

<b>1. INTRODUCTION</b>	<b>4</b>
1.1 PURPOSE	5
1.2 INTENDED AUDIENCE	5
1.3 CONCLUSION	6
<b>2. INCEPTION OF IIT 360</b>	<b>6</b>
2.1 INTRODUCTION	6
2.2 EXISTING STORY	7
2.3 INCEPTION PROCEDURE	7
2.3.1 IDENTIFY THE CLIENT OF OUR PROJECT	8
2.3.2 ICEBREAKING	8
2.3.3 IDENTIFYING THE STAKEHOLDERS OF THE IIT 360	8
2.4 CONCLUSION	9
<b>3. ELICITATION OF IIT 360</b>	<b>9</b>
3.1 INTRODUCTION	9
3.2 ELICITING REQUIREMENTS	10
3.2.1 COLLABORATIVE REQUIREMENTS GATHERING	10
3.2.2 QUALITY FUNCTION DEPLOYMENT	10
Offline Availability of Information:	11
Get Updated Routine:	11
Morning Class Alarm:	11
Notification:	11
Reminder:	11
3.2.3 USAGE SCENARIO	12
USER:	12
AUTHENTICATION AND AUTHORIZATION:	12
DYNAMIC CLASS ROUTINE:	13
Class Cancellation:	13
Class Swapping :	13
Class Addition:	14
Class Rescheduling:	14
MORNING CLASS ALARM:	15
NOTIFICATION:	15
<b>4. SCENARIO BASED MODELING</b>	<b>18</b>
4.1 INTRODUCTION	18
4.2 DEFINITION OF USE CASE	19

4.3 USE CASE DIAGRAMS	20
Use Case Diagram level: 0	20
Use Case Diagram level: 1	21
Use Case Diagram level: 1.1	22
Use Case Diagram level: 1.1.1	24
Use Case Diagram level: 1.1.2	25
Use Case Diagram level: 1.1.3	28
Use Case Diagram level: 1.1.3.1	29
Use Case Diagram level: 1.1.5	31
Use Case Diagram level: 1.2	32
Use Case Diagram level: 1.2.1	33
4.4 ACTIVITY DIAGRAMS	35
Activity Diagram ID: 01	35
Activity Diagram ID: 02	36
Activity Diagram ID: 03	37
Activity Diagram ID: 04	38
Activity Diagram ID: 05	39
Activity Diagram ID: 06	40
Activity Diagram ID: 07	41
Activity Diagram ID: 08	42
Activity Diagram ID: 09	43
Activity Diagram ID: 10	44
Activity Diagram ID: 11	45
Activity Diagram ID: 12	46
Activity Diagram ID: 13	47
Activity Diagram ID: 14	49
Activity Diagram ID: 15	50
Activity Diagram ID: 16	51
Activity Diagram ID: 17	52
Activity Diagram ID: 18	53
Activity Diagram ID: 19	54
Activity Diagram ID: 20	55
<b>5. DATA-BASED MODELLING OF IIT 360</b>	<b>56</b>
5.1 Data Modelling Concept	56
5.1.1 Data Objects	56
5.1.1.1 Data Object Identification	56
5.1.1.2 Final Data Objects	60
5.2 DATA OBJECT RELATIONSHIPS	61

5.3 ENTITY RELATIONSHIP DIAGRAM	62
5.4 SCHEMA DIAGRAM	63
<b>6. CLASS-BASED MODELING OF IIT 360</b>	<b>68</b>
6.1 CLASS BASED MODELING CONCEPT	68
6.2 Noun List for CMS	68
6.3 Verb List of CMS	70
6.4 GENERAL CLASSIFICATION	70
Table:1 General Classification	71
6.5 SELECTION CRITERIA	73
Table:2 Selection Criteria	73
6.6 ATTRIBUTE AND METHOD IDENTIFICATION	75
6.6.1 For Mobile Application Classes:	75
Table:3 Mobile Application Classes	75
6.6.2 For Admin Panel Classes:	78
Table:4 Admin Panel Classes	78
6.6.3 For Server-side Classes:	78
Table:5 Server Side Classes	78
6.7 Analyzing Classes	80
6.8 CRC Card	80
6.9 Class Card	84
For Mobile Application Classes:	84
Table:6 Class Card for IITian	84
Table:7 Class Card for Faculty	85
Table:8 Class Card for Staff	85
Table:9 Class Card for Authentication	86
Table:10 Class Card for Alarm	87
Table:11 Class Card for Notice	87
Table:12 Class Card for News&Events	88
Table:13 Class Card for Notification	88
Table:14 Class Card for EventReminder	89
Table:15 Class Card for Routine	89
Table:16 Class Card for Research	90
Table:17 Class Card for Project	90
Table:18 Class Card for Assignment	91
Table:19 Class Card for Document	91
Table:20 Class Card for Project	92
Table:21 Class Card for Achievement	92
Table:22 Class Card for Academic	93

Table:23 Class Card for Research Area	93
Table:24 Class Card for Admin	94
Table:25 Class Card for Content Manager	95
Table:26 Class Card for Semester Routine	95
Table:27 Class Card for AccountManager	96
Table:28 Class Card for RoutineManager	97
Table:29 Class Card for GetContent	98
Table:30 Class Card for AddContent	98
Table:31 Class Card for UpdateContent	99
Table:32 Class Card for DeleteContent	100
<b>7. FLOW ORIENTED MODEL OF IIT 360</b>	<b>101</b>
DATA FLOW DIAGRAMS:	101
DFD level: 0	101
DFD level: 1	101
DFD level: 2	102
DFD level: 2	103
DFD level: 2	104
DFD level: 2	105
DFD level: 2	106
DFD level: 2	107
<b>8. BEHAVIORAL MODELING OF IIT 360</b>	<b>109</b>
8.1 Event Identification Table	109
8.2 STATE TRANSITION DIAGRAMS	115
8.3 SEQUENCE DIAGRAM:	129

# 1. INTRODUCTION

This chapter is a part of our software requirement specification for the project “IIT 360”. In this chapter we will focus on the intended audience for this project.

## 1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of our mobile application IIT 360. It contains the functional, non-functional and the supporting requirements and establishes a requirement’s baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## 1.2 INTENDED AUDIENCE

Our intended audiences are:

1. Students
2. Teachers
3. Anyone who wants to know about IIT

This SRS will be used to verify whether the developer team has created a product that is acceptable to the desired users. The developers will use this SRS to plan milestones and a delivery date, and ensure that they are on track during development of the system. The designers will use this SRS as

a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs. The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements. The testers will use this SRS to derive test plans and test cases for each documented requirement.

## 1.3 CONCLUSION

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

## 2. INCEPTION OF IIT 360

In this chapter, the Inception part of the SRS will be discussed briefly.

### 2.1 INTRODUCTION

**IIT 360** is a cross platform mobile application which will serve multiple purposes namely

- a. Ease the life of IITians by providing them what they need in their daily lives like:
  - i. Dynamic Class Routine
  - ii. Morning Class Alarm
  - iii. Reminder and Notification
- b. Making information available for all in the mobile app.

### 2.2 EXISTING STORY

Current situation is shown below:

1. Information about IIT is only available online.
2. Students are too much dependent on facebook in case of knowing about any changes of class schedules.
3. Students tend to miss or make late in morning classes because they forget to set the alarm seldom.
4. Students seldom remain unnoticed about events and assignments.
5. Students tend to forget about events and assignments.

## 2.3 INCEPTION PROCEDURE

At the beginning of our project, we entered the inception stage. This stage includes how the project will be started and their **scope** and **limitations**. The main goal of this phase is to **identify the requirements, demand** and establish some sort of **mutual understanding** between the software team and the stakeholders of the IIT 360. In order to make this phase effective followed some steps namely:

- a. Identifying the client of our project
- b. Icebreaking
- c. Identifying the stakeholders of IIT 360
- d. Identifying the multiple viewpoints of stakeholders

### 2.3.1 IDENTIFY THE CLIENT OF OUR PROJECT

At first, we identified the location from where we will start our expedition. Our location was Institute of Information Technology, University of Dhaka. The most important stakeholder of our project are the Students. The teachers are also equally important. Besides, general people are also our clients. We have analyzed our requirements with the consent of both of them.

### 2.3.2 ICEBREAKING

Icebreaking refers to diminishing the communication barrier between you and the other person. It is a crucial part since it decides the acceptance of our proposal. We started this phase by talking with them with context free languages. Their behavior, response to our question or willing to accept the new course management system solely depends on this phase.



### 2.3.3 IDENTIFYING THE STAKEHOLDERS OF THE IIT 360

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its installation. IIT 360 has a limited number of stakeholders. Identification of the stakeholders was done from the information provided by the teachers and the students. It turned out that we don't have any stakeholders other than the clients.

## 2.4 CONCLUSION

The primary goal of this project is to model and design a software for those people who are related to a course. For these reasons, the software will be designed in such a way that it will bring comfort to the client who will use it. The software will be simple and user friendly. Otherwise, it will not be appreciated by the clients. The software will be designed in such a way that it takes very little time to manage. To make this software project successful, collaboration with stakeholders is the main priority. To sum up, what they want, how the software will work and how it can be more efficient than the earlier times is the main concern in this phase.

