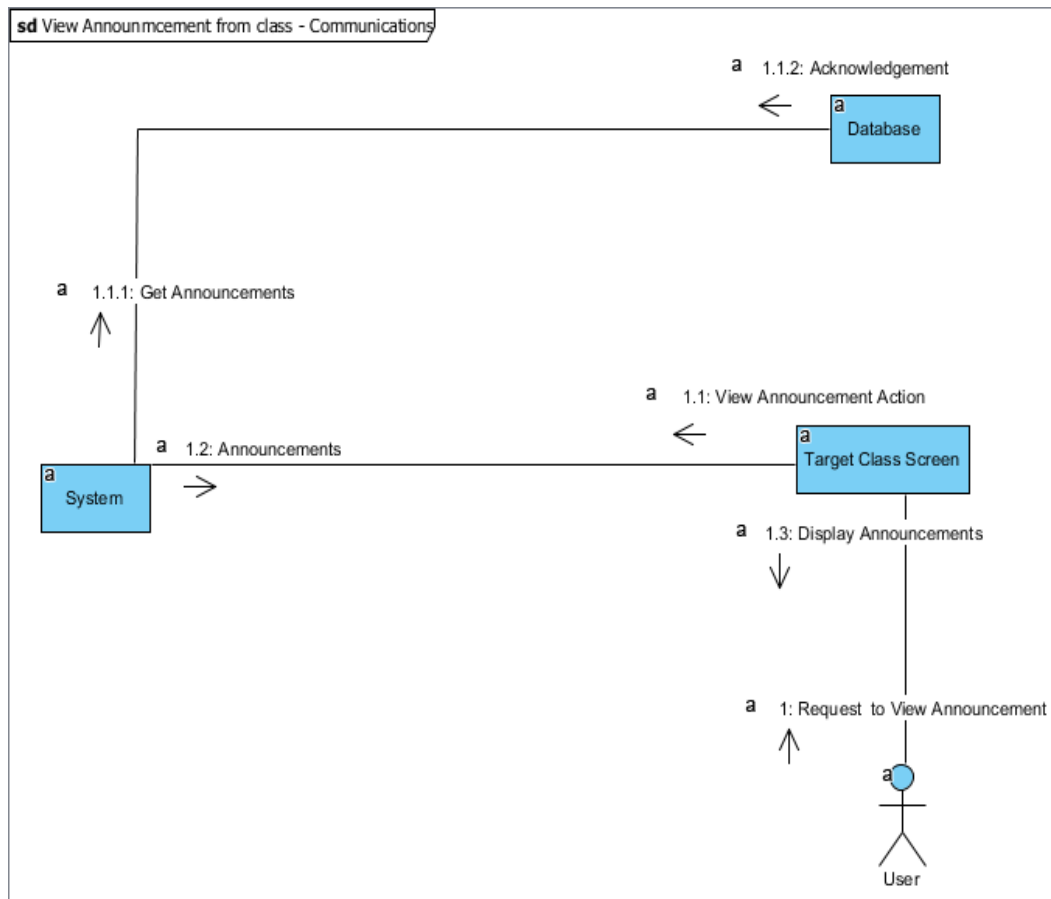


Figure 42: Collaboration Diagram View Announcement from Class

Above diagram describes the communication between objects in view announcements in virtual class process as follows

- User class instance was created
- Virtual class announcement instance was created
- User class instance put class id to virtual class announcement instance
- Virtual class announcement put virtual class id to BOL database instance
- BOL database instance returned announcements to virtual class announcement instance

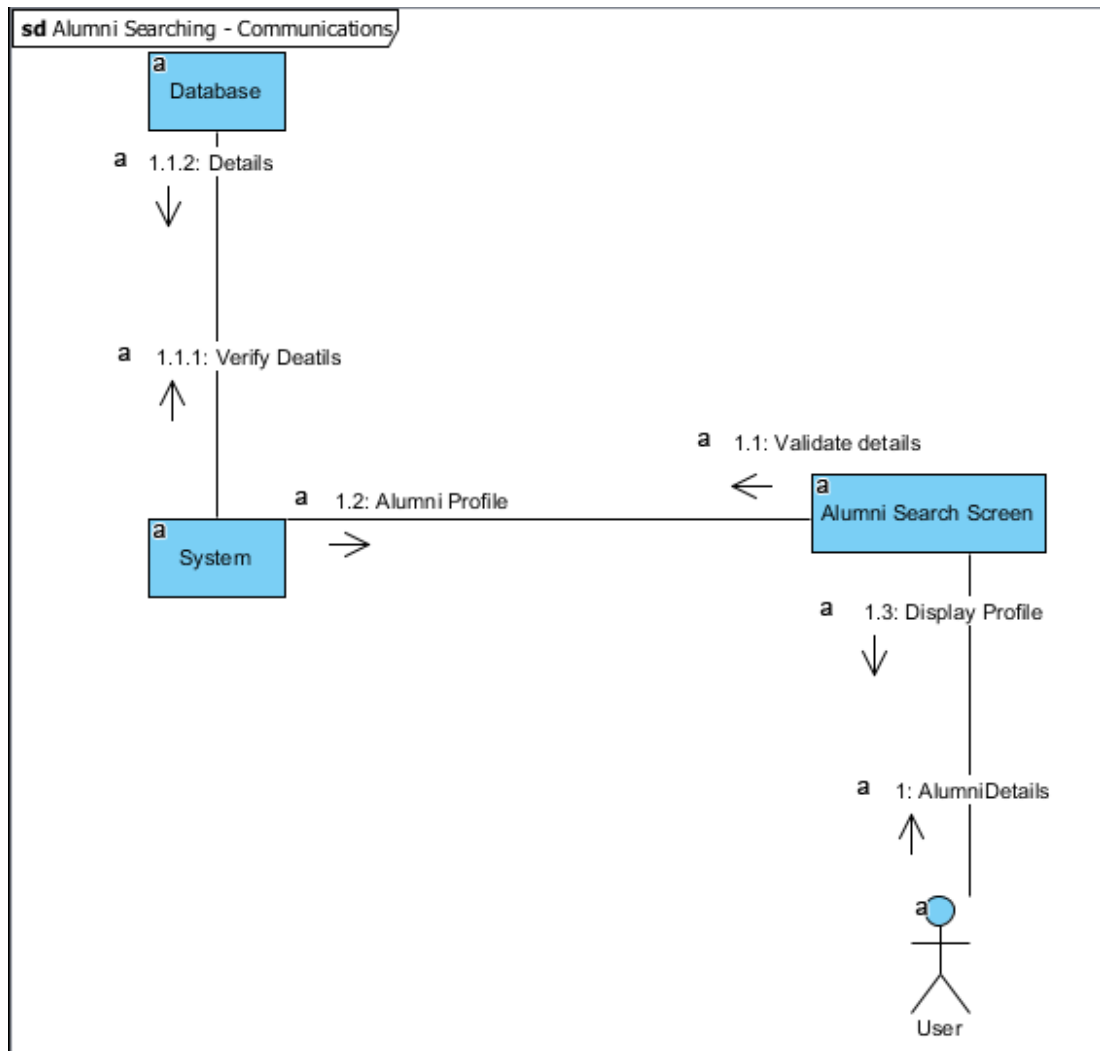


Figure 43: Collaboration Diagram Alumni Search

Above diagram describes the communication between objects in alumni searching process as follows

- User instance was created
- Alumni instance was created
- BOL database instance was created
- User instance put user id to alumni instance
- Alumni instance put user id to BOL database instance
- BOL database returned alumni details to alumni instance

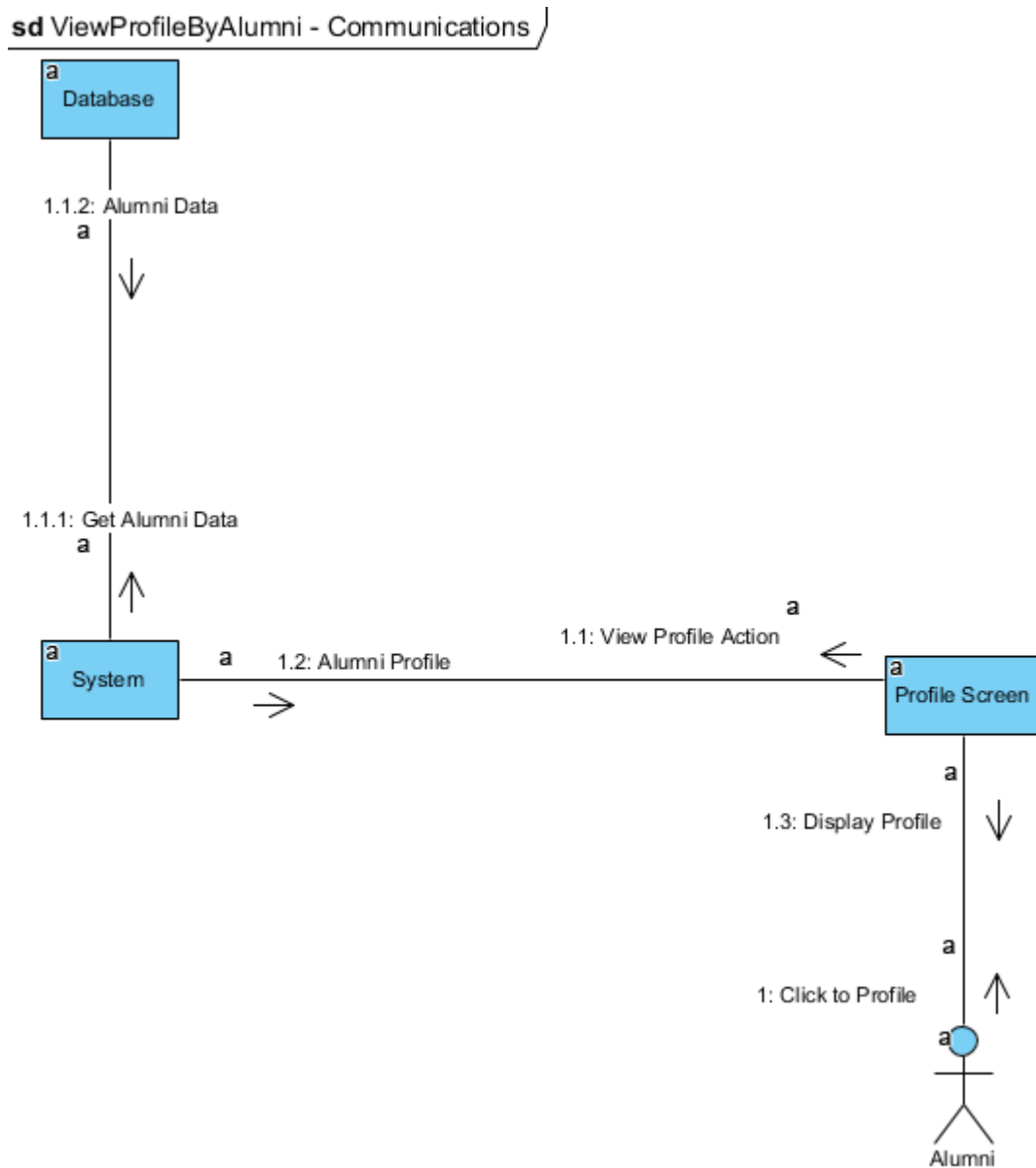


Figure 44: Collaboration Diagram View Profile by Alumni

Above diagram describes the communication between objects in alumni profile open process as follows

- Alumni instance was created
- BOL database instance was created
- Alumni instance put user id to BOL database instance
- BOL database returned alumni details to alumni instance

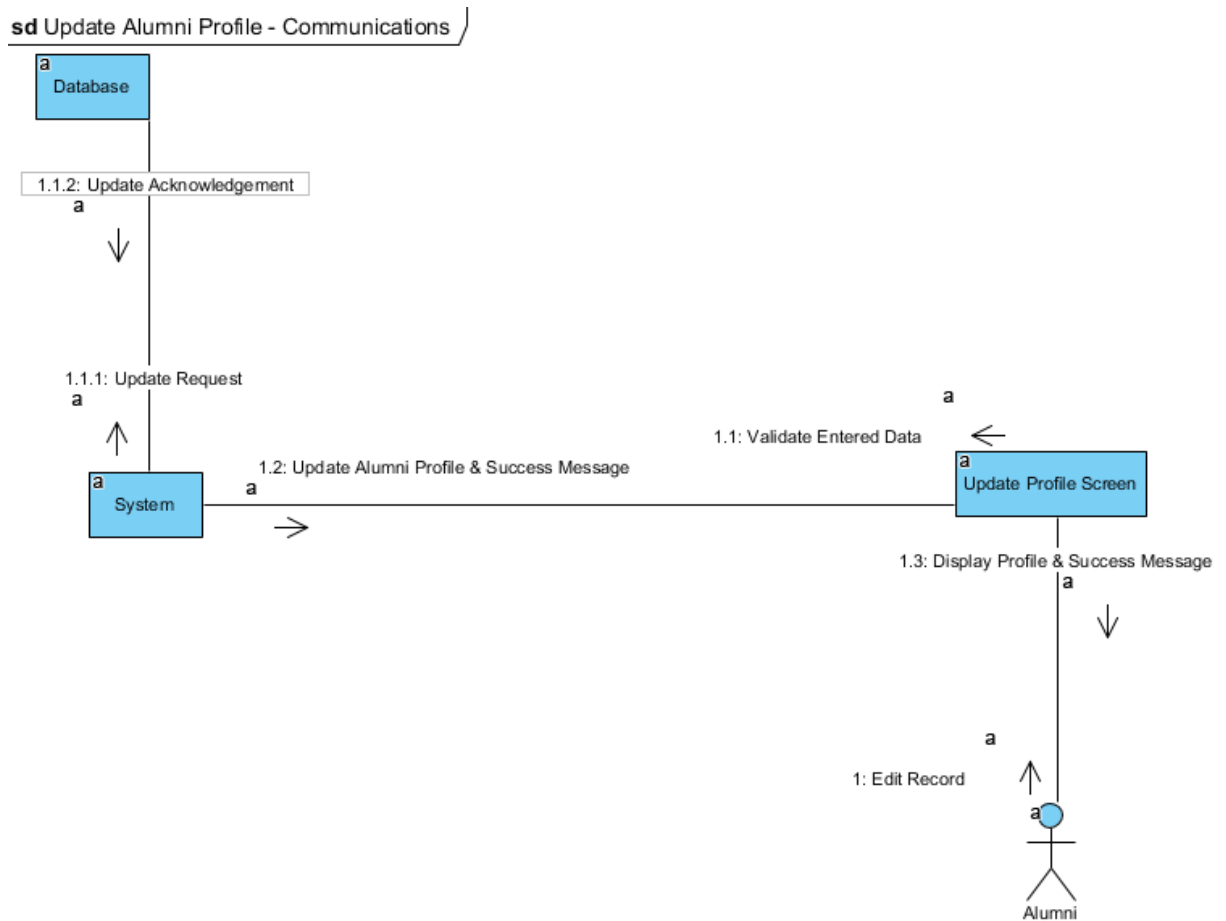


Figure 45: Collaboration Diagram Update Alumni Profile

Above diagram describes the communication between objects in update alumni profile process as follows

- Alumni instance was created
- BOL database instance was created
- Alumni instance put updated details to BOL database instance
- BOL database returned updated alumni details to alumni instance

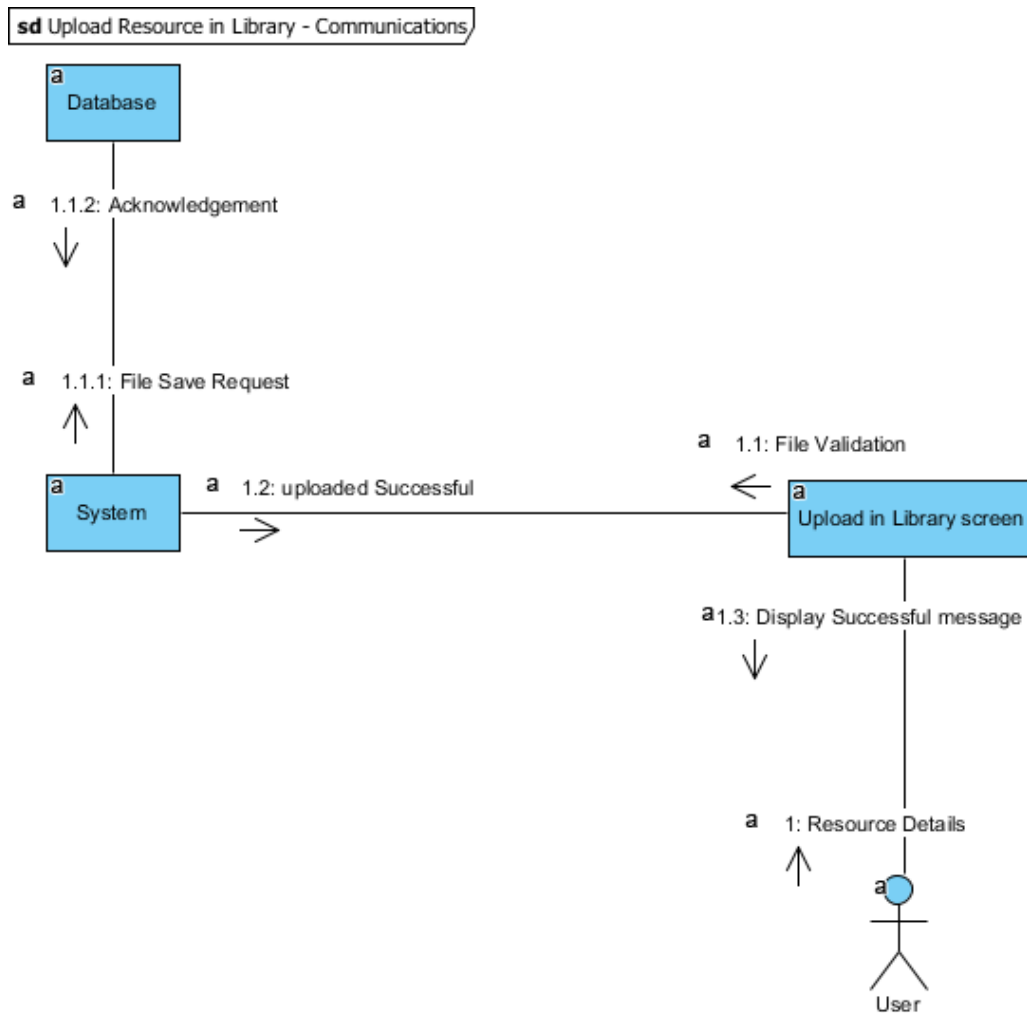


Figure 46: Collaboration Diagram Resource in Library

Above diagram describes the communication between objects in upload resources in library process as follows

- User instance was created
- Library instance was created
- BOL database instance was created
- User instance put resource id to library instance
- Library instance put resource id to BOL database instance
- BOL database instance returned file path to library instance

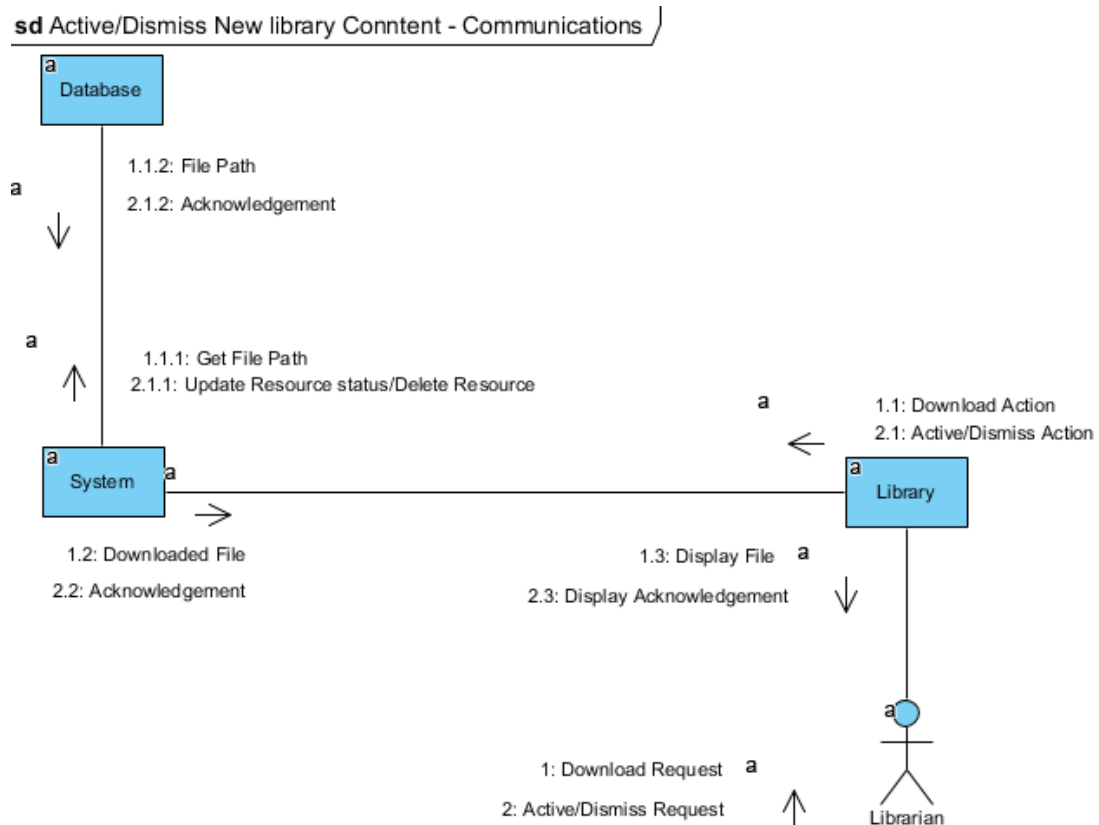


Figure 47: Collaboration Diagram Active/Dismiss New Content from Library

Above diagram describes the communication between objects in Active/dismiss new content of library process as follows

- Librarian instance was created
- BOL database instance was created
- Librarian instance put updated status to BOL database instance or delete record from BOL database instance
- BOL database returned remaining contents

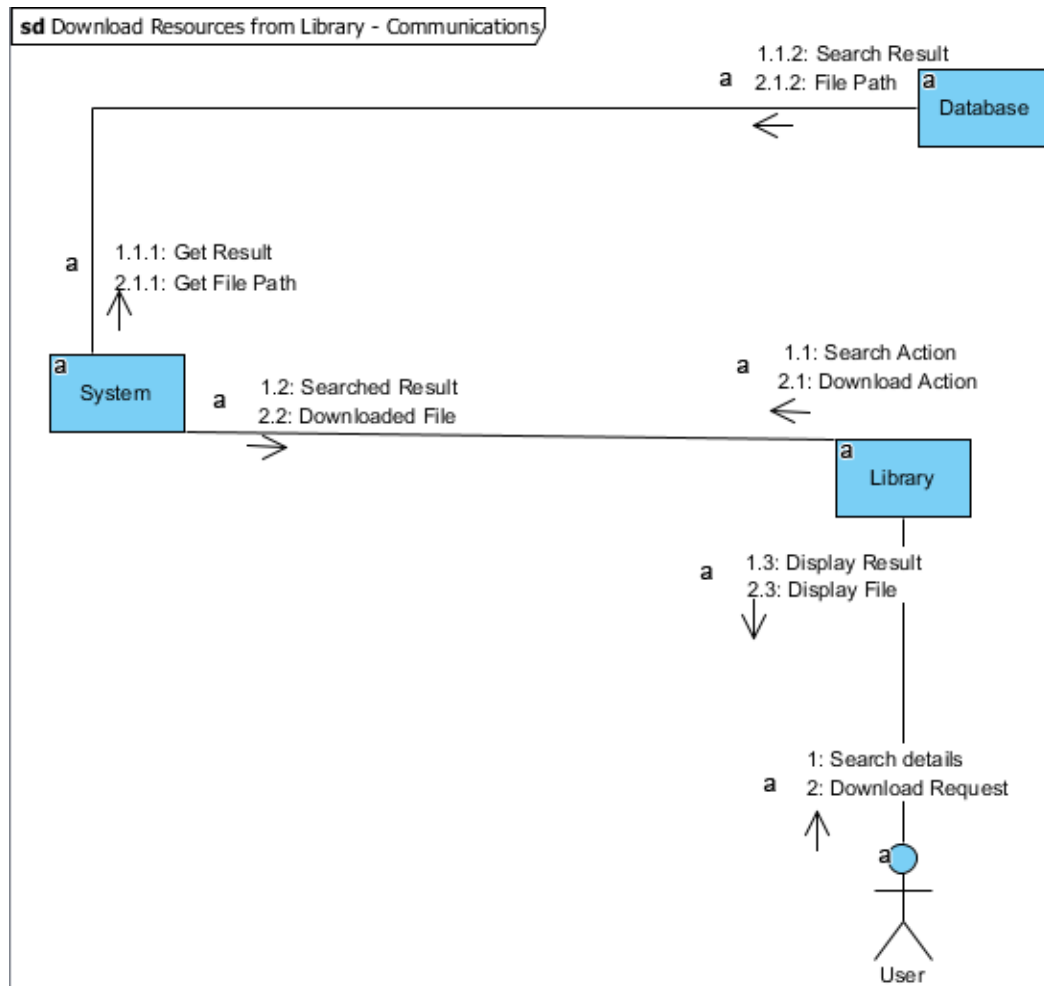


Figure 48: Collaboration Diagram Download Resource from Library

Above diagram describes the communication between objects in download resources from library process as follows

- User instance was created
- Library instance was created
- BOL database instance was created
- User instance put resource id to library instance
- Library instance put resource id to BOL database instance
- BOL database instance returned file path to library instance

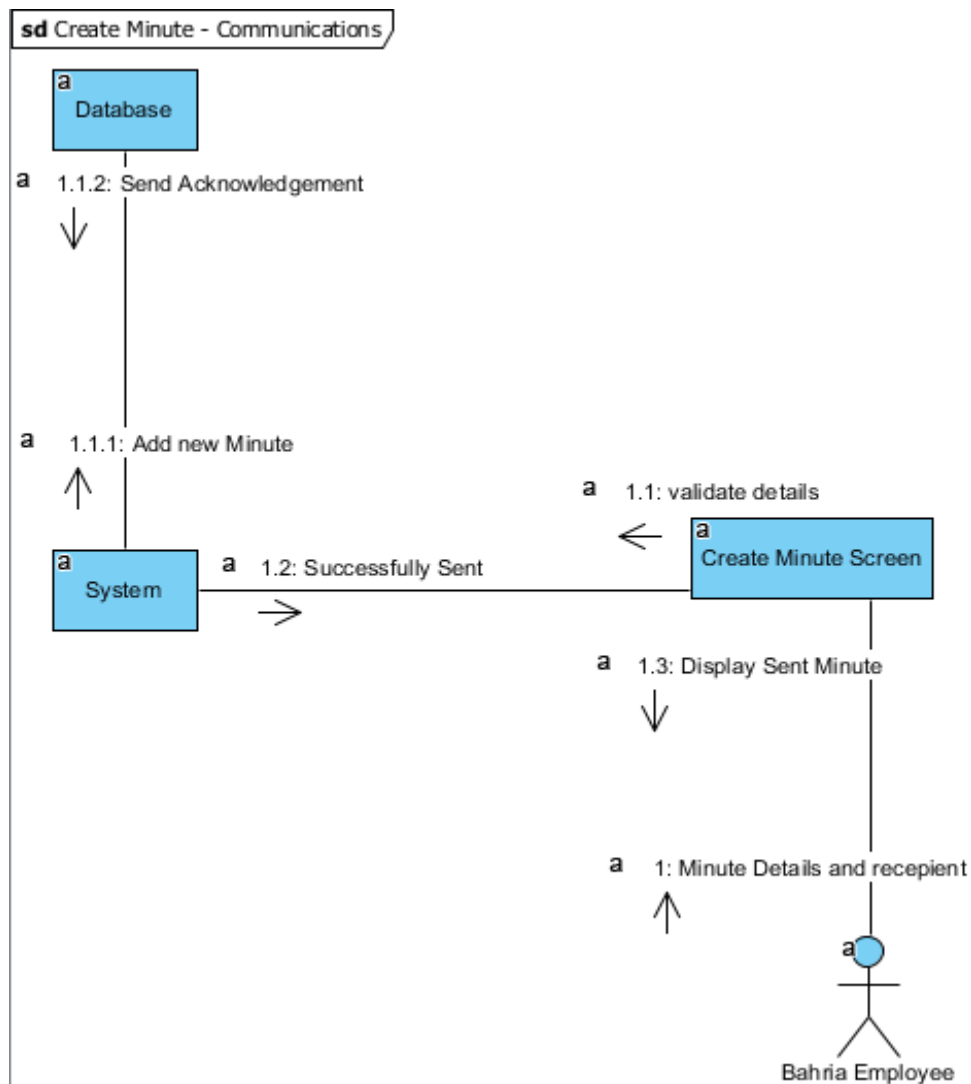


Figure 49: Collaboration Diagram Create a Minute

Above diagram describes the communication between objects in create minute process as follows

- User instance was created
- minute instance was created
- BOL database instance was created
- User instance put minute details to minute instance
- Minute instance put details to BOL database instance

