Final Year Project Management System

By:

Muhammad Muneeb
7757
Umair Qamar
10776
Bilal Hassan
7471

Supervised by:
Zeeshan Sabir

Faculty of Computing
Riphah International University, Islamabad
Fall 2015

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of

the Degree of

Bachelors of Science in Software Engineering

Faculty of Computing
Riphah International University, Islamabad

Date: [date of external examination]

Final Approval

This is to certify that we have read the report submitted by ***name of student(s) (CMS #)***, for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Software Engineering (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Software Engineering (BSSE).

Committee:

1 _____

[Name Supervisor]
(Supervisor)

2 _____

[Name of External Examiner]
(External Examiner)

3 _____

[Name of HOD/chairman]
(Head of Department/chairman)

Declaration

We hereby declare that this document "**Final Year Project Management System**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **Mr. Zeeshan Sabir**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

_____

Muhammad Muneeb

7757

_____

Umair Qamar

10776

_____

Bilal Hassan

7471

Dedication

Conventionally we would like to dedicate our project to our parents and to all of our teachers who ever taught us and also to also our Mentor Dr. Sheheryar Malik and to all of our batch fellows and seniors who helped us a lot during the whole degree.

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We do express our heartfelt gratitude in deep humility to the H.O.D of the Faculty of computing Assistant Professor Abdul Mateen, who has provided us with all facilities to conduct or project and immense co-operation. We do take the opportunity to acknowledge and honor the contribution of our supervisor Dr. Shahryar Malik (Assistant Professor) Riphah international university who has guided us all along by his wise lead, benevolent direction, suggestions and time worthy interaction with us. We express our sincere thanks to Mr. Muhammad Saud Khan for his encouragement and for academic guidance regarding the completing this project. The project would have never been a success without the support of our family, our seniors, friends and Supervisor who have technically as well as morally supported us throughout the project.

Muhammad Muneeb

7757

Umair Qamar

10776

Bilal Hassan

7471

Abstract

Many of the projects fail because of barriers of communication between supervisor and students. Students mostly face the problem of meeting the supervisor, timeline and deadline mostly supervisors don't have the updated document of the project. Project management tool is automated software that keeps track and repository of all projects and manages all the processes involved in projects. Final Year Project Management System manages current and previous final year projects. All activities/processes involved in final year projects will be managed. Final Year Project Management System follows complete final year project's lifecycle. It handles Project repository and their documents, it keep track of Notifications and deadlines.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Chapter 1:
Introduction

Final Year Project Management System is a web based solution designed for Riphah International University which automates tasks of Final Year Projects.
FYP Management System manages all the processes involved in Final year projects and also keep repository of all projects.

1.1.Background

Project management tool is automation of all the tasks involved in Final Year Project lifecycle.

1.2.Motivations and Challenges

There are some systems implemented in some universities of Pakistan, but most of the Universities handle FYP task manually or through their university's Virtual Learning Environment (VLE) or their Course Portals.
In Riphah International University there is no customized solution available, most of the tasks are handled manually and some with help of Moellim VLE. This motivate us to build and automated system to handle almost all the tasks of FYP lifecycle.

1.3.Goals and Objectives

To create an automated system that follows complete final year project's lifecycle. And maintains repository of all projects.

1.4.Solution Overview

Final Year Project Managements system follows complete final year project's life cycle. It is automation of manual tasks in FYP. It maintains Project repository and their documents. It handles Notifications and deadlines. It is fully automated system with online availability. It will also maintain student and supervisor meeting logs. It also allows external examiner to review and evaluate projects

1.5.Report Outline

This project report covers in detail all the aspects of the system. For the sake of understanding and clarity, this report is divided into following chapters:

- Chapter 1: Introduction
- Chapter 2: Literature Survey
- Chapter 3: Requirement Analysis
- Chapter 4: System Design

Chapter 2:
Literature/Market Survey

Chapter 2:
Literature/Market Survey

2.1. Introduction

The market survey tells us that are very few universities have system to handle Final year projects while most of universities in Pakistan either uses their default Virtual Learning Environment (VLE) to deal with Final Year Project tasks or handle all tasks manually. However there is no custom made solution available for management of Final Year

2.2. Technologies Overview

System made for different Universities are web-based. These system follow their university guidelines and rules.