## 3. ELICITATION OF IIT 360

After discussing the Inception phase, we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

### 3.1 INTRODUCTION

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the stakeholders. We have faced many difficulties, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to shortage of time and volatility of the stakeholders. Though it is not easy to gather requirements within a very short time, we have surpassed these problems in an organized and systematic manner.

### 3.2 ELICITING REQUIREMENTS

We have seen the Question and Answer (Q&A) approach in the previous chapter, where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation of Work Products

### 3.2.1 COLLABORATIVE REQUIREMENTS GATHERING

We have met with many students in the Inception phase. Besides, we have met with 1 teacher. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with some stakeholders (who are acting a vital role in the whole process) more than once to elicit the requirements.

### 3.2.2 QUALITY FUNCTION DEPLOYMENT

#### **1. Offline Availability of Information:**

- a. Everyone who will have the mobile application will be able to access general information about IIT offline.

#### **2. Get Updated Routine:**

- a. The IITians will be able to view the updated routine. The dynamic routine will handle all the changed schedules and notify everyone when a change occurs in the routine. Change can be of 4 types:
  - b. Cancel Class
  - c. Add Extra Class
  - d. Swap Class
  - e. Reschedule Class

#### **3. Morning Class Alarm:**

- a. Morning Alarms will help IITians to wake up on time. The alarm will go off a few hours prior to the first class.

#### **4. Notification:**

- a. The app will notify IITians about given assignments and events that are going to take place in IIT and let IITians know more about those events and assignments.

#### **5. Reminder:**

- a. IITians will be reminded about the event on event day.

### **3.2.3 USAGE SCENARIO**

#### **USER:**

- IITian (anyone with IIT mail/ authentic users)
- Guest

#### **AUTHENTICATION AND AUTHORIZATION:**

- Users can sign up using the IIT mail and their given password.
- An OTP will be sent to their mail to confirm their email address.
- To complete sign up, IITians(except faculty members) will have to provide some information:
  - Name
  - Program(BSSE/MSSE)
  - Semester
  - Photo(Optional)
- They may change this information later.

- To sign in, they will have to provide iit mail and password
- Faculty members will be automatically identified and they won't need to provide the latter information.
- Users will be considered as guests if they do not sign in.
- Once IITians sign out, they will be considered as guest users again.
- IITians(having IIT mails) can view information about IIT and also access the special features like routine, alarm, notification and reminder.
- Guest users will only be able to view information about IIT.

## **DYNAMIC CLASS ROUTINE:**

- Class Routine can be seen based on IITian's semester
- Faculty Members can also see their routine
- Any permanent change can be done only by the admin
- CR's can make temporary changes in class routine based on faculty member's demand
- Temporary changes in routine (by CR) is classified into 4 categories:

- ***Class Cancellation:***

Classes can be cancelled anytime by the teacher

- ***Input:***

1. Semester (Course Code will appear based on semester)
2. Course Code (Drop Down List)
3. Date (Date Picker)

- ***Class Swapping :***

Any two classes can be swapped with their respective time slots

- ***Input:***

1. Semester
2. Course Code 1
3. Date 1
4. Course Code 2
5. Date 2

- ***Check:***

1. Validity Check (Whether Course X has schedule on Date X)
2. Accommodation Check (New slot has enough time to accommodate the Course)

- ***Class Addition:***

Extra Classes can be added to the routine

- ***Input:***

1. Semester
2. Course Code
3. Date
4. Start Time
5. End Time

- ***Check:***

1. Accomodation Check

- ***Class Rescheduling:***

Any class that will be taken on a later date

- ***Input:***

1. Semester
2. Course Code
3. Previous Date
4. New Date
5. New Start Time
6. New End Time

- ***Check:***

1. Validity Check
2. Accomodation Check

- Routine will be refreshed after any sort of update.
- Notification will come for any routine change.

## **MORNING CLASS ALARM:**

- Alarm will go off a few hours before the first class of every day.
- The time of the alarm can be set by the user i.e. how long before the first class the user wants to wake up (maybe 1,2 or 3 hours)
- Alarm of any day can be turned off
- Teachers will also get alarm prior to their first class in the same manner
- Alarm will be refreshed after any kind of routine change(as long as the user is online)

## NOTIFICATION:

- Both Faculty members and Students will be notified (push/pull) if any event gets uploaded.
- Both Faculty members and Students will be notified (push/pull) if any change occurs in routine.
- Both Faculty members and Students will be notified if any notice gets uploaded.

## REMINDER:

- Both Faculty members and students will be reminded in the morning if any event is going to take place or any assignment needs to be submitted on that day.

## INFORMATION ABOUT IIT:

- Admin can change and add any new information from the database because nothing will be hard coded in the mobile app, everything will be fetched from the database making it very easy and efficient to use.
- **About IIT**
  - Institution
    - Welcome to IIT
    - Why IIT
    - Mission
  - Aims and Objectives
  - History
  - Faculty



- Present
  - Former
  - Officers and Staffs
- **Academic**
  - Undergraduate Studies
    - Information
    - Admission
  - Graduate Studies
    - MSSE
      - Information
      - Admission
    - MIT
      - Information
      - Admission
    - PGDIT
      - Information
      - Admission
  - Training Programs
    - Web Design
      - Information
      - Admission
    - Web Programming
      - Information
      - Admission
    - Office Applications
      - Information
      - Admission
    - Matlab-Origin-Latex
      - Information
      - Admission
    - Mobile Application
      - Information
      - Admission
- **Notices**
  - General
    - Scholarship
    - Registrar Office

- Projects
- Undergraduate Studies
  - BSSE
  - MSSE
- Graduate Studies
  - MSSE
  - MIT
  - PGDIT
- Training Programs
  - Web Design
  - Web Programming
  - Office Applications
  - Matlab-Origin-Latex
  - Mobile Application
- ***Life in IIT***
  - News and Events
  - Upcoming Events
  - Achievements
- ***Research***
  - Optimization
  - Cloud Computing
  - Image Processing and Pattern Recognition
  - Information Retrieval
  - Wireless Network and Security
  - Software Engineering

## 4. SCENARIO BASED MODELING

This chapter describes the Scenario Based Model for the “**Course Management System**”

### 4.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

### 4.2 DEFINITION OF USE CASE

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system. Actors either consume/ produce/ modify information.

## **Primary Actor**

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software. The actors who do more than one job(consume/produce/ manipulate information ) are our primary actors.

## **Secondary Actor**

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

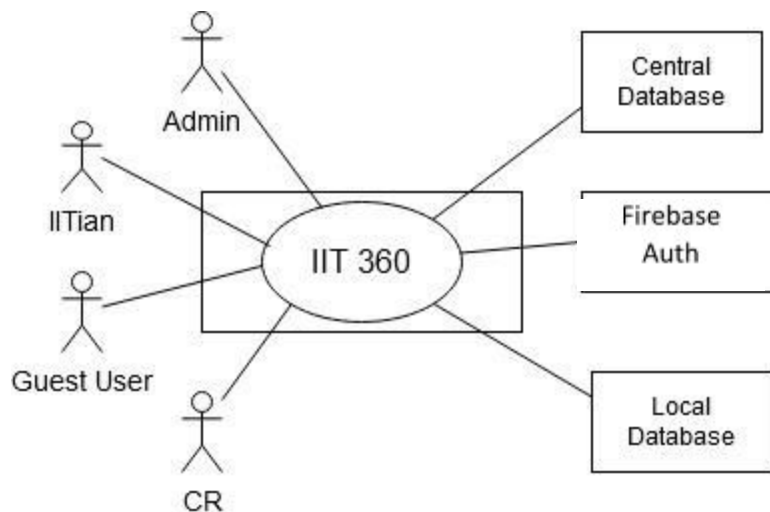
## 4.3 USE CASE DIAGRAMS

Use Case diagrams give a non-technical view of the overall system.

