****

**COMSATS Institute of Information Technology,**

**Park Road, Chak Shahzad, Islamabad Pakistan**

**Automation of FYP Process of CIIT - II**

***By***

**Abdul Wahab Mazhar Shah CIIT/FA11-BCS-209/ISB**

**Ayza Habib CIIT/FA11-BCS-225/ISB**

***Supervisor*Dr. Muhammad Asim Noor**

***Bachelor of Science in Computer Science (2011-2015)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

** COMSATS Institute of Information Technology,**

**Park Road, Chak Shahzad, Islamabad Pakistan**

**Automation of FYP Process of CIIT - II**

***By***

**Abdul Wahab Mazhar Shah CIIT/FA11-BCS-209/ISB**

**Ayza Habib CIIT/FA11-BCS-225/ISB**

***Supervisor*Dr. Muhammad Asim Noor**

***Bachelor of Science in Computer Science (200x-20xx)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

** COMSATS Institute of Information Technology,**

**Park Road, Chak Shahzad, Islamabad Pakistan**

**Automation of FYP Process of CIIT - II**

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelors of Science in Computer Science (2011-2015)***

**By**

**Abdul Wahab Mazhar Shah CIIT/FA11-BCS-209/ISB**

**Ayza Habib CIIT/FA11-BCS-225/ISB**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Abdul Wahab Mazhar Shah Ayza Habib

--------------------------- ---------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Automation of FYP Process of CIIT - II” was developed by “**Abdul Wahab Mazhar Shah (CIIT/FA11-BCS-209)**” and “**Ayza Habib (CIIT/FA11-BCS-225**)” under the supervision of “Dr. Muhammad Asim Noor” and co supervisor “Mr. Zaheer ul Hussain Sani” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**Executive Summary**

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility ir-respective of the fact that any event may happen or not. Moreover they are usually pointing at a specific user defined locations so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn’t require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Muhammad Asim Noor” and our Co-Supervisor “Mr. Zaheer ul Hussain Sani”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Abdul Wahab Mazhar Shah Ayza Habib

--------------------------- ---------------------------

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **CIIT** | COMSATS Institute of Information Technology |
| **LMS** | Learning Management System |
| **FYP** | Final Year Project |
| **FYPAS** | Final Year Project Automation System |
| **PCM** | Project Committee Member |
| **PCS** | Project Committee Secretary |
| **PC** | Project Committee |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 2

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Porject Background](#_Toc268523780) 3

[1.4 Literature Review](#_Toc268523780) 3

[1.5 Methodology and Software Life Cycle](#_Toc268523782) 5

**Problem Definition** 6

[2.1 Purpose](#_Toc268523795) 7

[2.2 Product Functions](#_Toc268523796) 7

[2.3 Proposed Architecture](#_Toc268523804) 7

[2.4 Project Deliverables](#_Toc268523806) 8

[2.5 Operating Environment](#_Toc268523807) 8

[2.6 Assumptions and Dependencies](#_Toc268523808) 8

**Requirement Analysis** 9

[3.1 Functional Requirments](#_Toc268523823) 10

[3.2 Non – Functional Requirments](#_Toc268523825) 11

[3.2.1 Usability](#_Toc268523787) 11

[3.2.2 Reliability](#_Toc268523787) 11

[3.2.3 Performance](#_Toc268523787) 11

[3.2.4 Supportability](#_Toc268523787) 11

[3.2.5 Design Constraints](#_Toc268523787) 11

[3.2.6 Licensing Requirements](#_Toc268523787) 11

[3.3 Use case Model](#_Toc268523823) 12

[3.3.1 Use Case Diagarm](#_Toc268523787) 12

[3.3.2 Actors Discription](#_Toc268523787) 16

[3.3.3 Use Case Discription](#_Toc268523787) 17

**The Design** 21

[4.1 UML Structural Diagrams](#_Toc268523830) 22

[4.1.1 Component Diagram](#_Toc268523787) 22

[4.1.2 System Component Diagram](#_Toc268523787) 24

[4.1.3 Package Diagram](#_Toc268523787) 25

[4.1.4 Deployment Diagram](#_Toc268523787) 26

[4.2 UML Behavioral Diagrams](#_Toc268523830) 27

[4.2.1 Activity Diagrams](#_Toc268523787) 27

[4.2.2 State Machine Diagrams](#_Toc268523787) 29

[4.3 UML Interaction Diagrams](#_Toc268523830) 30

[4.3.1 Sequence Diagrams](#_Toc268523787) 30

[4.4 Node Structure](#_Toc268523830) 31

[4.5 Communication Design Protocol](#_Toc268523830) 32

**Imlementation** 33

[5.1 Communication Protocol Implementation](#_Toc268523830) 34

[5.2 PC Application Implementation](#_Toc268523830) 36

[5.3 Embedded Application Implementation](#_Toc268523830) 38

[5.4 Wireless Sensor Application Implementation](#_Toc268523830) 51

**Testing and Evaluation** 52

[6.1 Verification](#_Toc268523830) 53

[6.1.1 Functional Testing](#_Toc268523787) 53

[6.1.2 Static Testing](#_Toc268523787) 57

[6.2 Validation](#_Toc268523830) 57

[6.3 Usability Testing](#_Toc268523830) 57

[6.4 Unit Testing](#_Toc268523830) 57

* 1. [Integration Testing](#_Toc268523830) 57

6.6 [System Testing](#_Toc268523830) 57

**GUI** 58

**Future Work** 70

**References** 72

**List of Figures**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

# Chapter 1: Introduction

This chapter provides the overview of the project. The first paragraph of every chapter should provide the chapter summary.

# Brief

The system is an upgrade of currently used CIITFYP Portal developed by FYP students last year. So it’s an upgrade or related version of an already built system. Proposed system is extracted from this existing system however some optimizations, improvements and enhancements are required by the relevant stakeholders. Students will be able to submit their documents and work online within the given deadlines. PC Members and Supervisor can view student’s work online, grade their work, make changes if any, send notifications and can communicate with the students. This system will be developed using ASP.NET Framework with backend C# as the server side language with MS SQL Server for the database. HTML and JavaScript will be used on client side.

# Relevance to Course Modules

**1.2.1 Introduction to Computer Programming**

This course is appropriate to our project because it provides a basis for programming like Program control if and else statements; For, while and do while loops; Switch statements; Compound statements; Functions and Parameters; Arrays declaration; Array passing to functions; String; Strings manipulating; pointers; Recursion; Introduction to abstract data type; Structure declaration and initialization; Accessing members of Structure; File processing reading, writing, randomly accessing data structures.

**1.2.2 Object Oriented Programming**

All the main concepts of Object Oriented Programming were used in this System like Inheritance, Constructors, and Objects etc. We try to make our code in professional and systematic way. All the Coding is done in “Classes” also we make different Packages according to functionality of Classes.

**1.2.3 Web Engineering**

All the web concepts and languages we studied in this course will be implemented in this project. For example Server side and client side validations, data insertion to database, data retrieval from the database etc… to generate reports. HTML, CSS, JavaScript will be used for client side validations and the advance concepts we studied in this course will be helpful to make our system more interactive and user friendly.