2.3. Summary

In Riphah International University Final Year Project Management is handled by Moellim VLE. But it is not enough to handle Final Year Projects processes, since it is designed to handle course subjects and not the final year project. So keeping that in mind we are going to create this Final Year Project Management System

Chapter 3:
Requirement Analysis

3.1. Introduction

Project management tool is automated software that keeps track and repository of all projects and manages all the processes involved in projects. Final Year Project Management System manages current and previous final year projects. All activities/processes involved in final year projects will be managed

3.2. Problem Scenarios

Every semester many of the projects fail because of barriers of communication between supervisor and students. Students mostly face the problem of meeting the supervisor, timeline and deadline mostly supervisors don't have the updated document of the project. Students usually don't have idea of projects that have already been made, since there is no repository of such projects are available

3.3. Functional Requirements

Final Year Project Management System has different users with different requirements as below

No.	Requirement	Detail
Student		
3.3.1.	Repository	Repository will contain all projects
3.3.2.	Make Group	Student can make a new group or join an existing group
3.3.3.	Choose Supervisor	Student will be able to choose from available list of supervisors
3.3.4.	Upload Deliverables	Student will be able to upload different artifacts of their projects
Supervisor		
3.3.5.	Choose group	Supervisor will choose groups to supervise from his request list
3.3.6.	Schedule Meetings	Supervisor will be able to schedule meetings with

		students
3.3.7.	Evaluate	Supervisor can evaluate SDP-1
3.3.8.	Add Meeting Logs	Supervisor will add meeting logs with students to the system
Coordinator		
3.3.9.	Create/Manage batch	Coordinator will create a new batch or manage existing created batch
3.3.10.	Register Student	Coordinator will register students to the batch
3.3.11.	Manage/Create groups	Coordinator will manage already created groups or create groups of students who are not with any group
3.3.12.	Set Deadlines	Coordinator will set deadlines to deliver different artifacts of projects
3.3.13.	Generate Notifications	Coordinator will generate different notifications related to the FYP and these notifications will be sent to all students
3.3.14.	Send Emails	Coordinator will be able to send emails to all the users of the system
3.3.15.	Register External Examiner	Coordinator will register external examiners
3.3.16.	Manage Faculty and their roles	Coordinator will register new faculty or manage existing faculty and set their roles
3.3.17.	Set Configurations	Will set configurations for each Batch
3.3.18.	Manage Repository	Will manage repository

Table 3.4.1 Functional Requirements

3.4. Non-Functional Requirements

Requirement	Description
Availability	The system is available 24/7 except in the case of unexpected condition like failure of database etc.
Usability:	System is user friendly, responsive, mobile friendly and it is designed so that interaction of user will be maintained.
Accessibility	Since this system is mobile friendly and responsive so users will be able to access from every device with internet connection.
Modifiability	Our system is prone to future changes. There will be a panel to change and set rules (i.e. Maximum members in a group etc.).Users can be added or removed. Records can be edited also.
Failure Management	To avoid failure we will use server validation using PHP and JavaScript validation on client side. System will also maintain a cloned database so if any failure occur it will be able to restore the database
Consistency	To ensure consistency we are using same CSS stylesheets on almost all pages