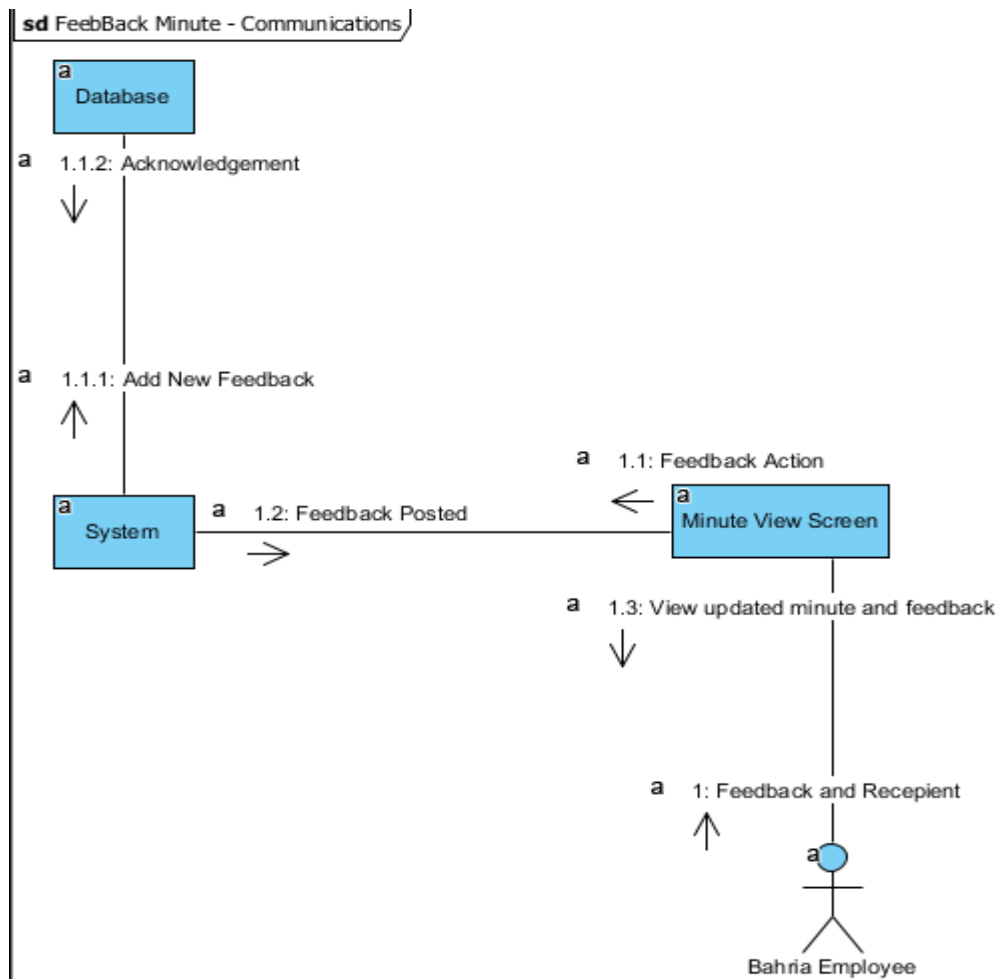


Figure 50: Collaboration Diagram Feedback on a Minute

Above diagram describes the communication between objects in feedback of a minute process as follows

- User instance was created
- minute instance was created
- feedback minute was created
- BOL database instance was created
- User instance put feedback details and minute instance put minute id to feedback minute instance
- Feedback Minute instance put details to BOL database instance

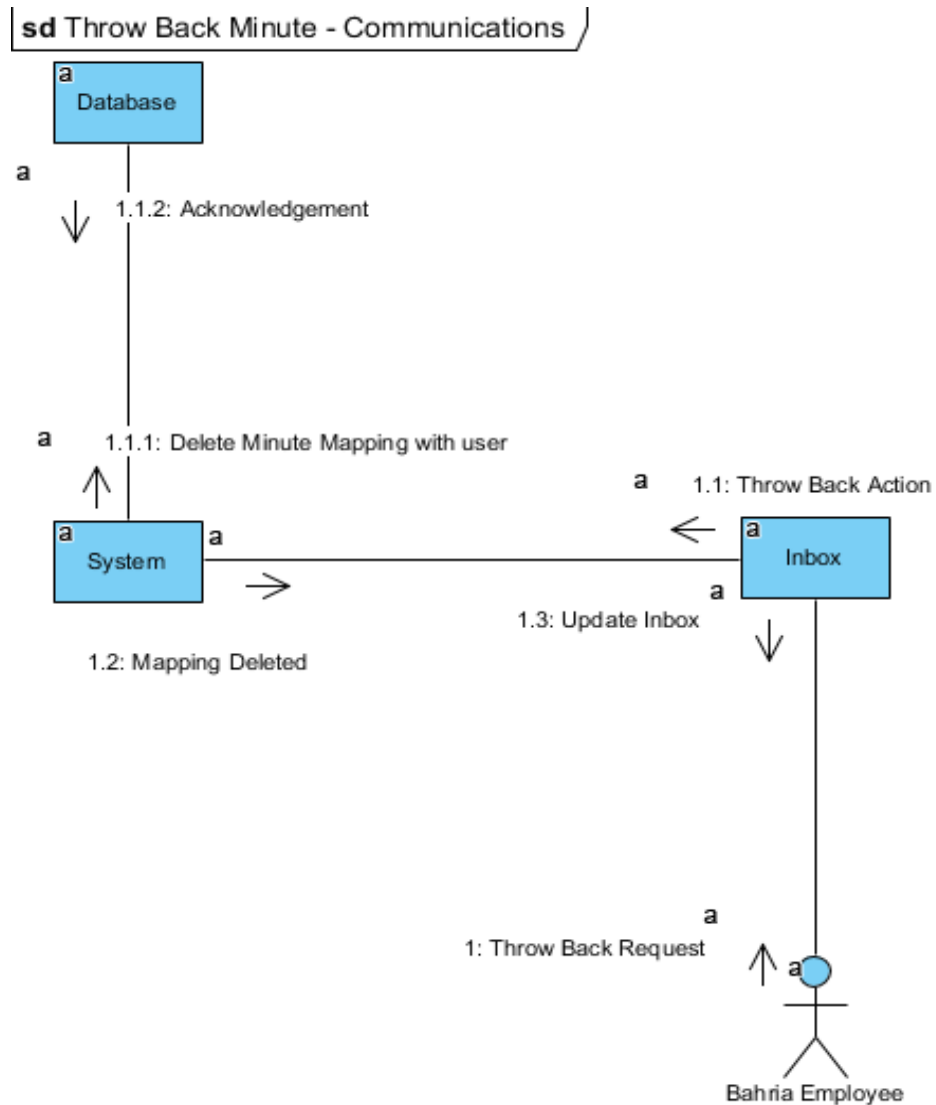


Figure 51: Collaboration Diagram Throw Back a Minute

Above diagram describes the communication between objects in throwback a minute process as follows

- User instance was created
- minute instance was created
- BOL database instance was created
- User instance fire throwback action
- BOL database instance discard user mapping

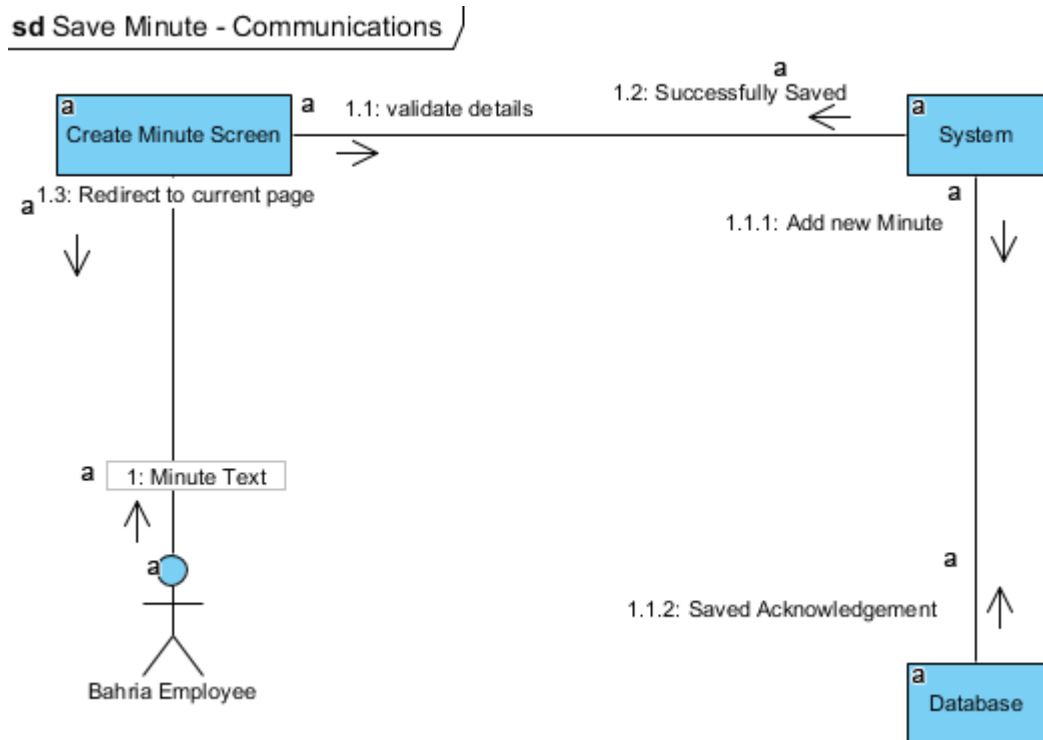


Figure 52: Collaboration Diagram Save a Minute

Above diagram describes the communication between objects in save minute process as follows

- User instance was created
- minute instance was created
- BOL database instance was created
- User instance put minute details to minute instance
- Minute instance put details to BOL database instance

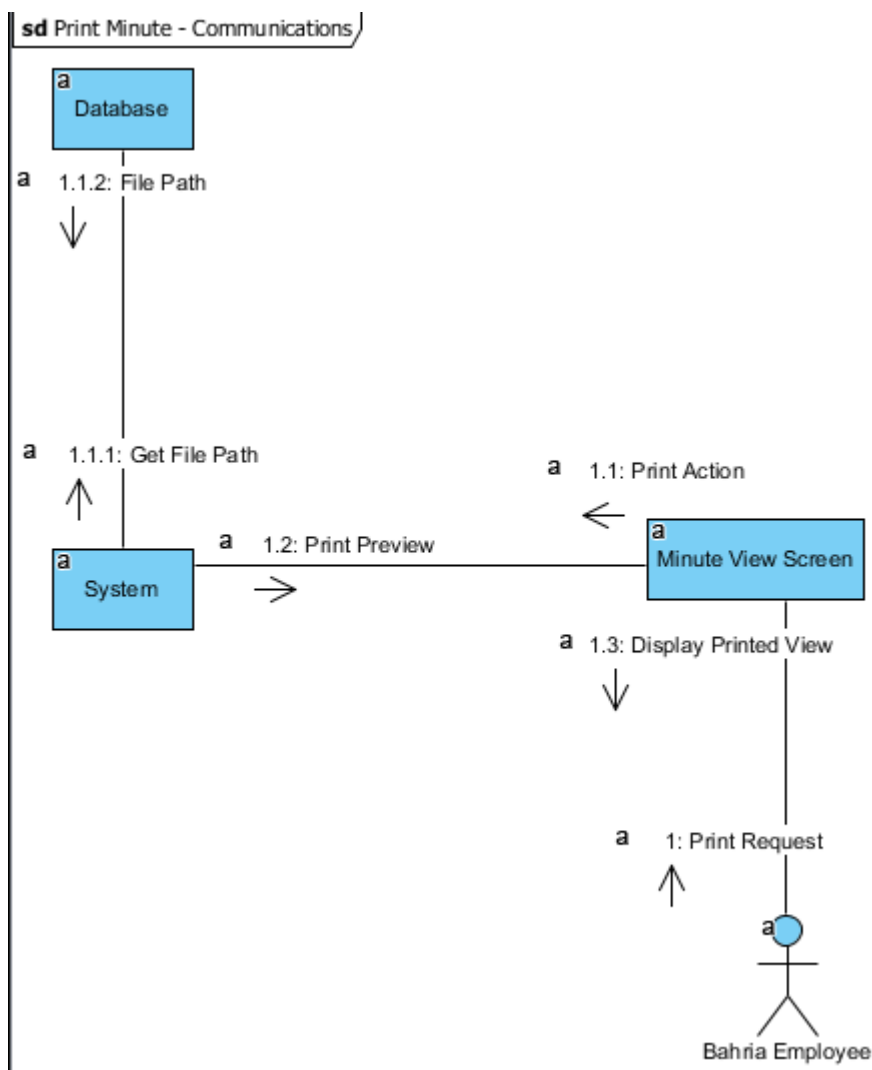


Figure 53: Collaboration Diagram Print a Minute

Above diagram describes the communication between objects in printing a minute process as follows

- User instance was created
- minute instance was created
- BOL database instance was created
- User instance put minute id to minute instance
- Minute instance put minute id BOL database instance
- BOL database instance put minute details to minute instance

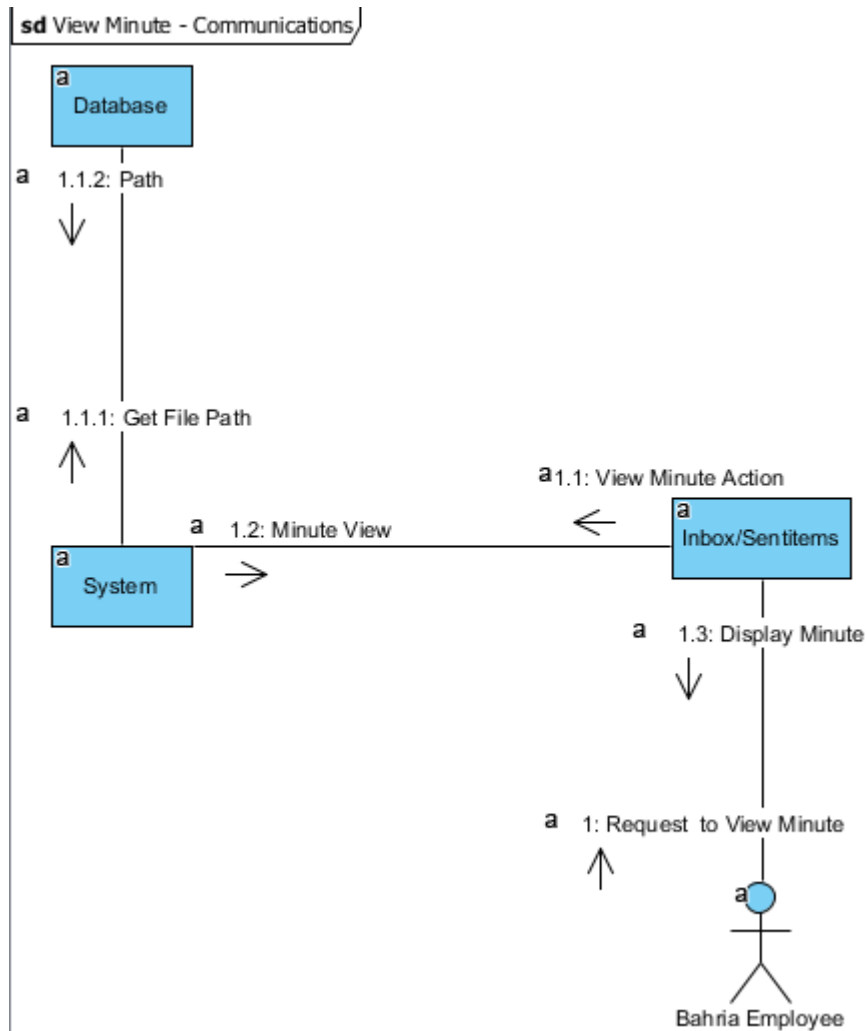


Figure 54: Collaboration Diagram View a Minute

Above diagram describes the communication between objects in viewing a minute process as follows

- User instance was created
- minute instance was created
- BOL database instance was created
- User instance put minute id to minute instance
- Minute instance put minute id BOL database instance
- BOL database instance put minute details to minute instance

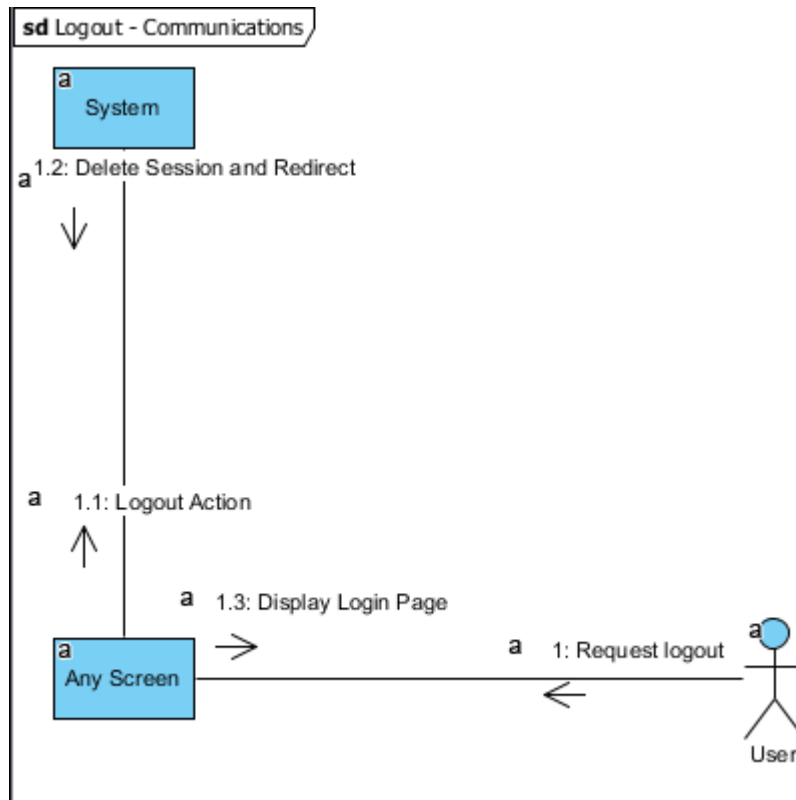


Figure 55: Collaboration Diagram Logout

Above diagram describes the communication between objects in logout process as follows

- User created instance and session was deleted
- BOL database connection was disconnect

3.5 Domain Model

A domain model is a conceptual model of the domain that incorporates both behaviour and data. Domain Modelling is a way to describe and model real world entities and the relationships between them, which collectively describe the problem domain space. Derived from an understanding of system-level requirements, identifying domain entities and their relationships provides an effective basis for understanding and helps practitioners design systems for maintainability, testability, and incremental development. Following is the Domain Model of Bahria One Link (BOL).

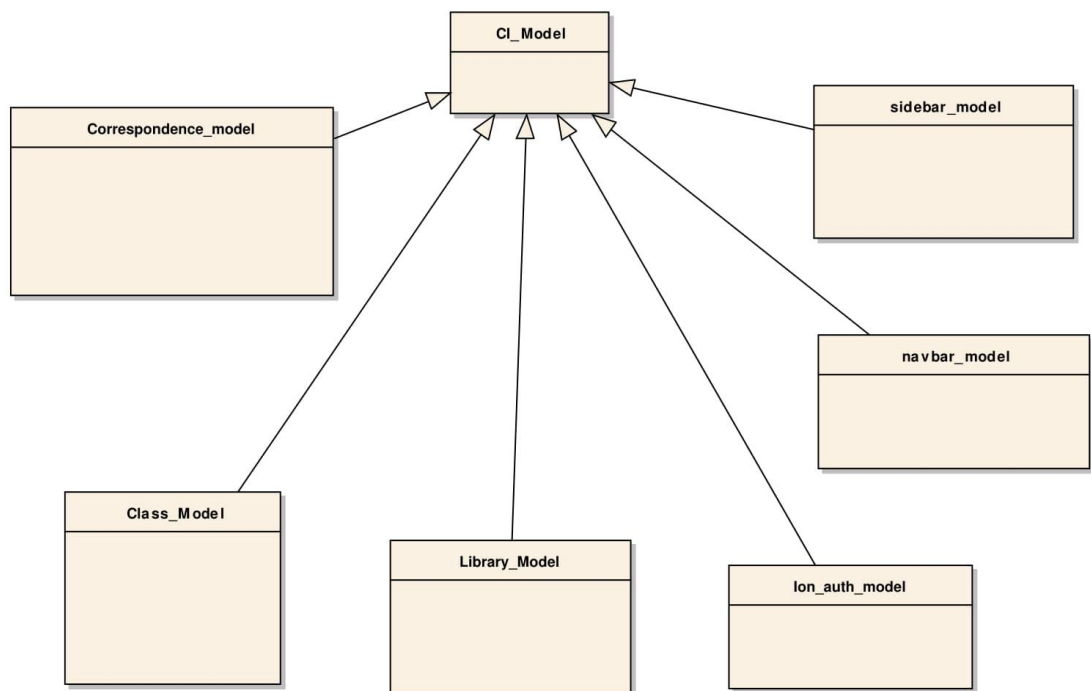


Figure 56: Domain Model - 1

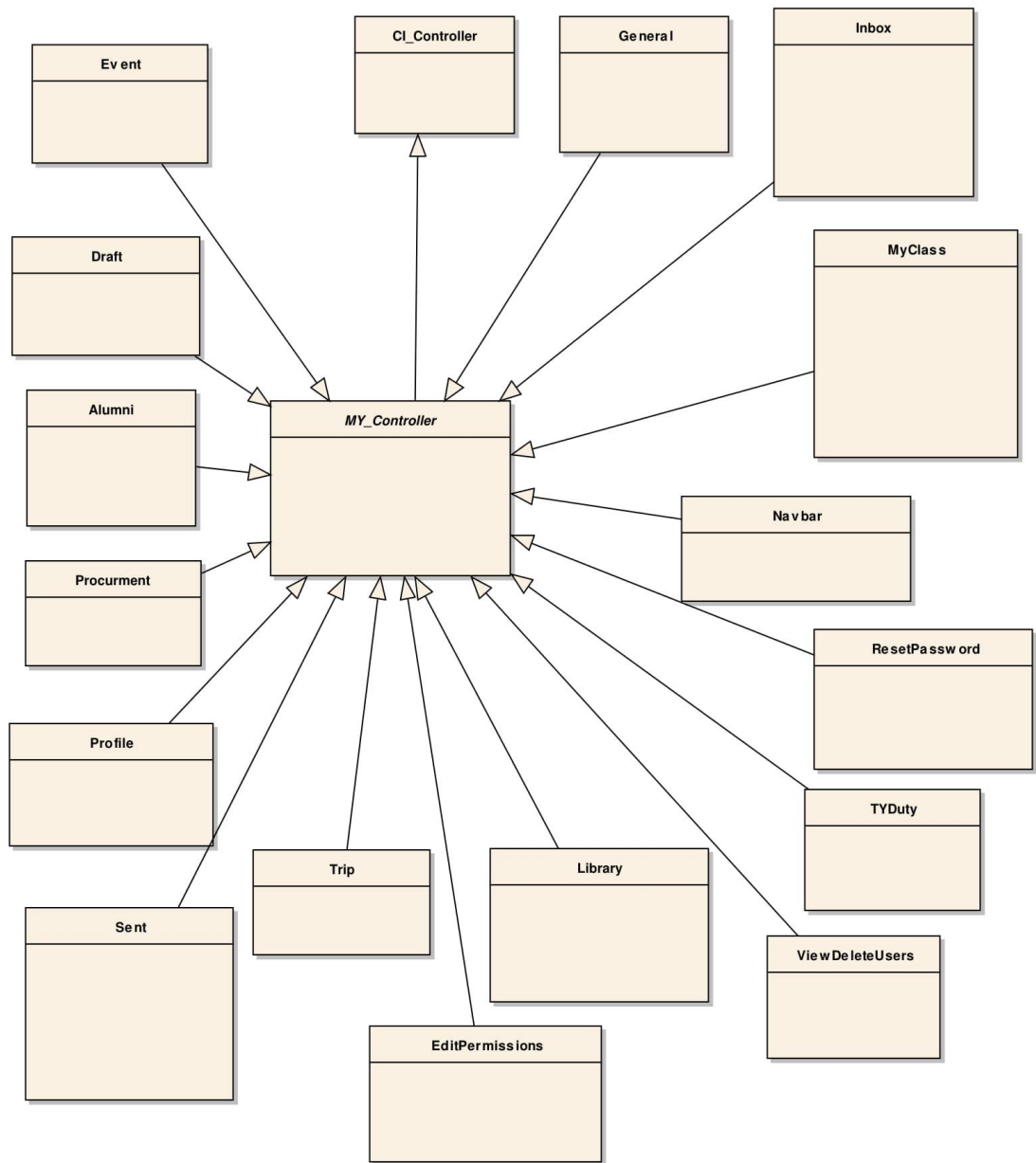


Figure 57: Domain Model - 2

3.6 Design Class Diagram

Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

Capsules are also stereotyped classes, used to represent concurrent threads of execution in real-time systems. In such cases, other design classes are 'passive' classes, used within the execution context provided by the 'active' capsules. When the software architect and designer choose not to use a design approach based on capsules, it is still possible to model concurrent behaviour using 'active' classes.

Active classes are design classes, which coordinate and drive the behaviour of the passive classes - an active class is a class whose instances are active objects, owning their own thread of control.

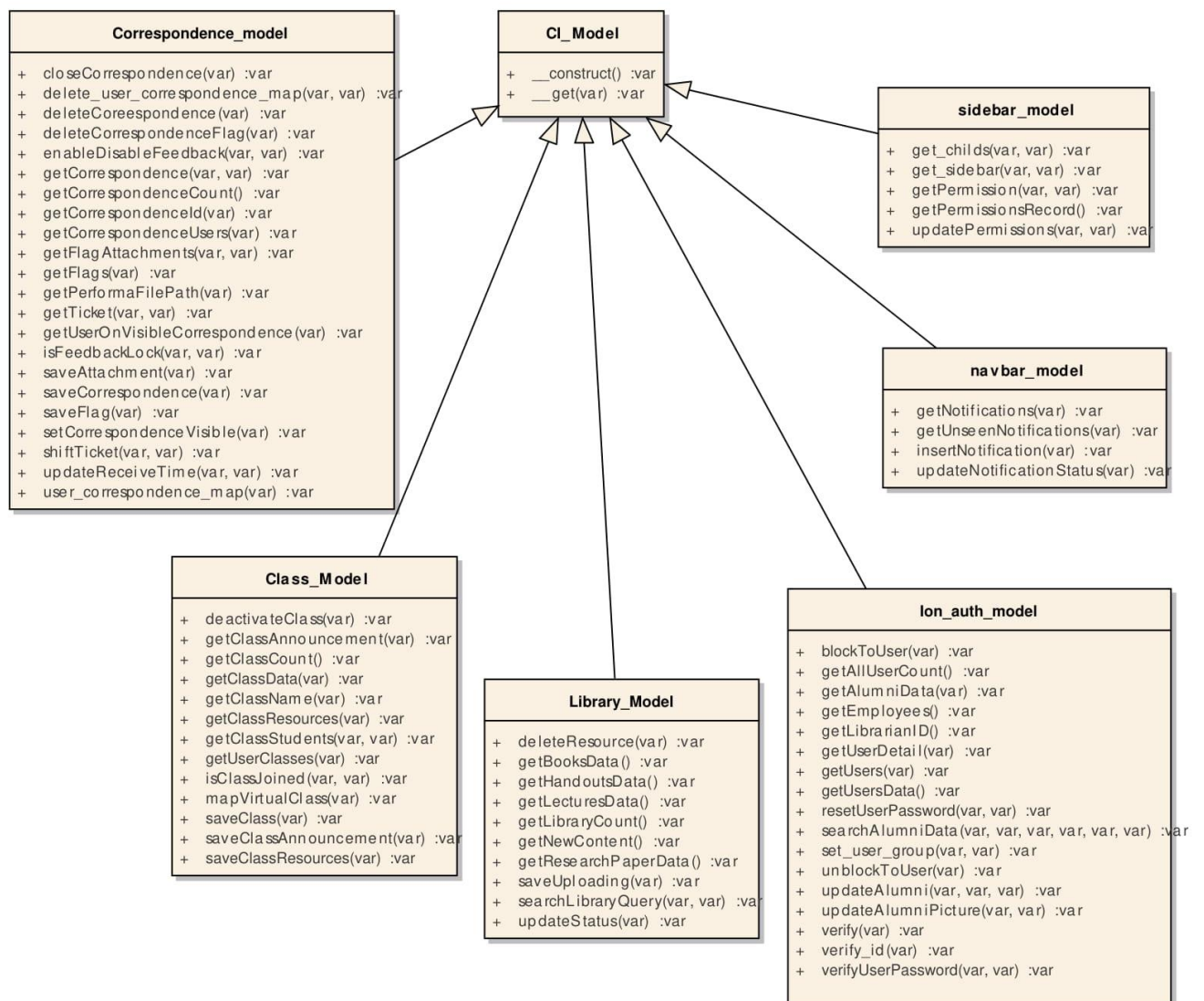


Figure 58: Class Diagram - 1

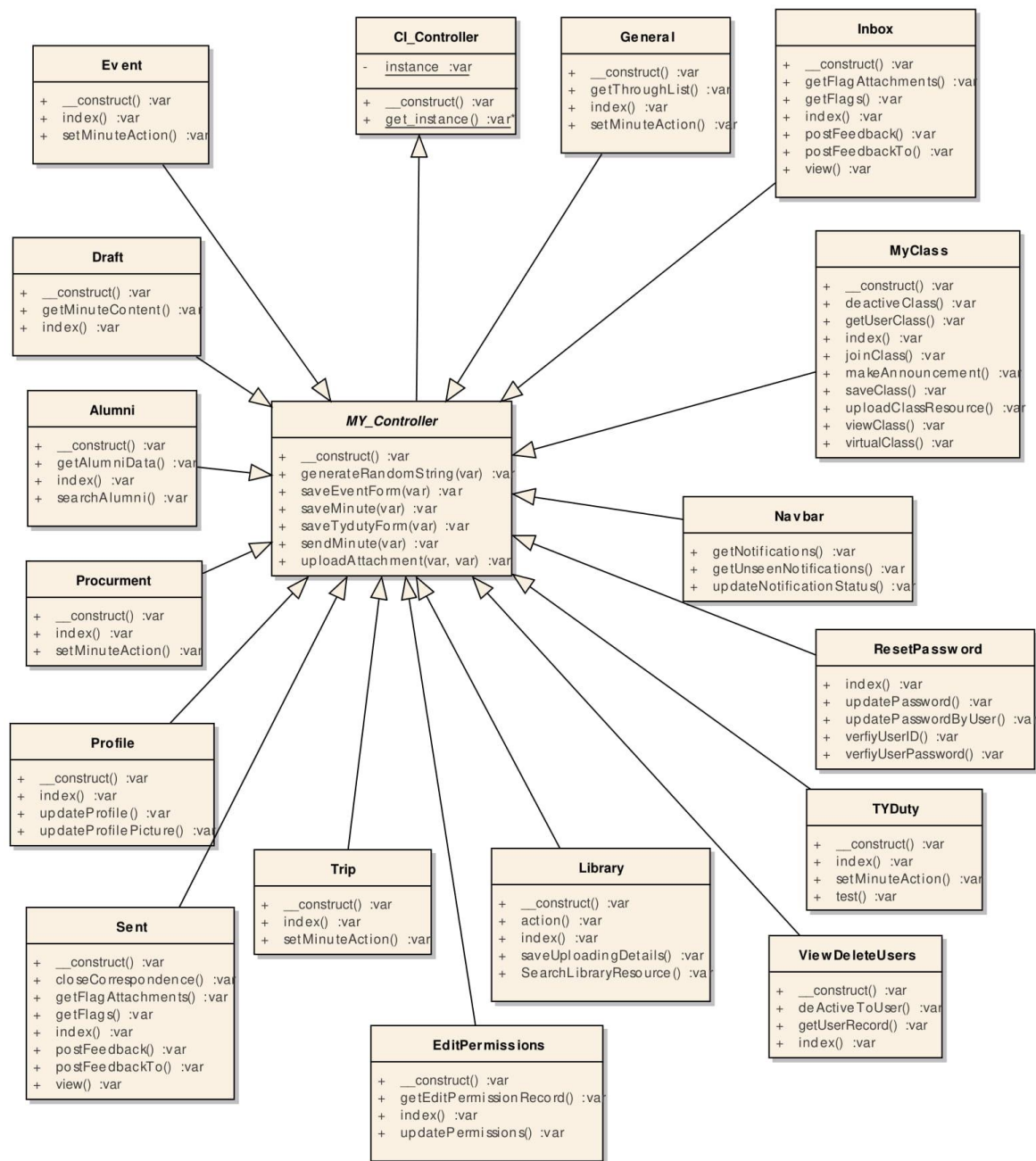


Figure 59: Class Diagram - 2

3.7.2 Internal Model

‘

Users

Table 29: User Internal Modal

Attribute	Type
ID (PK)	Int
User ID	Varchar(20)
Password	Varchar(50)
Name	Varchar(25)
Email	Varchar(20)
Type	Varchar(20)
Designation	Varchar(20)
Department	Varchar(20)
Active	Boolean