**1.2.4 Human Computer Interaction**

Interface is the basic requirement of every system which helps the user to communicate with the system effectively. In our course Human Communication Interface (HCI) we learnt the basic things regarding interface and affordance of the user before designing the interface for a specific system. These concepts proved really helpful for us.

**1.2.5 Visual Programming**

A visual programming language (VPL) is any programming language that lets users create programs by manipulating program elements graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols used either as elements of syntax or secondary notation. We are developing a system based upon the above mentioned things which we learnt in this course.

**1.2.6 Database Management System**

Database works as a backbone for any type of applications where information storage is involved. Almost every application now a days use databases in which data mining, web applications are on top. We will use MS SQL Server for the database of our application. The core concepts of this course like writing SQL Quires, store procedures, creating views writing triggers, relations ships between tables etc… we will be implementing in our system.

# Project Background

The system is an upgrade of currently used CIITFYP Portal developed by FYP students last year. So it’s an upgrade or related version of an already built system. Proposed system is extracted from this existing system however some optimizations, improvements and enhancements are required by the relevant stakeholders. To meet the requirements of the relevant stakeholders we planned to upgrade the portal that aggregates information from multiple sources and makes that information available to various users by enhancing department’s efficiency and productivity. The system will provide a platform to students and supervisors to communicate easily during all the vital processes of final year project. Students will be able to submit their documents and work online within the given deadlines. PC Members and Supervisor can view student’s work online, grade their work, make changes if any, send notifications and can communicate with the students.

# Literature Review

Current trend that is being used by the students to submit different milestones to their supervisors is in hard forms. For instance student submit hardcopies of SRS/SDS/Scoped document which are checked by PC Members and remarks are given on portal as to correct or change that content and resubmit it again there is a need of improvement in marking of the documents and their evaluation. Currently there is no mean of communication between students and project supervisors however communication in an education portal environment involves students, supervisors and PC members. The portal can provide its users with easier access to information as well as with information that is more relevant to them as Lack of delayed communication can hamper progress and slow down the rate at which work is done. Other modules like Project directory and conference room reservations require work to be done.

# Analysis from Literature Review (in the context of your project)

The portal will be able to minimize the work load of the users. Students will able to communicate with their supervisors and vice versa, student will be able to upload their milestones and the respective supervisors able to review and grade the reports and send it back to them.

# Methodology and Software Lifecycle for This Project

A brief discussion of methodology and SDLC model selected for this project.

* + 1. Rationale behind Selected Methodology

We are going to use Rapid Prototyping technique. In rapid prototyping, interactive prototypes are developed which can be quickly replaced or changed as per supervisor’s call. With this technique it is quite possible with adjusting to new requirements as more knowledge is gained and the project progresses. The process is flexible, which does not require to have difficult planning from the very beginning rather it can adapt as the project evolves (new features can be added or current being improved).

**Chapter 2: Problem Definition**

This chapter discusses the precise problem to be solved. It should extend to include the outcome.

# Problem Statement

For any university to manage large number of Final Year Projects manually is a challenging task. There are so many tasks included in the process which are to be achieved throughout the final year. Handling this process manually may involve several problems like student may not be able to submit their documents within deadline; they may miss their presentations; they may not get updates properly from the concerned authorities and so many administrative issues like management of all hard form documents submitted by all the groups; presentation scheduling; documents submission announcements; notifications to students, communication with their respective supervisors etc...

# Deliverables and Development Requirements

Deliverables and development requirements.

**2.2.1 Deliverables**

***2.2.1.1 Project Management Schedule***

A complete Project Management Schedule of all the activities that we have performed in the development of this project.

***2.2.1.2 Project Report***

A complete Project Report that includes Software Requirements Specification, Software Design Specification, GUI Mockups, Test Cases, and other major tasks that we will or have performed.

***2.2.1.3 Source Code***

A CD package including the source code, documentation and presentations.

***2.2.1.4 Product Manual (Optional)***

A complete product Manual that includes the configurations of this project, it also includes the user guide that tells how to use this system.

**2.2.2 Development Requirements**

Following are the requirements that are pre-requisite for running this system.

***2.2.2.1 OS Requirements***

Windows XP with SP3, Windows Vista, Windows 7 or Updated version of Windows.

***2.2.2.2 Application Requirements***

Minimum requirement of system to run is Microsoft Visual Studio 2012 or Updated version of Microsoft Visual Studio.

# Current System (if applicable to your project)

A brief description of an existing system. Figure 2.1 is the sample figure, please follow the same style of numbering and caption for the whole report.



**Figure 2.1: Sample picture**

The following table (Table 2.1) is sample table; please follow the same style of numbering and caption for the whole report.

**Table ‎2.1: Sample Table**

|  |  |  |
| --- | --- | --- |
| **Header 1** | **Header 2** | **Header 3** |
| Text | Text | Text |
|  |  |  |

The following list style is the sample to consistently follow in the whole report.

* List items 1
* List items 2

# Chapter 3: Requirement Analysis

Software Requirements Specification (SRS) report should be included in this chapter.

# Use Cases

**3.1.1 Upload Milestone**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-1 | |
| **Use Case Name:** | Upload Milestone | |
| **Actors:** | | Student |
| **Description:** | | This requirement is used to upload the milestones (SRS/SDS/Report/Presentations) by the students. |
| **Trigger:** | | User is required to Click on “Upload” button to proceed. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have uploaded the document successfully. |
| **Normal Flow:** | | 1. User access the Documents section of the system. 2. The system displays all the options related to that section. 3. Submit your project document appears on screen. 4. User specifically upload the particular document. 5. The system validates the information and determines that validation is passed. 6. The system displays a dialog box to notify user about submission. 7. Success message will be shown and the milestone will be added for evaluation. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Internet connectivity error may occur. * All the required field for uploading a document must be filled, else the error message will be displayed. * File is already uploaded. |
| **Includes:** | | Login Use case. |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

**3.1.2 View Announcements**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-2 | |
| **Use Case Name:** | View Announcements | |
| **Actors:** | | Student |
| **Description:** | | The purpose of this use case is to allow the user to view the announcements about the latest activities regarding project. |
| **Trigger:** | | After successful login this use case triggers the latest announcements for the user to view all the current activities. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. * At least one event should have been created about the announcements. |
| **Post conditions:** | | The user will view latest announcements of his/her profile. |
| **Normal Flow:** | | 1. User will hover on “Announcements” button. 2. User will select “latest Announcements” from the dropdown. 3. System will show the details of the event. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Required announcement cannot be viewed |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user has basic computer knowledge. |
| **Notes and Issues:** | | N/A |