### Use Case Diagram level: 0

Name: IIT 360

ID: 01



**Description:** The use case “IIT 360” at level 0 shows the top-most view of the project and all the actors, namely:

1. Admin
2. IITian
3. CR
4. Guest User
5. Central Database
6. Firebase Auth
7. Local Database

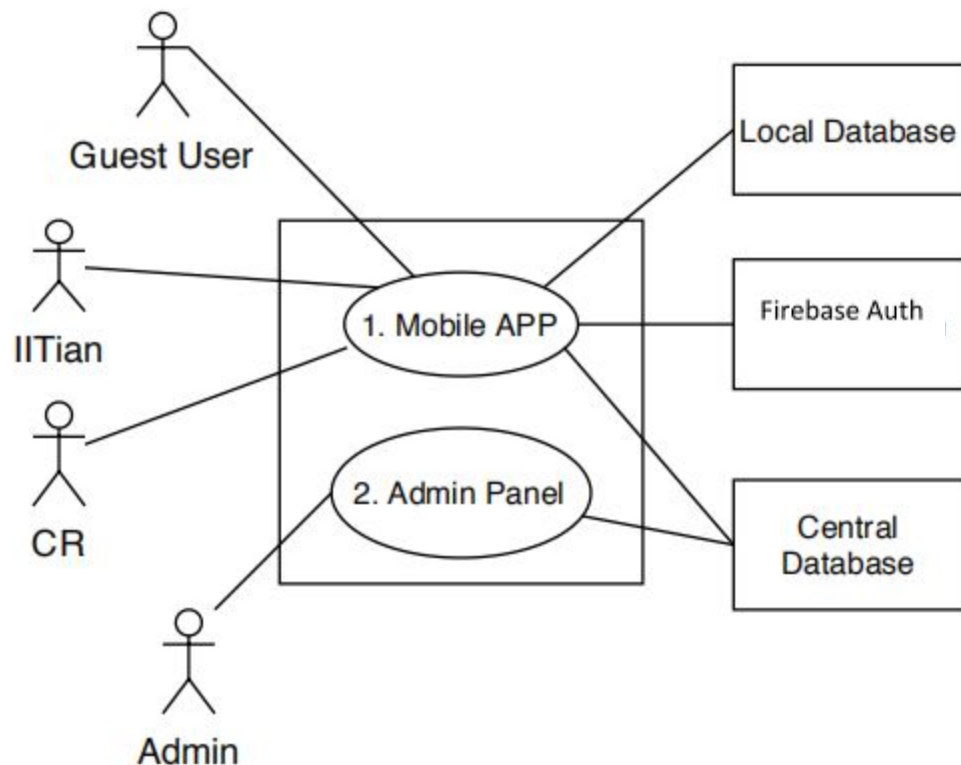
**Primary Actors:** Admin, Iltian, CR, Guest User

**Secondary Actor:** Central Database, Local Database, Firebase Auth

### Use Case Diagram level: 1

Name: Sub-systems of IIT 360

ID: 02



**Description:** IIT 360 is divided into 2 main subsystems:

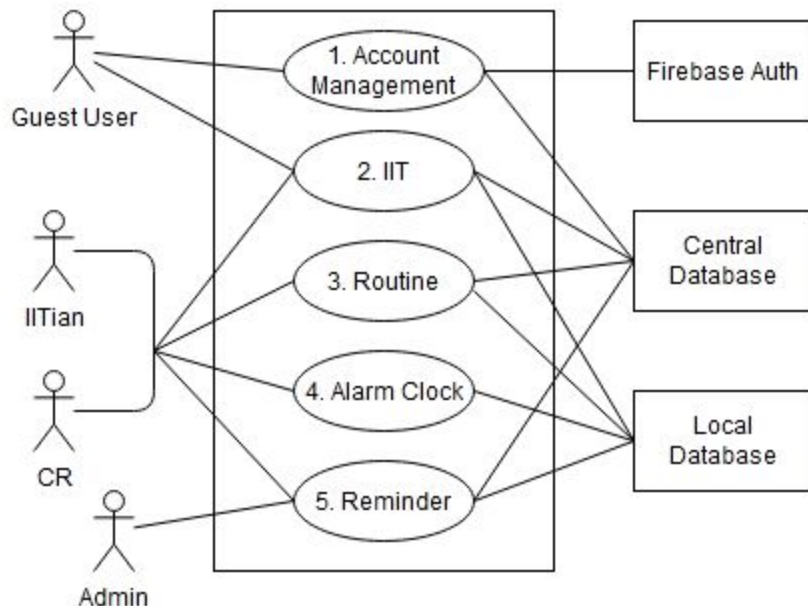
- 1. Mobile APP:** This sub-system covers all the functionalities of our mobile app. This sub-system is further divided in level 1.1.

- 2. Admin Panel:** Maintaining and modifying the information of central database is the responsibility of this sub-system. This sub-system is further divided in level 2.1.

### Use Case Diagram level: 1.1

Name: Mobile App

ID: 03



**Description:** The Mobile App is divided into 5 components. Their action-reply is discussed below:

- 1.1.1. **Account Management:** This component will manage all sorts of user account related issues. This module will be further divide in level 1.1.1
- 1.1.2. **IIT:** This component covers all the functionalities regarding the IIT website. This component is further divided in level 1.1.2

- 1.1.3. **Routine:** This component manages the “Dynamic Routine” feature of the app. It is further divided in level 1.1.3.
- 1.1.4. **Alarm Clock:** This component manages the “Good Morning Alarm” feature of the app.

**User Action:** User will set how much time before the earliest class of the day he wants his alarm to go off.

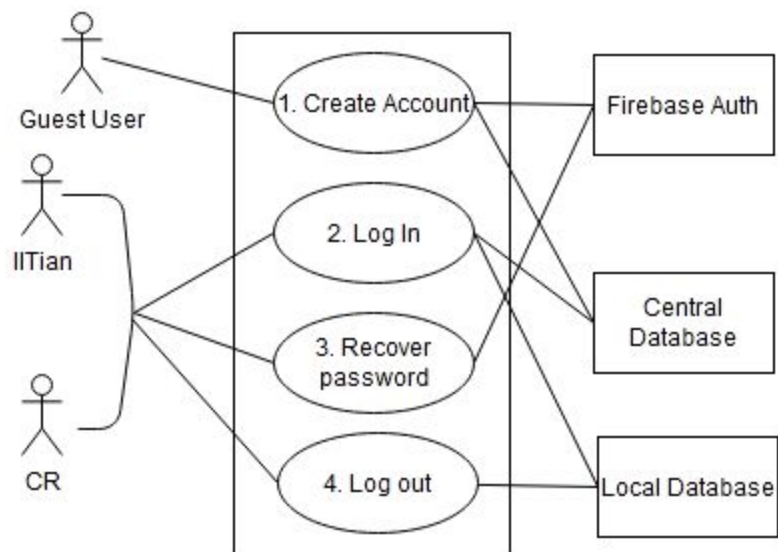
**System Reply:** System will sync with the “Routine” component to set the alarm at user defined time before the earliest class & adapt if that class is rescheduled.

- 1.1.5. **Reminder:** This component manages the Reminder & Notification feature of the app. It is further decomposed at level 1.1.5

### Use Case Diagram level: 1.1.1

Name: Account Management

ID: 04





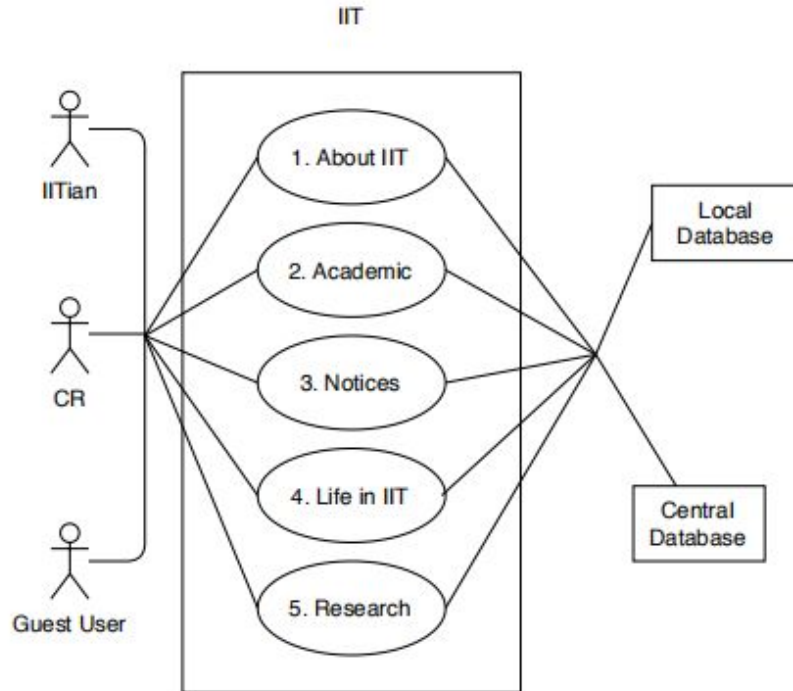
**Description:** The Account management is divided into 5 components.  
Their action-reply is discussed below:

- 1.1.1.1.      **Create Account:** This sub module will handle account creation.  
                 **User Action:** User will provide his username, mail address &  
                 account password.  
                 **System Reply:** If valid, a new account will be created.
- 1.1.1.2.      **Log In:** This sub module will be used for logging in.  
                 **User Action:** User will provide his mail address and password.  
                 **System Reply:** If valid, user will be logged in.
- 1.1.1.3.      **Recover Password:** This sub module will handle password  
                 recovery issues  
                 **User Action:** User will provide his mail address.  
                 **System Reply:** After validity check, user will be allowed to set a  
                 new password
- 1.1.1.4.      **Log out:** This is a simple module for logging out  
                 **User Action:** User will choose to log out  
                 **System Reply:** User will be logged out and will become a guest  
                 user

**Use Case Diagram level: 1.1.2**

Name: IIT

ID: 05



**Description:** This level shows all the main components to cover the IIT website. The components & their Action-Reply is given below.

- 1.1.2.1 **About IIT:** Shows all the information about IIT, including Institution, History, Aims & Objectives, Faculty, Officers & Staffs.