Privileges

Table 30: Privileges Internal Modal

Attribute	Type
ID (PK)	Int
Side_Bar id(fk)	Int
Group ID (FK)	int
Permissions	Varchar(10)

Side Bar**Table 31: Side Bar Internal Modal**

Attribute	Type
ID (PK)	Varchar(20)
Name	Varchar(25)
Order	int
Parent ID	int
Page Name	Varchar(20)
Controller	int
Sub_Controller	int
Category	Varchar(5)
Icon Name	Varchar(20)

Groups**Table 32: Groups Internal Modal**

Attribute	Type
ID (PK)	Varchar(20)
Name	Varchar(50)
Description	Varchar(25)

User Notification**Table 33: User Notification Internal Modal**

Attribute	Type
ID (PK)	int
User_id(fk)	Varchar(50)
Type	Varchar(20)
Date_Time	DateTime
Link	Varchar(20)
Description	Varchar(25)
Status	Boolean

Virtual Class

Table 34: Virtual Class Internal Modal

Attribute	Type
ID (PK)	int
Class_Code	Varchar(50)
Class_Name	Varchar(20)
Class_Desc	Varchar(20)
Creator	Varchar(50)
isActive	Boolean
Campus	Varchar(20)
Total_Assignments	int

User-Class Mapping

Table 35: User Class Mapping Internal Modal

Attribute	Type
ID (PK)	int
User_Id(FK)	Varchar(50)
Virtual_Class_id (fk)	int

Virtual Class Resource

Table 36: Virtual Class Resource Internal Modal

Attribute	Type
ID (PK)	int
Virtual_Class_id (fk)	int
Type	Varchar(20)
Path_Resource	VarChar(50)
Actual File Name	Varchar(50)
Random File Name	Varchar(50)

Virtual Class Announcement

Table 37: Virtual Class Announcement Internal Modal

Attribute	Type
ID (PK)	int
Virtual_Class_id (fk)	int
Description	Varchar(20)
Date Time	DateTime

Library

Table 38: Library Internal Modal

Attribute	Type
ID (PK)	int
Name	Varchar(50)
Type	Varchar(20)
Author	VarChar(20)
Faculty	VarChar(20)
Date_Time	DateTime
Path_File	VarChar(50)
Uploader	Varchar(13)
Status	Boolean

Alumni**Table 39: Alumni Internal Modal**

Attribute	Type
ID (PK)	int
User_Id(Fk)	Varchar(50)
Batch	Varchar(20)
Department	VarChar(20)
Degree_Level	VarChar(20)
Campus	VarChar(20)
Email	VarChar(20)
Current_Job	VarChar(20)
Official_Phone	VarChar(20)
Picture Name	Varchar(20)
Picture_path	VarChar(50)
Mobile	Varchar(20)

Correspondence**Table 40: Correspondence Internal Modal**

Attribute	Type
ID (PK)	int
Type	Varchar(50)
Subject	Varchar(20)
Path_File	Varchar(50)
Intiate_Date_Time	DateTime

User-Correspondence Mapping

Table 41: User Correspondence Mapping Internal Modal

Attribute	Type
ID (PK)	int
CS_Id(fk)	int
User_Id(fk)	Varchar(20)
Side_Bar_Id(fk)	int

Correspondence-Feedback

Table 42: Correspondence Feedback Internal Modal

Attribute	Type
ID (PK)	int
Parent_CS_Id	int
Path_File	Varchar(20)

Correspondence-Trail

Table 43: Correspondence Trail Internal Model

Attribute	Type
ID (PK)	int
User_id	Varchar(15)
Minute_Type	Varchar(20)
Minute_Number	int

CHAPTER 4

IMPLEMENTATION AND METHODOLOGY

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.

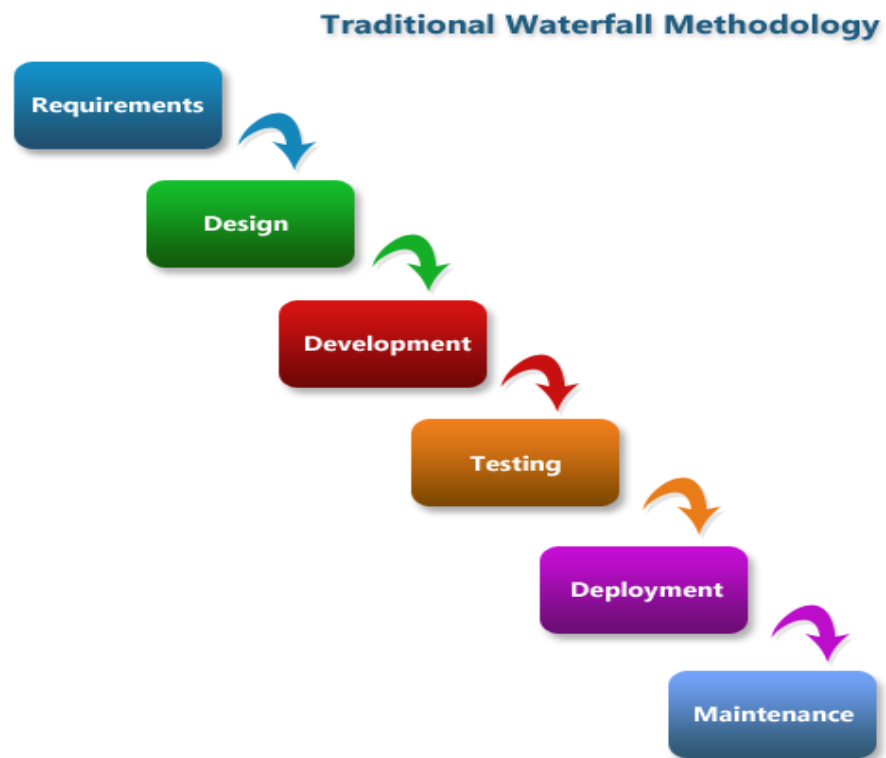


Figure 61: Waterfall Model

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.

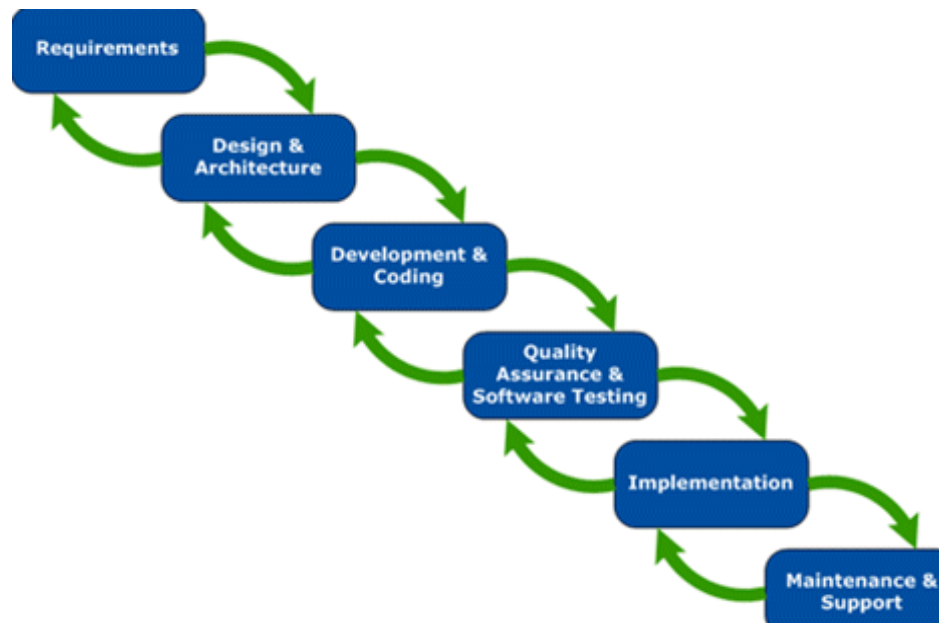


Figure 62: Agile Model

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.

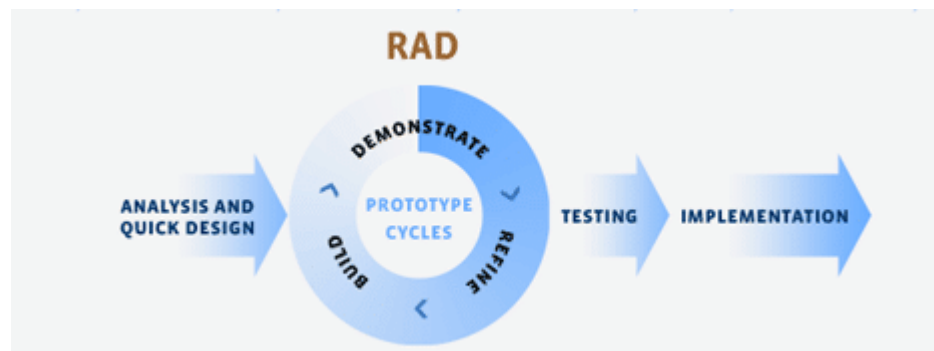


Figure 63: RAD Model

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.

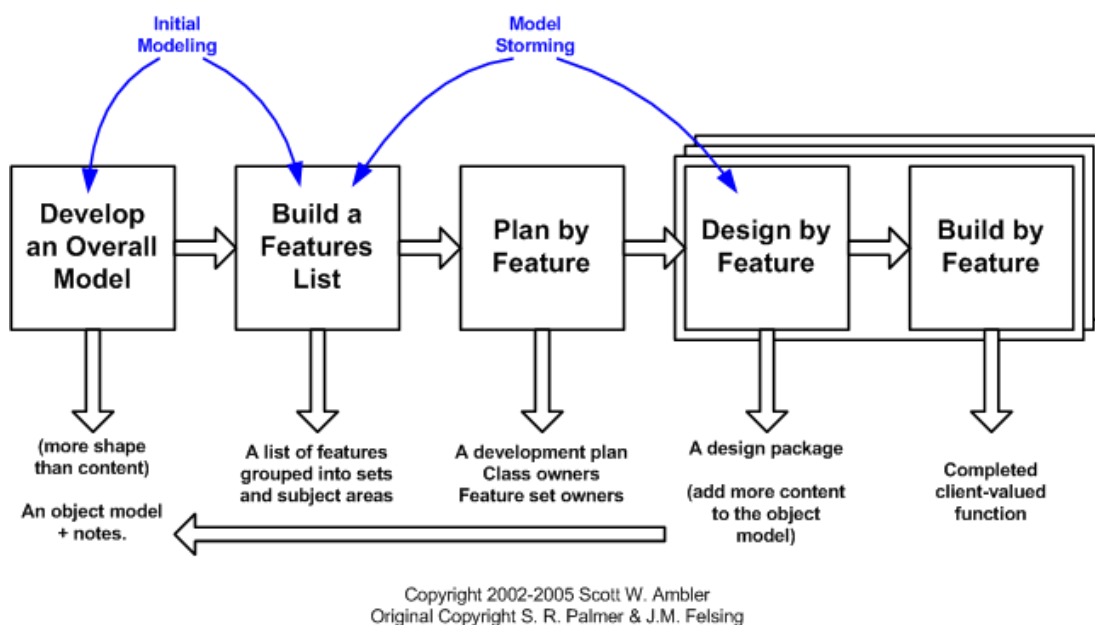


Figure 64: FDD Model

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, 3].

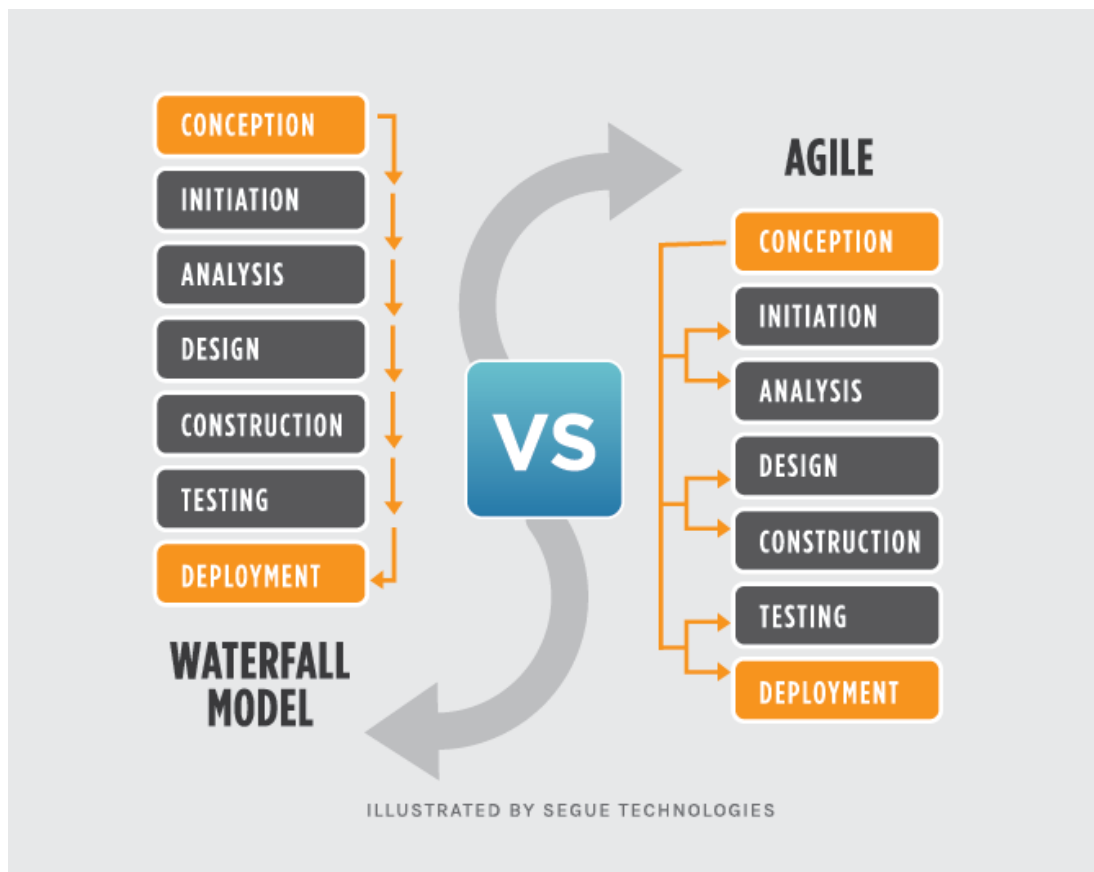


Figure 65: Waterfall vs. Agile

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 HOD Computer Science for the conformance of the requirements of Correspondence System. We show up some screen prototypes to the HOD and he point

out some changes in the requirement that causes extension in scope of correspondence system. We again made the requirement chart with changed requirements. This phase ends by making domain model and class diagram.

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 planed according to their dependencies between features in terms of classes involved, balancing load across class owners, and complexity of the features to be implemented.

4.1.3.4 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 [4, 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. In this section, we discuss the critical code of our application that we designed to full-fill the system requirement.

We design an algorithm that supports multiple flags having multiple attachments in each flag, which is a high priority requirement of correspondence system (see requirement chart CS-R1). Moreover, Codignitor (php framework that we followed) does not provide any functionality that supports multiple attachments. Following is the screen shot of code that we have made for multiple attachments.

```

$_FILES['upl'] = $_FILES[$flag];
$len= count($_FILES['upl']['name']);
for($i=0;$i<$len;$i++){
    $actualName = str_replace(' ', '', $_FILES['upl']['name'][$i]);
    $randomName = $this->generateRandomString().$actualName;
    $_FILES[$flag]['name'] = $randomName;
    $_FILES[$flag]['type'] = $_FILES['upl']['type'][$i];
    $_FILES[$flag]['tmp_name'] = $_FILES['upl']['tmp_name'][$i];
    $_FILES[$flag]['error'] = $_FILES['upl']['error'][$i];
    $_FILES[$flag]['size'] = $_FILES['upl']['size'][$i];
    $config['upload_path'] = './uploads/';
    $config['allowed_types'] = 'gif|jpg|png|pdf|docx|pptx';
    $config['max_size'] = 100000;
    $config['max_width'] = 5500;
    $config['max_height'] = 3500;
    $config['max_height'] = 3500;
    $this->load->library('upload', $config);
    if(!$this->upload->do_upload($flag)){
        $error = $this->upload->display_errors();
        return $error;
    }
}

```

Figure 66: Multi Upload Algorithm - 1

Above code uploads multiple attachments in one flag and for multiple flags there exists a loop that calls this function for each flag as shown below

```
for($i=0;$i<$count;$i++){  
    if($check == 'true'){  
        $check = $this->uploadAttachment('F/'.chr($ascii),$query);  
    }  
}
```

Figure 67: Multi Upload Algorithm - 2

4.2.2 Technical Aspect of the BOL

Since the Bahria One Link (BOL) is a web application, both web server and database management system are necessary. A high-level system architecture diagram is illustrated below. The system consist of front-end and back-end and is categorised into five layers.

4.2.3 Back-end Technologies

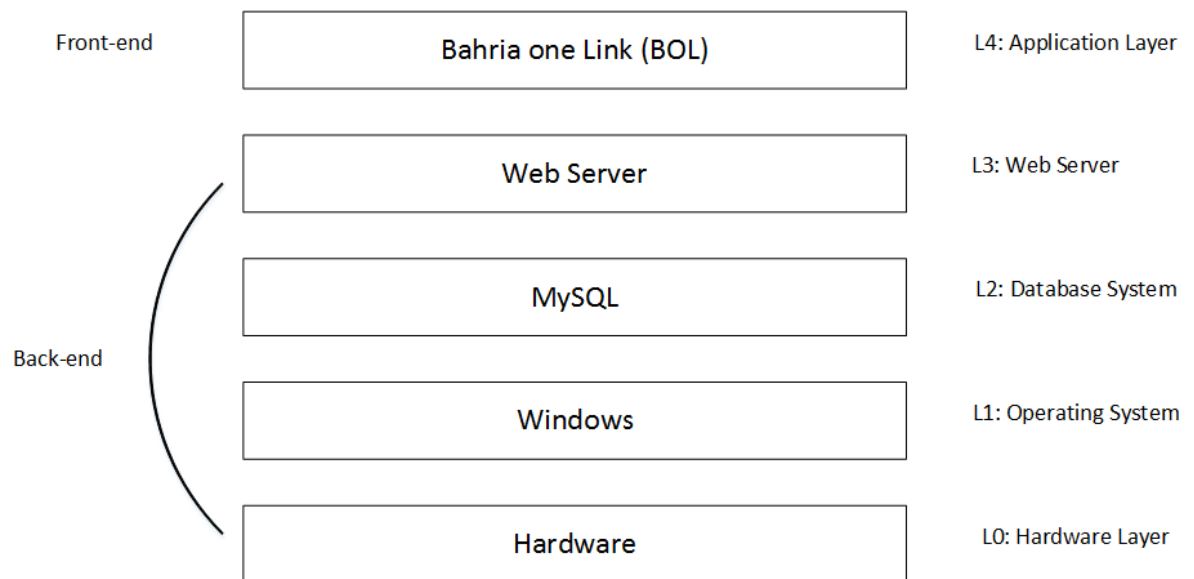


Figure 68: Application Architecture

The details of each layer in the architecture above figure discussed below.

L0: Hardware layer contains necessary hardware (Personal Computer) for the BOL to deploy.

L1: BOL will deploy under the Windows operating system, which operated above the hardware layer

L2: MySQL is chosen as the database system. It is relational database system that is renowned for its reliability and data integrity.

L3: The BOL uses the code-Ignitor web framework as its back-end system.

4.2.4 Front-end Technologies

Table 44: Front End Technologies

Technology	Major functions on a web page
HTML	Outline the structure of a web, it serves as the building blocks
CSS	Set the visual style to provide a better user interface
JavaScript/JQuery	Provide interactive elements

4.2.5 Basic structure of user Interface

As this application is developing for a university so, the interface of pages should be look like theme of Content management system (CMS). Following is the basic structure of user interface.

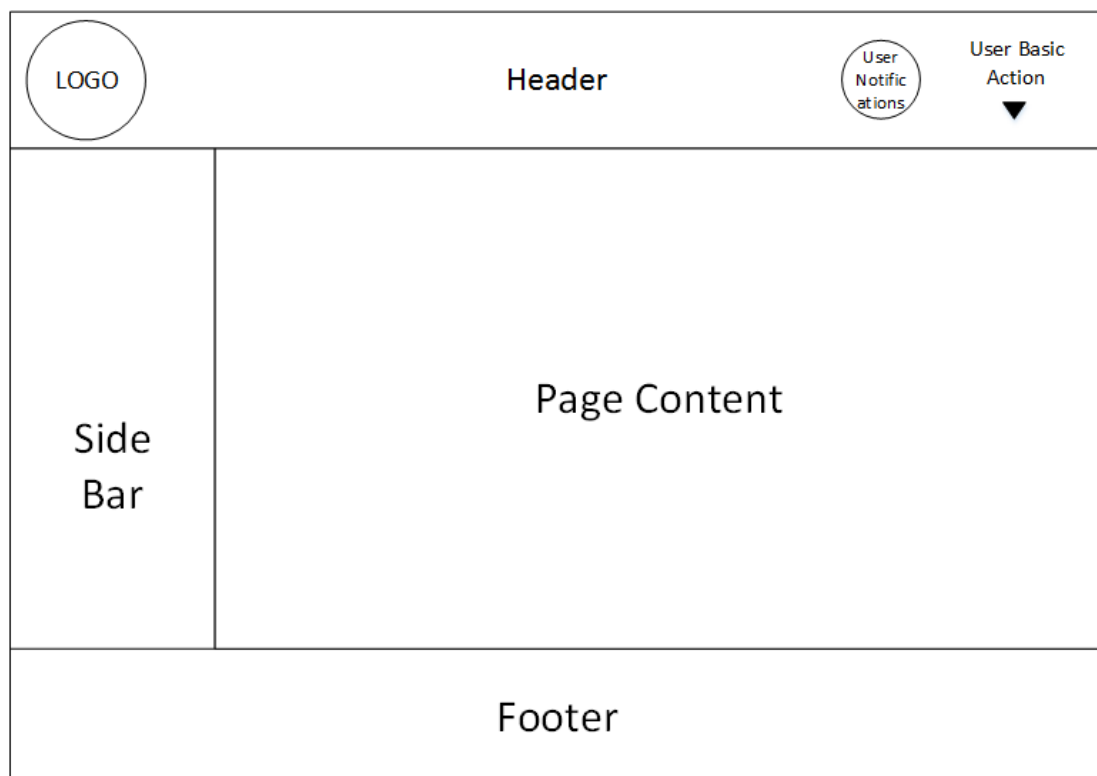


Figure 69: User Interface Structure

4.2.6 Understand the flow of developed system

This section describes the flow of events or user actions in Bahria One Link. User could understand all the valid and invalid actions in BOL application. This section includes general flow of BOL application and module wise flow also.

4.2.6.1 Bahria One Link (Complete)

When user access Bahria One Link login screen will show up. Registered user enter the credentials and non-registered user go to the registration screen. After login, user will redirect to the authorized home screen and after registration user will redirect to the login screen. Following flow chart explains the general flow of system.

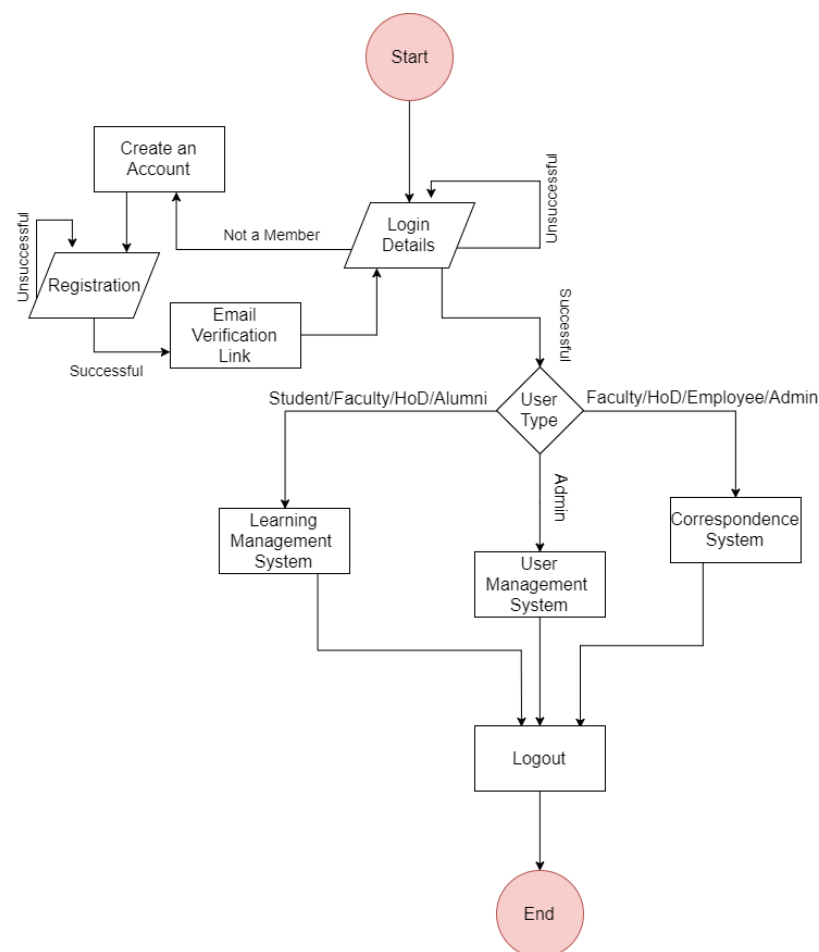


Figure 70: Flow Chart BOL

4.2.6.2 User Management System (Module 1)

Admin can perform number of actions in user management system (UMS). After login admin have to click on user management system to access it (see section 5.1.4). Admin can view virtual class, library, users and correspondence statistics on UMS home and navigate to any other screen to perform other actions on user and user groups as shown in below flow chart and sub-flowcharts.