Table 3.4.2 Non-Functional Requirements

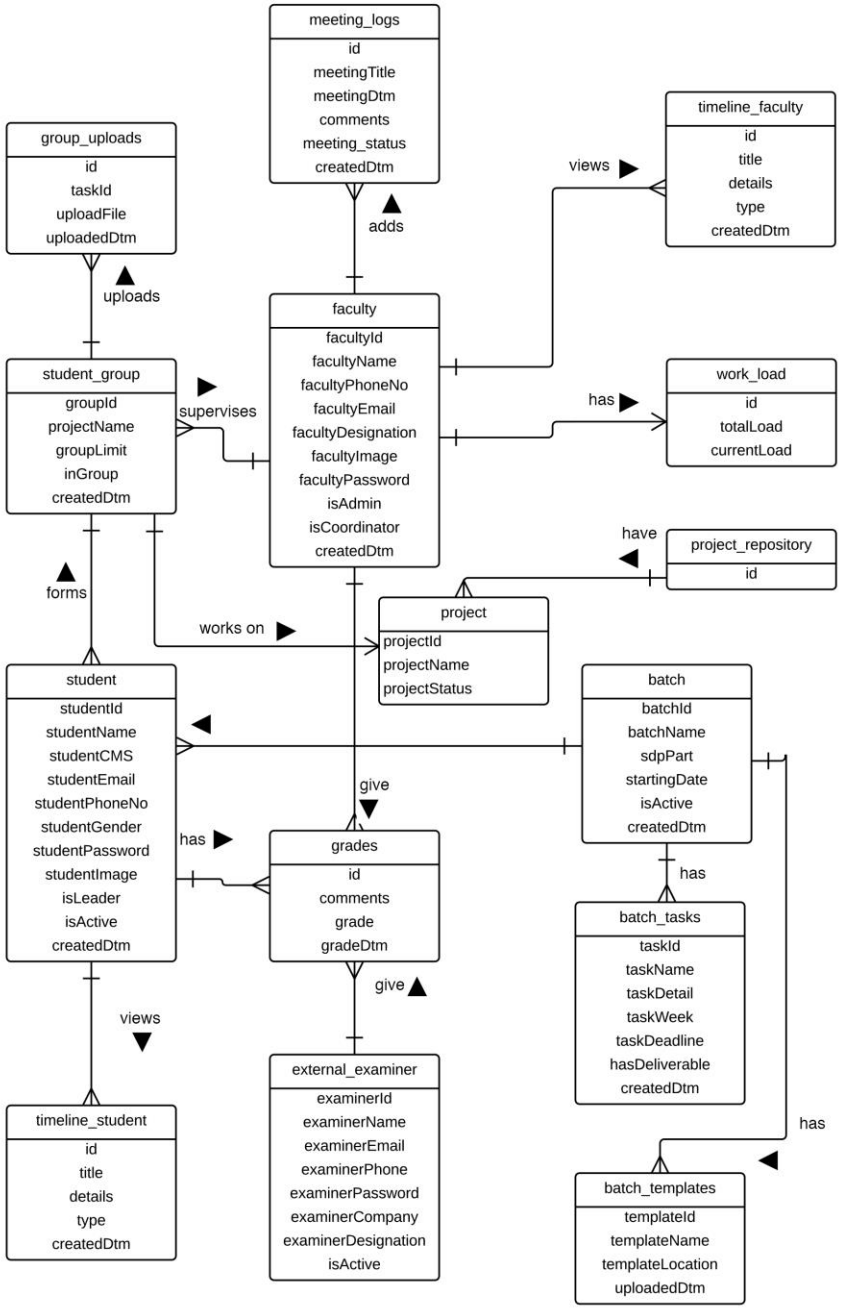
Chapter 4:
System Design

4.1. Introduction

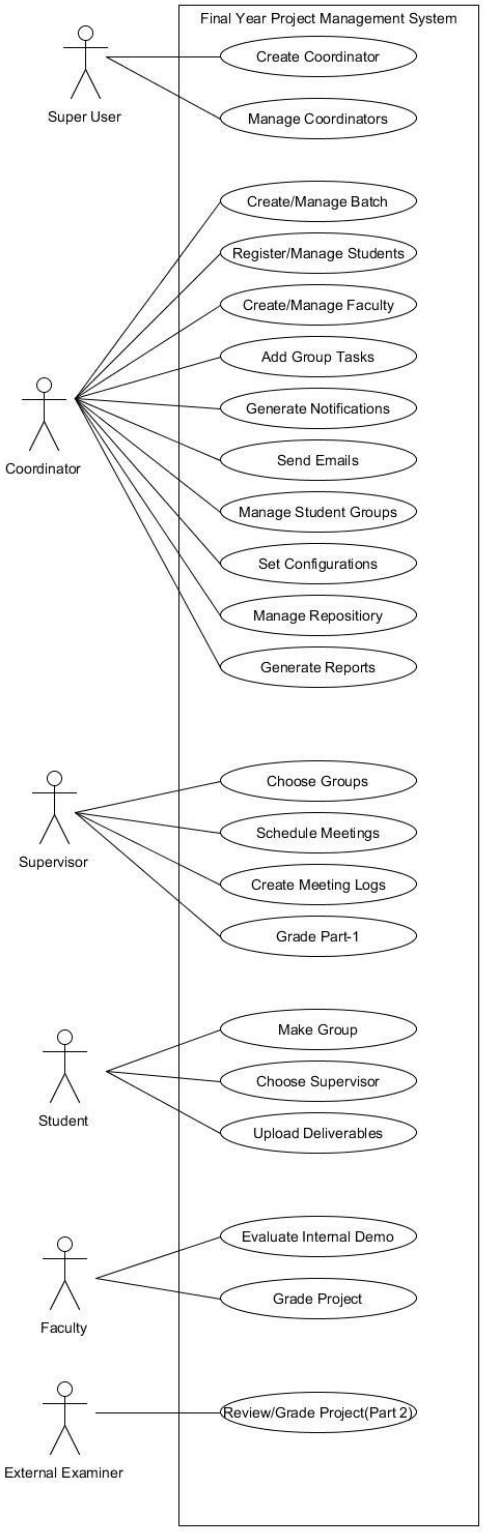
This chapter provides the detail design of “Final Year Project Management System” that encloses its entities, information flow and event flows. To understand the system in detail this chapter provides Use-case, Details of Use-cases and Entity Relationship diagrams that starts from basic architecture to complex systems information and event flows.