**User Action:** User chooses to see “About IIT”

**System Reply:** The app gets the information from the local database if the internet connection is off, otherwise it shows the updated information from the central database.

- 1.1.2.2 **Academic:** Shows all the “Academic” information, including Undergraduate Studies, Graduate Studies and Training programs.

**User Action:** User chooses to see “Academic”

**System Reply:** The app gets the information from the local database if the internet connection is off, otherwise it shows the updated information from the central database.

- 1.1.2.3 **Notices:** Shows all kinds of Notices i.e. General, Projects, Undergraduate Studies, Graduate Studies and Training Programs.

**User Action:** User chooses to see “Notices”

**System Reply:** The app gets the information from the local database if the internet connection is off, otherwise it shows the updated information from the central database.

- 1.1.2.4 **Life in IIT:** Shows the information regarding News & Events, Upcoming Events & Achievements.

**User Action:** User chooses to see “Life in IIT”

**System Reply:** The app gets the information from the local database if the internet connection is off, otherwise it shows the updated information from the central database.

- 1.1.2.5 **Research:** Shows information about research that is going on at IIT.

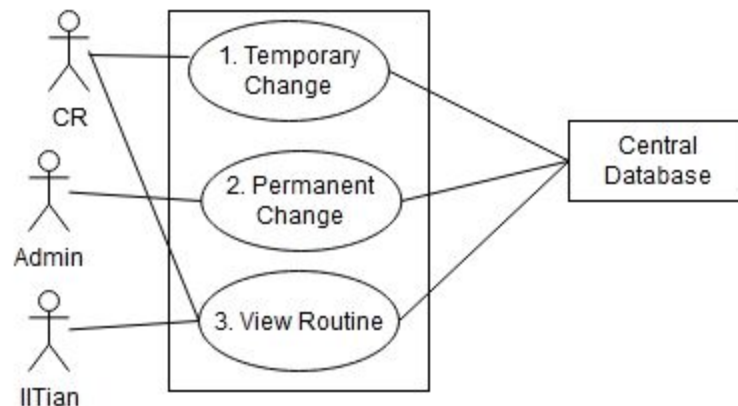
**User Action:** User chooses to see “Research”

**System Reply:** The app gets the information from the local database if the internet connection is off, otherwise it shows the updated information from the central database.

### Use Case Diagram level: 1.1.3

Name: Routine

ID: 06



**Description:** This module covers the Routine feature and handles the Temporary and Permanent schedule changes in daily routine.

- 1.1.3.1. **Temporary Change:** This sub-module is responsible for handling temporary routine changes such as- Class cancellation, Class rescheduling, Extra class or class swap. This module is further divided in level 1.1.3.1
- 1.1.3.2. **Permanent Change:** This will handle all permanent changes in the routine within a semester.  
**User Action:** The Admin will provide information for the new schedule from the admin panel.  
**System Reply:** If valid, the routine will be changed in the central database & local database will be updated accordingly when it is connected to the internet.
- 1.1.3.3. **View Routine:** This will handle displaying the routine of this week for instructors and students.

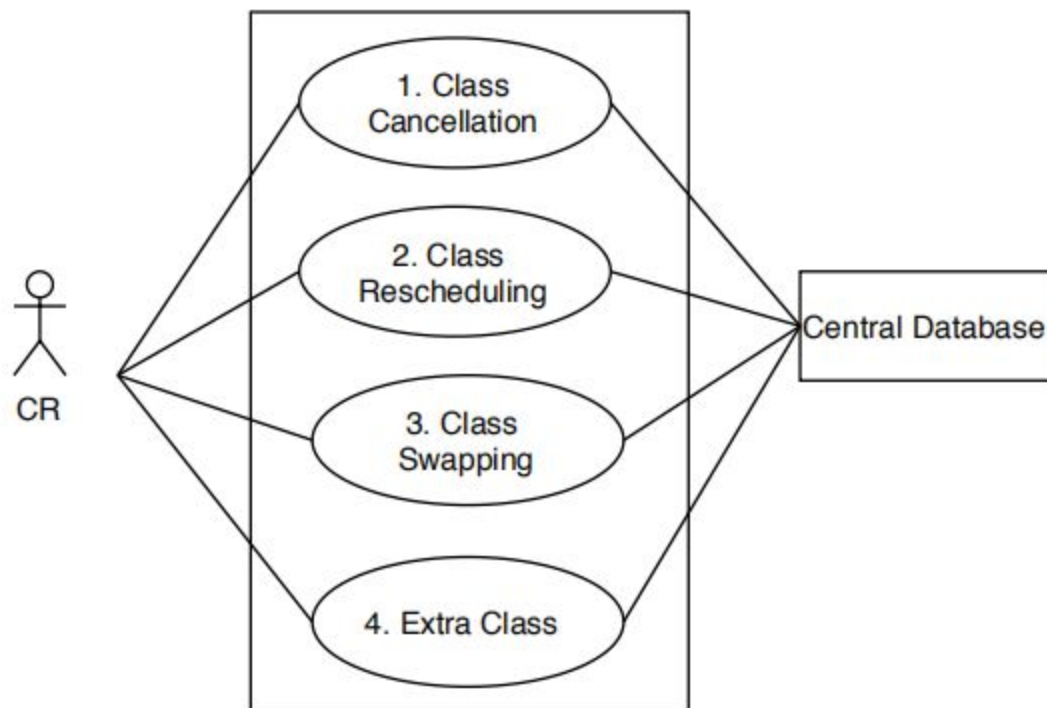
**User Action:** if the user is Instructor, he need not provide any additional info, but a student has to provide his/her semester info to view the routine.

**System Reply:** The system will provide routine of this week

### Use Case Diagram level: 1.1.3.1

Name: Temporary Routine Change

ID: 07



**Description:** This module shows the sub-modules for temporary routine change modules. The action-replies of its sub-modules are given below.

1.1.3.1.1 **Class Cancellation:** This sub-module will handle class cancellation related schedule change.

**User Action:** The CR will provide course code and date to cancel a class

**System Reply:** If valid, the class will be cancelled on that date

1.1.3.1.2 **Class Rescheduling:** This module will handle class rescheduling related changes.

**User Action:** The CR will provide course code, old and new date, begin and end time to reschedule a class.

**System Reply:** If valid, the class will be rescheduled for the new date

1.1.3.1.3 **Class Swapping:** This module is will handle the swapping of class related changes

**User Action:** The CR will provide the course codes and their dates

**System Reply:** If valid, the class the classes will be swapped

1.1.3.1.4 **Extra Class:** This module is will handle allotting extra class related changes

**User Action:** The CR will provide course code, date, begin and end time to allot an extra class.

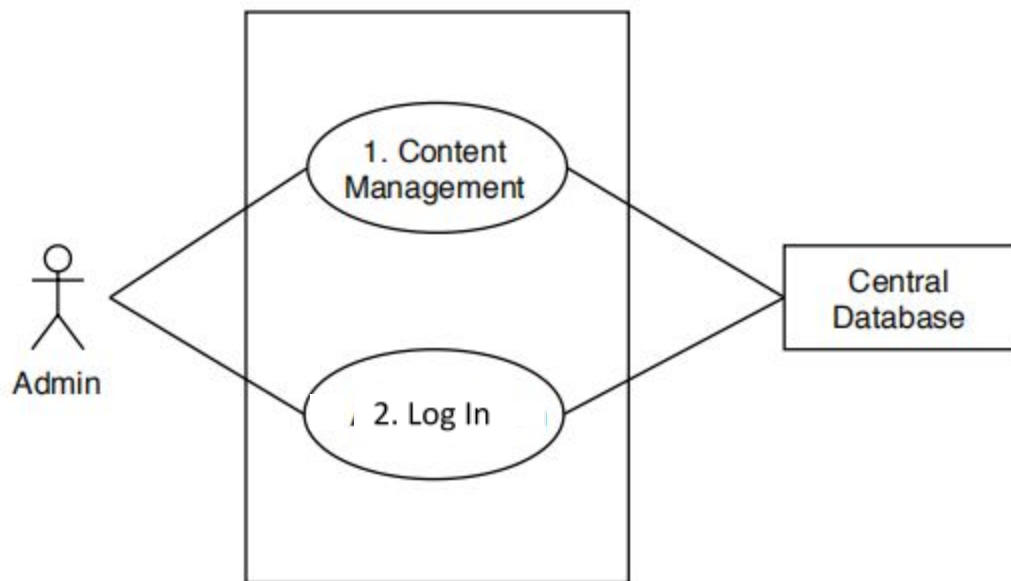
**System Reply:** If valid, the class will be allotted on that day.



## Use Case Diagram level: 1.2

Name: Admin Panel

ID: 09



**Description:** This module organizes the administrative functions of the system with the help of 3 sub-modules. These sub modules are further described below:

1.2.1 **Content Management:** This sub-module manages all the contents of this app. This sub-module will be further divided and described in level 1.2.1.1

1.2.2 **Log In:** This module is for authenticating as an Admin in Admin Panel.