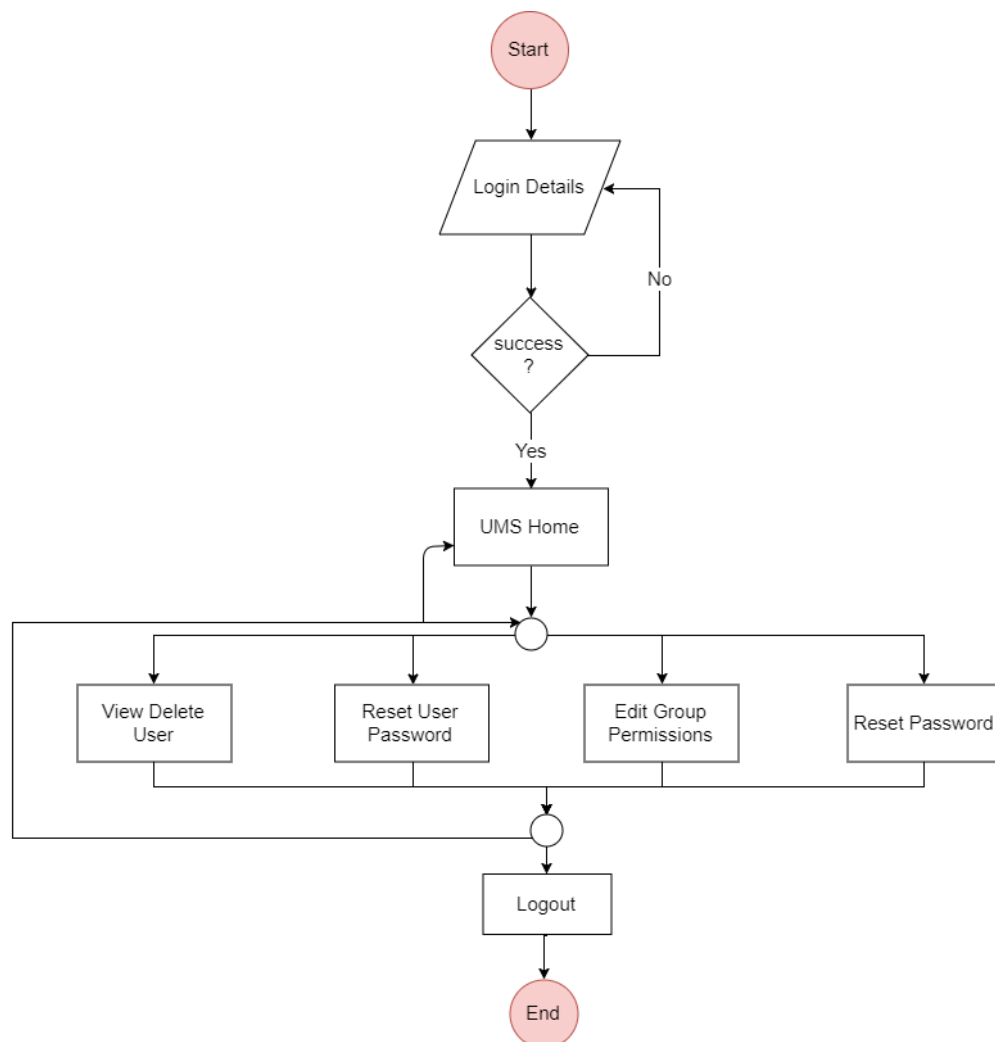


Figure 71: Flow Chart User Management System

4.2.6.2.1 UMS – View Delete User

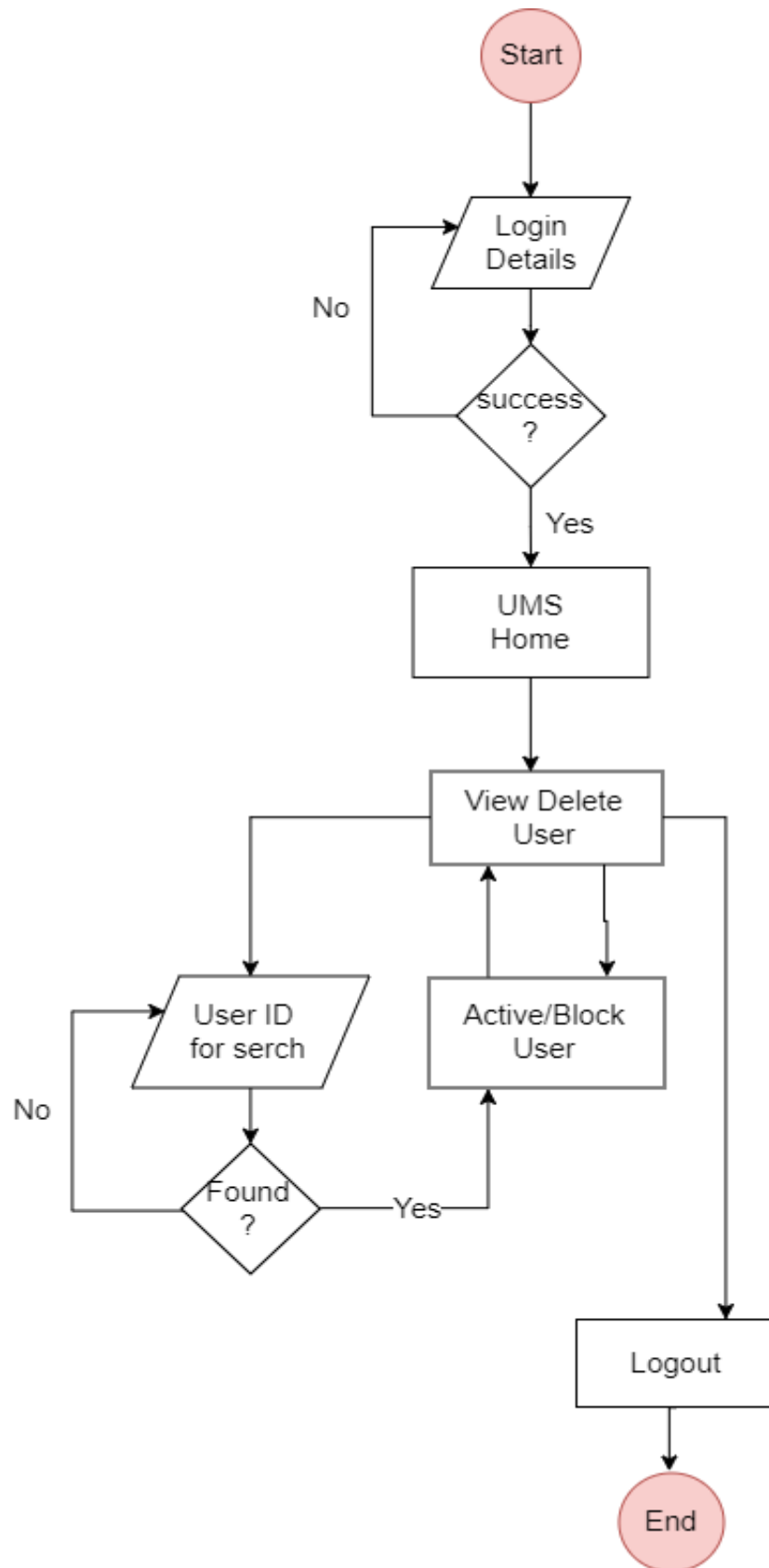


Figure 72: Flow Chart View Delete User UMS

4.2.6.2.2 UMS – Edit Group Permissions

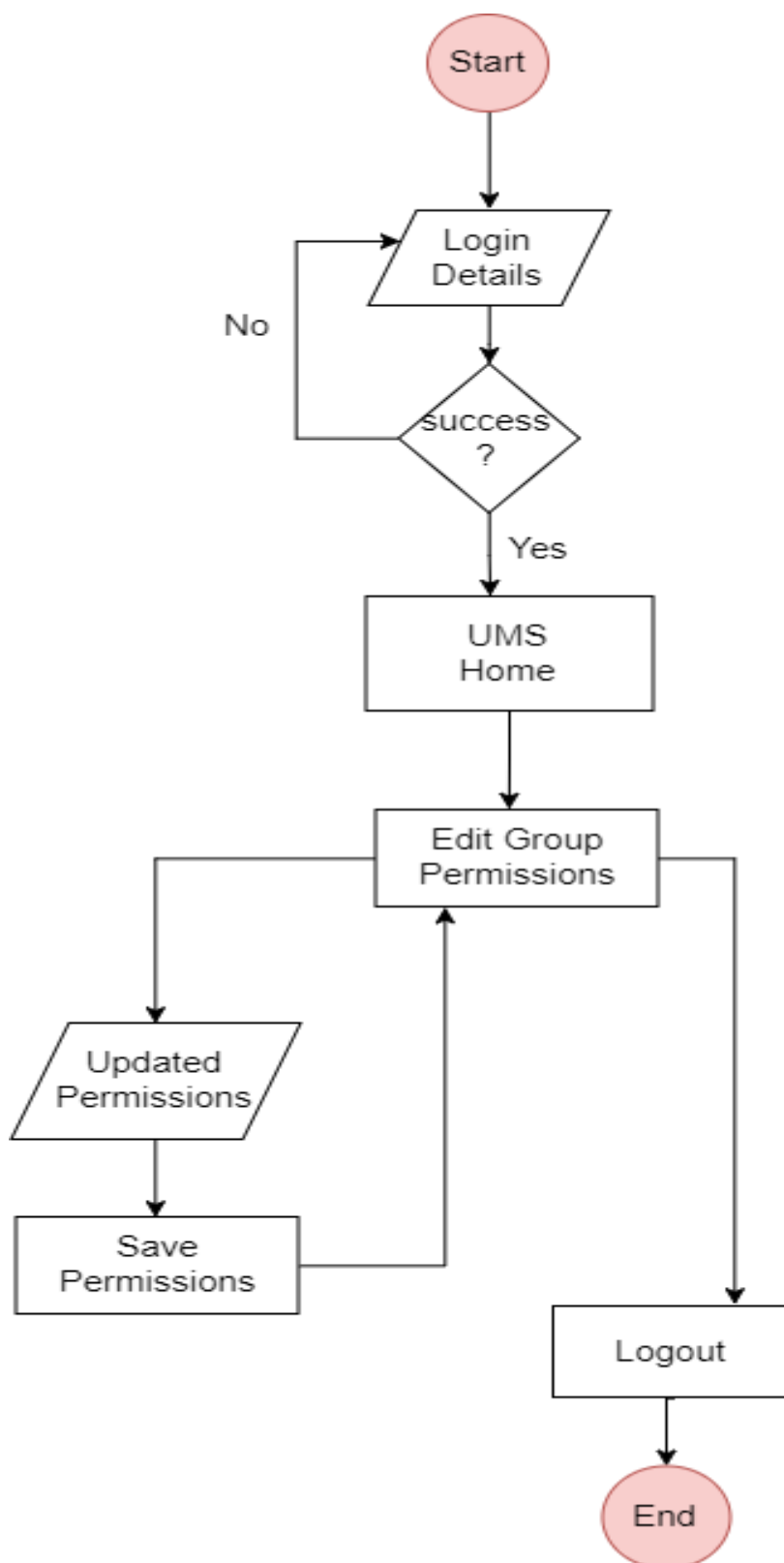


Figure 73: Flow Chart Edit Group Permissions UMS

4.2.6.2.3 UMS – Reset User Password

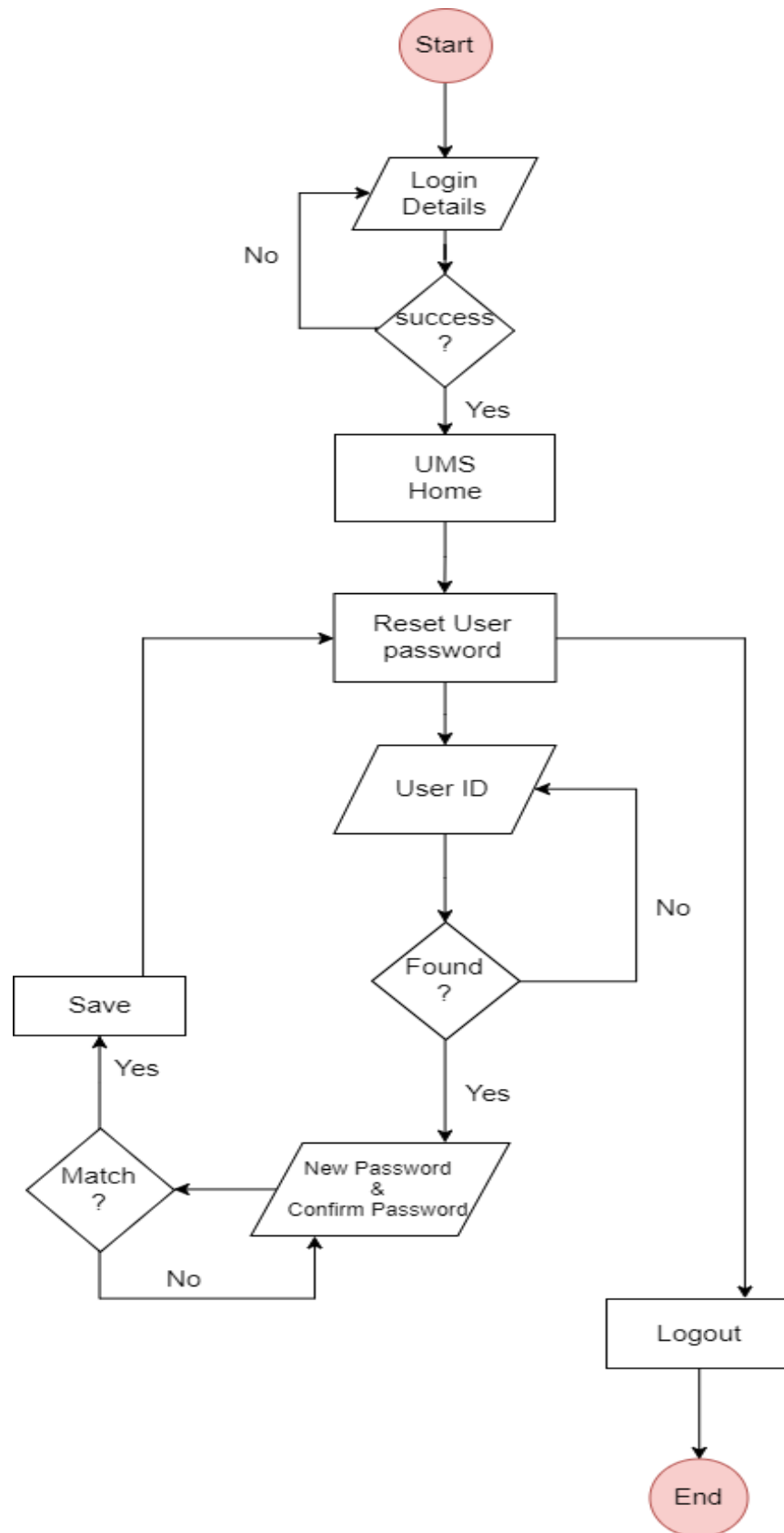


Figure 74: Flow Chart Reset User Password UMS

4.2.6.2.4 Reset Password

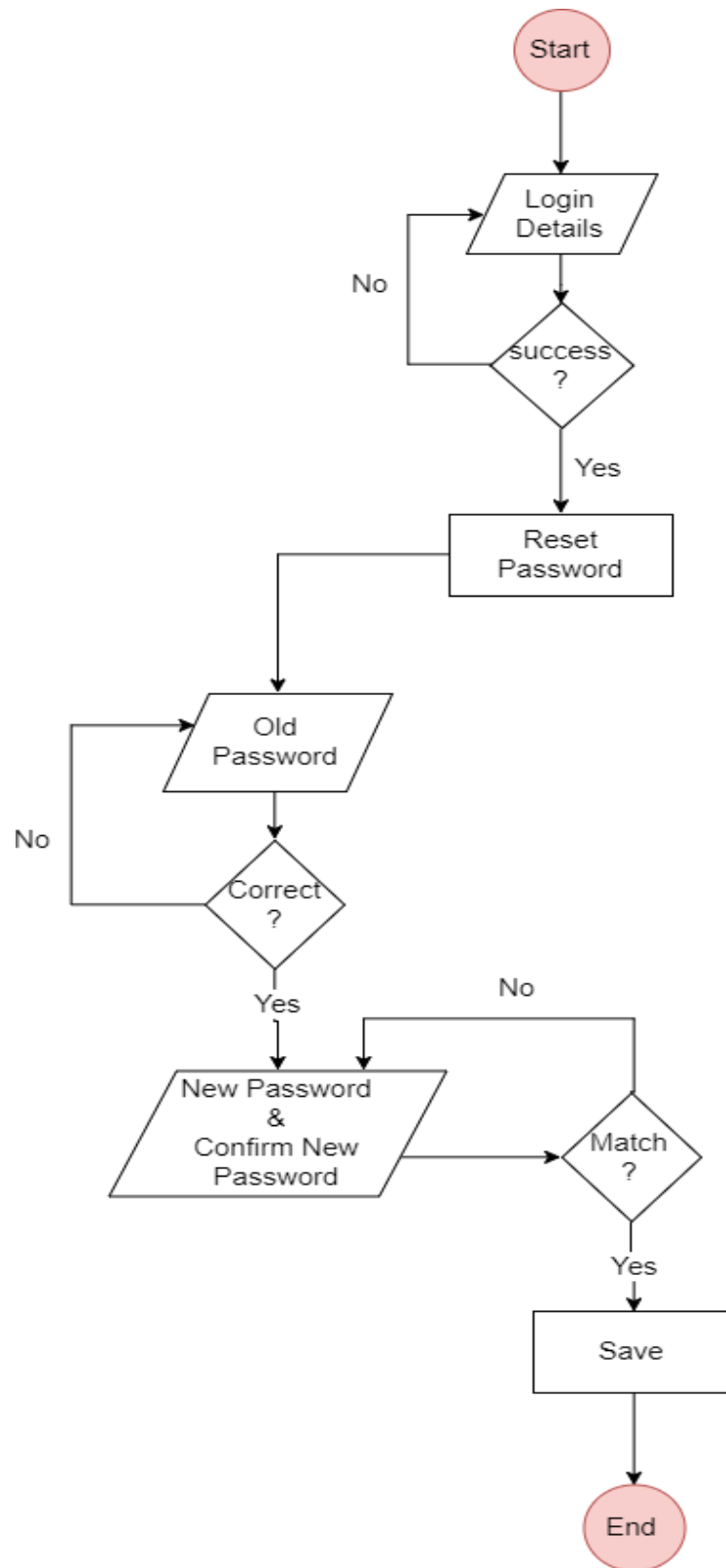


Figure 75: Flow Chart Reset Password

4.2.6.3 Learning Management System (Module 2)

Student, Alumni and faculty can perform number of actions in learning management system (LMS) mostly related to learning. After login faculty have to click on learning management system to access it (see section 5.1.1). Alumni can also maintain its profile in LMS. All user can navigate to any authorized screen from home screen to perform different actions as shown in below flow chart and sub-flowcharts.

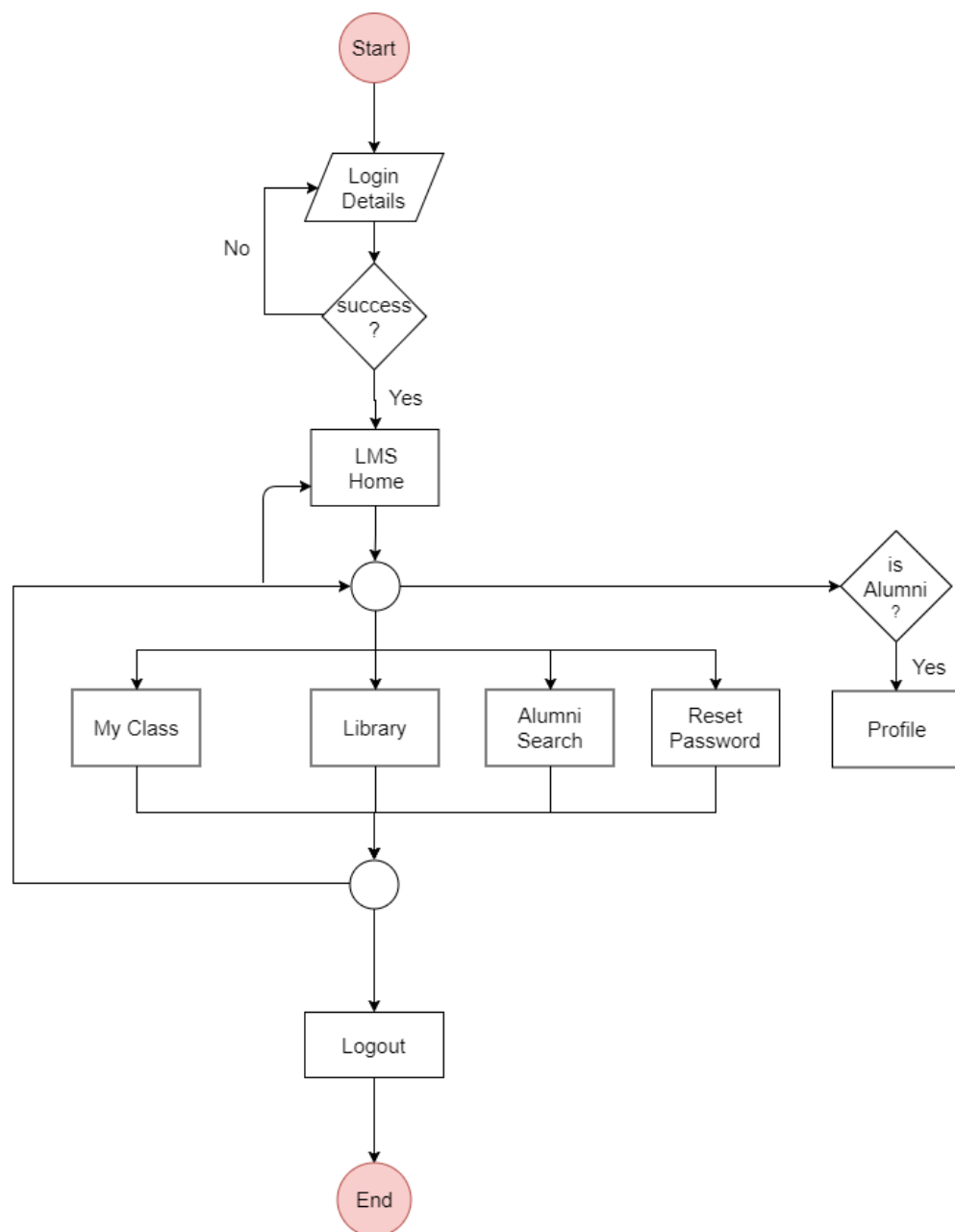


Figure 76: Flow Chart Learning Management System

4.2.6.3.1 LMS – My Class

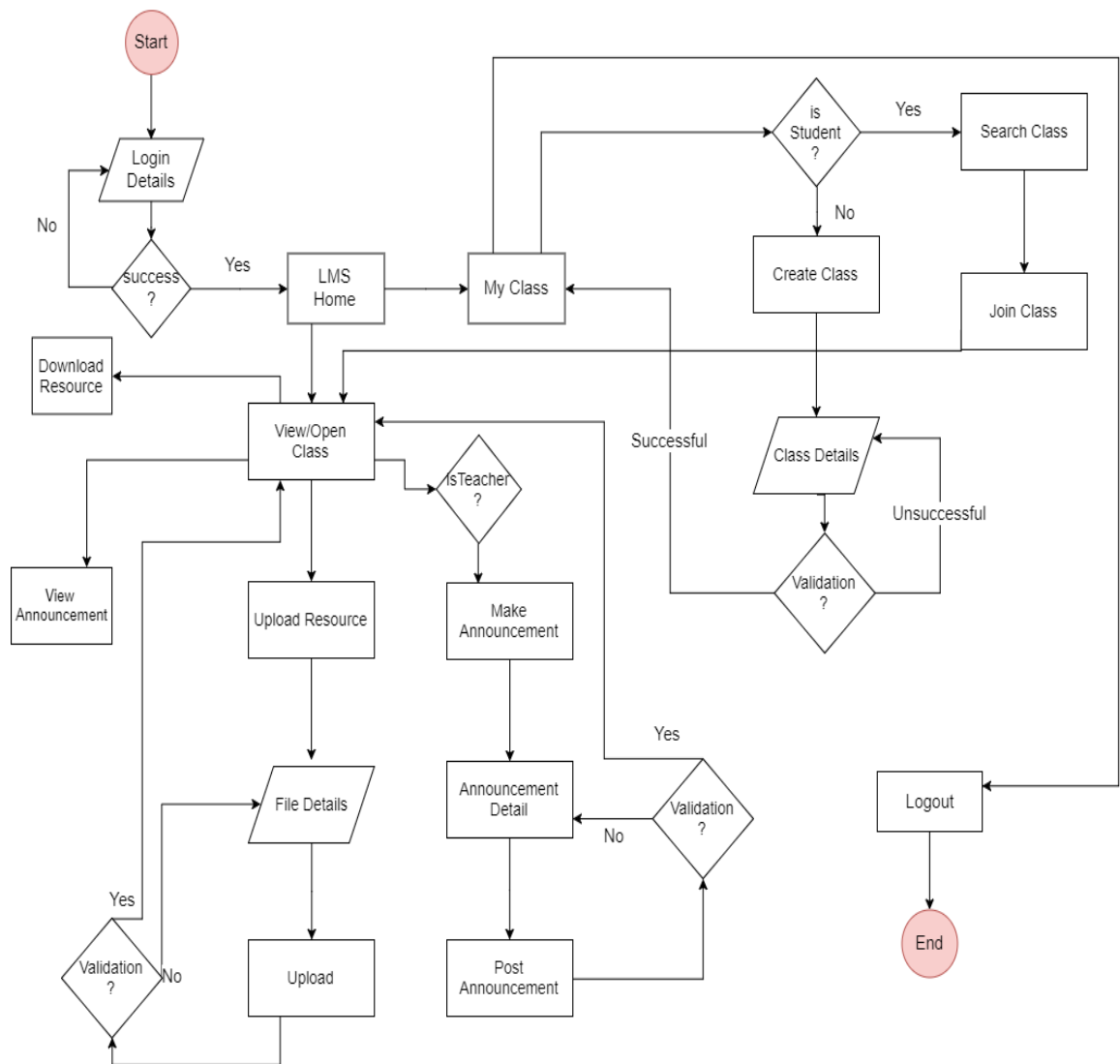


Figure 77: Flow Chart My Class LMS

4.2.6.3.2 LMS – Library

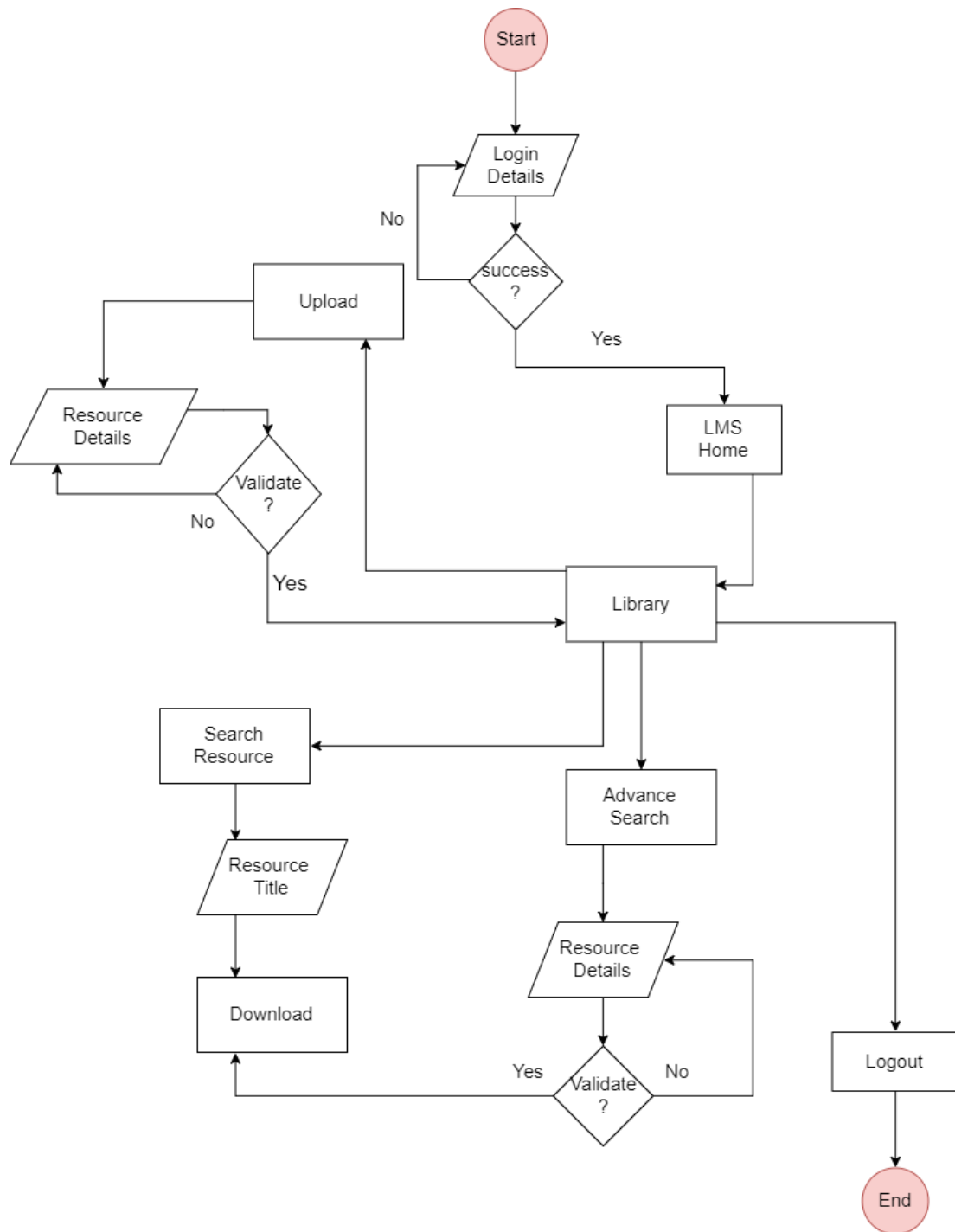


Figure 78: Flow Chart Library LMS

4.2.6.3.3 LMS – Alumni Search

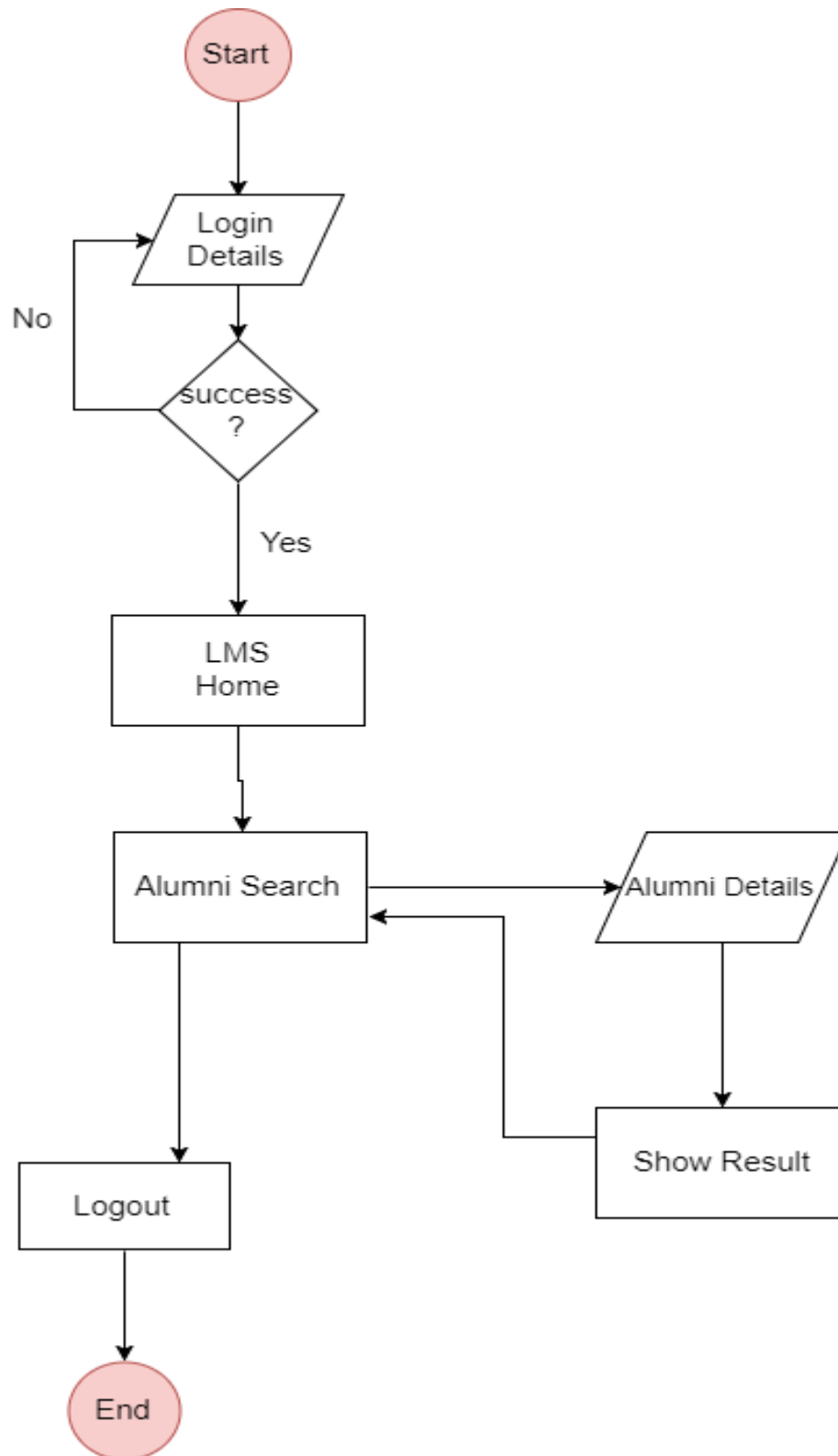


Figure 79: Flow Chart Alumni Search LMS

4.2.6.3.4 Reset Password

See Section 4.2.6.2.4

4.2.6.3.5 LMS – Profile

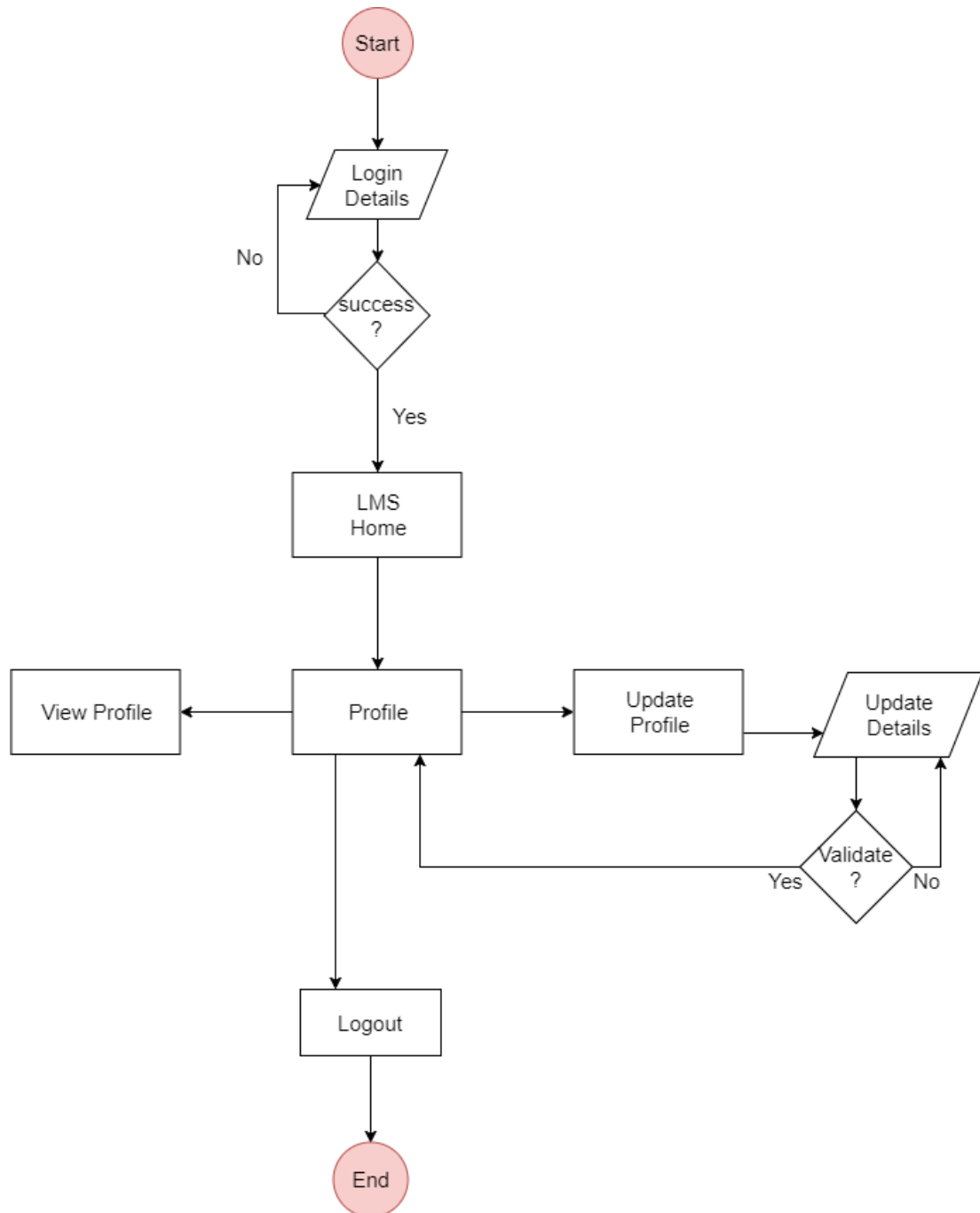


Figure 80: Flow Chart Profile LMS

4.2.6.4 Correspondence System (module 3)

Employees of Bahria can perform actions related to internal communication. After login faculty have to click on correspondence system to access it (see section 5.1.4). User can navigate to any authorized screen from home screen to perform different actions as shown in below flow chart and sub-flowcharts.