**3.1.3 Search Project by Research Group**

|  |  |  |
| --- | --- | --- |
| **Use Case**  **ID:** | UC-3 | |
| **Use Case**  **Name:** | Search Project by Research Group | |
| **Actors:** | | Admin/Faculty/Student |
| **Description:** | | This use case describes how the user can search for a project using research group name. |
| **Trigger:** | | User select Research group search option. |
| **Preconditions:** | | 1. User must log in to perform this function.  2. Availability of research group name. |
| **Post conditions:** | | 1. User have successfully searched for the required research field. |  |
| **Normal Flow:** | | 1. User clicks on search field of research group name.  2. User enters his search criteria in the fields and click search  3. System validates all search information entered by user  4. System displays a list of all the saved research group names.  5. User selects the desired option. |
| **Alternative**  **Flows:** | | N/A |
| **Exceptions:** | | Research Group not listed or spellings mistake while typing. |
| **Includes:** | | N/A |
| **Special**  **Requirements:** | | N/A |
| **Assumptions:** | | N/A |
| **Notes and**  **Issues:** | | N/A |

**3.1.4 Change Forgot Password**

|  |  |  |
| --- | --- | --- |
| **Use Case**  **ID:** | UC-4 | |
| **Use Case**  **Name:** | Change Forgot Password | |
| **Actors:** | | Student |
| **Description:** | | User needs to retrieve their password in case of forgetting |
| **Trigger:** | | User select forgot password option |
| **Preconditions:** | | * Users account exist in database |
| **Post conditions:** | | * Password retrieved successfully |
| **Normal Flow:** | | 1. User selects forgot option from the login panel  2. User provide his email address in the text box  3. System checks for the data against provided email address in  database  4. System generates an email with reset link against the provided  email address  5. System prompt the user that resetting details have been sent to their  account  6. User open the reset link form their email  7. System prompt for new password  8. User enters the data  9. User click on submit button  10. System validates the form  11. System stores data in database  12. System shows a dialog box notifying user that password has been  changed successfully. |
| **Alternative**  **Flows:** | | N/A |
| **Exceptions:** | | 9a. In step 9 of normal flow if user enters invalid data  1. System compares password fields  2. If data is incorrect system prompt the user to reenter data  3. Use case resumes from step 10 |
| **Includes:** | | N/A |
| **Special**  **Requirements:** | | N/A |
| **Assumptions:** | | User account already exist in database |
| **Notes and**  **Issues:** | | N/A |

**3.1.5 Create Schedule**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-5 | |
| **Use Case Name:** | Create Schedule | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to create schedule for presentations of final year students. |
| **Trigger:** | | User is required to Click on “schedule management” control to proceed. |
| **Preconditions:** | | * User must login to perform this function. * User must have authentication rights. |
| **Post conditions:** | | User have created the schedule successfully. |
| **Normal Flow:** | | 1. User access the schedule management section of the system. 2. The system displays options related to that section. 3. View create option mode that appears on screen and click the create button. 4. User can view all the parameters, fill them and proceed next. 5. Newly created schedule will be added to the list. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Internet connectivity error may occur. * Schedule already present * Clashes among groups in schedule |
| **Includes:** | | N/A |
| **Special Requirements:** | | * User must be logged in admin * 2. Project must be added. |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.6 Search Project by Date**

|  |  |  |
| --- | --- | --- |
| **Use Case**  **ID:** | UC-6 | |
| **Use Case**  **Name:** | Search Project by Date. | |
| **Actors:** | | Admin/Student/Supervisor /PC Convener |
| **Description:** | | This requirement is used to search projects within specific date range. |
| **Trigger:** | | User select search project by date option. |
| **Preconditions:** | | * Internet is working * Application is running * User is logged in * Projects are being viewed |
| **Post conditions:** | | Projects are displayed successfully within the desired date range. |
| **Normal Flow:** | | 1. User clicks on search project by date option 2. User enters his search keyword dates in the field. 3. User enters search button 4. System displays a list of all projects within the particular date range. |
| **Alternative**  **Flows:** | | In step 4 of normal flow if information entered in search field  Returns an error.   1. System prompts the user about error 2. User reenter the search information and click on search button |
| **Exceptions:** | | * No project exist within the required date. * User must input valid date. |
| **Includes :** | | N/A |
| **Special Requirements:** | | N/A |
| **Assumptions:** | | User enter correct information for searching criteria. |
| **Notes and**  **Issues:** | | N/A |

**3.1.7 Reserving a Room**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-7 | |
| **Use Case Name:** | Reserving a Room for Weekly Meetings | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to reserve a room for the weekly meetings that are held. |
| **Trigger:** | | User will be required to click on Room Management control to proceed |
| **Preconditions:** | | * User must be logged in to perform this action * User should have authentication rights |
| **Post conditions:** | | User has successfully reserved the room |
| **Normal Flow:** | | 1. User accesses the Room Management section of the system. 2. The system display options related to that section. 3. User searches for the availability of room. 4. Room is booked successfully. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Room is already booked |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in as admin |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.8 Searching for Room’s Availability**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-8 | |
| **Use Case Name:** | Searching for Room’s Availability | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to search for the availability of room. |
| **Trigger:** | | User will be required to click on Room Management control to proceed. |
| **Preconditions:** | | * User must be logged in to perform this action * User should have authentication rights. |
| **Post conditions:** | | User has successfully reserved the room. |
| **Normal Flow:** | | 1. User accesses the Room Management section of the system. 2. The system display options related to that section. 3. User searches the room. 4. Available room displayed. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Room is not available |
| **Includes:** | | N/A |
| **Special Requirements:** | | User |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.9 View Comments**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-9 | |
| **Use Case Name:** | View Comments | |
| **Actors:** | | Student |
| **Description:** | | This requirement is used to view the comments given by the PC members. |
| **Trigger:** | | User will be required to click on “Remarks” option to proceed |
| **Preconditions:** | | * User must be logged in to perform this action |
| **Post conditions:** | | User successfully views the comments |
| **Normal Flow:** | | 1. User accesses the Remarks option. 2. User selects the required milestone. 3. User clicks on view comments option next to the project name to view. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * No comments are available |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in as student |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.10 Send Email**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-10 | |
| **Use Case Name:** | Send Email | |
| **Actors:** | | Administrator |
| **Description:** | | This requirement is used to send an email to students regarding latest announcements. |
| **Trigger:** | | User will be required to click on Announcement section to proceed. |
| **Preconditions:** | | * User must be logged in to perform this action * User should have authentication rights |
| **Post conditions:** | | Email is sent successfully |
| **Normal Flow:** | | 1. Admin logged in 2. User clicks on announcement option 3. He may enter new Announcement message. 4. When all modifications have been done by admin he clicks the   commit button   1. All changes will be saved if no validation errors 2. User select option of all students to whom notifications should be sent 3. User click on send button 4. System sends notification 5. Notification can be seen by only those users whose email id is provided. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | * Unavailability of Email ID |
| **Includes:** | | N/A |
| **Special Requirements:** | | Availability of Email ID |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.11 Send Message**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-11 | |
| **Use Case Name:** | Send Message | |
| **Actors:** | | Student |
| **Description:** | | This requirement is used as a mode of communication between student and supervisor. |
| **Trigger:** | | User will be required to click on send message button |
| **Preconditions:** | | * User must be logged in to perform this action * User should have authentication rights |
| **Post conditions:** | | Message is sent successfully |
| **Normal Flow:** | | 1. User clicks on my project 2. User selects send message option next to supervisor’s name 3. User types in the message and sends it. |
| **Alternative Flows:** | | 1. User can go to inbox option 2. User can select message option from there |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | N/A |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.12 Grade Milestones**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-12 | |
| **Use Case Name:** | Grade Milestones | |
| **Actors:** | | Admin/supervisors |
| **Description:** | | This requirement is used to mark different milestones and evaluate them along with their presentations. (SRS/SDS/Report/Presentations) |
| **Trigger:** | | User is required to Click on “Marking category” option to proceed. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have uploaded the marks successfully. |
| **Normal Flow:** | | 1. User access the Marking Category section of the system. 2. The system displays all the options related to that section. 3. User views the list of milestones that appears on screen. 4. User selects the option of milestone to be marked. 5. User specifically upload the marks of the particular document. 6. The system validates the information and determines that validation is passed. 7. The system displays a dialog box to notify user about submission of marks. 8. Success message will be shown and the milestone will be marked. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