**User Action:** Admin will provide his username & password.

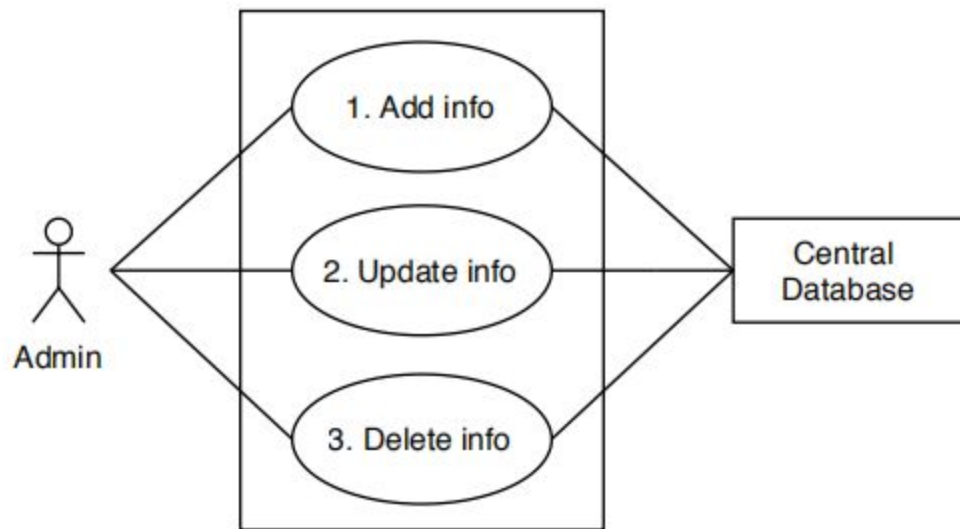
**System Reply:** If the username and password is authentic, the system will give him access to the Admin Panel.



## Use Case Diagram level: 1.2.1

Name: Content Management

ID: 10



**Description:** The module handles the content related management of the app. Its sub-modules are described below.

1.2.1.1. **Add info:** Admin can add new info related to IIT

**User Action:** Admin will provide new info

**System Reply:** New info will be added to the central database.

1.2.1.2. **Update info:** Admin can update existing info related to IIT

**User Action:** Admin will provide updated info

**System Reply:** Updated Info will be added to the central database.

1.2.1.3. **Delete info:** Admin can delete info related to IIT

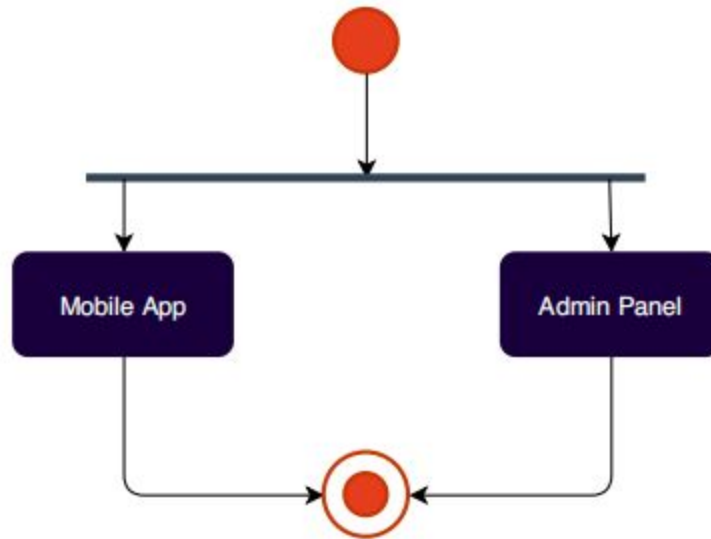
**User Action:** Admin will select which info will be deleted.

**System Reply:** The selected info will be deleted from central database

## 4.4 ACTIVITY DIAGRAMS

**Activity Diagram ID: 01**

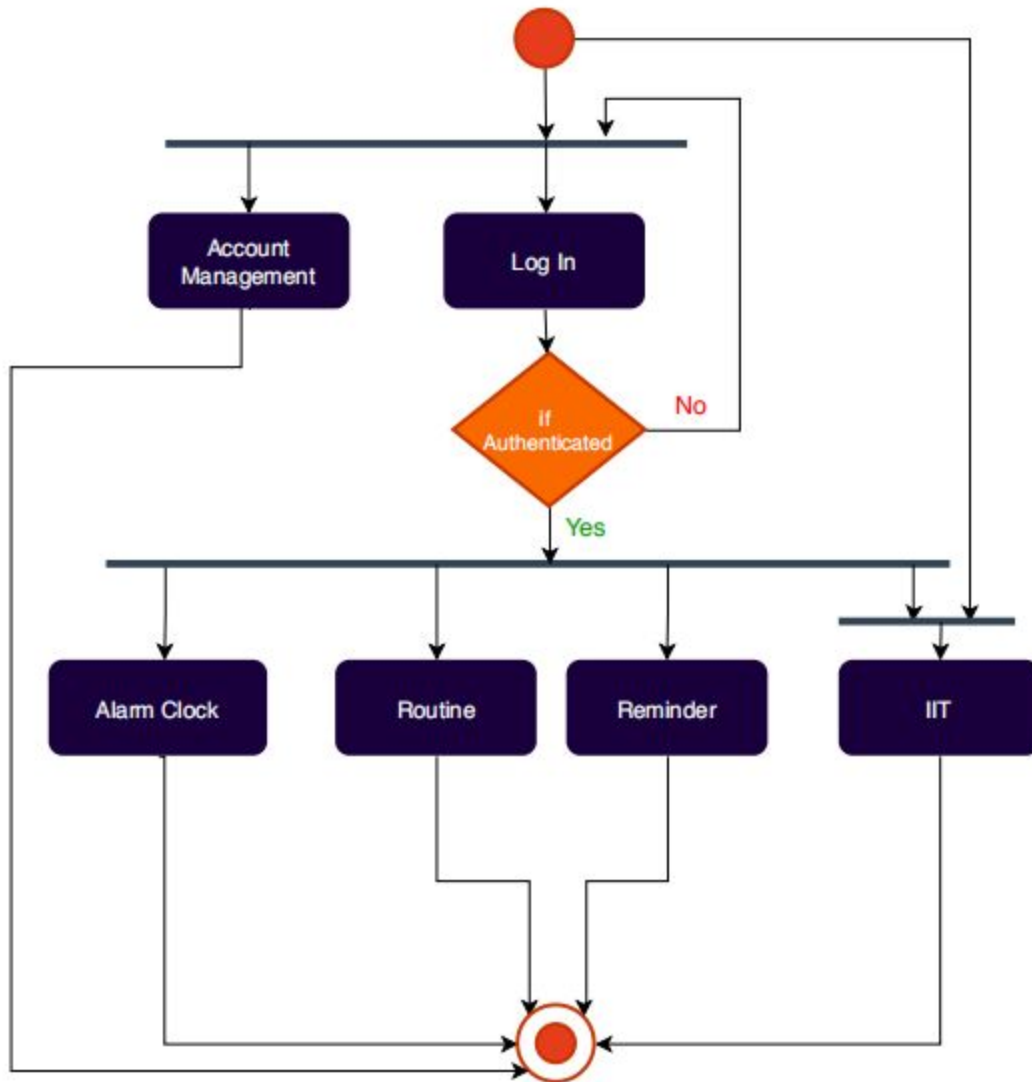
**Name:** IIT 360



**Description:** This is the activity diagram of corresponding use case diagram level 1, ID: 2. It is the top-level activity view of IIT 360.

## Activity Diagram ID: 02

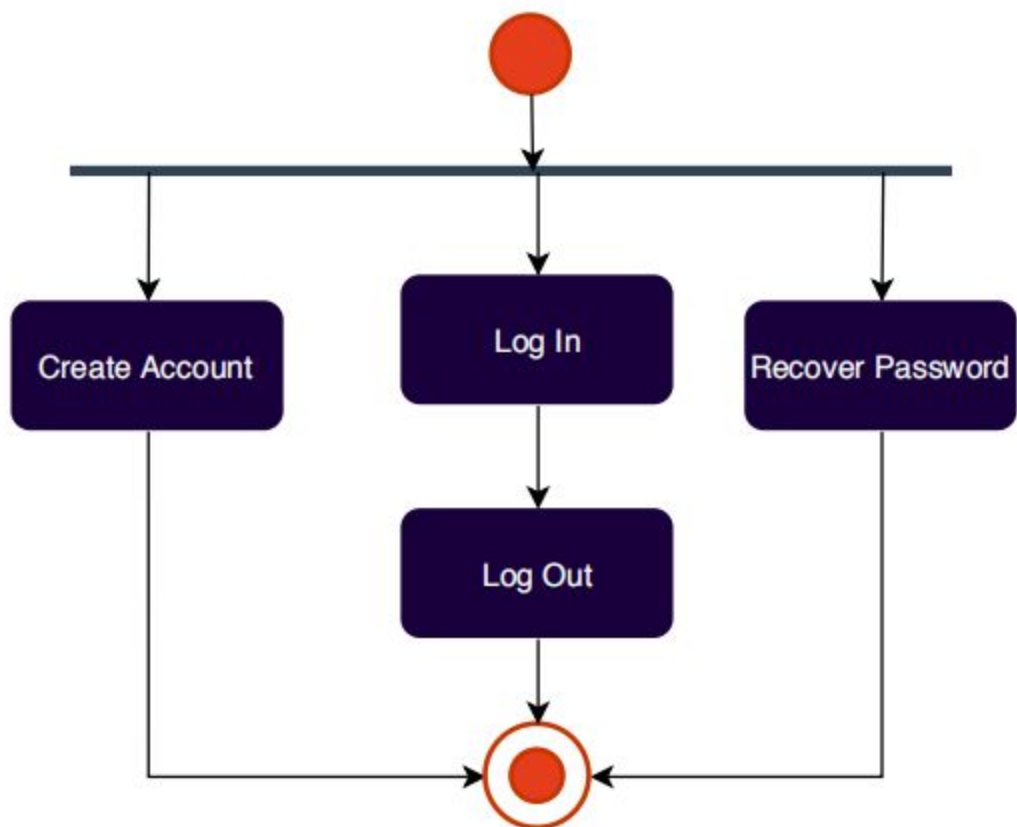
Name: Mobile App



**Description:** This is the activity diagram of corresponding use case diagram level 1.1, ID: 3. It shows the broad level activities of the app and user activities before and after logging in as an IITian.