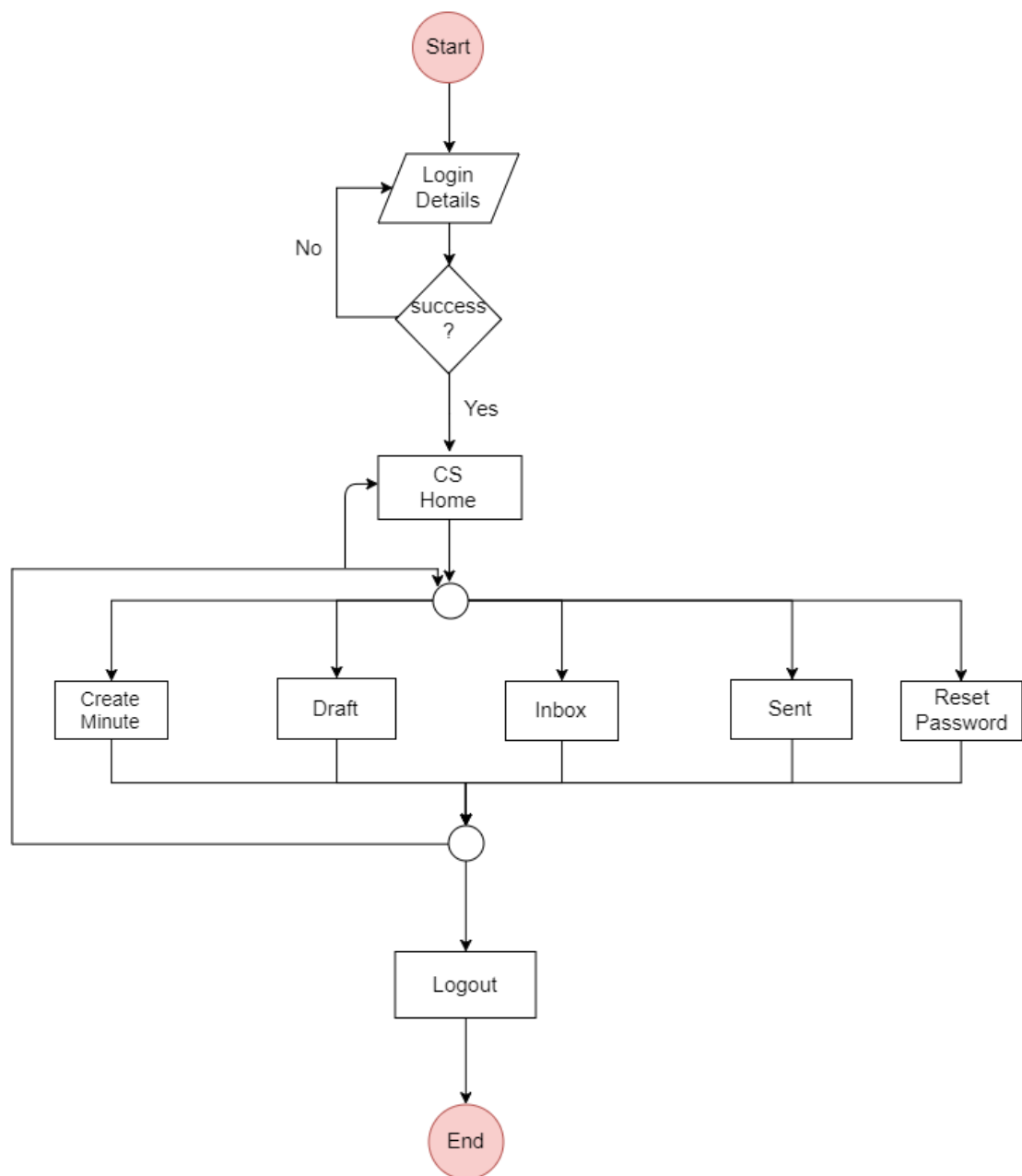


Figure 81: Flow Chart Correspondence System

4.2.6.4.1 CS – Create Minute

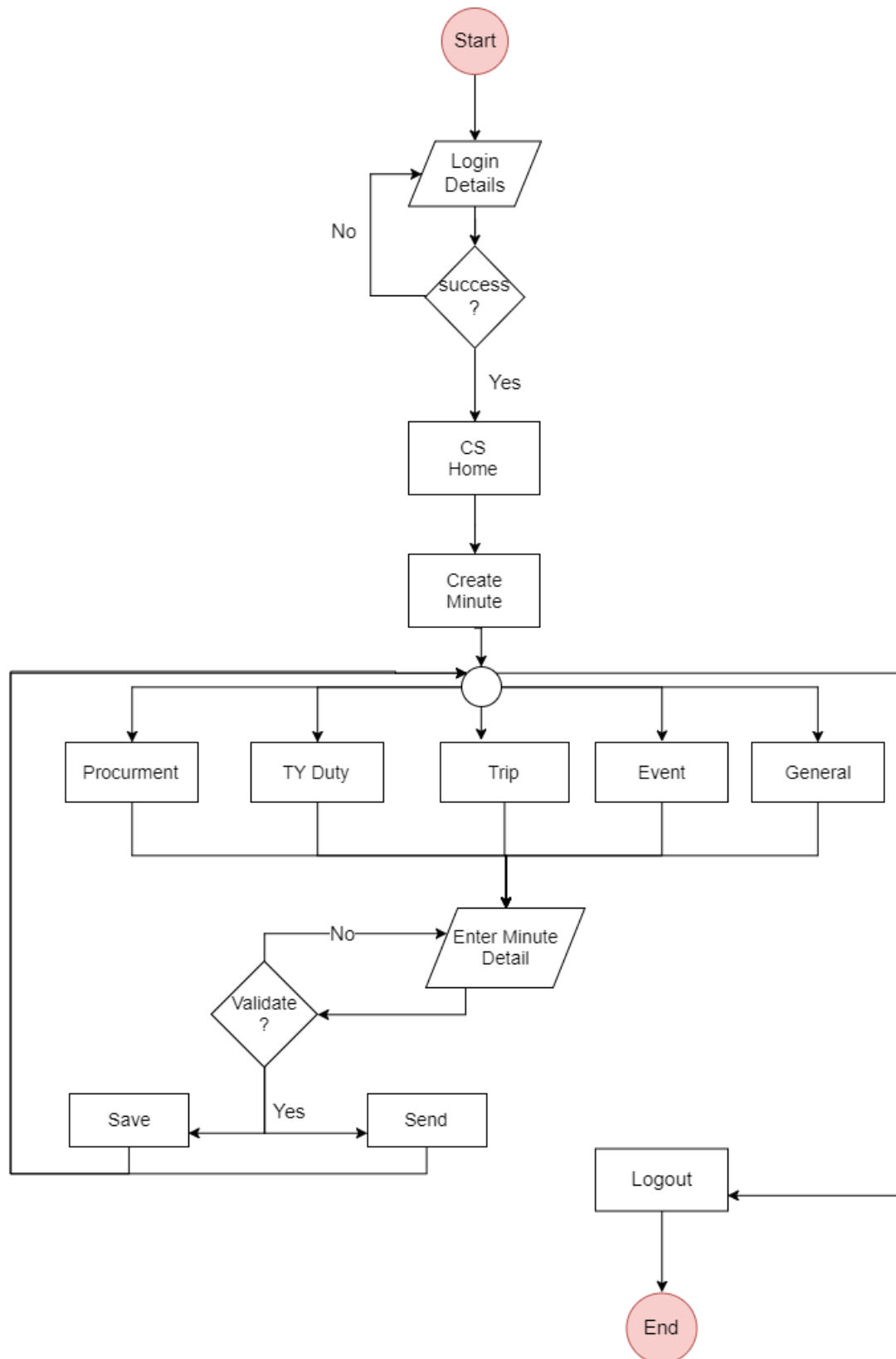


Figure 82: Flow Chart Create Minute CS

4.2.6.4.2 CS – Draft

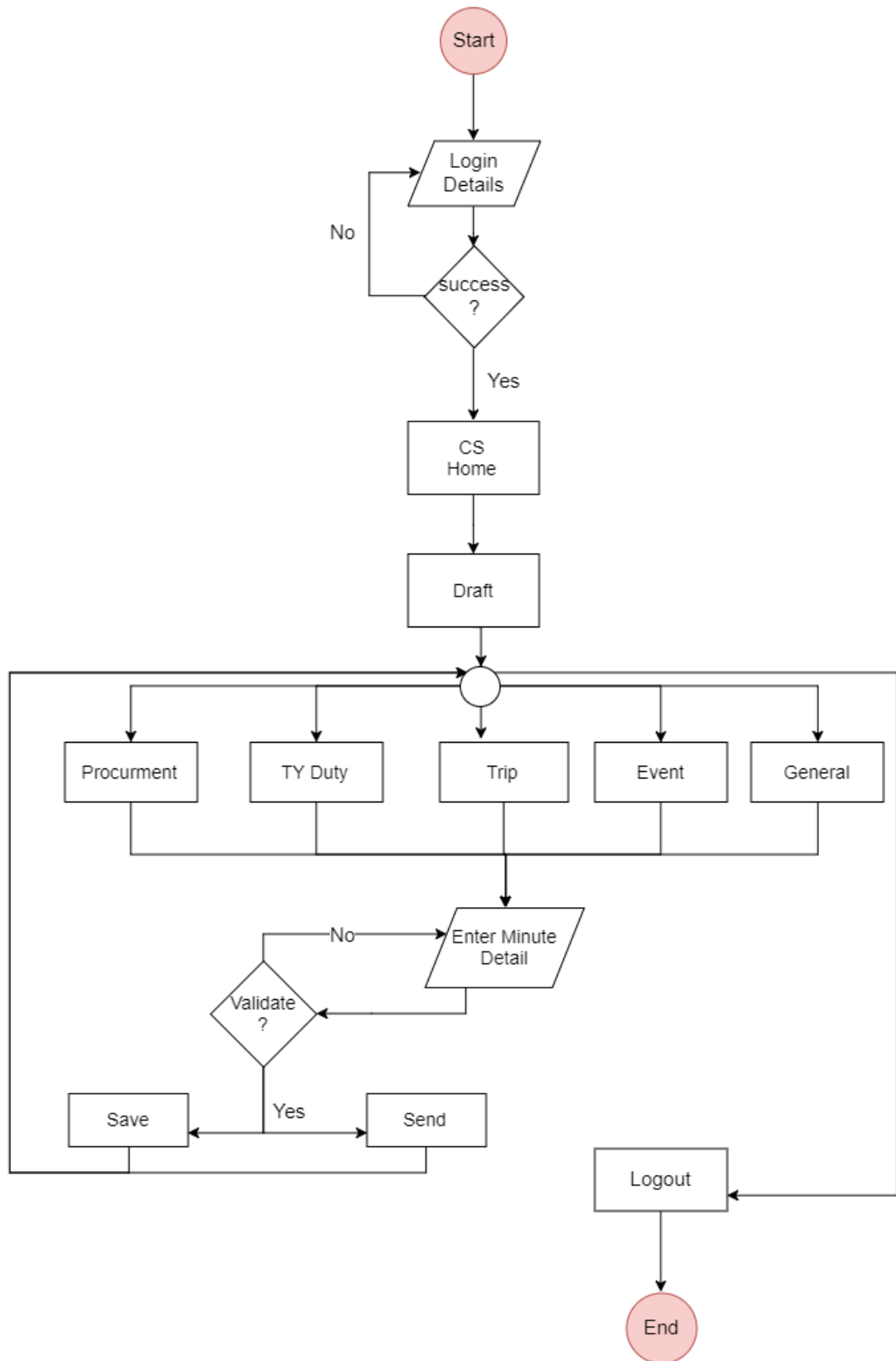


Figure 83: Flow Chart Draft CS

4.2.6.4.3 CS – Inbox

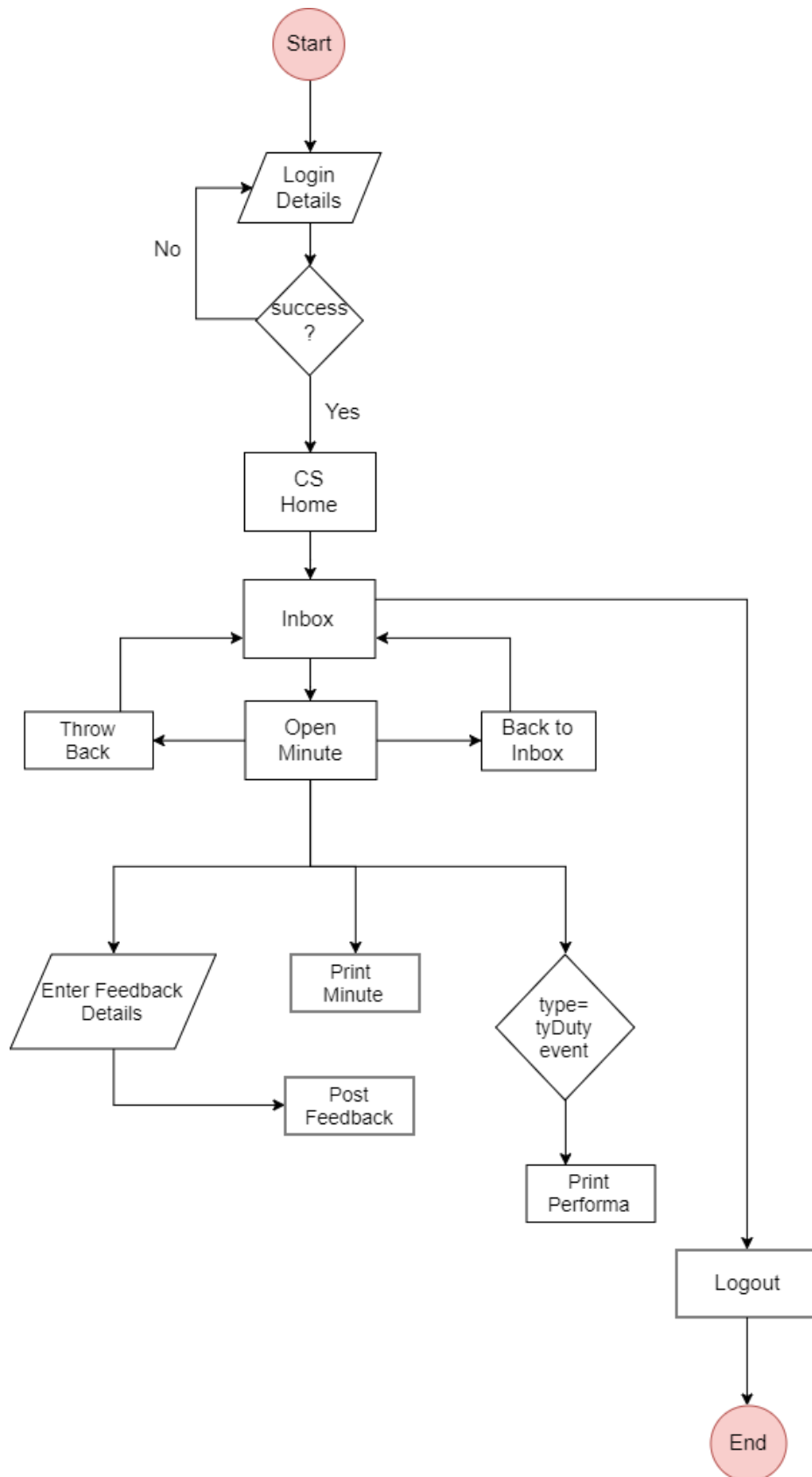


Figure 84: Flow Chart Inbox CS

4.2.6.4.4 CS – Sent

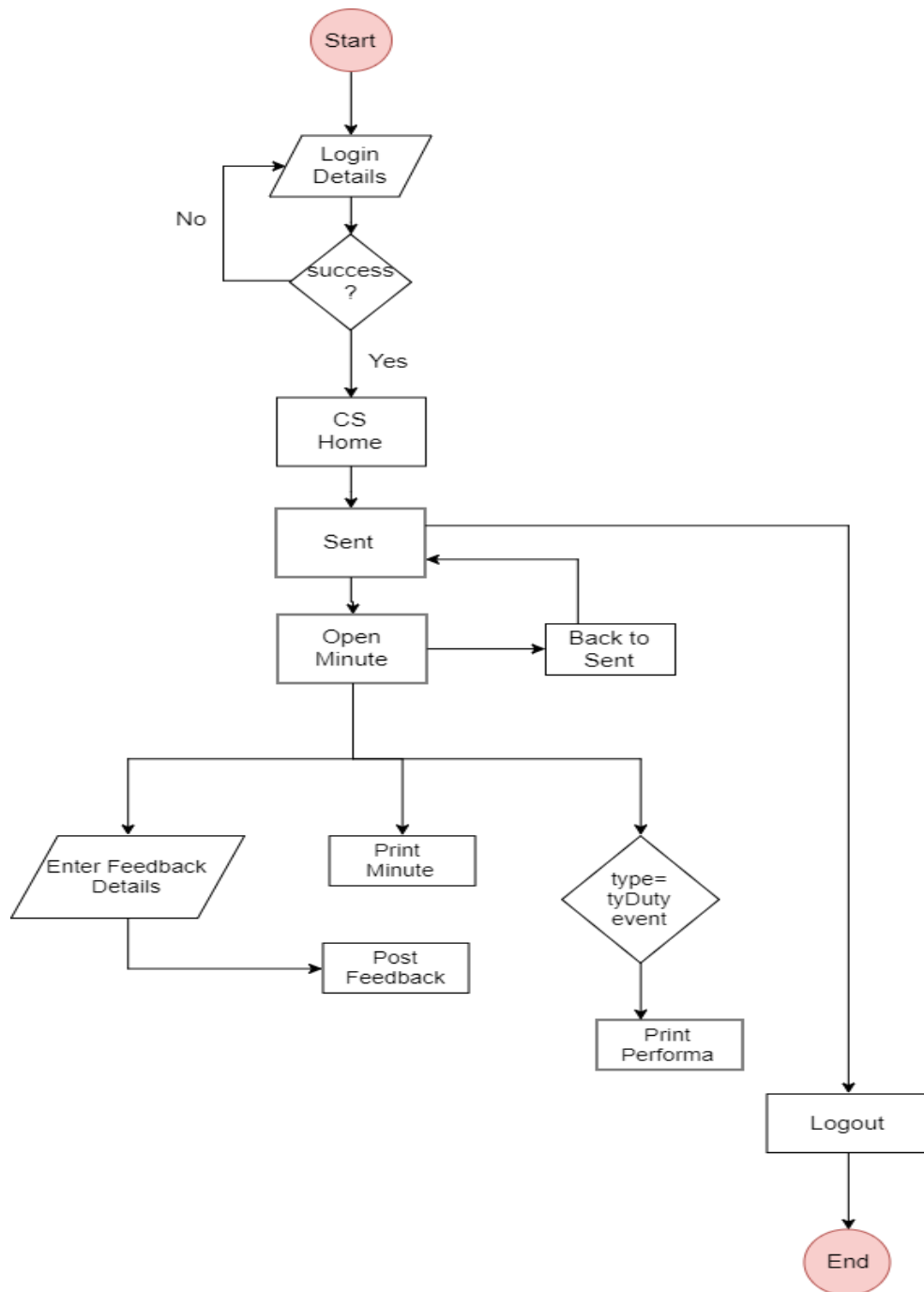


Figure 85: Flow Chart Sent CS

4.2.6.4.5 Reset Password

See Section 4.2.6.2.4

CHAPTER 5

USER MANUAL

5.1 Getting started

5.1.1 User Access, Roles and Privileges

The users of the application are the employee and student of the Bahria University, each user related to a specific group, and each group has some privileges [6].

The table below explains the different levels of the access rights and corresponding explanation for each group. There are three type of systems to access in BOL application.

Table 45: Module Authorization and Privileges

Type of System	Authorized Personnel and their rights
Learning Management System	<p>Faculty that can create virtual class, upload study related material in both e-library and virtual class, make announcement in virtual class, and search for Alumni.</p> <p>Student that can join virtual class, upload study related material in both e-library and virtual class and search for Alumni.</p>

	<p>Alumni that have all the privileges like Student but also have an Alumni Profile to maintain.</p> <p>Librarian that have access to the library and is responsible for grant permission to new uploaded resources.</p>
Correspondence System	<p>It used by all employees of Bahria University that can send, receive, view and print correspondence according to their privileges. Employees includes:</p> <ul style="list-style-type: none"> • Director • Deputy Director • HODs • Faculty • Student Advisors • Account officer • Librarian • Examination Department • Finance Officer • Procurement Officer • Admission officer • PO SRC • Network Engineer
User Management System	<p>System Engineer who is administrator of Bahria One Link that can reset passwords on users' requests, edit permissions and view all details of users uses it.</p>

5.1.2 How to access BOL

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

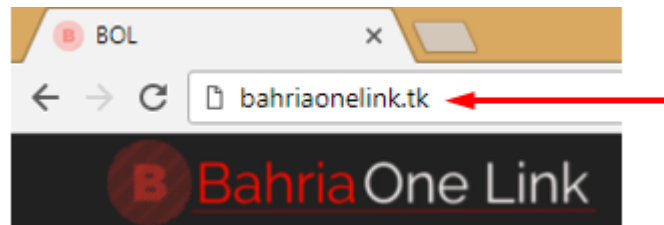


Figure 86: How to Access BOL

5.1.3 Login & Registration at BOL

After accessing BOL, Login screen (as shown below) will display.

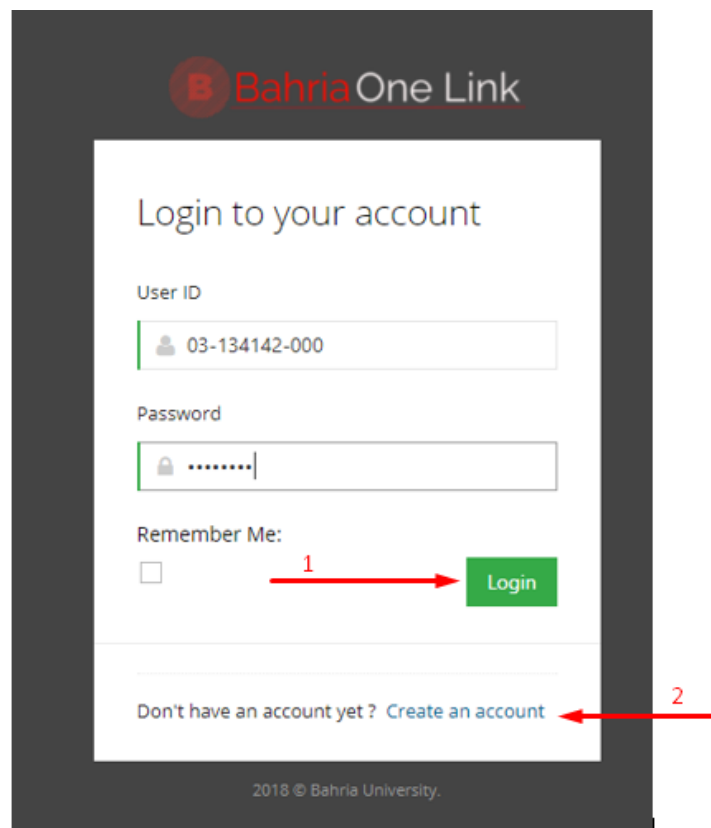


Figure 87: Login Screen

- 1) Enter login credentials if you are already registered at BOL and click on Login button
- 2) Click on 'Create an account' to registered at BOL. After click on 'Create an account' registration form will show as in below figure

Figure 88: SignUp Screen

- 1) Fill the registration form by providing university identity number (i.e. Student Enrolment /Employee ID) and a new password of minimum 8-character length then 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

- 2) By click on 'Back' button, you will go back to Login screen

5.1.4 Role based authorized home screens

As discussed in section 5.1.1 there are roles of every user. Following are home screen of each user type

- 1) Student will go to the home screen of Learning Management System (LMS) after successful login that will be discussed thoroughly in section 5.2
- 2) Employee of Bahria university other than the faculty and system engineer will go to the home screen of Correspondence System (CS) after successful login that will be discussed thoroughly in section 5.3
- 3) Faculty of Bahria University will go to a screen that have choice between LMS and CS as shown in below figure

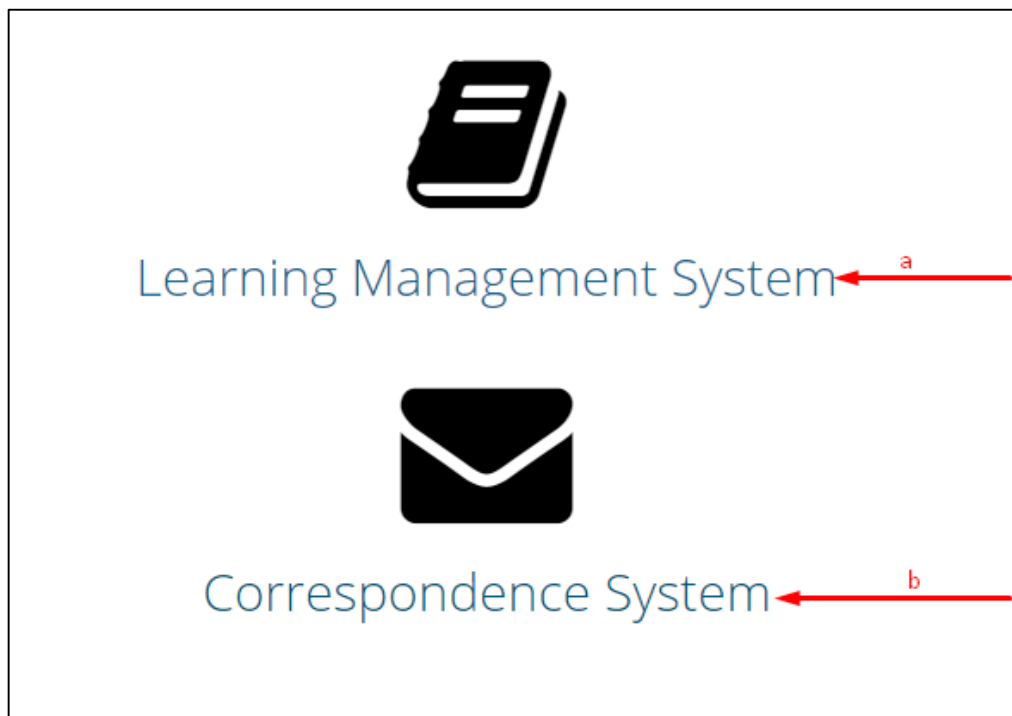


Figure 89: Home Screen for Faculty and HODs

- a) By click on 'Learning Management System' faculty member will redirect to 'LMS home' that will further explain in section 5.2
 - b) By click on 'Correspondence System' faculty member will redirect to 'CS home' that will further explain in section 5.3
- 4) System Engineer of Bahria University will go to a screen that have choice between User Management System (UMS) and Correspondence System (CS) as shown in below figure

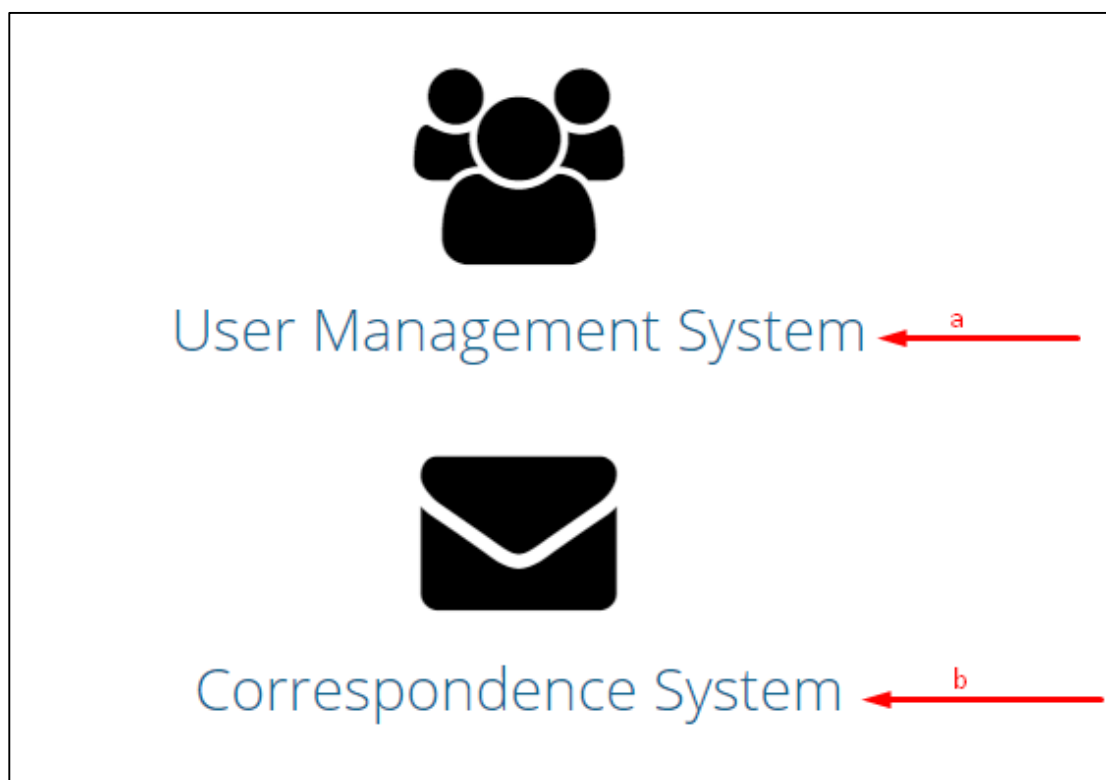


Figure 90: Home Scree for Admin

- a) By click on 'User Management System' system engineer will redirect to 'UMS home' that will further explain in section 5.4
- b) By click on 'Correspondence System' system engineer will redirect to 'CS home' that will further explain in section 5.3

5.1.5 Introduction to Navigation Bar

Navigation bar contains the user notification list and user basic functionalities that includes ‘Reset Password’ and ‘Logout’ as shown in below figure.