**3.1.13** **Select Externals**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-13 | |
| **Use Case Name:** | Select Externals | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to select externals randomly for final project presentation. |
| **Trigger:** | | User is required to Click on “External Selection” option to proceed. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | Externals assigned successfully |
| **Normal Flow:** | | 1. User access the “External” section of the system. 2. The system displays all the options related to that section. 3. User views the option that appears on screen. 4. User selects the option of assigning externals to the list of projects. 5. Success message will be shown “Externals assigned successfully”. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

**3.1.14 View New Projects**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-14 | |
| **Use Case Name:** | View New Projects | |
| **Actors:** | | Admin/supervisors/students |
| **Description:** | | This requirement is used to display all the new projects added at one place. |
| **Trigger:** | | User is required to Click on “New Projects” option to view the list of all the new projects to proceed. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have viewed the list of new projects. |
| **Normal Flow:** | | 1. User access the Project section of the system. 2. The system displays all the options related to that section. 3. User views the new projects that appears on screen. 4. User selects the option of new projects. 5. The system specifically shows list of all the new projects that are recently added. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

**3.1.15 View Old Projects**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-15 | |
| **Use Case Name:** | View Old Projects | |
| **Actors:** | | Admin/supervisors/students |
| **Description:** | | This requirement is used to display all the old projects at one place. |
| **Trigger:** | | User is required to Click on “Old Projects” option to view the list of all the previous projects to proceed. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have viewed the list of Old projects. |
| **Normal Flow:** | | 1. User access the Project section of the system. 2. The system displays all the options related to that section. 3. User views the old projects that appears on screen. 4. User selects the option of old projects. 5. The system specifically shows list of all the old projects. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | N/A |
| **Notes and Issues:** | | N/A |

**3.1.16 Display Name and Registration Number**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-16 | |
| **Use Case Name:** | Display Name and Registration Number | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to display name along with the registration number of all the students who have been assigned the projects in the list of view all projects. |
| **Trigger:** | | User is required to Click on “view all projects” option to view the complete name of the student to whom project have been assigned. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have viewed the list of all the projects assigned to students along with their name and registration numbers. |
| **Normal Flow:** | | 1. User access the Project section of the system. 2. The system displays all the options related to that section. 3. User selects the option of view all projects. 4. The system specifically shows list of all the projects with the name and registration number of the student who have been assigned the project. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |
|  | |  |

**3.1.17 Manage Milestone Status**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-17 | |
| **Use Case Name:** | Manage Milestone Status | |
| **Actors:** | | Admin |
| **Description:** | | This requirement is used to display the status of the milestone if it is cleared or is in process. |
| **Trigger:** | | User is required to Click on “Document Management” option to manage the status of various documents. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have a maintained the status of all the documents. |
| **Normal Flow:** | | 1. User access the Document Management section of the system. 2. The system displays all the options related to that section. 3. User selects the option of view project status. 4. The system specifically shows list of all the documents and the statuses being assigned to them. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

**3.1.18 View Saved Projects**

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-18 | |
| **Use Case Name:** | View Saved Projects | |
| **Actors:** | | Student |
| **Description:** | | This requirement is used to display the saved project to user’s wish list to view it later. |
| **Trigger:** | | User is required to Click on “view all projects” option to view and add projects to wish list. |
| **Preconditions:** | | * Internet is connected. * User must login to perform this function. |
| **Post conditions:** | | User have added |
| **Normal Flow:** | | 1. User access the Project section of the system. 2. The system displays all the options related to that section. 3. User selects the option of view all projects. 4. The system specifically shows list of all the projects and add them to list to be saved. |
| **Alternative Flows:** | | N/A |
| **Exceptions:** | | N/A |
| **Includes:** | | N/A |
| **Special Requirements:** | | User must be logged in. |
| **Assumptions:** | | The user must have the related knowledge to operate this task. |
| **Notes and Issues:** | | N/A |

# Functional Requirements

|  |  |
| --- | --- |
| Project Initiation.View Announcements | The purpose of this use case is to allow the user to view the announcements about the latest activities regarding project. |
| Project Initiation.Sending Email | This requirement is used to send an email to students regarding latest announcements. |
| Project Initiation.Search Project by Research Group | This use case describes how the user can search for a project using research group name. |
| Project Initiation.Forgot Password | User needs to retrieve their password in case of forgetting |
| Project Initiation.Schedule Creation | This requirement is used to create schedule for presentations of Final Year Projects. |
| Project Initiation.Search Project by Date | This requirement is used to search projects within specific date range. |
| Project Initiation.Random Selection of Externals | This requirement is used to select externals randomly for final project presentation. |
| Conference Room Reservation.Reserving a Room for Weekly Meetings | This requirement is used to reserve a room for the weekly meetings that are held. |
| Conference Room Reservation.Searching for Room’s Availability | This requirement is used to search for the availability of room |
| Communication.Sending Message | This requirement is used as a mode of communication between student and supervisor. |
| Document Management.Upload Milestone | This requirement is used to upload the milestones by the students on this portal |
| Document Management.Grading Milestones | This requirement is used to mark different milestones and evaluate them along with their presentations. (SRS/SDS/Report/Presentations) |
| Project Directory.View Comments | This requirement is used so that students can view the comments by their supervisor |
| Project Directory.View New Projects | This requirement is used to display all the new projects added at one place. |
| Project Directory.View Old Projects | This requirement is used to display all the old projects at one place. |
| Project Directory.Display Name and Registration Number | This requirement is used to display name along with the registration number of all the students who have been assigned the projects in the list of view all projects. |
| Project Directory.Manage Milestone Status | This requirement is used to display the status of the milestone if it is cleared or in process. |
| Project Directory.View Saved Projects | This requirement is used to display the saved project to user’s wish list to view it later. |

# Non-Functional Requirements

|  |  |
| --- | --- |
| **NFR Name** | **NFR Description** |
| **Usability** | **Understandability**  Interface elements should be easy to understand since we are following standard Windows element in our interface.  **Learnability**  The system is basically for the students, faculty, supervisors and project committee who will be using different modules of the system.  **Operability**  The interface will show the login option which will decide the mode for the user. |
| **Reliability** | **Accuracy**  Accuracy of the system will be specified with the type of user has given its input.  **Bugs**  Critical bug can be categorized as a complete loss of data which can be possible if any system file is corrupted. System cannot operate until file is repaired or system is reinstalled. |
| **Performance** | **Response Time** The response time of the system depends on the internet connection and underlying hardware.  **Resource utilization** As the system will be an online application so it will not be consuming much memory and processing. |
| **Supportability** | **Naming Conventions**  Standard naming conventions of C# will be used in the development of this project. |

Chapter 4: Design and Architecture

This chapter will discuss the design and architecture of your system.