### Activity Diagram ID: 03

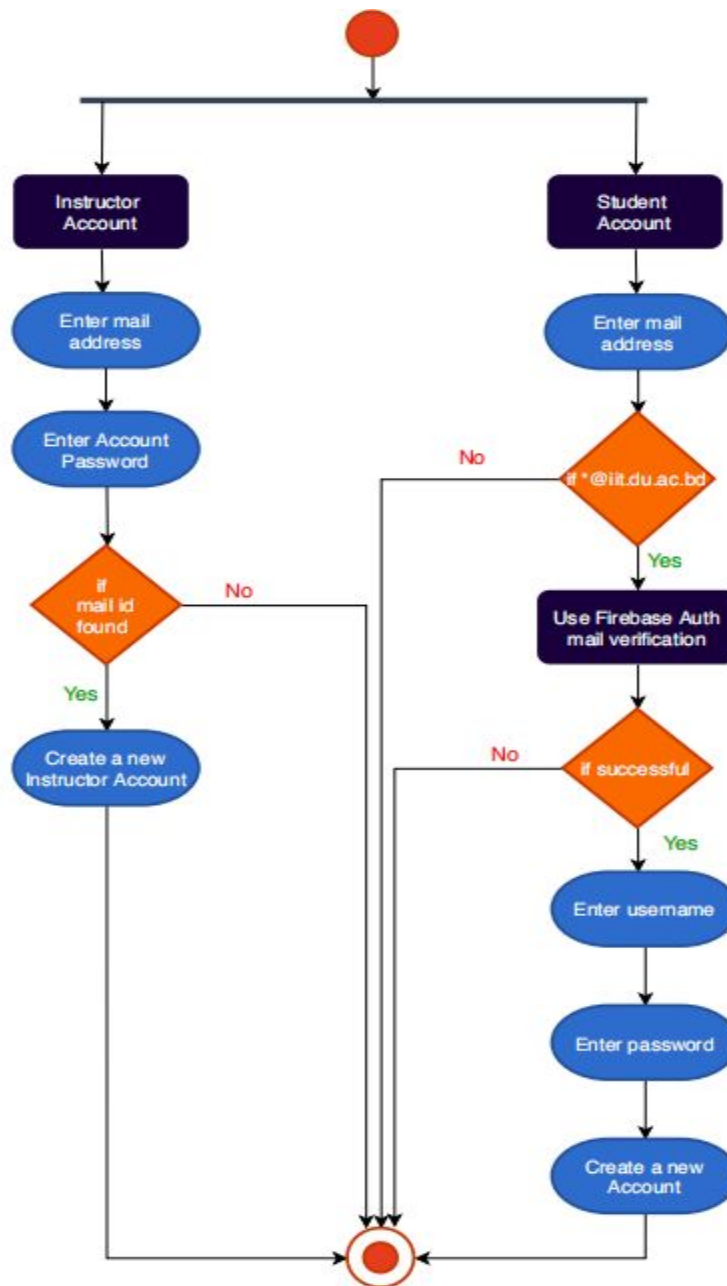
**Name:** Account Management



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.1, ID: 4. It shows all the activities of the Account Management module.

## Activity Diagram ID: 04

Name: Create Account



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.1.1, ID: 4. It shows the series of activities for creating a new instructor or student account.

## Activity Diagram ID: 05

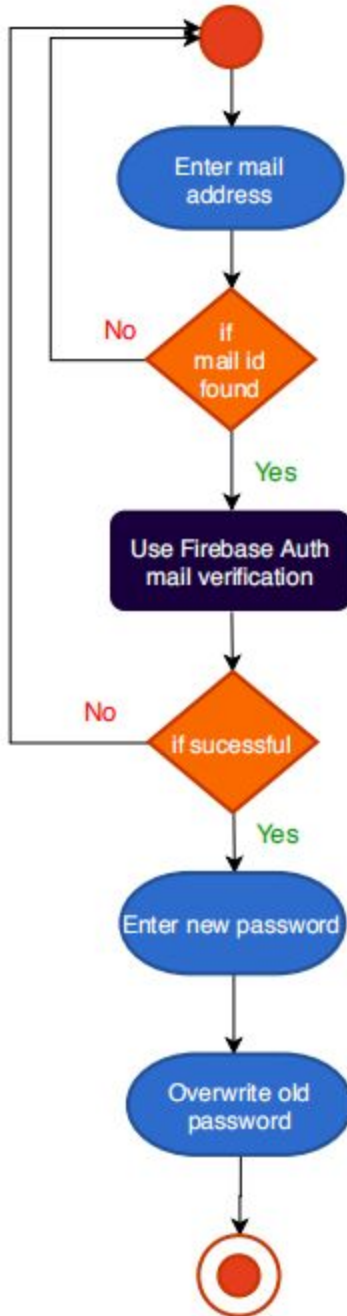
Name: Log In



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.1.2, ID: 4. It shows the steps for logging in using mail id and password.

## Activity Diagram ID: 06

**Name:** Recover password

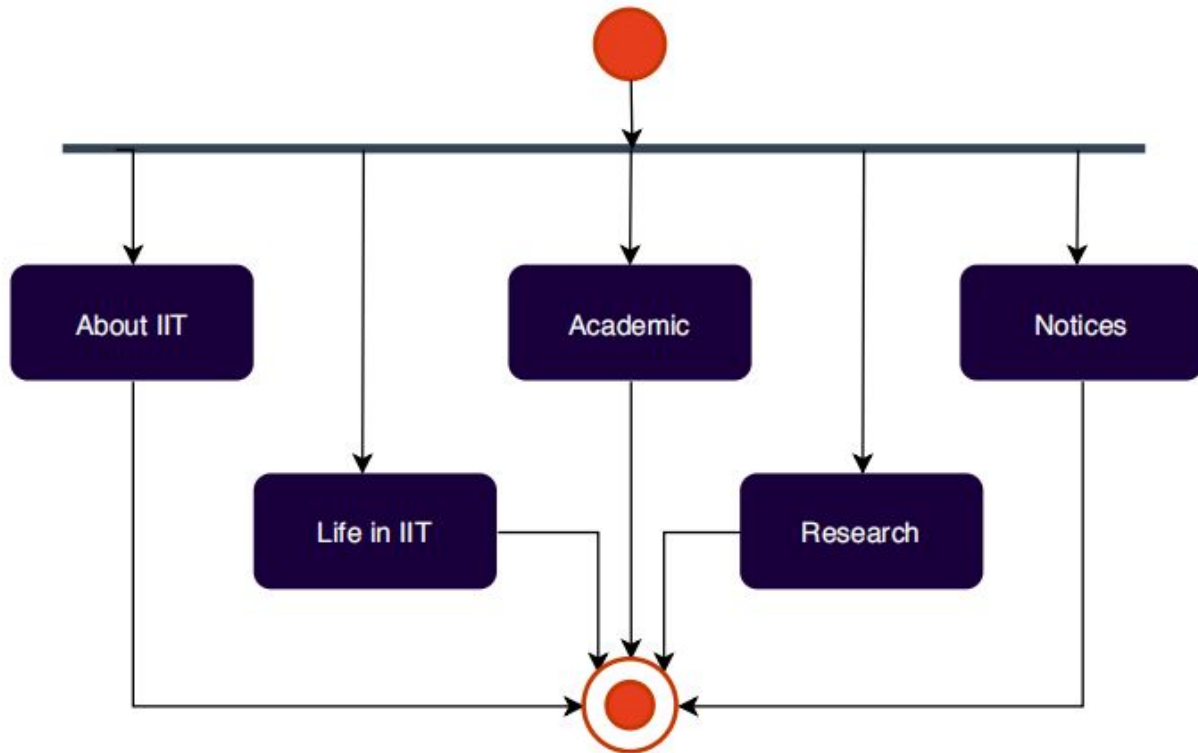


**Description:** This is the activity diagram of corresponding use case diagram level 1.1.1.3, ID: 4. It shows the steps for recovering a password.