4.2. Architectural Design

4.2.1. Entity Relationship Diagram



4.2.2. Use-case Diagram



4.3. Detailed Design

4.3.1 Detailed Use-Cases

<table>
<tr><td colspan="2">Use case: Create Batch</td></tr>
<tr><td colspan="2">Brief Description: Coordinator create batch by setting its year and starting date</td></tr>
<tr><td colspan="2">Primary actors: Coordinator</td></tr>
<tr><td colspan="2">Preconditions: Coordinator is logged in</td></tr>
<tr><td colspan="2">Main Flow:</td></tr>
<tr><td><u>Actor:</u>
1. Coordinator enters batch and year
2. Coordinator submits the form</td><td><u>System:</u>
3. System reads the inputs
4. System creates a batch</td></tr>
<tr><td colspan="2">Post condition: A batch is successfully created</td></tr>
<tr><td colspan="2">Alternative flow:</td></tr>
<tr><td><u>Actor:</u>
1.a) Batch already exists</td><td><u>System:</u>
Systems does not create new batch</td></tr>
<tr><td colspan="2">Special Requirements:</td></tr>
</table>

<table>
<tr><td colspan="2">Use case: Register Students</td></tr>
<tr><td colspan="2">Brief Description: The coordinator will register new students to newly created batch</td></tr>
<tr><td colspan="2">Primary actors: Coordinator</td></tr>
<tr><td colspan="2">Preconditions: Coordinator is logged in and Batch is created</td></tr>
<tr><td colspan="2">Main Flow:</td></tr>
<tr><td><u>Actor:</u>
1. Coordinator will enter details of students
2. Coordinator will select the batch
3. Coordinator submits the form.</td><td><u>System:</u>
4. System reads the inputs
5. System validates the inputs
6. System adds the student to the database</td></tr>
<tr><td colspan="2">Post condition: A student is registered to a batch successfully</td></tr>
<tr><td colspan="2">Alternative flow:</td></tr>
<tr><td><u>Actor:</u>

2.a) There is no batch created
3.a) User is already registered</td><td><u>System:</u>
System generates error
System does not register student</td></tr>
<tr><td colspan="2">Special Requirements:</td></tr>
</table>

Use case: Set Deadlines	
Brief Description**:** Coordinator will set deadlines for each batch	
Primary actors: Coordinator	
Preconditions: Coordinator is logged in and batch is selected	
Main Flow:	
Actor: 1. Coordinator enters deadline date, type of deadline, and details of deadline 2. Coordinator submits the deadline	System: 3. System reads the inputs 4. System creates deadline for the batch
Post condition: A deadline is successfully created and added to the batch	
Alternative flow*:*	
Actor: 1.a) Coordinator enters wrong input	System: Systems does not create deadline
Special Requirements:	

Use case: Generate Notifications	
Brief Description**:** Coordinator will generate notification that will be sent to selected users	
Primary actors: Coordinator	
Preconditions: Coordinator is logged in	
Main Flow:	
Actor: 1. Coordinator enters type of notification, select users, enter details of notification 2. Coordinator sends the notifications	System: 3. System reads the inputs 4. System creates a new notification and send it to selected users
Post condition: A notification is generated and sent to selected users timelines	
Alternative flow*:*	
Actor: 1.a) Coordinator enters wrong input	System: Systems does not creates notifications
Special Requirements:	