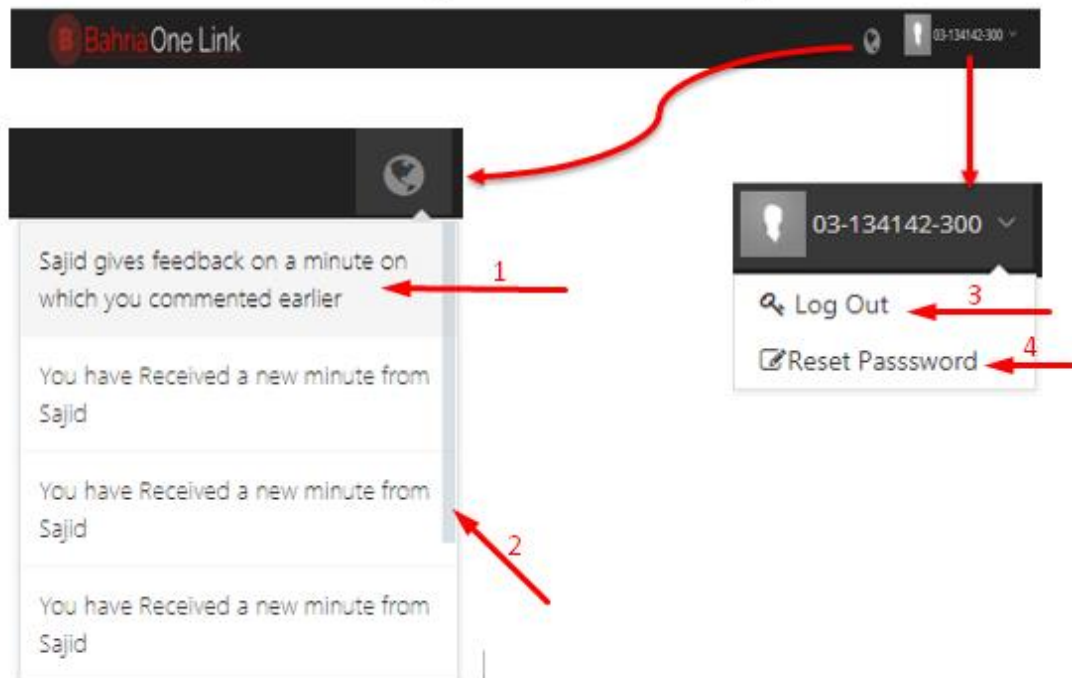


Figure 91: Introduction to Navigation Bar

- 1) By click on the notification it will redirect to the related page e.g. Inbox, Sent etc.
- 2) When the notification list exceeds the height of notification window a scroll bar helps to scroll between notifications
- 3) By click on ‘Logout’ user will redirect to ‘Login’ page and required to re-logged in to access the system
- 4) Reset password option helps user to renew the password by giving old and new passwords as shown in below figure

Old Password *

Old Password

New Password *

Enter New Password

Confirm New Password *

Enter Confirm New Password

Reset Password

a

Figure 92: Reset Password by User

- a) By click on 'Reset Password' button new password updates the old password and from now user will logged in by entering the new password

5.2 Learning Management System (LMS)

As discussed in section 5.1.1 Learning Management System is used by faculty, student and alumni and in section 5.1.4 all authorized home screens are also discussed. Now in this section usage of Learning Management System will be discussed that starts from 'Home' Screen as shown in below figure.

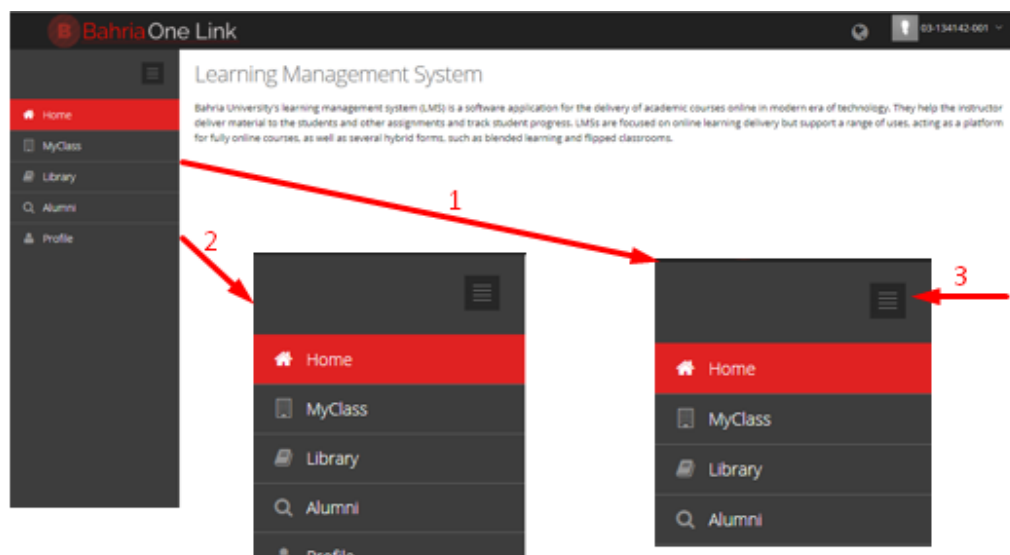


Figure 93: LMS Home Screen

- 1) This sidebar shows to the user of type Student and faculty and both can visit to any page given in sidebar by click on it, for example if student wants to view his/her classes or use any feature (according to privileges) related to class just click on 'My Class'
- 2) This sidebar shows to the user of type Alumni. All the things are same as in sidebar show to student and alumni but alumni can also maintain his/her profile
- 3) This toggle button use to shrink the sidebar as shown in below figure

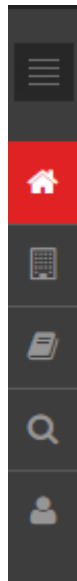


Figure 94: Shrink Side bar

5.2.1 MyClass

After click on 'MyClass' following screen will be show

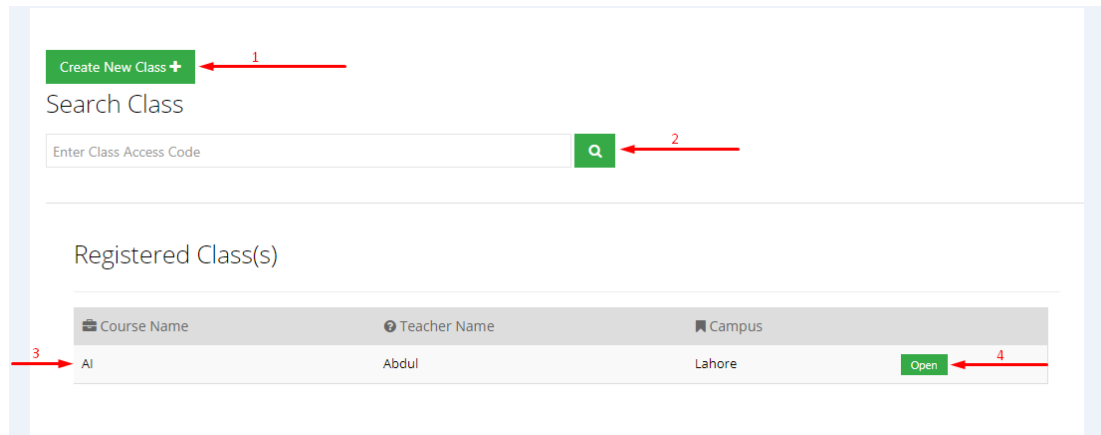


Figure 95: MyClass View

- 1) This button is only visible to the user of type faculty. By click on 'Create New Class' a form will show in which teacher enter the class details
- 2) This field only visible to the user of type student. Student enter the class access code and click on the search button
- 3) In this table all the classes in which user is registered will show up
- 4) By click 'Open' button user can open the class

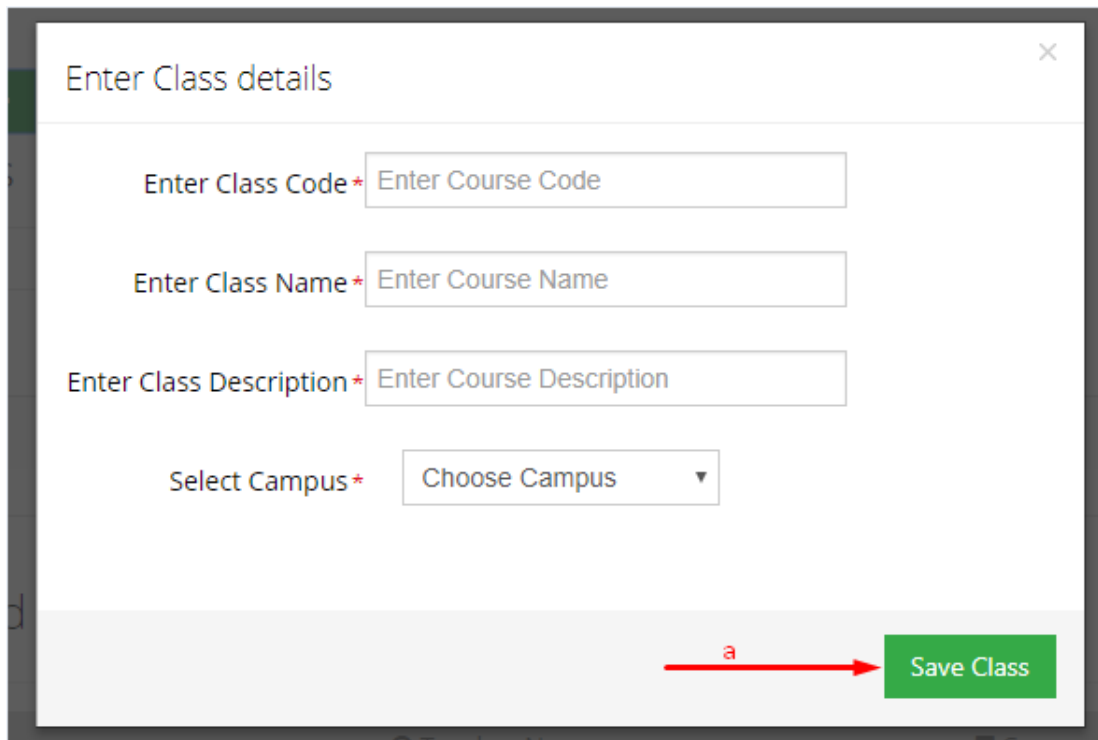
5.2.1.1 Create a new class

- 1) New class can be create by click on the button 'Create New Class' as shown below



Figure 96: Create New Class Button

- 2) After click on the 'Create New Class' class detail form will be show as follows



Enter Class details

Enter Class Code * Enter Course Code

Enter Class Name * Enter Course Name

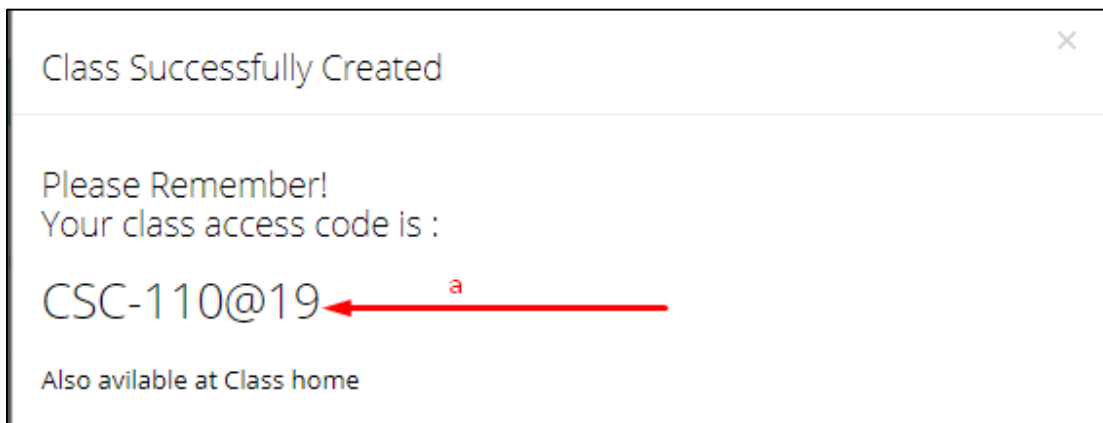
Enter Class Description * Enter Course Description

Select Campus * Choose Campus

Save Class

Figure 97: Create New Class Screen

- a) After fill the form click on the 'Save Class' button to crate the class
- 3) After click on 'Save Class' successful class creation and class access code message will display as shown below



Class Successfully Created

Please Remember!
Your class access code is :

CSC-110@19

Also avilable at Class home

Figure 98:Class Create Message

- a) This class access code teacher gives to students for class searching and this code also available in that class's home

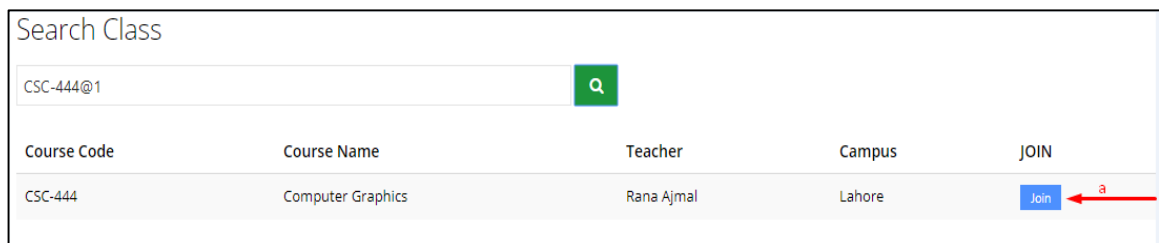
5.2.1.2 Search and join a class

- 1) To search a class write class access code given by the teacher in the search field as shown below



Figure 99: Class Access Code Screen

- 2) After enter the class access code click on the search button
 3) After click on the search button class will record will be shown as below



Course Code	Course Name	Teacher	Campus	JOIN
CSC-444	Computer Graphics	Rana Ajmal	Lahore	Join

Figure 100: Search Class Result Screen

- a) In the searched class record click on the 'Join' button to join that class
 4) After click on the 'Join' button successful class joined message will display and that class will be show in registered classes table

5.2.1.3 Open a class

- 1) To open a class click on the 'open' button in the registered class table as shown below

Registered Class(s)			
Course Name	Teacher Name	Campus	
SQA	Abdul	Karachi	Open
Cloud	Abdul	Lahore	Open

Figure 101: Registered Classes Screen

2) After click on the 'Open' button class view will display as shown below

Software

abc

- Resources
- Announcements
- Submitted Assignments

Book(s) [Upload Resource](#)

Lecture(s)

Assignment(s)

Assignment 1	Apple1.docx	Submit Assignment
Assignment 2	352284030-DW-Assignment1.docx	Submit Assignment

Figure 102: Registered Class View after Open

5.2.1.4 Upload and Download Resource from class

1) To upload a resource in class click on the 'Upload Resource' button as shown below



Figure 103: Upload Resource Button

- 2) After click on the 'Upload Resource' button file detail form will show up as below

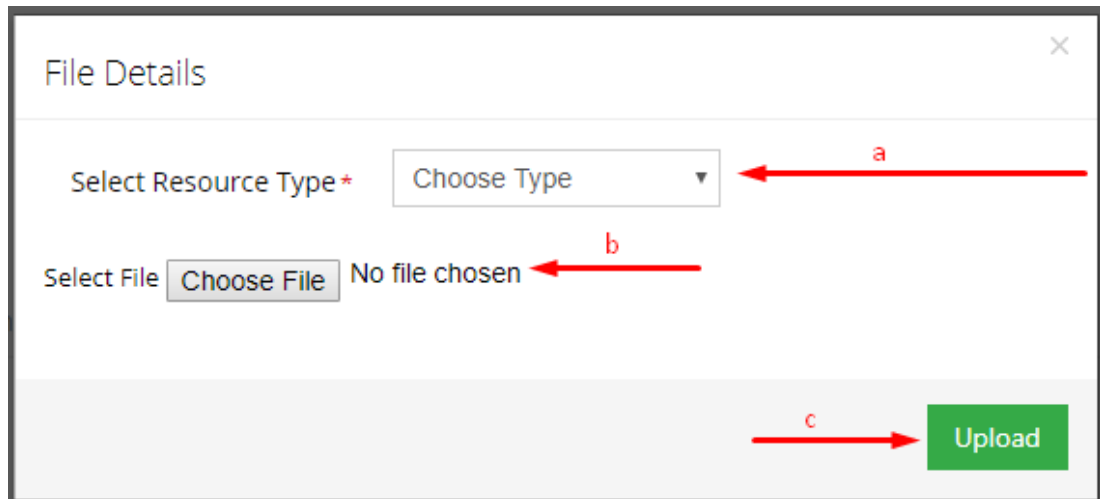


Figure 104: Upload Resource in Class

- a) Choose Resource type from the list
 - b) Choose the file from your computer
 - c) Click on 'Upload' button
- 3) After click on 'Upload' button the success message will show up and resource will show in class resources as shown below

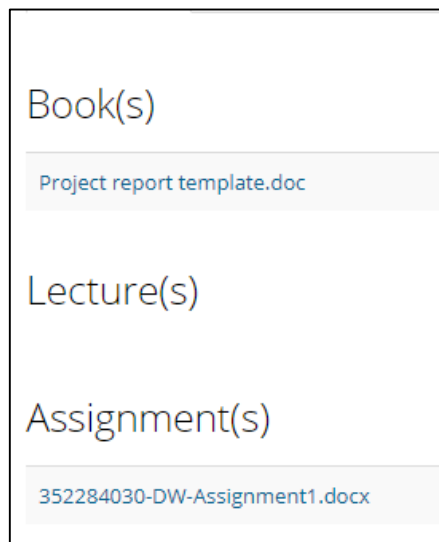


Figure 105: Virtual Class Resources View

- 4) To download resource just click on the resource at class home screen

5.2.1.5 Submit assignment in class

- 1) To submit assignment in class click on the ‘Submit Assignment’ button in front of any assignment as shown below

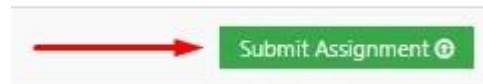


Figure 106: Submit Assignment button

- 2) After click on the ‘Submit Assignment’ button file upload form will show up as below

The image shows a 'File Details' dialog box. At the top, it says 'File Details' with a close button (X). Below that, there is a 'Select File' label, a 'Choose File' button, and the text 'No file chosen'. At the bottom right of the dialog, there is a green 'Submit' button.

Figure 107: Assignment Submit Form

- 3) Upload the Assignment file by click on ‘Choose File’ button and click on ‘Submit’ button
- 4) After click on ‘Submit’ button the assignment will show in submitted assignment section as shown below

Resources	Announcements	Submitted Assignments
≡ Assignment 1		
03-134142-001 submitted Submitted Assignment 1 at 2018-05-10 01:30:50		DOWNLOAD
≡ Assignment 2		
03-134142-001 submitted Submitted Assignment 2 at 2018-05-10 01:33:17		DOWNLOAD

Figure 108: Assignment Submitted Section

- 5) Teacher can download submitted assignments by click on ‘DOWNLOAD’ link

5.2.1.6 View / Make Announcement in class

- 1) To make announcement in class click on 'Announcement' tab
- 2) Click on 'Make Announcement' button in 'Announcement' tab
- 3) After click on 'Make Announcement' button announcement form will show up as below

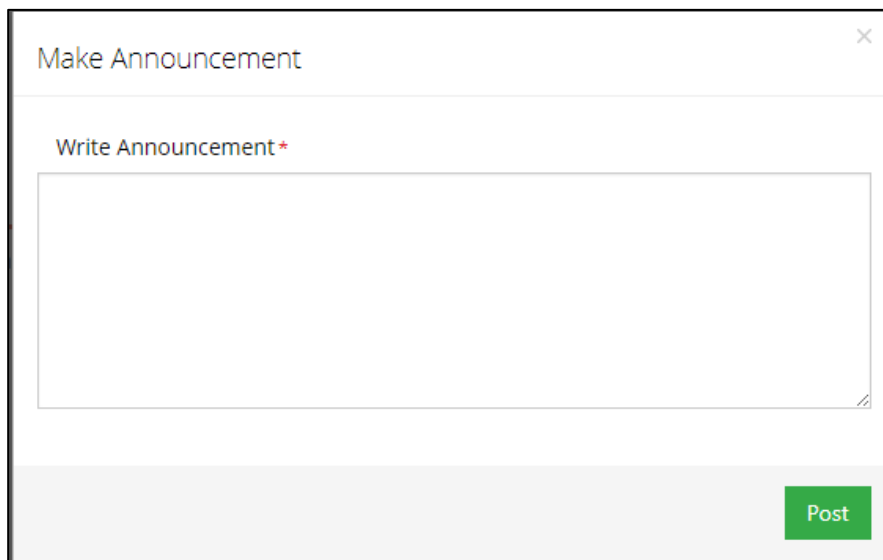


Figure 109: Make Announcement Screen

- 4) After write the announcement click on 'Post' button
- 5) After on 'Post' button announcement will show in 'Announcement' tab as shown below

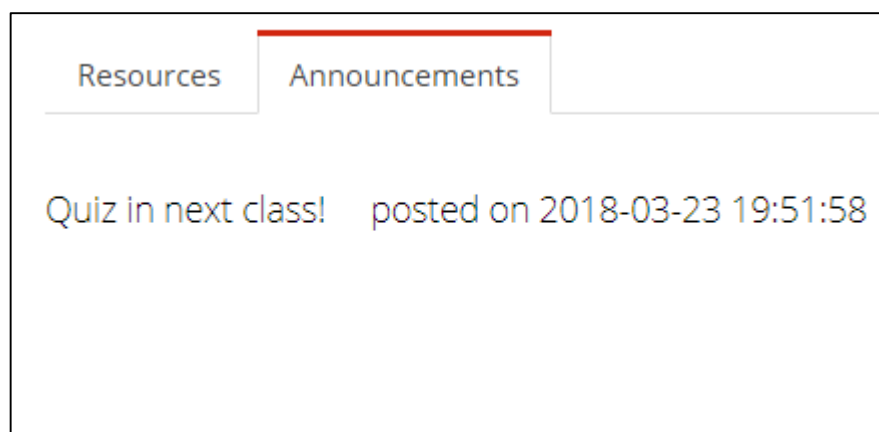


Figure 110: Announcement View

5.2.2 Library

After click on 'Library', following screen will be show

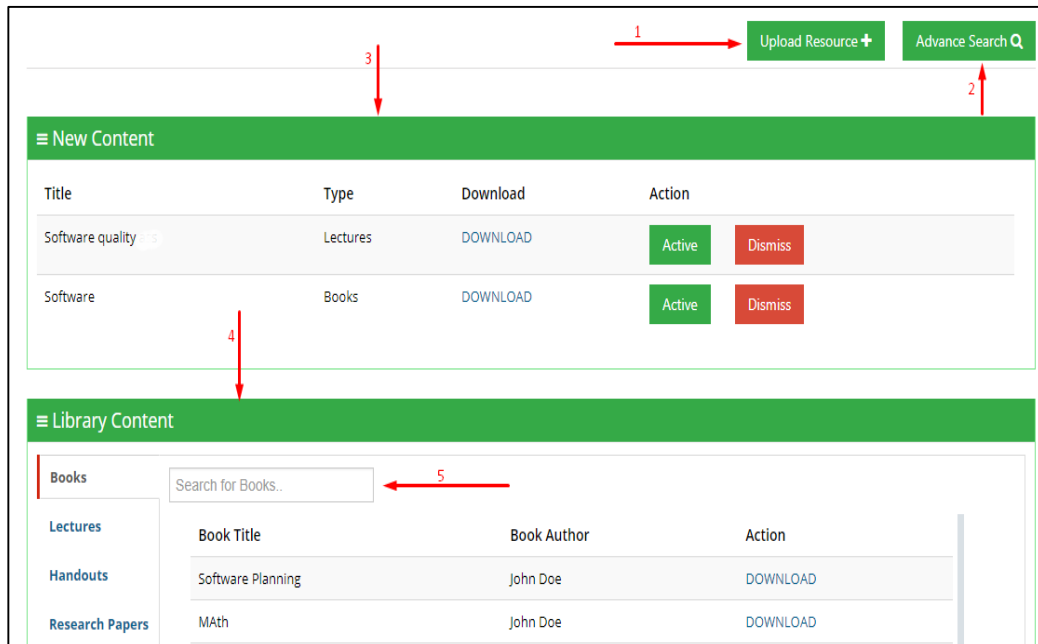


Figure 111: Library Home Screen

- 1) By click on 'Upload Resource' button user can upload resources in library
- 2) By click on 'Advance Search' button user can perform advance search in library by giving complete details of a resource
- 3) This table show all the new uploaded content in library require permission by librarian to be visible in library. This section only visible to user type librarian
- 4) This window show all the content available in library and by user can navigate through the content type by changing the tab for example Books, Lectures etc
- 5) This search field helps to perform live search by giving resource title
- 6) User can download any resource by click on 'DOWNLOAD' link

5.2.2.1 Upload and Download resource from Library

- 1) To upload a resource in class click on the 'Upload Resource' button as shown below

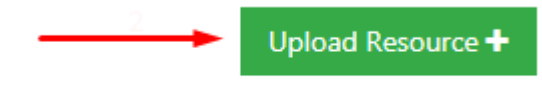


Figure 112: Upload Resource Button in Library

- 2) After click on the 'Upload Resource' button file detail form will show up as below