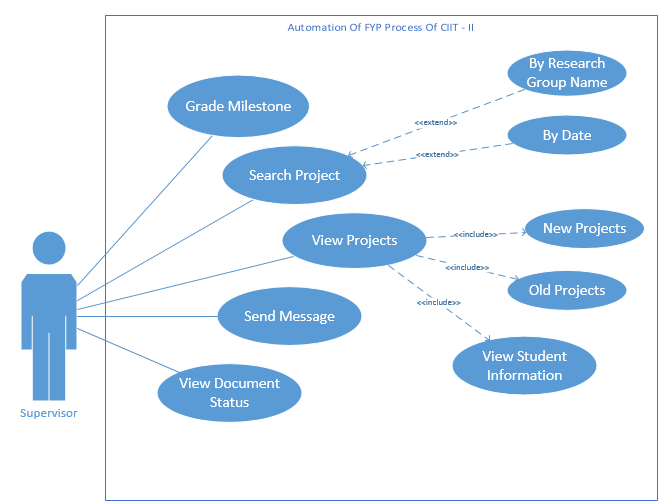
# System Architecture

Explain and justify the choice of system architecture for your project.

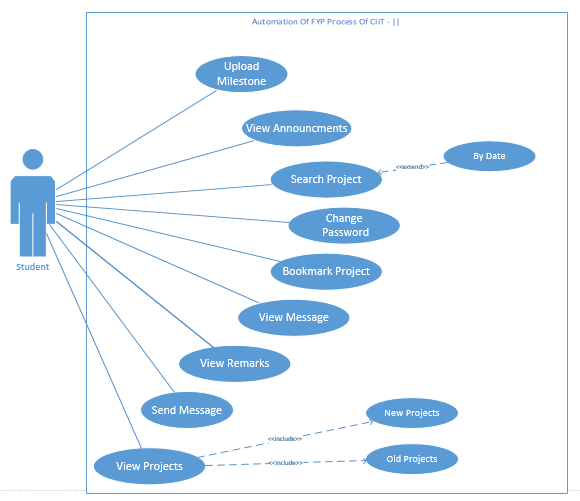
# System Design

* + 1. **UML Behavioral Diagram**

* + - 1. **Use Case Diagrams**

****

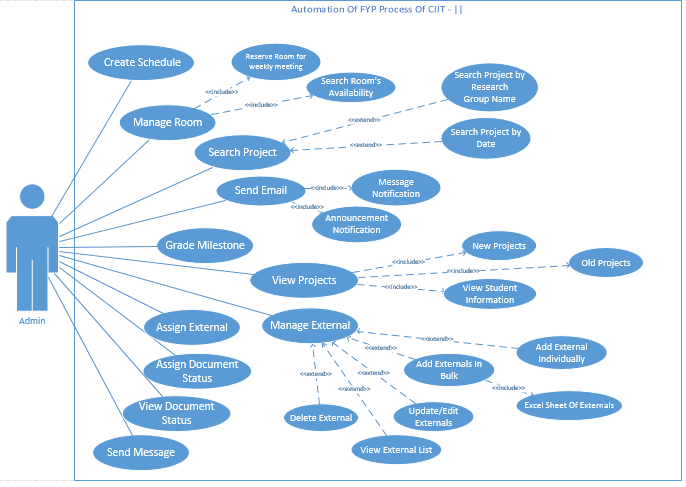
**Figure 3- 4.2.1.1 Use Case Diagram**



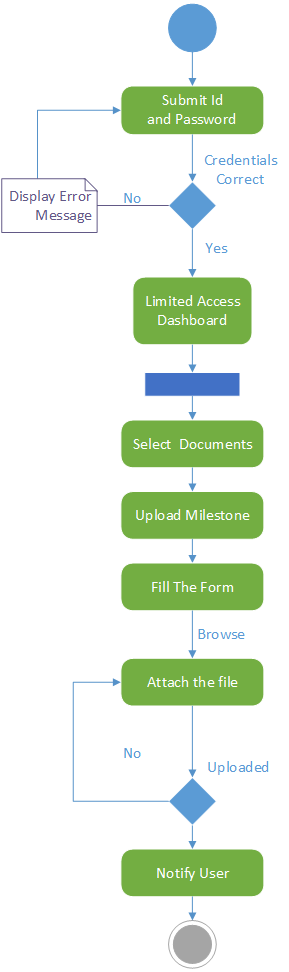
**Figure 4- 4.2.1.2 Use Case Diagram**

**Figure 4- 4.2.1.3 Use Case Diagram**

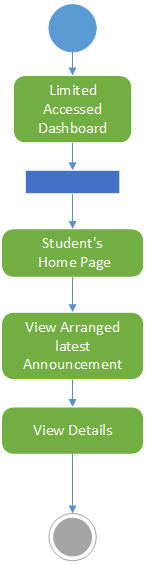
**4.2.2 Activity Diagrams**

**4.2.2.1 Student**

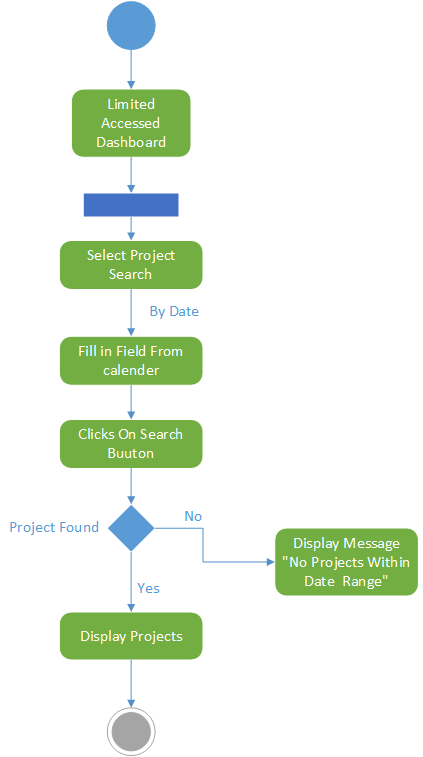
**4.2.2.1.1 Upload Milestone**

****

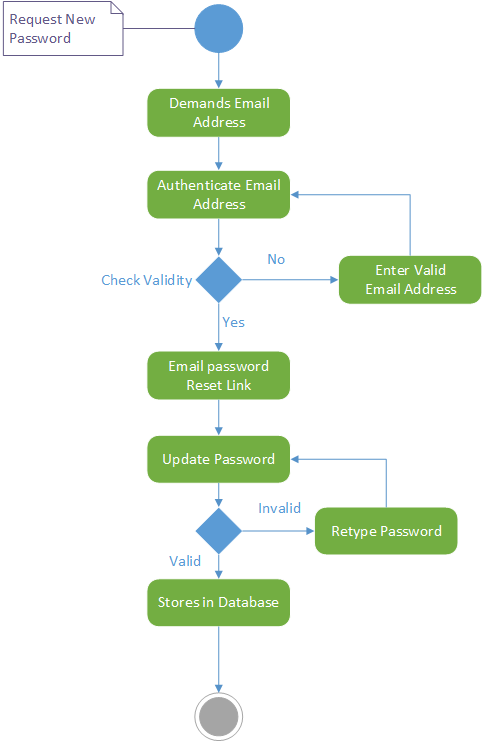