Use case: Choose Groups	
Brief Description**:** Supervisor will choose groups of students he wants to supervise from his request panel	
Primary actors: Supervisor	
Preconditions: Supervisor is logged in and students group have sent him request	
Main Flow:	
<u>Actor:</u> 1. Supervisor will open his request panel 3. Supervisor will accept the requests of groups he wants to supervise	<u>System:</u> 2. System shows list of groups which have sent him request 4. System will set the supervisor of group
Post condition: A supervisor has successfully selected a group to supervise	
Alternative flow*:*	
<u>Actor:</u> 3.a) Supervisor exceeded the limit of groups he's allowed to supervise	<u>System:</u> Systems system will not allow supervisor to accept any more requests
Special Requirements:	

Use case: Schedule Meetings	
Brief Description**:** Supervisor will set meeting schedule for students to meet	
Primary actors: Supervisor	
Preconditions: Supervisor is logged in and is supervising groups	
Main Flow:	
<u>Actor:</u> 1. Supervisor enters date and time for meeting, selects group to send 2. Supervisor enters meeting objectives 3. Supervisor submits the form	<u>System:</u> 4. System reads the inputs 5. System generates a meeting notification and sends it respective groups
Post condition: A meeting notification is sent from supervisor to selected group	
Alternative flow*:*	
<u>Actor:</u> 1.a) Supervisor entered wrong input	<u>System:</u> Systems does not sends notification
Special Requirements:	

Use case: Join Group	
Brief Description**:** Student will select a group to join	
Primary actors: Student	
Preconditions: Student is logged in and group is already created by another student	
Main Flow:	
Actor: 1. Student selects a group from list	System: 2. System reads the inputs 3. System will send request notification to the group leader
Post condition: A student is successfully added to required group	
Alternative flow*:*	
Actor: 1.a) There are no groups created 1.b) Group limit is exceeded	System: System shows error and ask student to create a new group System does not add student to the group
Special Requirements:	

Use case: Choose Supervisor	
Brief Description**:** Student will choose supervisors from list of available supervisor	
Primary actors: Student	
Preconditions: Student is logged in and has formed a group	
Main Flow:	
<u>Actor:</u> 1. Student opens list of available supervisors 3. Student selects supervisor	<u>System:</u> 2. System will display list of supervisors 4. System sends this groups request to the selected supervisor
Post condition: A group request has been sent to the supervisor	
Alternative flow*:*	
<u>Actor:</u> 1.a) There is no supervisor available	<u>System:</u> Systems asks user to contact with course coordinator
Special Requirements:	