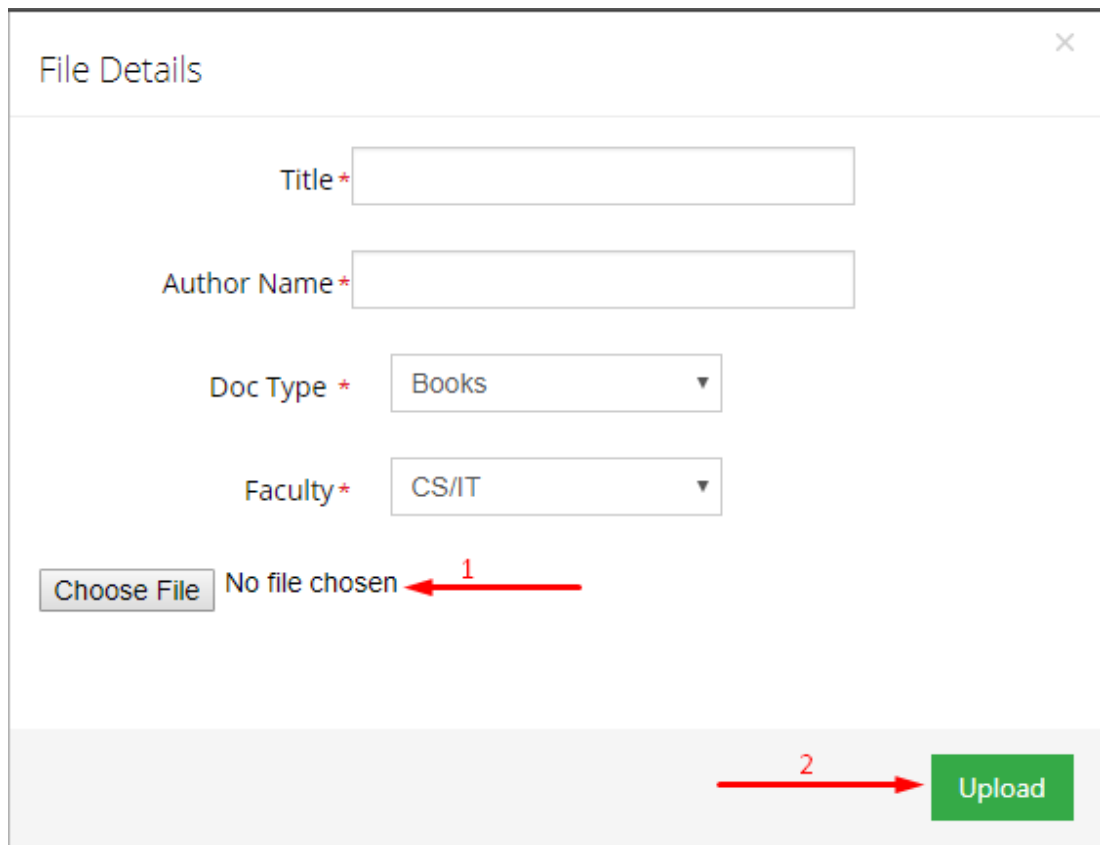


Figure 113: Upload Resource in Library

- a) After enter detail of resource choose the resource file from your computer
- b) After choosing the file click on 'Upload' button to upload the file

- 3) After click on 'Upload' button the success message will show up and resource will show in library resources as shown below
- 4) To download resource just click on the resource at library home screen

5.2.2.2 Advance search in Library

- 1) To perform advance search in library, click on the 'Advance Search' button as shown below

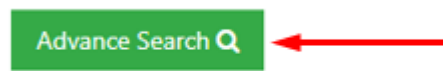


Figure 114: Advance Search Button

- 2) After click on the 'Advance search' button file detail form will show up as below

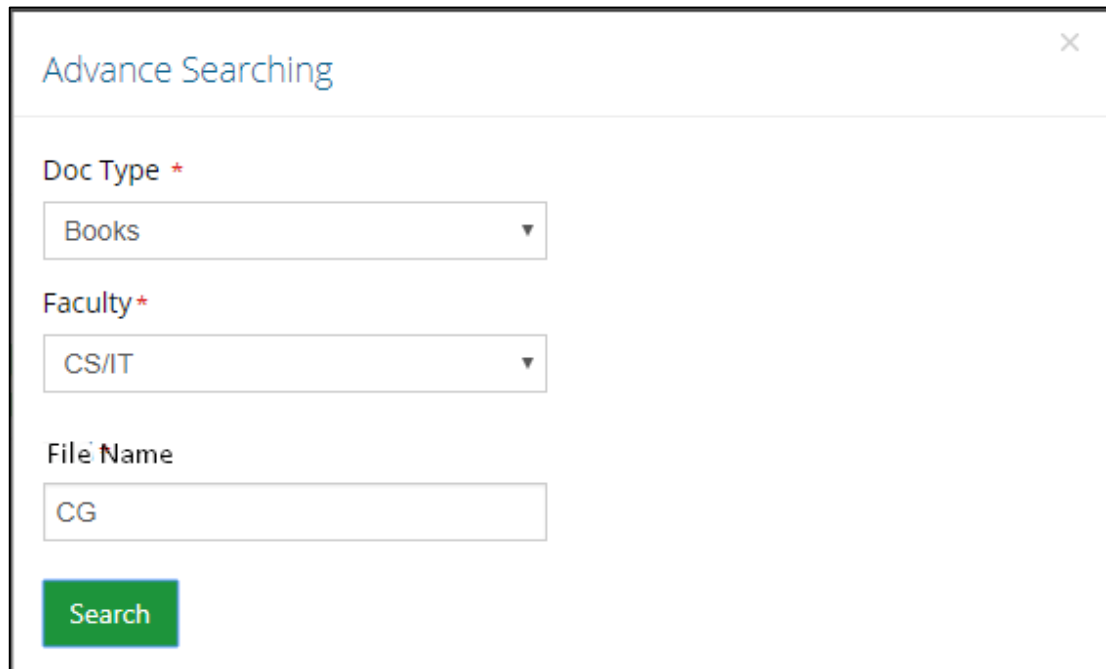


Figure 115: Advance Search Screen

- 3) After filling the form click on 'Search' button

- 4) After click on 'Search' button resource record will show in a table as shown below

Title	Author	Download
CG	John Doe	DOWNLOAD

Figure 116: Search Result Screen

5.2.2.3 Active/Dismiss in library

- 1) Click on 'Library' from sidebar
- 2) In 'New Content' section all the content require permission will be shown up as in following figure

New Content			
Title	Type	Download	Action
Software quality	Lectures	DOWNLOAD	Active Dismiss
Software	Books	DOWNLOAD	Active Dismiss

Figure 117: New Uploaded Content Approve/Disapprove in Library

- 3) Click on the 'Download' link to view the resource and verify it as shown below

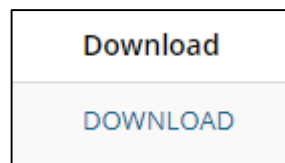


Figure 118: Download Link in Library

- 4) After verify the content click on 'Active' button if resource is fit to be visible in library as shown below

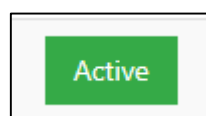


Figure 119: Library Content Active Button

OR

Click on ‘Dismiss’ button if resource is not fit to be visible in library as shown below

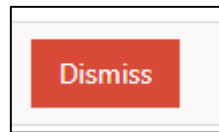


Figure 120: Library Content Dismiss Button

5.2.3 Alumni

After click on ‘Alumni’, following screen will be show

Figure 121: Alumni Advance Search Screen

- 1) In Alumni Search form ‘Student Name’ and ‘Campus’ are compulsory fields
- 2) After fill mandatory fields and other known fields click on ‘Search’ button
- 3) After click on ‘Search’ button record will show up in tabular form as shown below

Enrollment	Student Name	Batch	Degree Level	Campus	Current Job	Official Phone	Mobile	
03-134142-001	Abdul	2014	Undergraduate	Lahore	Analyst at Systems	8888	11111	View

Figure 122: Alumni Search Result Screen

5.2.4 Profile

After click on 'Profile', following screen will be show

Figure 123: Alumni Profile View

- 1) 'Profile Info' tab shows the profile information of Alumni and his/her picture
- 2) User can update his/her profile information and picture form 'Update Profile' tab

5.2.4.1 Update profile Information

- 1) Click on 'Update Profile' tab on Profile
- 2) Under 'Update Profile' tab Click on 'Personal Info' tab as shown below

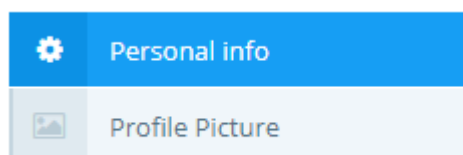
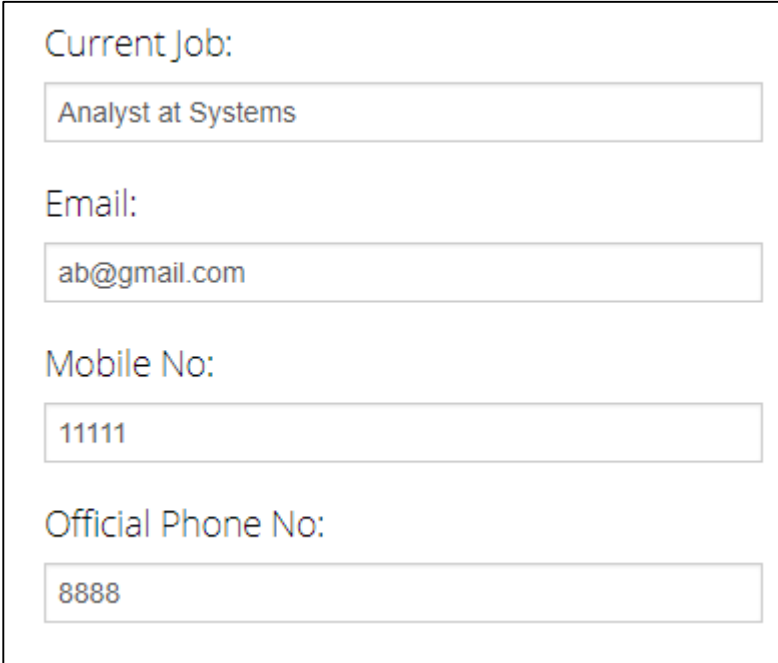


Figure 124: Edit User Profile Tab

3) Update editable fields as shown below



Current Job:
Analyst at Systems

Email:
ab@gmail.com

Mobile No:
11111

Official Phone No:
8888

Figure 125: Update User Information Screen

4) Click on 'Update' button at end of profile to update the profile as shown below



Figure 126: Update User Profile Button

5.2.4.2 Update Profile Picture

- 1) Click on 'Update Profile' tab on Profile
- 2) Under 'Update Profile' tab Click on 'Profile Picture' tab as shown below

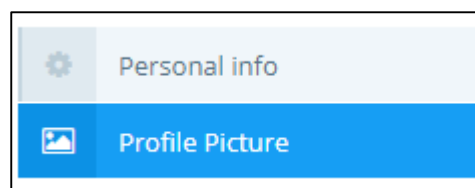


Figure 127: Edit User Profile Tab

3) After click on

- 4) 'Profile Picture' tab choose picture by click on 'Choose file' button
- 5) After choosing file click on 'Update' button as shown below

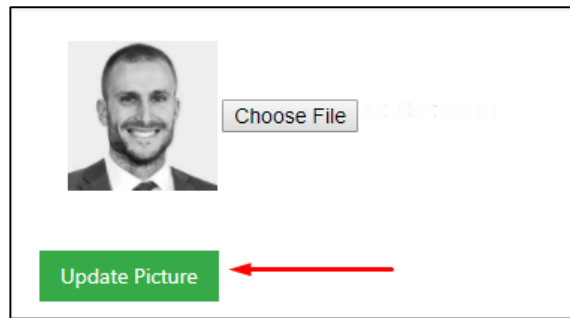


Figure 128: Update User Profile Picture

5.3 Correspondence System (CS)

As discussed in section 5.1.1 Correspondence System is used by employee of Bahria University and in section 5.1.4 all authorized home screens are also discussed. Now in this section usage of Correspondence System will be discussed that starts from 'Home' Screen as shown in below figure.

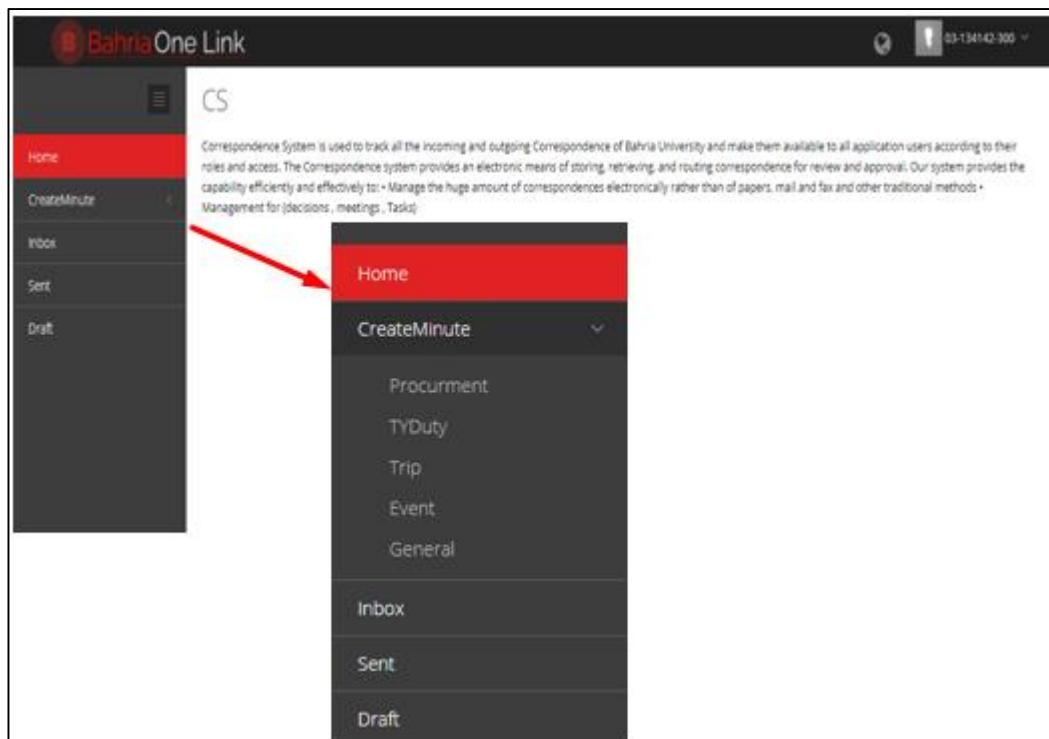


Figure 129: Correspondence Home Screen

In above diagram that sidebar show to the user of type employee and employee can visit to any page given in sidebar by click on it, for example if employee wants to

create correspondence or use any feature (according to privileges) just click on ‘Create Minute’ and then click on any type of minute.

5.3.1 Create Minute

After click on ‘Create Minute’ a list of minute type will show as below

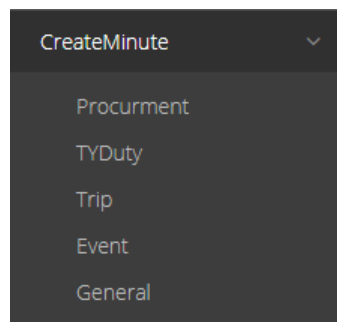


Figure 130: Minute List

Click on desired minute and that minute view will open (for example TY Duty as follows)

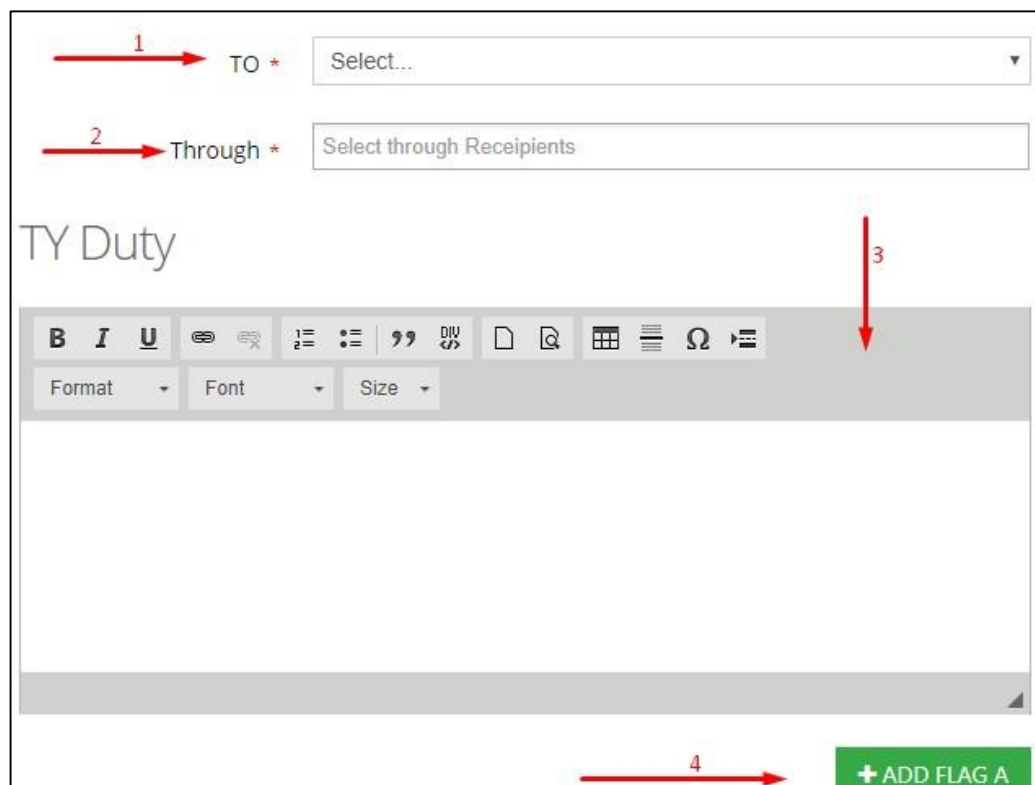


Figure 131: Create Minute Screen

- 1) List of recipients from which user have select the person to whom user want to send the minute at final
- 2) List of recipients from which user have select the persons through those user want to send the minute to the final person
- 3) Editor in which user write the minute
- 4) By click on 'Add Flag' button user able to add flag of attachments

If minute type have Performa then Performa will also show after editor as shown below

The screenshot shows a form titled "Temporary Duty Performa" with a blue header. Below the header, there are several input fields with labels and asterisks indicating they are required:

- Name * (text input: Enter Name)
- Designation/Appointment * (text input: Enter Designation)
- Nature of duty * (text input: Enter Nature Of Duty)
- Satation of duty to be visited * (text input: Enter Station of Duty)
- Date of Departure * (text input: Enter Date of Departure)
- Expected duration of stay at the proposed station of duty * (text input: Enter Expected Duration)
- Mode of conveyance * (dropdown menu: Choose Mode)

Figure 132: Performa Attached with Minute

5.3.1.1 Send a Minute

- 1) Open any desired type of minute and select the final recipient as shown below

The screenshot shows a dropdown menu for selecting recipients. The dropdown is open, showing a list of roles:

- Select...
- Select...
- hod:Computer Science
- hod:Management Sciences
- SystemEngineer:IT
- director:Admin
- deputyDirector:Admin

Figure 133: Recipient List

- 2) After selecting final recipient select the recipients for through list as shown below



Figure 134: Recipient List

- 3) Write minute in the editor as shown below



Figure 135: Minute Input Area

- 4) Add flags for attachments by click on 'Add Flag' button (if required) as shown below

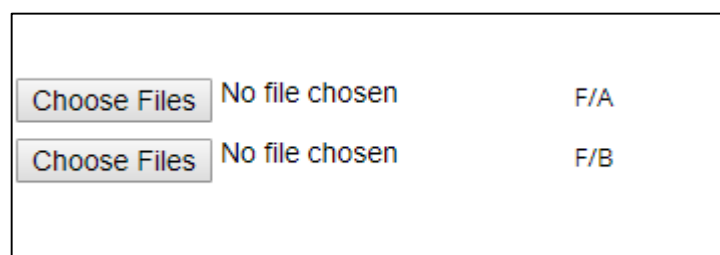


Figure 136: Minute Flags without Attachments

- 5) Choose files for as flag attachments (if flags added) as shown below

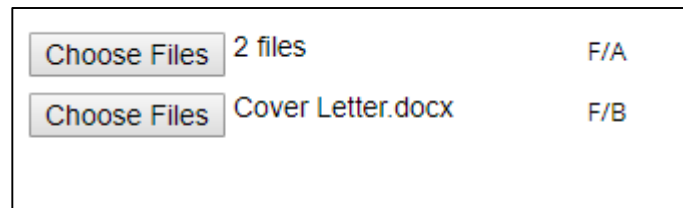


Figure 137: Minute Flags with Attachments

- 6) Fill the given Performa if minute type has Performa
- 7) Finally click on 'Send' button to send the minute as shown below



Figure 138: Sent Minute Button

- 8) After Click on the 'Send' button minute can be seen in 'Sent' from sidebar

5.3.1.2 Save a Minute

- 1) Open any desired type of minute
- 2) Write minute in the editor as shown below



Figure 139: Minute Input Area

- 3) Click on 'Save' button to save the minute as shown below

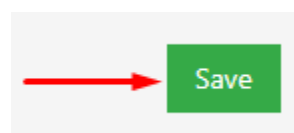
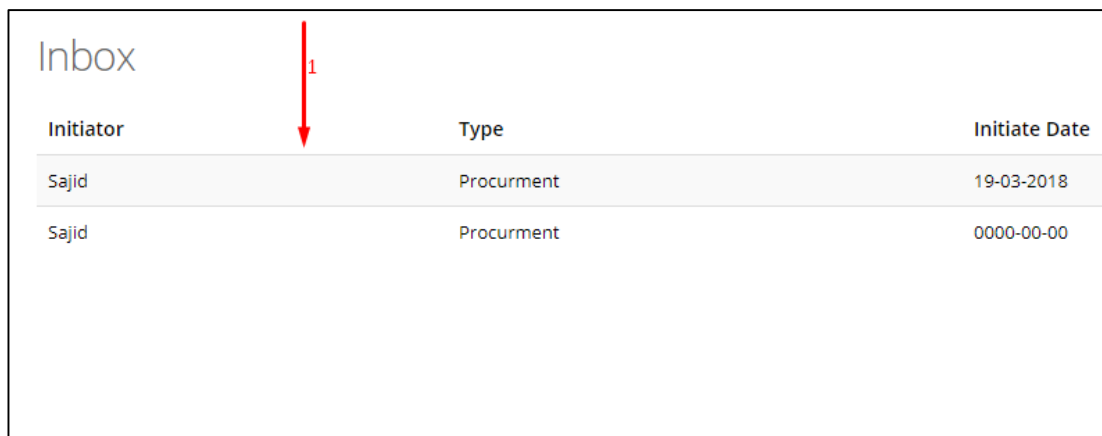


Figure 140: Save Minute Button

- 4) After Click on the 'Save' button minute can be seen and re-edited in 'Draft from sidebar

5.3.2 Inbox

After click on 'Inbox', following screen will be show



Initiator	Type	Initiate Date
Sajid	Procurement	19-03-2018
Sajid	Procurement	0000-00-00

Figure 141: Inbox View

- 1) All received minutes of user and each can be open by click on it

5.3.2.1 Open a received minute

- Click on 'Inbox' from sidebar
- Click on any desired minute
- After click on any minute following screen will show up

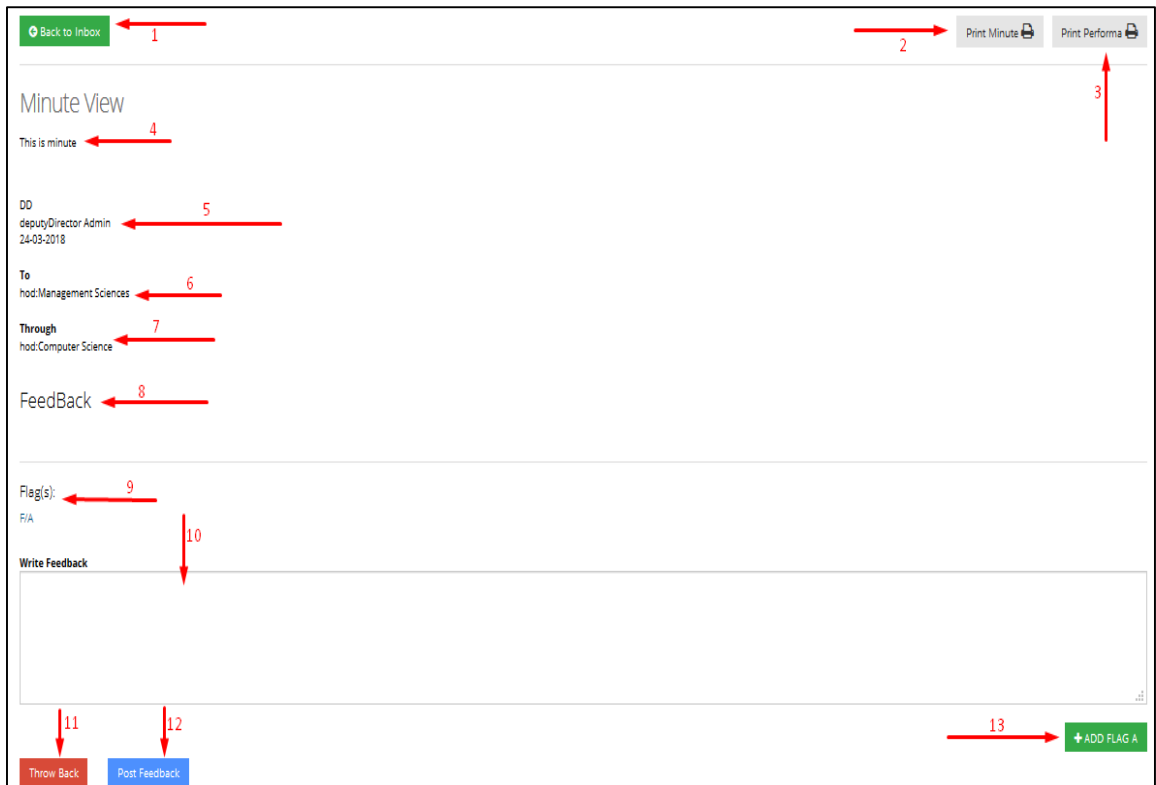


Figure 142: Received Minute View

- 1) By click on 'Back to Inbox' button (as shown below) user get back to the list of received minutes in Inbox



Figure 143: Back to Inbox Button

- 2) By click on 'Print Minute' button (as shown below) user is able to print the minute

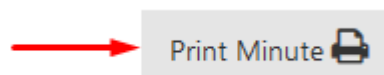


Figure 144: Print Minute Button

- 3) If minute have Performa then 'Print Performa' button (as shown below) helps to print the Performa attached with the minute



Figure 145: Print Performa Button

- 4) Main body of the minute as shown below

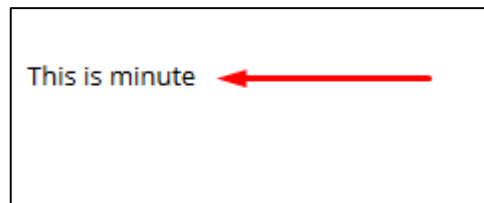


Figure 146: Received Minute Text

- 5) Details of sender as shown below

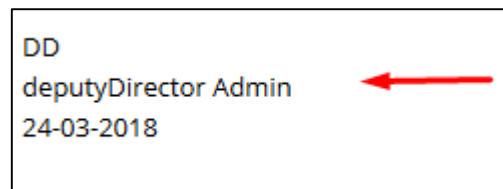


Figure 147: Sender Detail in Received Minute

- 6) Final recipient as shown below

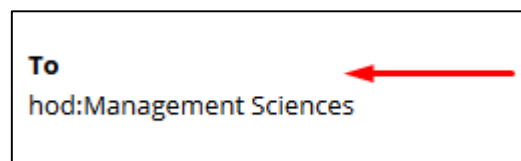


Figure 148: Final Recipient in Received Minute

- 7) Recipient list through which minute sent to final recipient (see point 6)

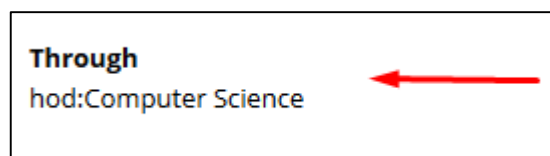


Figure 149: Through Recipient of Received Minute

- 8) Feedback list shows up here given by all recipients as shown below



Figure 150: Feedback List in Received Minute

- 9) Attached flags with the minute and each flag have at least one attachment that can be seen by click on the flag as shown below

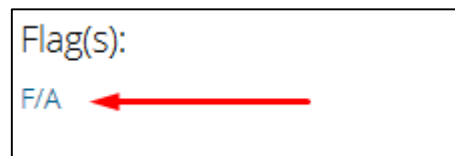


Figure 151: Flags in Received Minute

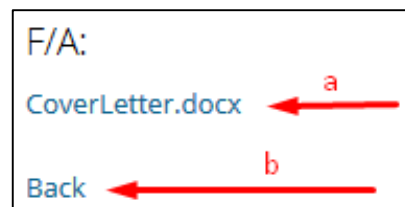


Figure 152: Attachments in flag in Received Minute

- a) Attachment inside the flag
b) By click on 'Back' link user get back to the flags list

- 10) Text area to write a feedback as shown below

Write Feedback

Figure 153: Feedback Input Area

- 11) If minute is not related to the user, then by clicking on 'Throw Back' button (as shown below) sender will get a notification and that minute will discard from the received list of the user



Figure 154: Throw Back Button

- 12) After writing feedback in the text area (see point 10) click on 'Post Feedback' button (as shown below) to post the feedback

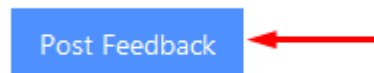


Figure 155: Post Feedback Button

- 13) 'Add Flag' button (as shown below) will add flags in minute with feedback posting



Figure 156: Add Flags Button

5.3.2.2 Giving feedback of received minute

- 1) Open any desired minute from 'Inbox'
- 2) Write the feedback in feedback area as shown below

Write Feedback

This is feedback

Figure 157: Filled Feedback Input Area

- 3) After writing feedback click on 'Post Feedback' button to post the feedback as shown below

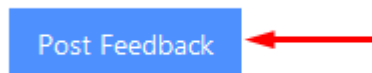


Figure 158: Post Feedback Button

- 4) After click on 'Post Feedback' button feedback will be show up in feedback list as shown below



Figure 159: Feedback View

5.3.2.3 Giving Feedback to a specific person

- 1) Open any desired minute from 'Inbox'
- 2) Click on 'Post To' button as shown below



Figure 160: Post To button

- 3) After click on 'Post To' button feedback dialog will show up as below

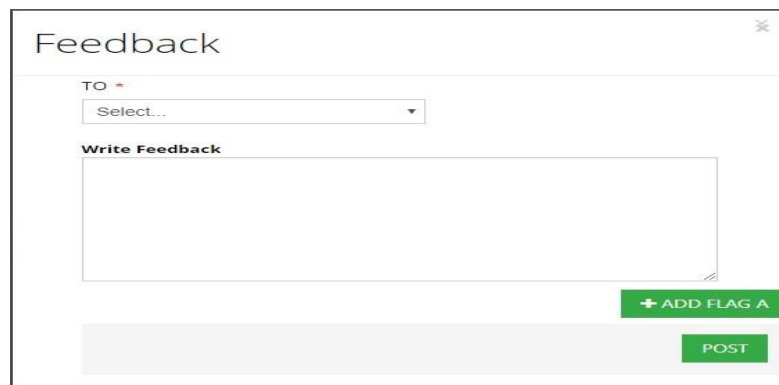


Figure 161: Feedback Dialog box

- 4) Write feedback and select the recipient from 'To' dropdown
- 5) Finally click on 'Post' button