## Activity Diagram ID: 07

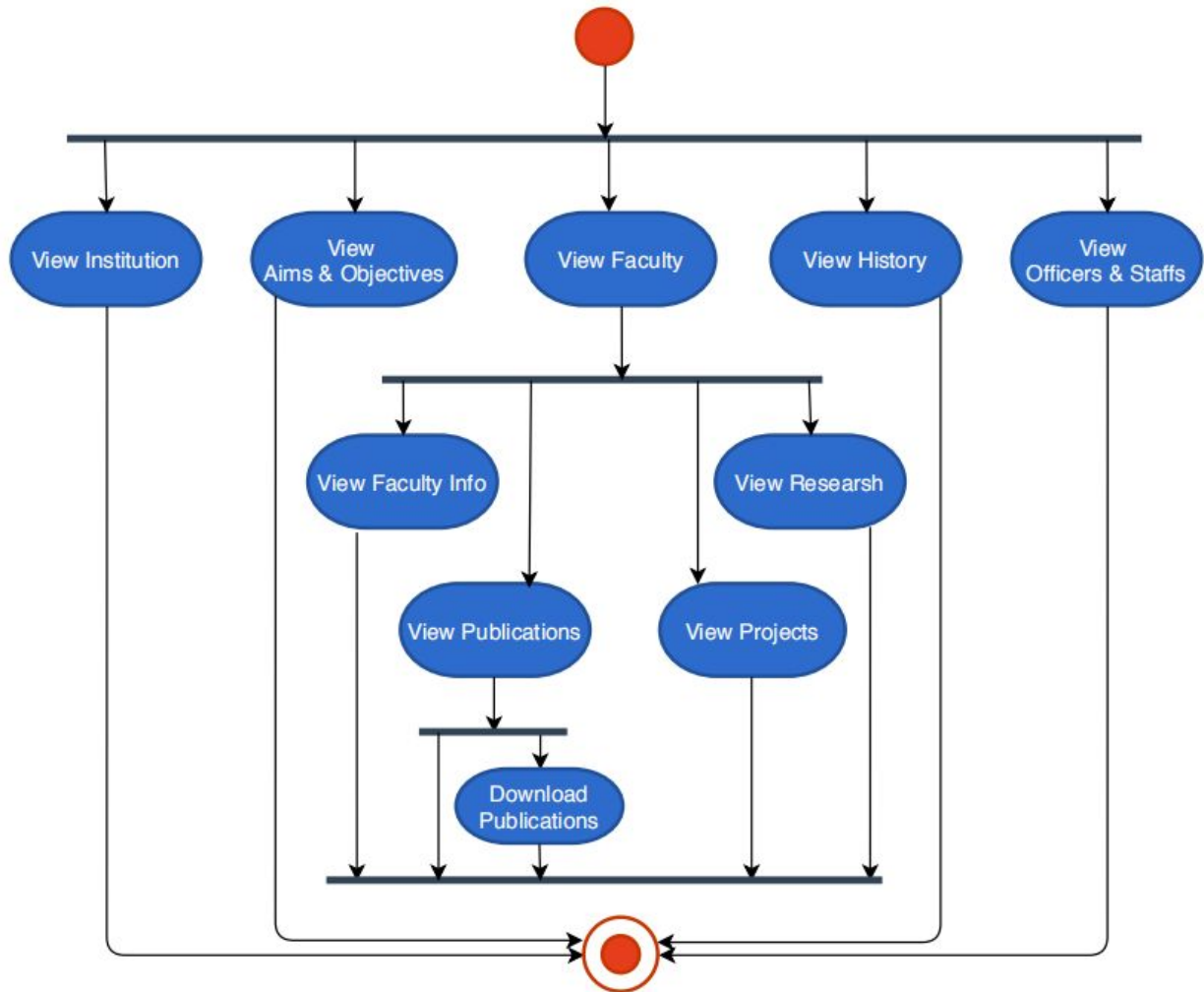
Name: IIT



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2, ID: 5. It shows the activities within the IIT feature.

## Activity Diagram ID: 08

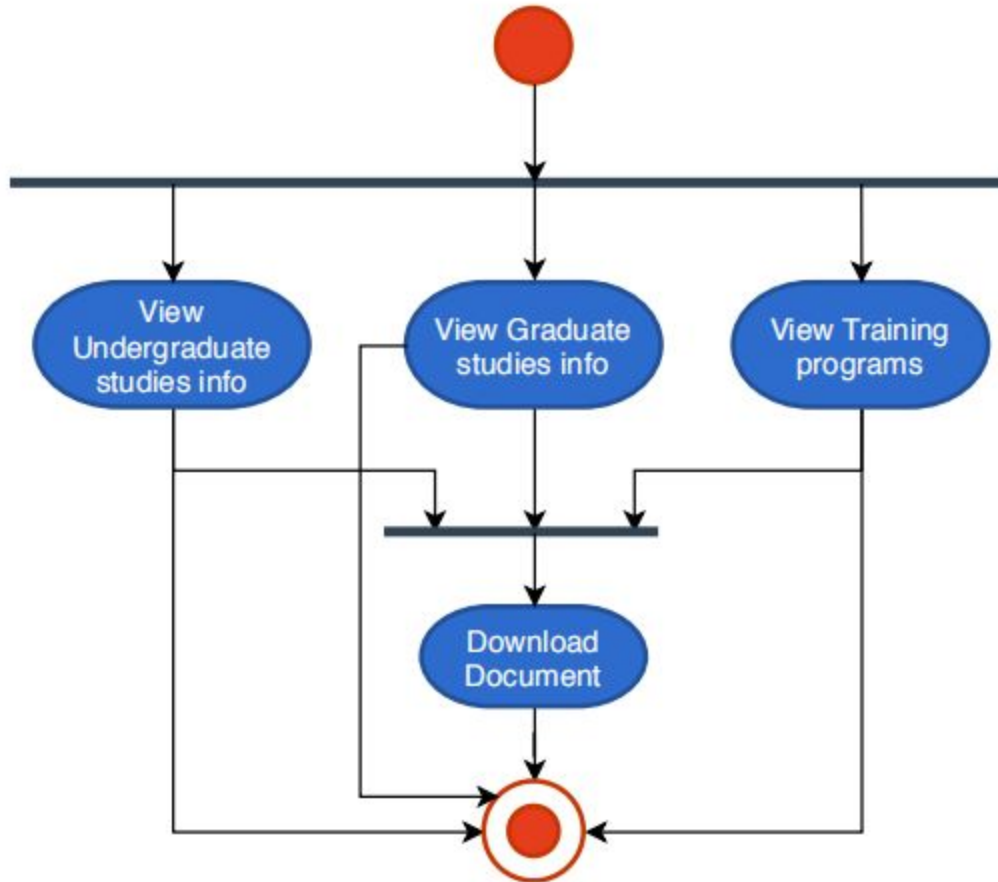
Name: About IIT



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2.1, ID: 5. It shows the user activities in “About IIT”.

## Activity Diagram ID: 09

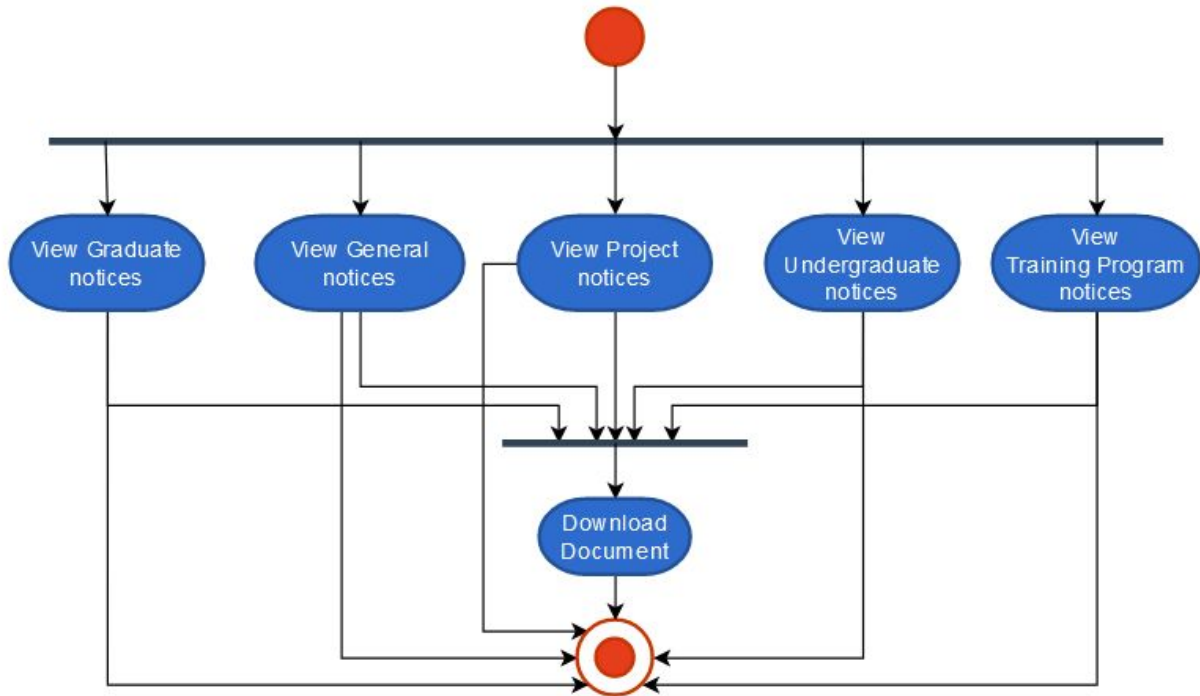
Name: Academic



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2.2, ID: 5. It shows the user activities in “Academic”.