View Announcements



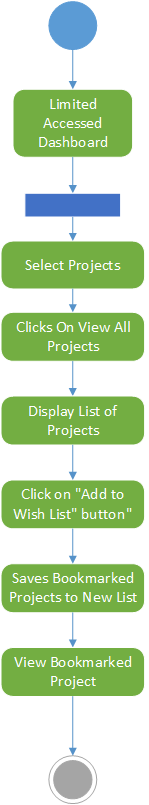
Search Project



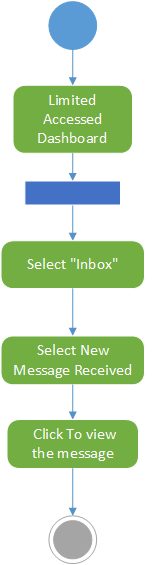
Change Password



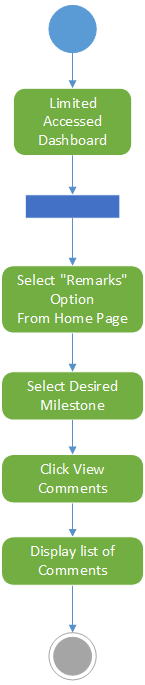
Bookmark Project



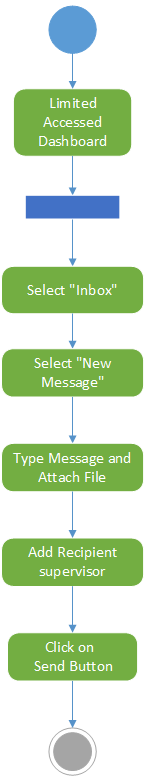
View Message



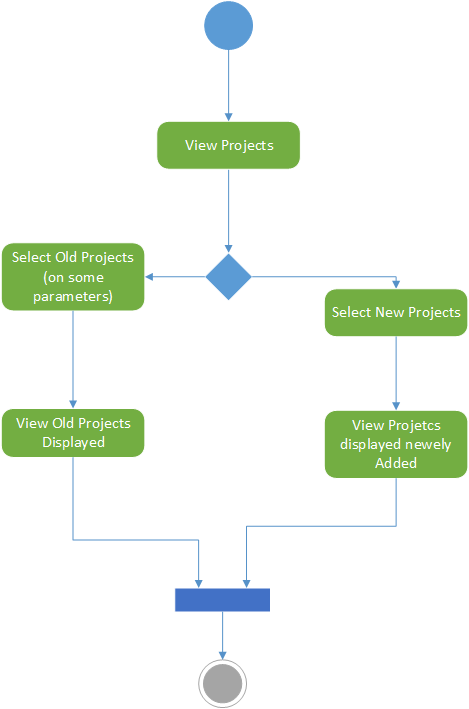
View Remarks



Send Message

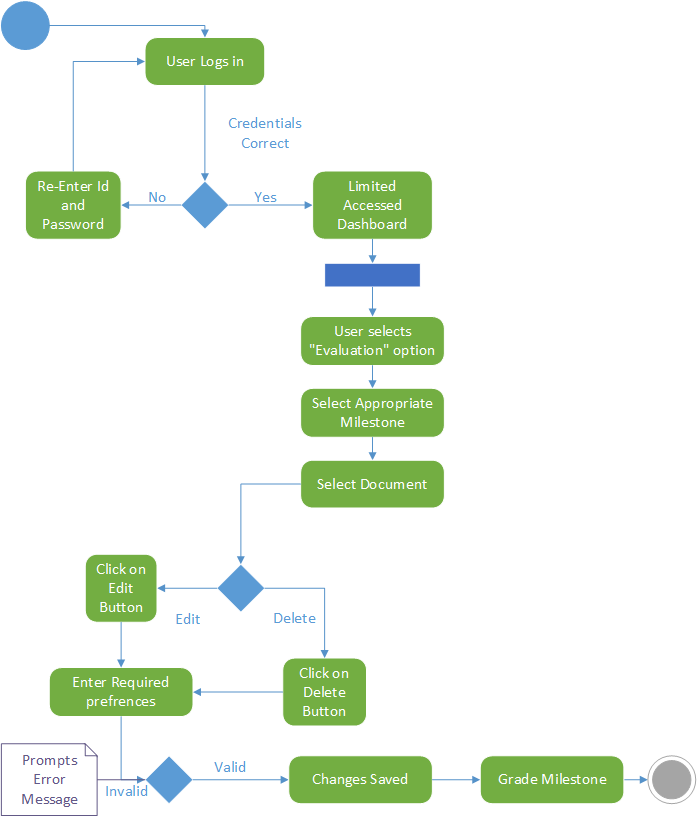


View Projects

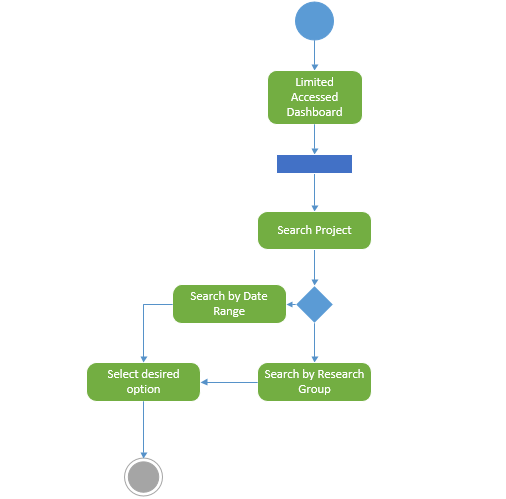


**4.2.2.2 Supervisor**

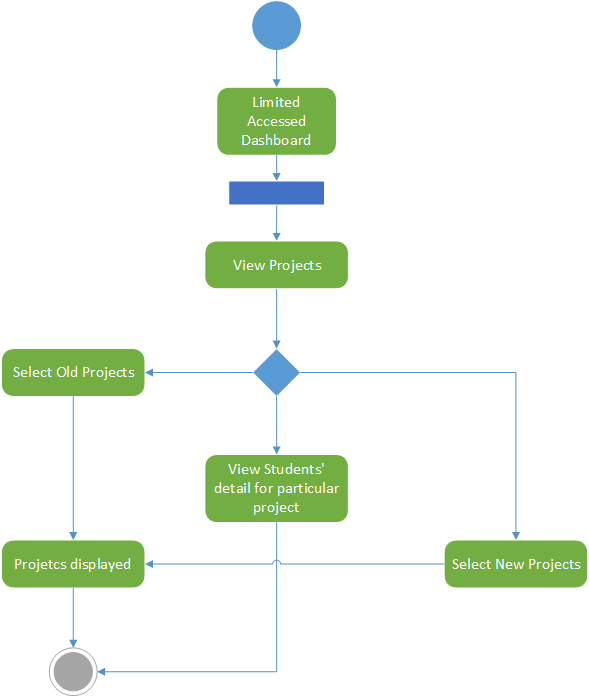
**Grade Milestone**

****

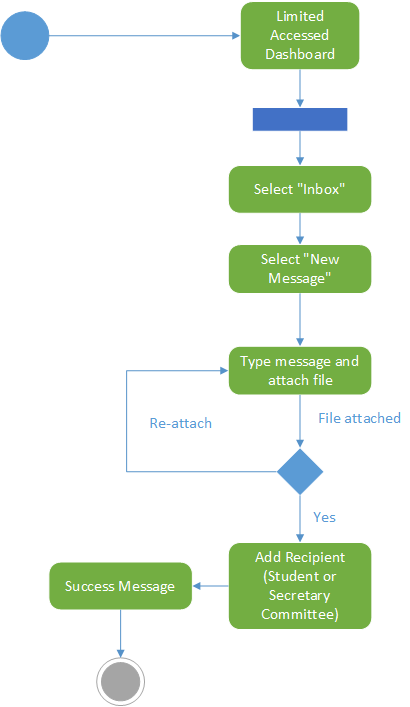
Search Project



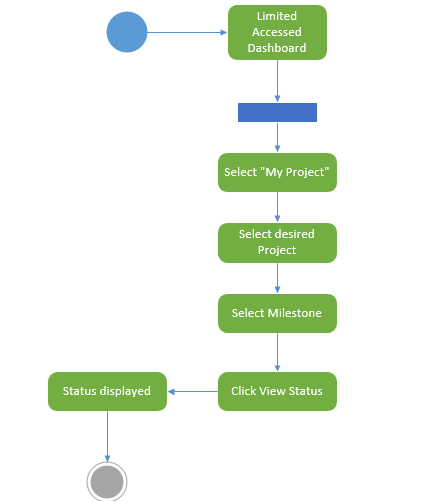
View Projects



Send Message