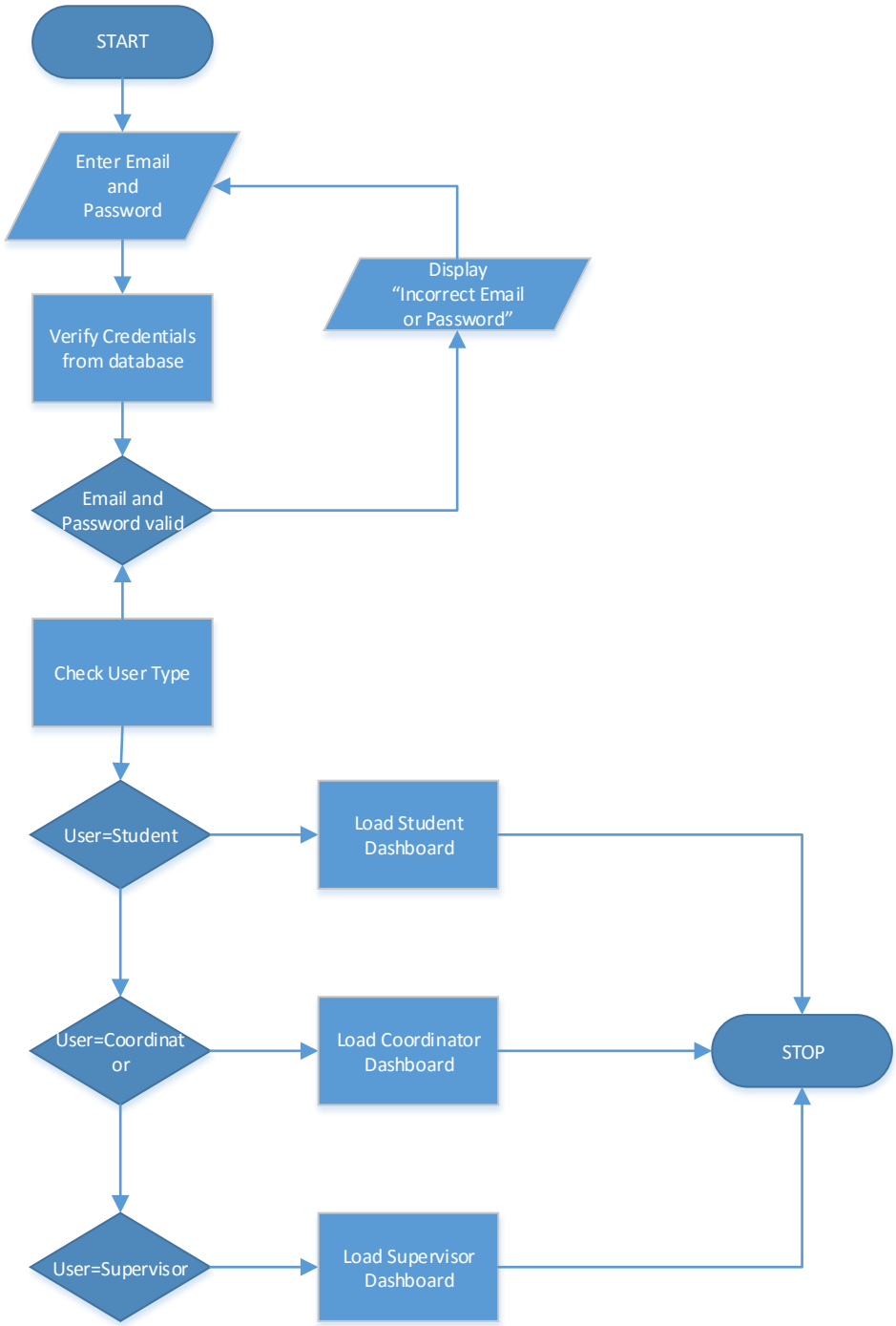
Use case: Grade Students	
Brief Description**:** External Examiner will grade SDP-2 students	
Primary actors: External Examiner	
Preconditions: External Examiner is logged in	
Main Flow:	
Actor: 1. External Examiner selects group to Grade 2. External Examiner enters Grade and submit	System: 3. System reads the inputs 4. System sets the grade for selected group
Post condition: A group has been successfully graded	
Alternative flow*:*	
Actor: 2.a) External Examiner enters wrong input in Grade	System: Systems does not sets grade
Special Requirements:	

Chapter 5: Implementation

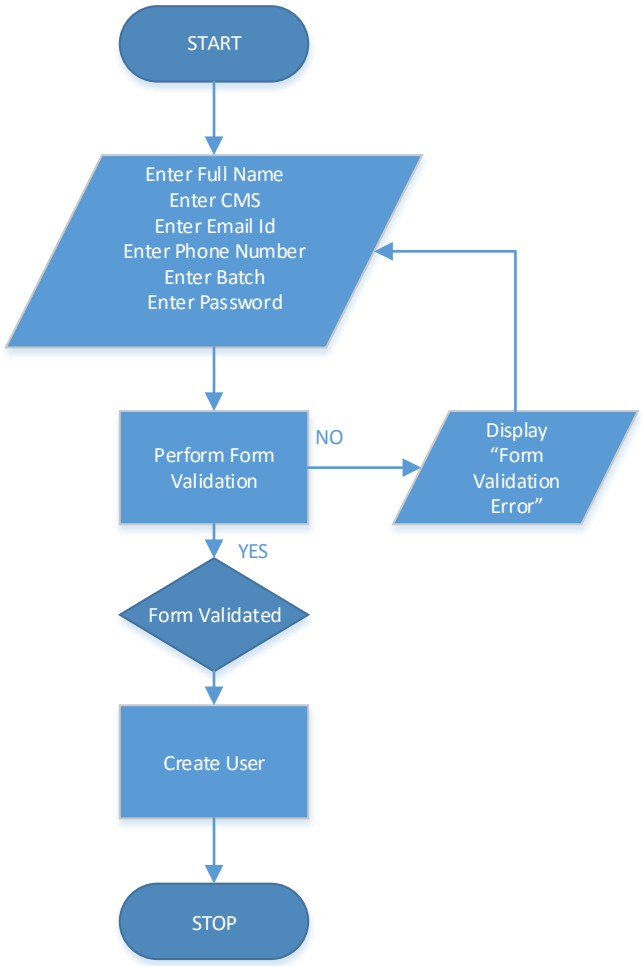
5.1. Flow Control/Pseudo codes

The system has Student Module, Supervisor Module and Coordinator module. All users have their different roles and responsibilities.