5.3.2.4 Print a received minute and/or Performa

- 1) Open any desired minute from 'Inbox'
- 2) Click on 'Print Minute' button to print the minute or click on 'Print Performa' button to print the Performa as shown below

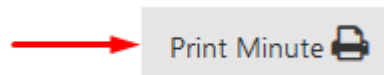


Figure 162: Print Button in Received Minute

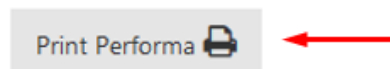


Figure 163: Print Performa Button in Received Minute

- 3) After click on 'Print Minute' button print window will open as shown below

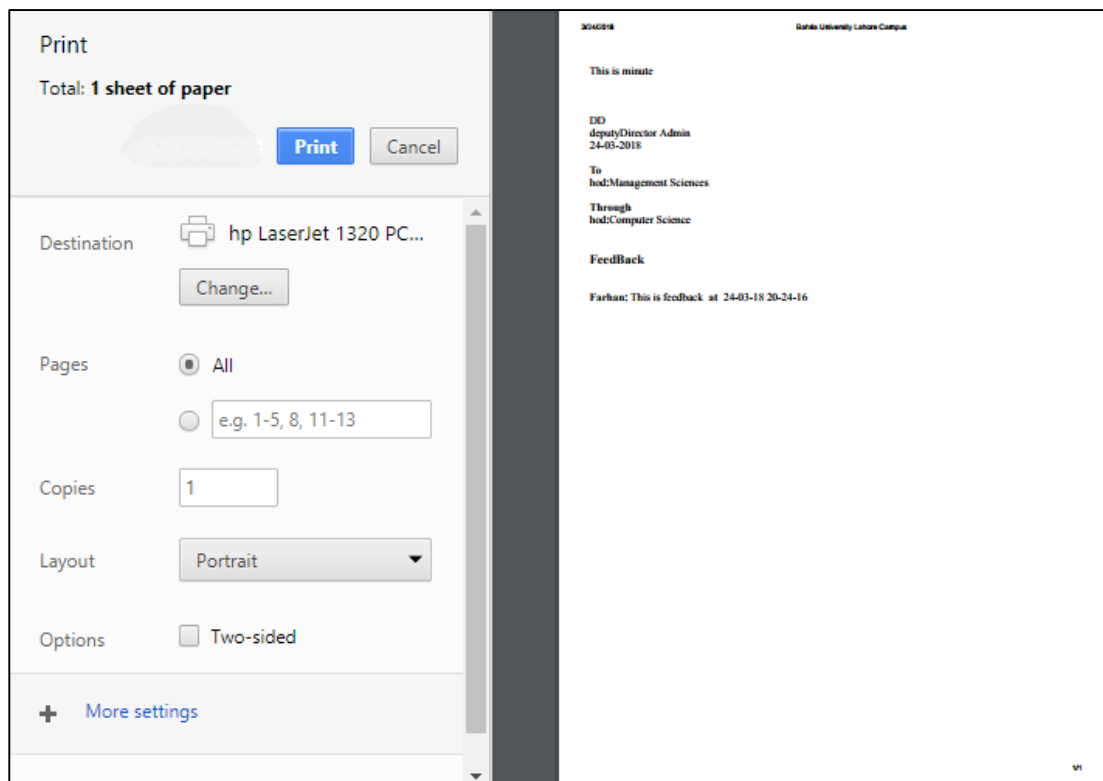


Figure 164: Print Minute Screen in Received Minute

- 4) Click on 'Print' button to print as shown below



Figure 165: Print Button

5.3.3 Sent

After click on 'Sent', following screen will be show

Sent		
Initiator	Type	Initiate Date
Farhan	General	19-03-2018

Figure 166: Sent Minute View

- 1) All sent minutes of user and each can be open by click on it

5.3.3.1 Open a sent minute

- Click on 'Sent' from sidebar
- Click on any desired minute
- After click on any minute following screen will show up

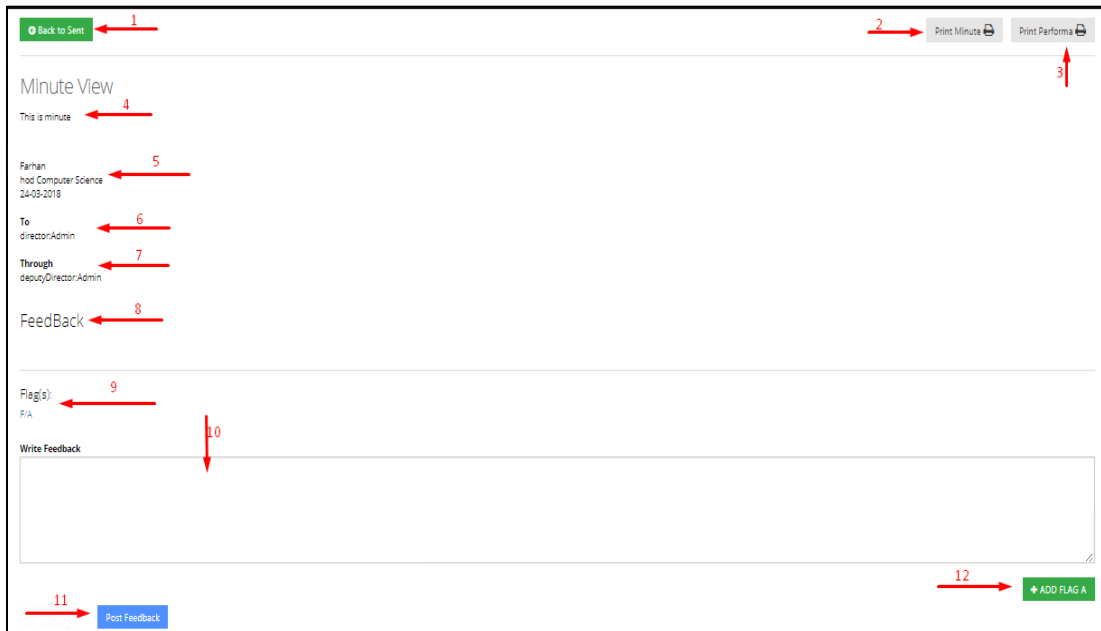


Figure 167: Sent Minute View

- 1) By click on 'Back to Sent' button (as shown below) user get back to the list of sent minutes in Sent

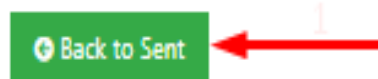


Figure 168': Back to Sent Button

- 2) By click on 'Print Minute' button (as shown below) user is able to print the minute

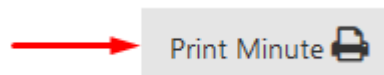


Figure 169: Print Minute Button in Sent Minute

- 3) If minute have Performa then 'Print Performa' button (as shown below) helps to print the Performa attached with the minute



Figure 170: Print Performance Button in Sent Minute

- 4) Main body of the minute as shown below

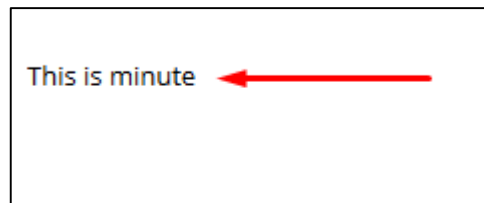


Figure 171: Sent Minute Text

- 5) Details of sender as shown below

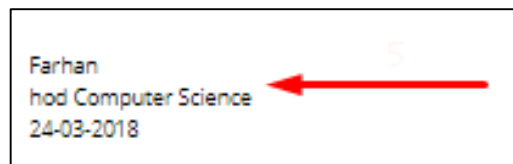


Figure 172: Sender Detail in Sent Minute

- 6) Final recipient as shown below

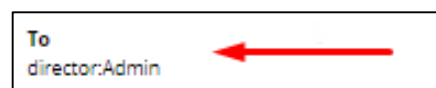


Figure 173: Final Recipient in Sent Minute

- 7) Recipient list through which minute sent to final recipient (see point 6)

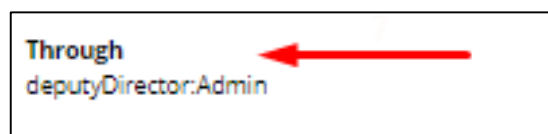


Figure 174: Through Recipient in Sent Minute

- 8) Feedback list shows up here given by all recipients as shown below

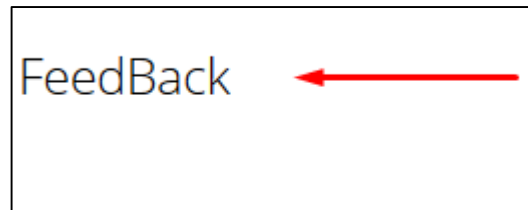


Figure 175: Feedback in Sent Minute

- 9) Attached flags with the minute and each flag have at least one attachment that can be seen by click on the flag as shown below

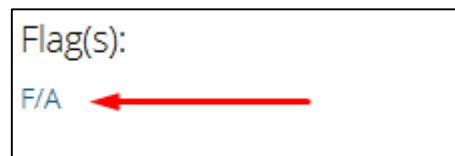


Figure 176: Flags in Sent Minute

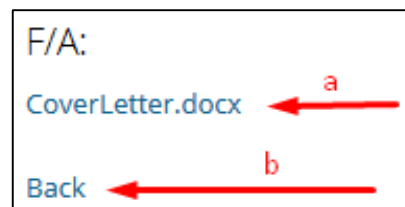


Figure 177: Attachments in flag in Sent Minute

- a) Attachment inside the flag
- b) By click on 'Back' link user get back to the flags list

- 10) Text area to write a feedback as shown below

Write Feedback

Figure 178: Feedback Input Area in Sent Minute

- 11) After writing feedback in the text area (see point 10) click on 'Post Feedback' button (as shown below) to post the feedback

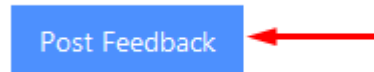


Figure 179: Post Feedback Button in Sent Minute

- 12) 'Add Flag' button (as shown below) will add flags in minute with feedback posting



Figure 180: Add Flags Button in Sent Minute

5.3.3.2 Giving feedback of Sent minute

- 1) Open any desired minute from 'Sent
- 2) Write the feedback in feedback area as shown below

Write Feedback
This is feedback

Figure 181: Filled Feedback Input area in Sent Minute

- 3) After writing feedback click on 'Post Feedback' button to post the feedback as shown below

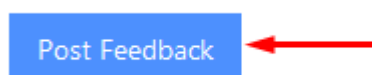


Figure 182: Post Feedback Button in Sent Minute

- 4) After click on 'Post Feedback' button feedback will be show up in feedback list as shown below



Figure 183: Feedback List in Sent Minute

5.3.3.3 Print a Sent minute and/or Performa

- 1) Open any desired minute from 'Sent
- 2) Click on 'Print Minute' button to print the minute or click on 'Print Performa' button to print the Performa as shown below

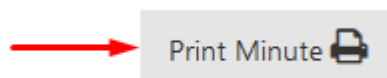


Figure 184: Print Minute Button in Sent Minute

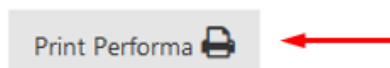


Figure 185: Print Performa Button in Sent Minute

- 3) After click on 'Print Minute' button print window will open as shown below

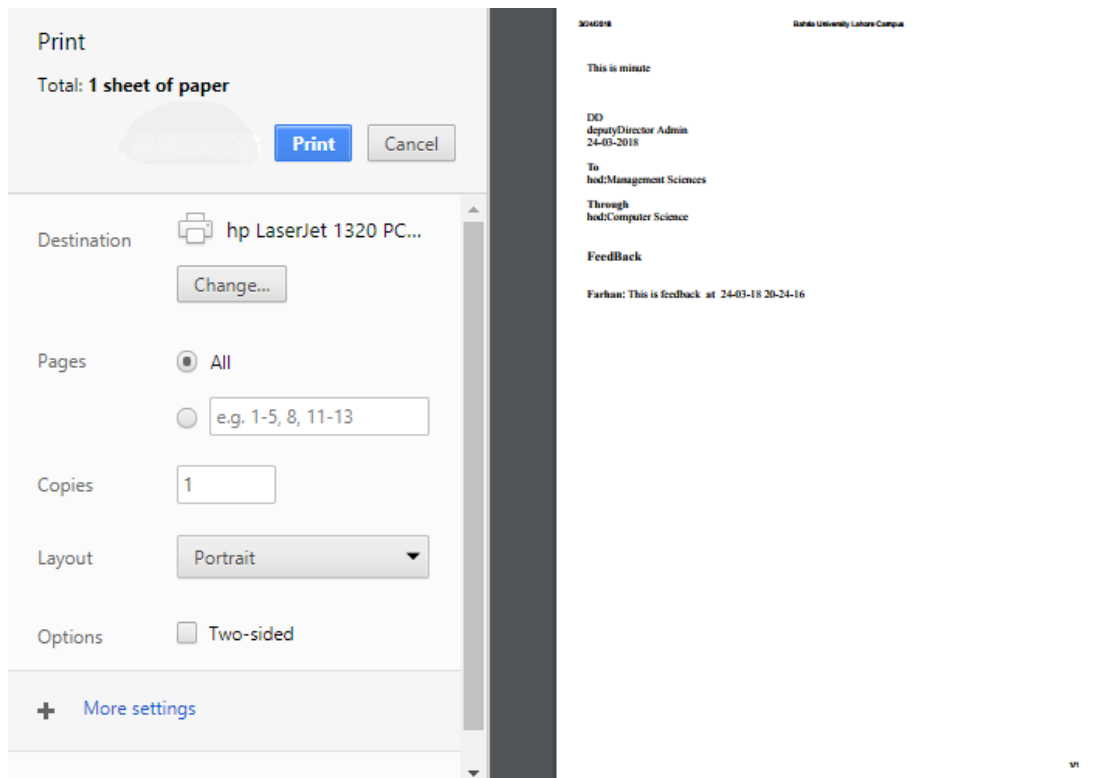


Figure 186: Print Minute View in Sent Minute

- 4) Click on 'Print' button to print as shown below



Figure 187: Print Button

5.3.3.4 Close a minute

- 1) Open any desired minute from 'Sent'
- 2) Click on 'Close Correspondence' button as shown below



Figure 188: Close Correspondence Button

- 3) After click on 'Close correspondence' button confirmation dialog will show up as below



Figure 189: Correspondence Close Confirmation Dialog

- 4) Click on 'Yes' button to close the correspondence

5.3.4 Draft

After click on 'Draft', following screen will be show

Draft		
Saved	Type	Date
Saved	General	20-03-2018

Figure 190: Draft View

- 1) All saved minutes of user and each can be edit by click on it

5.3.4.1 Edit a saved minute

- Click on 'Draft' from sidebar
- Click on any minute that to be edit
- After click on minute it will open in the editor of related minute type as shown below

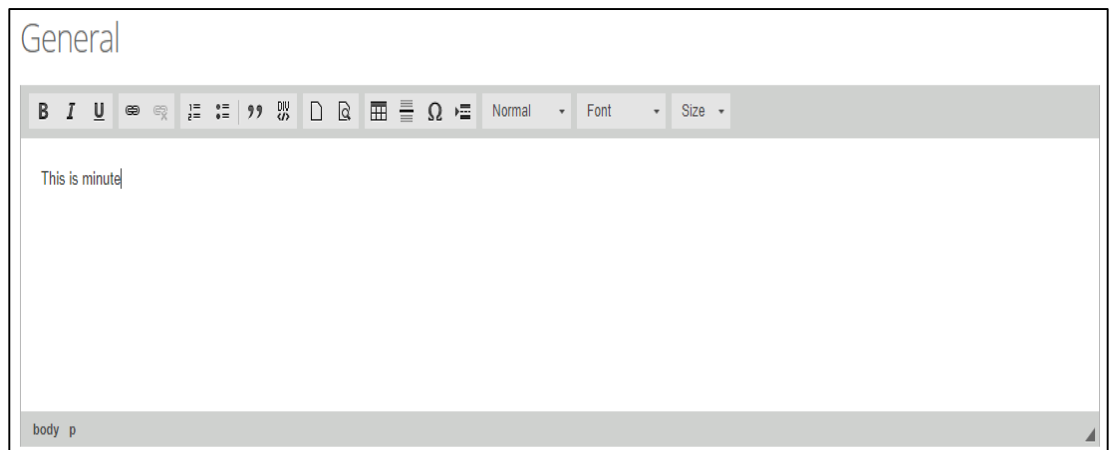


Figure 191: Filled Minute Input Area

- After editing minute, user can send or save the minute

5.4 User Management System (UMS)

As discussed in section 5.1.1 User Management System is used by System Engineer (that is the Admin of BOL) and in section 5.1.4 all authorized home screens are also discussed. Now in this section usage of User Management System will be discussed that starts from 'Home' Screen as shown in below figure.

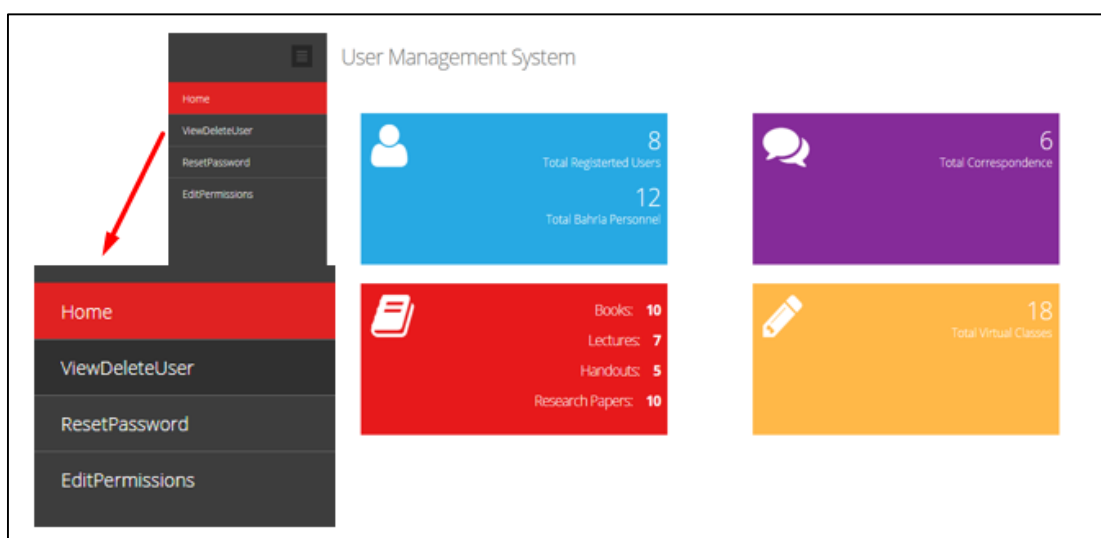


Figure 192: Admin Home Screen

In above diagram, that sidebar show to the user of type Admin and Admin can visit to any page given in sidebar by click on it, for example if Admin wants to reset password of a user then just click on 'ResetPassword'. Moreover, admin can see statistics of Users, Correspondence, Library and Virtual Classes on Home screen of UMS.

5.4.1 ViewDeleteUser

After click on 'ViewDeleteUser', following screen will be show

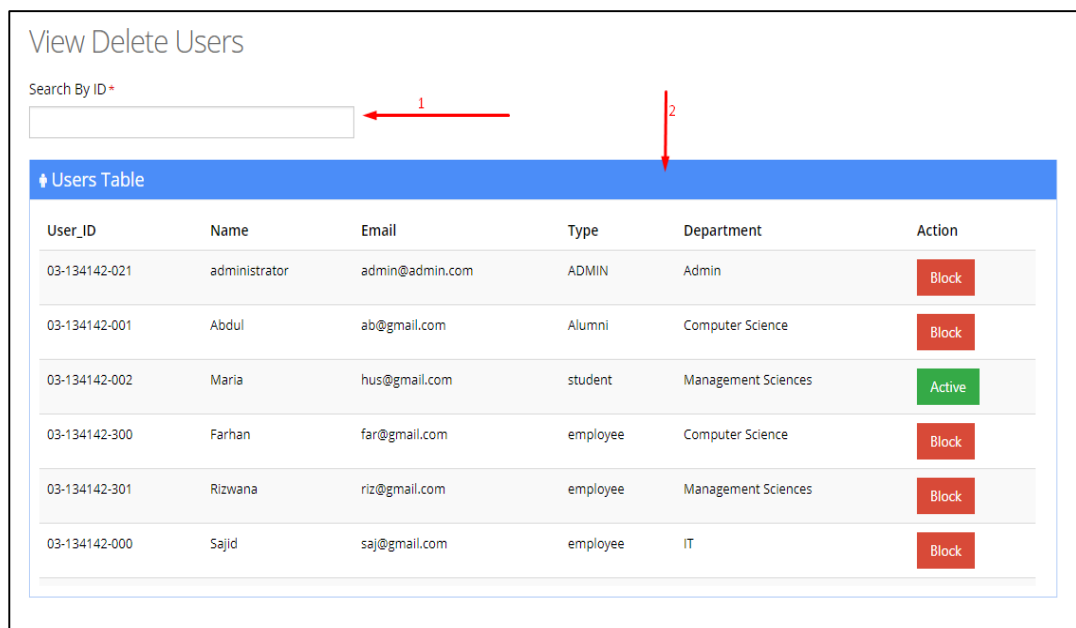


Figure 193: View Delete User Screen

- 1) Search field that provides live user search by user id from the below user table in diagram
- 2) User table having all users details with their account status

5.4.1.1 Search a User

- 1) Click on 'ViewDeleteUser' from sidebar
- 2) Type the user id in the 'Search By ID' field as shown below

Figure 194: Search User Field

- 3) After typing user id user record will show in the table as shown below

View Delete Users

Search By ID *

03-134142-021

Users Table

User_ID	Name	Email	Type	Department	Action
03-134142-021	administrator	admin@admin.com	ADMIN	Admin	Block

Figure 195: Search User Result

5.4.1.2 Block/Active a User

- 1) Click on 'ViewDeleteUser' from sidebar
- 2) Type the user id in the 'Search By ID' field as shown below

OR

Figure 196: Search User Field

Scroll in the user table as shown below

User_ID	Name	Email	Type	Department	Action
03-134142-021	administrator	admin@admin.com	ADMIN	Admin	Block
03-134142-001	Abdul	ab@gmail.com	Alumni	Computer Science	Block
03-134142-002	Maria	hus@gmail.com	student	Management Sciences	Active
03-134142-300	Farhan	far@gmail.com	employee	Computer Science	Block
03-134142-301	Rizwana	riz@gmail.com	employee	Management Sciences	Block
03-134142-000	Sajid	saj@gmail.com	employee	IT	Block
03-134142-500	Amiad	ami@gmail.com	employee	Admin	Block

Figure 197: Registered Users

- 3) After finding the user click on the 'Active' button (if user is already blocked) or click on 'Block' button (if user is already active) as shown below

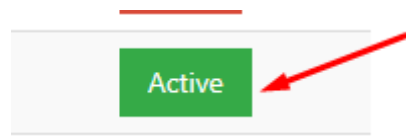


Figure 198: Active (Unblock) User Button

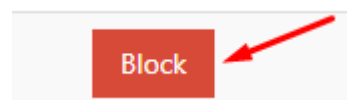


Figure 199: Block User Button

5.4.2 ResetPassword

After click on 'ResetPassword', following screen will be show

The screenshot shows a web form titled "Find User and Reset Password". It contains four main elements: a text input field for "Target User ID *", a text input field for "New Password *", a text input field for "Confirm New Password *", and a red "Reset Password" button with a magnifying glass icon. Red arrows with numbers 1, 2, 3, and 4 point to each of these elements respectively.

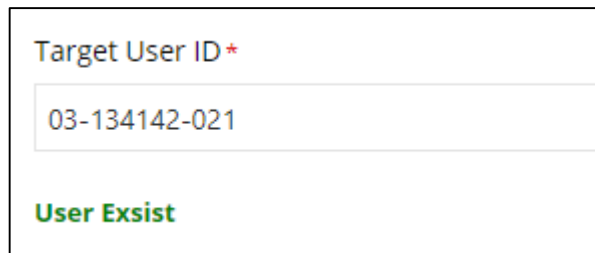
Figure 200: Reset User Password View

- 1) User id field in which admin enter target user id whom password is going to reset
- 2) Password field in which admin enter new password of the user
- 3) Password field in which admin re-enter the new password that make new password error free
- 4) By click on 'Reset Password' button newly entered password update the old password

5.4.2.1 Reset password of a user

- 1) Click on 'ResetPassword' from sidebar

- 2) Enter target user id in the field, if target user id is valid then it will show 'User Exist' message as shown below

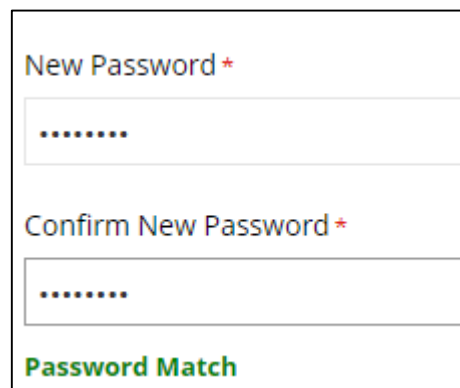


Target User ID *

User Exist

Figure 201: Check User Existence Filed

- 3) Enter new password and confirm new password for confirmation of password as shown below



New Password *

Confirm New Password *

Password Match

Figure 202: New and Conform Password Fields

- 4) Click on 'Reset password button' to reset the user password as shown below

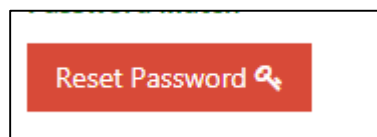


Figure 203: Reset Password Button by Admin

5.4.3 EditPermissions

After click on 'EditPermissions', following screen will be show

Edit Permissions

1

Edit Permission Table

Sidebar Name	System	Group Name	Edit Permissions	Action
Home	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
MyClass	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
MyChat	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
Library	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
Alumni	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
Home	CS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
CreateMinute	CS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>

Figure 204: Edit Permission View

- 1) Permissions table having all groups with their permissions on each sidebar

5.4.3.1 Edit Permission of a group

- 1) Click on 'EditPermissions' from sidebar
- 2) Scroll to target group whom permission have to be edited on a specific sidebar as shown below

MyClass	LMS	admin	<input checked="" type="checkbox"/> Add <input checked="" type="checkbox"/> Search <input checked="" type="checkbox"/> Upload <input checked="" type="checkbox"/> Update	<input type="button" value="Save"/>
---------	-----	-------	--	-------------------------------------

Figure 205: User Group Detail

- 3) Check/Un-Check the boxes to edit permissions of the group as shown below

<input checked="" type="checkbox"/> Add	<input type="checkbox"/> Search	<input checked="" type="checkbox"/> Upload	<input checked="" type="checkbox"/> Update
---	---------------------------------	--	--

Figure 206: User Group Permission Detail

- 4) Click on the 'Save' button to save the edited permissions as shown below



Figure 207: Save Button

5.5 Contact

In case of any quires/problem contact on the following emails

- jkazmi399@gmail.com
- cr.zahid@gmail.com

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

The aim of this project was to provide a System to Bahria University that improve learning processes and improve other communication processes. The development focused primarily on the usability of such an application, under strict usability guidelines, and the functionality needs of the user. It was recognised early in the project that a usable solution must utilise a database to drive the content of the application.

The system produced successfully met the usability and functional requirements of the user and gained their acceptance. The roll out of this application onto the World Wide Web shows the project was successful. The user testing and evaluation of the application did however highlight room for improvements and expansion. Although all the user requirements were. The application could therefore be developed further to include additional functionality and allow the user greater control.

REFERENCES

Books:

[1] Jeffrey L. Whitten, *System Analysis and Design Methods*, 7th Edition, McGraw-Hill, New York, 2007.

Technical Report and Theses:

[2] Paul Chapman, *Software Engineering with Web Development Frameworks*, BSc Computer Science Final Report, University of Leeds, Minerva, 2007.

[3] Gary Ingham, *A Content Management System to support small sports clubs*, BSc Computing (Industry) Final Report, University of Leeds, Minerva, 2006.

Electronic Sources from Internet:

[4] Wikidot, “*Agile-Feature Driven Development*”, 2011, Available at: sliitmscfdd.wikidot.com/introduction-to-fdd

[5] Martin, “*FDD & Web Development*”, 2003, Available at: www.featuredrivendevelopment.com/node/550

[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

Business Rules for Correspondence system

1. All the minutes that forwarded to Director Campus must be go through Deputy Director
2. Faculty members can forward minute(s) maximum to HOD level
3. All department heads can forward minute(s) maximum to Director Campus level
4. HODs of academic departments can forward minute(s) maximum to department Dean level
5. Minutes to Dean level must go through Deputy Director and Director

Requirement List

R1 → GUI should compatible with Chrome and Mozilla both.

R2 → Components of the project code will be tested alongside the implementation phase to ensure that they are functional.

R3 → Final, integrated project code will test to ensure that complete project is integrated well and functioning properly.

R4 → Display all the content after loading of web page.

R5 → Server response should be keep in mind while designing data retrieval algorithms.

User Management System (UMS)

R1 → User must be registered to access BOL

R2 → User login

R3 → Reset user password

R4 → Edit User Group Permissions

R5 → View/Search Users

R6 → Active/Block a User

R7 → All the passwords should save in encrypted form

Learning Management System (LMS)

R1 → Create a Virtual Class

R2 → Upload resources in virtual class

R3 → Download resource from virtual class

R4 → Open a Virtual Class

R5 → Join a virtual class

R6 → Make an Announcement in Virtual Class

R7 → View Announcements in virtual class

R8 → Search for Alumni

R9 → Upload study resource in library

R10 → Download resource from Library

R11 → There should be a popup window for uploading

R12 → There should be type of users like faculty, student, alumni etc.

R13 → Students who are graduated will be promoted automatically to the type of alumni

Correspondence System (CS)

R1 → Creating a new Minute

R2 → Give Feedback of a Minute

R3 → Printing a Minute

R4 → Viewing a Minute

R5 → Save a Minute

R6 → Correspondence system should be able to note the feedback time of receiver and use it at the time of printing.

R7 → In correspondence system only related people can access certain communication

R8 → In correspondence system privileges of sending correspondence should be strict to internal correspondence policy of Bahria University