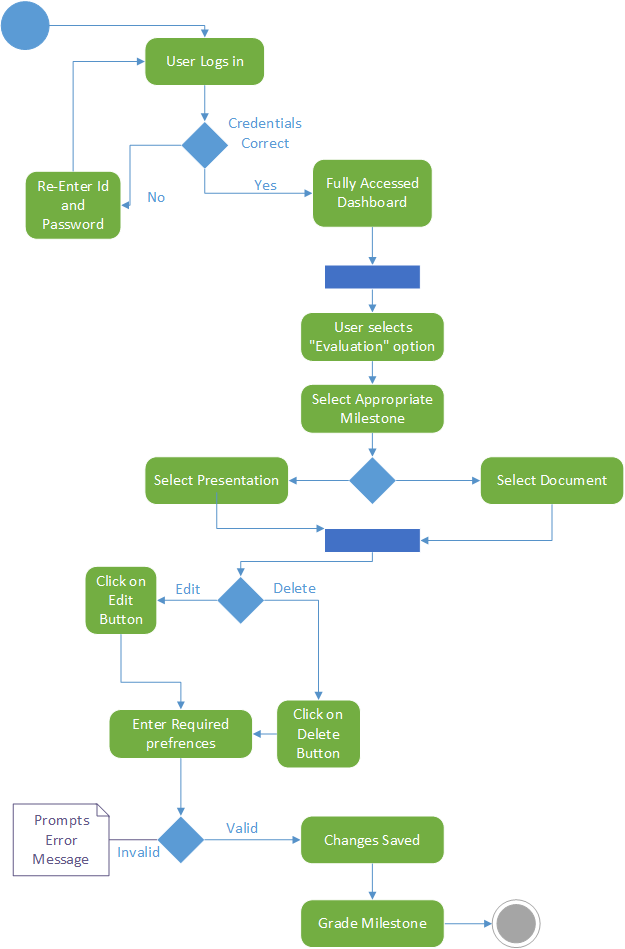


View Document Status

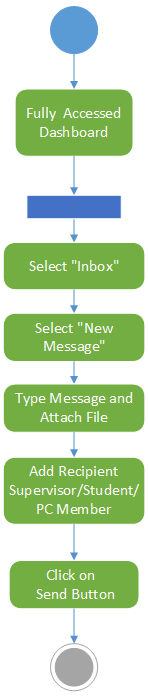


**4.2.2.3 Admin**

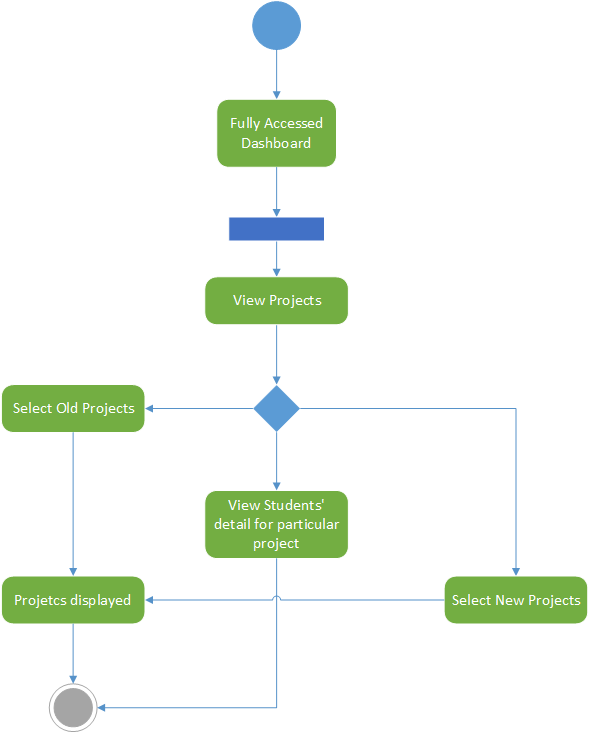
**Grade Milestone**

****

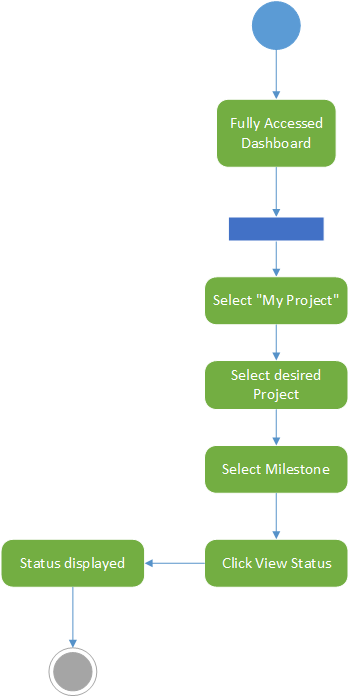
Send Message



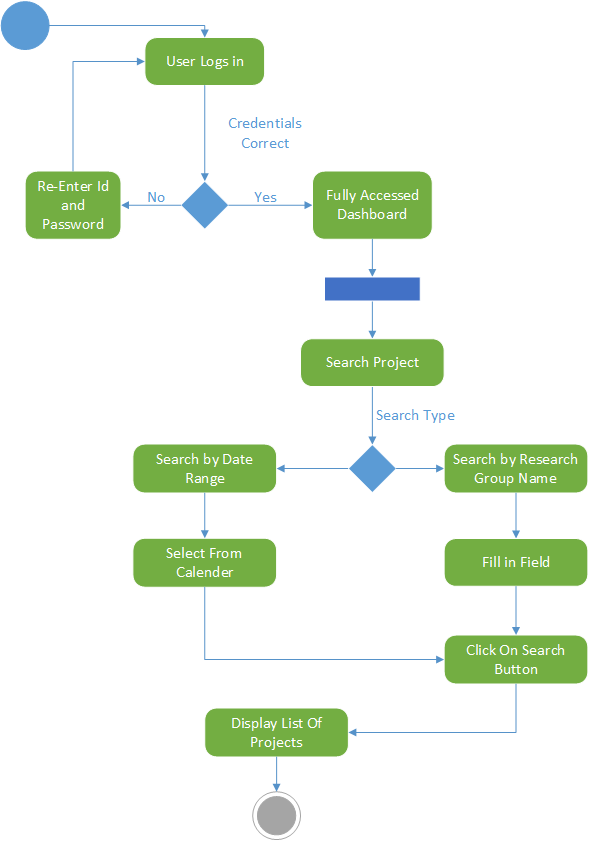
View Project (Old/New Projects)



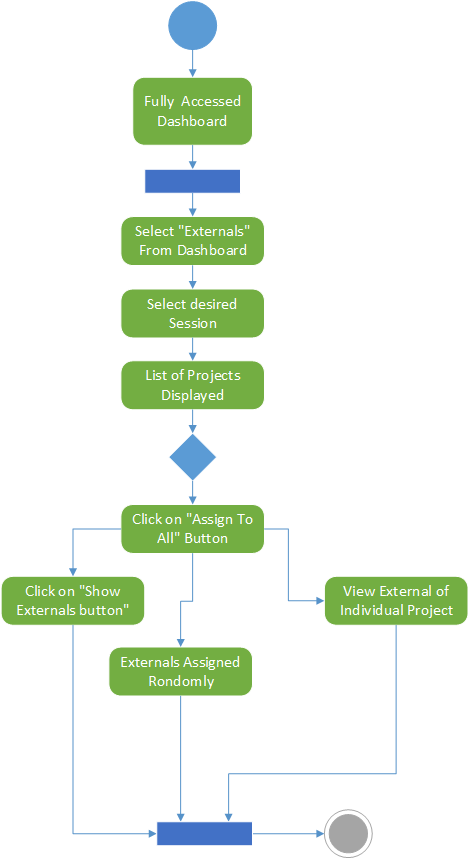
View Document Status



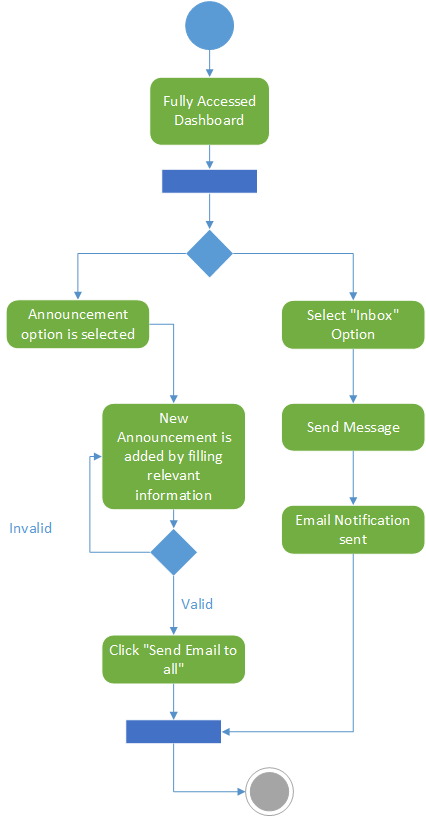
Search Project by Date and Research Group



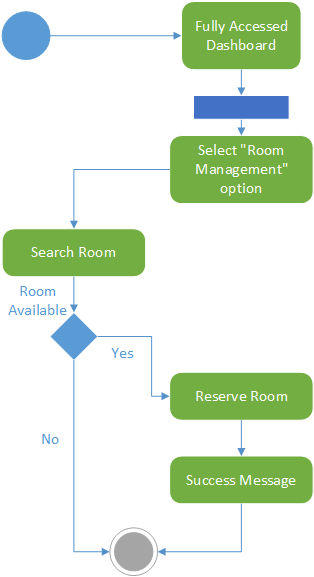
Assign External



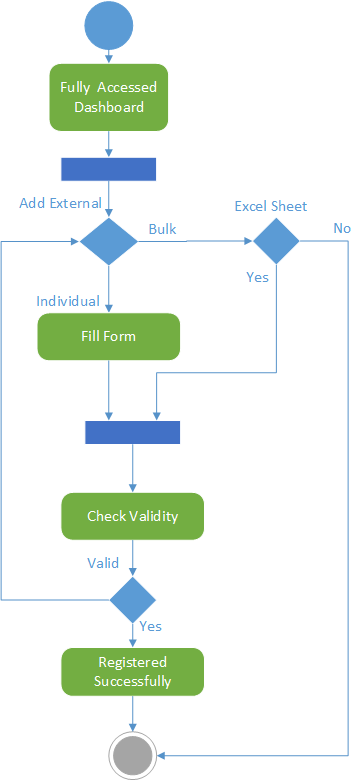
Send Email



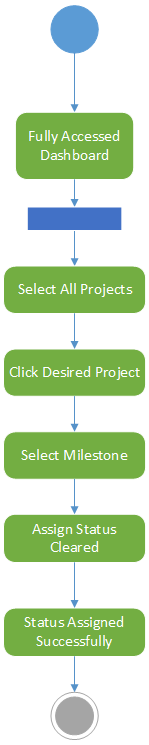
Room Management



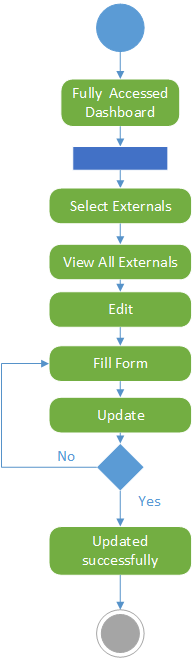
Manage External (Add individual and Bulk)



Assign Document Status (Scope, SRS cleared etc.)



Manage External (Edit)



# References

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

1 Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. (Journal paper example)

2 Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and Turu Ishada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 (paper on web) http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf

3 Ali Behforooz & Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. (book reference example)

4 Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed. (web site)