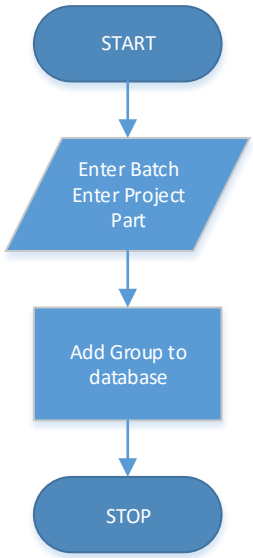
5.1.1 User Login Flow control



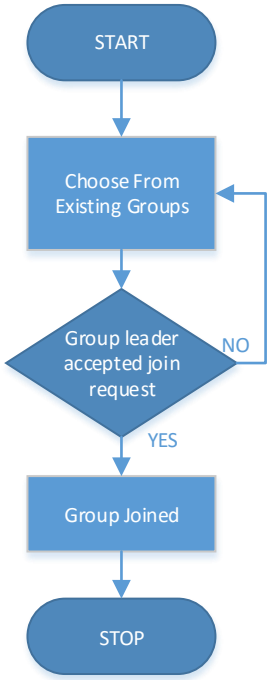
5.1.2 Register Students (Coordinator Role)



5.1.3 Create Group (Student Role)



5.1.4 Join Group (Student Role)



5.2. Components, Libraries, Web Services and stubs

We are using

- Bootstrap 3.3.5
- FPDF
- JQuery
- Font Awesome
- Kendo UI
- Sendgrid API

5.3. Best Practices / Coding Standards

HTML5

HTML5 is used since it is recommended by W3C

W3 Validation

All HTML pages are verified against the W3C validator to ensure that the markup is well formed.

Attributes and Tags

All tags and attributes are written in lowercase. Additionally, attribute values are also lowercase

Indentation

Proper indentation is used in PHP and HTML to reflect logical structure.

Externalized CSS

All the stylesheets are externalized and used in HEAD of HTML

JavaScript at footer

To make page load quickly as possible all JavaScript who perform their functionality after page has been loaded are placed in footer.

Responsiveness

Website is responsive and mobile friendly to deliver content to mobile devices. The principal objective is to improve the user experience of the Web when accessed from such devices.

5.4. Deployment Environment

Our system is PHP base web application so it will be deployed on a webserver.

5.5. Summary

In this chapter we have defined the work flow and pseudo code that describes the general Flow of the system. Also we have described the best practices for the languages that are being used to develop the system. All the coding conventions and standards that will be followed in the course of development are described in the chapter.

Chapter 6:
Testing and Evaluation

6.1. Introduction

Software development is a complex task. It is not possible to use any pre-defined approach that shall result in good quality software product. Software developers always try to develop bug free software products. But due to the deviation and complexity of software requirements it is not possible to develop a bug free product.

We can only remove bugs from software by testing it. Testing is meant to break the system and find bugs in it.

6.2. List of Test Scenarios

6.2.1 Test Case 1: User Login

Requirement Description:
User will enter email address and password to login. System should check the email, if the email is correct then the user will be login and if email is incorrect than email is invalid.

Input: password
Valid:

Email = ’saudkhan@riu.edu.pk‘

Invalid:

Email= ‘saudkhan’
Email = ‘ ’

Test Case #	1	2	3
Email		Saudkhan	saudkhan@riu.edu.pk
P. Tested	Email = ‘ ’	Email= ‘saudkhan’	Email = ’saudkhan@riu.edu.pk‘
Exp Output	Error	Error	OK

Input: password
Partition Input Values
Valid:

Password = 123

Invalid:

Password ≠ 123

Test Case #	1	2	3
Email		Saudkhan	saudkhan@riu.edu.pk
P. Tested	Email = ‘ ’	Email= ‘saudkhan’	Email = ’saudkhan@riu.edu.pk‘
Exp Output	Error	Error	OK

6.2.2 Test Case 2: Create Batch

Requirement Description:
Coordinator will created batch by selecting year and Batch (Fall or Spring) and start date of semester

Input: start date
Valid:

start date = 04/06/2016

Invalid:

start date = 123
start date = April 06,2016

Test Case #	1	2	3
start date	123	April 06,2016	04/06/2016
P. Tested	start date = 123	start date = April 06,2016	start date = 04/06/2016
Exp Output	Error	Error	OK

6.2.3 Test Case 3: Register Student

Requirement Description:
Coordinator will register students by filling details of students

Input: CMS
Valid:

CMS ≥ 000001 AND CMS ≤ 99999

Invalid:

Outside valid boundary class

Test Case #	1	2	3
CMS	0	123456	10776
Boundary	CMS ≥ 000001	CMS ≤ 99999	CMS ≥ 000001 AND CMS ≤ 99999
Exp Output	Error	Error	OK

Input: CMS
Valid:

CMS = [7471, 7757, 10776]

Invalid:

CMS = [abc, 123a, a123]

Test Case #	1	2	3
CMS	123a	abc	10776
P. tested	CMS = [123a]	CMS = [abc]	CMS = [7471]
Exp Output	Error	Error	OK

Input: Name
Valid:

Name = [Umair, Muneeb, Bilal, Ali]

Invalid:

Name = [Um@ir, Muneeb42]

Test Case #	1	2	3
Name	Muneeb42	Um@ir	Bilal
P. tested	Name = [Muneeb42]	Name = [Um@ir]	Name = [Bilal]
Exp Output	Error	Error	OK

Input: Name
Valid:

Name Length >= 8 & Name Length <= 20
Invalid:

Name Length <8
Name Length >20

Test Case #	1	2	3	4
Name		abc	Umair Qamar	abcdefghijklmnopqrstuvwxyz
P. tested	Name Length >= 8	Name Length >= 8	Name Length >= 8 & Name Length <= 20	Name Length <= 20
Exp Output	Error	Error	OK	Error

Input: Email
Valid:

Email = [umairqamar@live.com]

Invalid:

Email = [umairq.com, 123umairq]

Test Case #	1	2	3
Email	Umairq.com	123umairq	umairqamar@live.com
P. tested	Email != [umairqamar@live.com]	Email != [umairqamar@live.com]	Email = [umairqamar@live.com]
Exp Output	Error	Error	OK

Input: Phone Number
Valid:

Length = > 11 AND Length <=12

Invalid:

Length < 11 OR Length >12

Test Case #	1	2	3
Phone Number		03458521478999	03458541454
P. tested	Length < 11	Length >12	Length = > 11 AND Length <=12
Exp Output	Error	Error	OK

6.3. Performance and Evaluation

Following test cases were performed to make sure that the intended functionality and performance of “Hotel in Clouds” works properly. These types of results take developers to a decision where they may decide that system is acceptable or needs further refinement.

6.4. Summary

Testing is meant to break a system. We do testing to gain confidence in the correctness of a part or a product or to check if there are any errors in a part or a product. We do testing because we that we are going to make error.

Chapter 7:
Conclusion and Outlook

Chapter 7:
Conclusion and Outlook

7.1 Introduction

This chapter is summary of overall project, our achievement in this project. What we have learnt as working in team to build a Software system. The difficulties we faced during the whole project lifecycle and how we faced and solved this problem as a team by doing extensive research on that problem and finding multiple solutions to it and using an optimal solution that fits our need. Furthermore this chapter also discusses the future ideas that can be added to the project.

7.2 Achievements and Improvements

The major achievement in our eyes is the level of understanding that we as a team developed during the course of development. We as a team have learned many things related to the field of Web based systems and have gathered a lot of knowledge on it. As a team we have looked upon many problems and faced them together as a team and every problem taught us a new lesson academically and professionally. The project has helped us all develop individual skills in problem solving and researching. After the development this recommendation system we are able to understand how these systems work and how to develop such systems. Apart from academic perspective we all have developed skills to work as a team, to communicate and discuss problems and be responsible for what is we are saying.
We have learnt a lot about Web base technologies including PHP, JavaScript, JQuery Ajax, HTML 5, CSS 3, Bootstrap and different other tools and libraries

7.3 Critical Review

This project is a helping tool to students, supervisor and Coordinator that involve in Final Year Project Management System. This system is designed to make life easier for users that are related to Final Year Project

7.4 Future Recommendations/Outlook

This system is developed for Riphah International University to its requirements. In future this system can replace Moellim VLE system, for Final Year Project students. As this system is made to help and manage all the lifecycle of Final Year Project. Furthermore this project can be tweaked and changed to requirements of other universities.

7.5 Summary

FYP Management System helps students in their Final Year Project. There is no such system already available designed specifically for this purpose in Riphah International University. This system can help users of system in many way and make things easier for them.