## Activity Diagram ID: 10

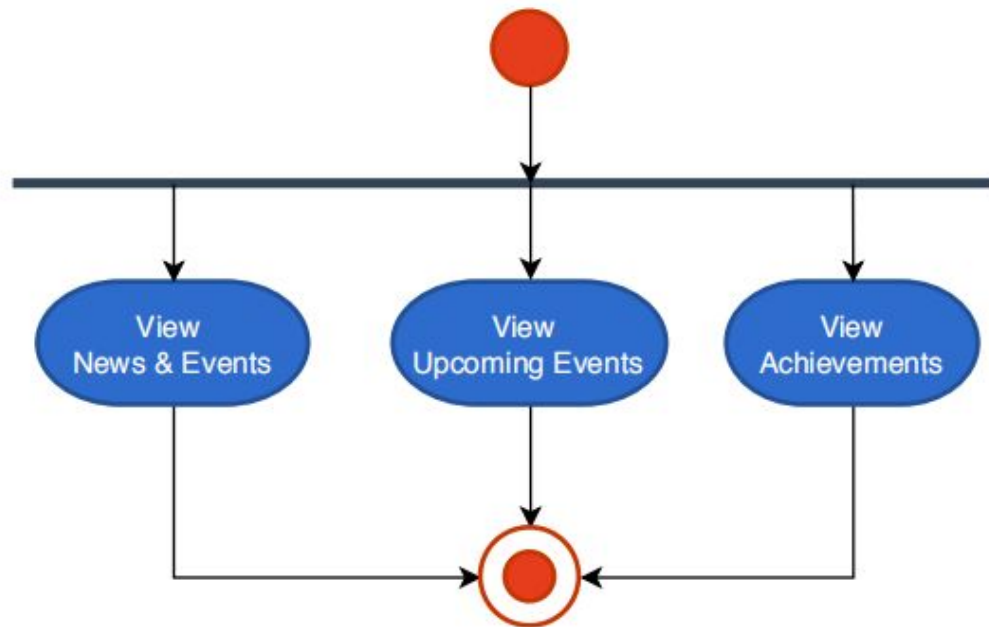
**Name:** Notices



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2.3, ID: 5. The activity flow of a user in “Notices” is shown here.

## Activity Diagram ID: 11

**Name:** Life in IIT



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2.4, ID: 5. It shows the activity for 3 sub sections of “Life in IIT”.

**Activity Diagram ID: 12**

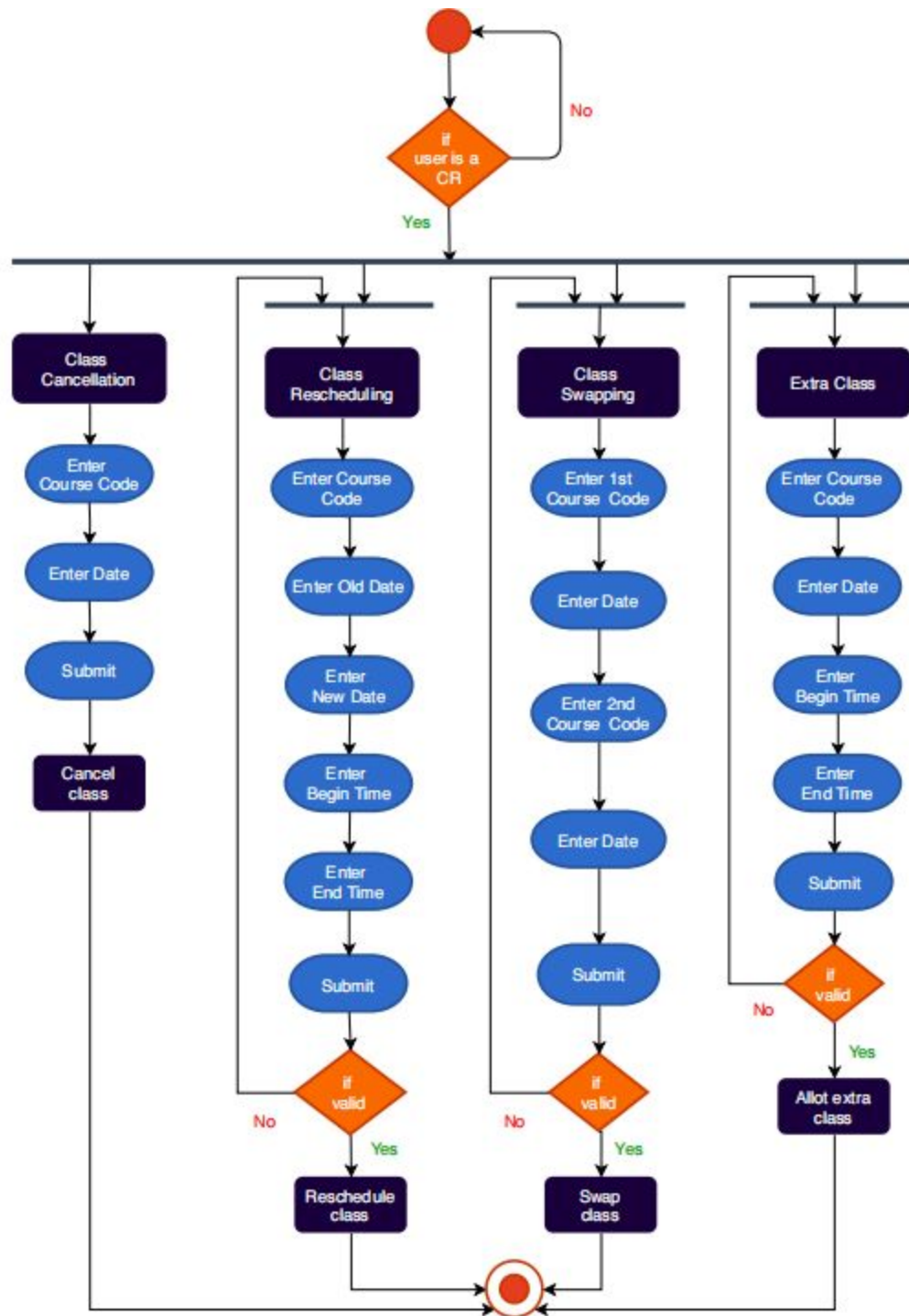
**Name:** Research



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.2.4, ID: 5. This shows a single activity for “Research” sub-section of “IIT”.

## Activity Diagram ID: 13

Name: Temporary Routine Change

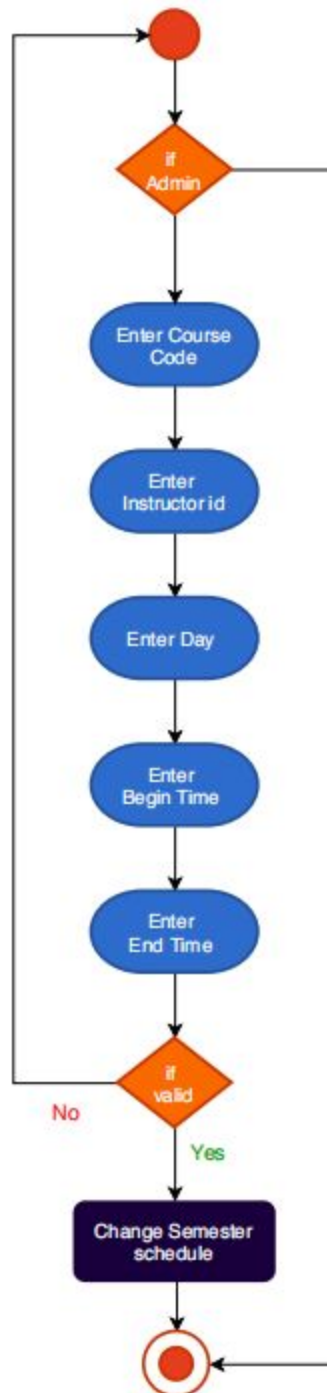


**Description:** This is the activity diagram of corresponding use case diagram level 1.1.3.1, ID: 7. It shows the necessary steps for each type of temporary routine change, i.e. Class Cancellation, Class Rescheduling, Class Swapping & Extra Class. It is important to note here that whether the user is a CR for a specific semester will be confirmed before he/she commits any change in routine.



## Activity Diagram ID: 14

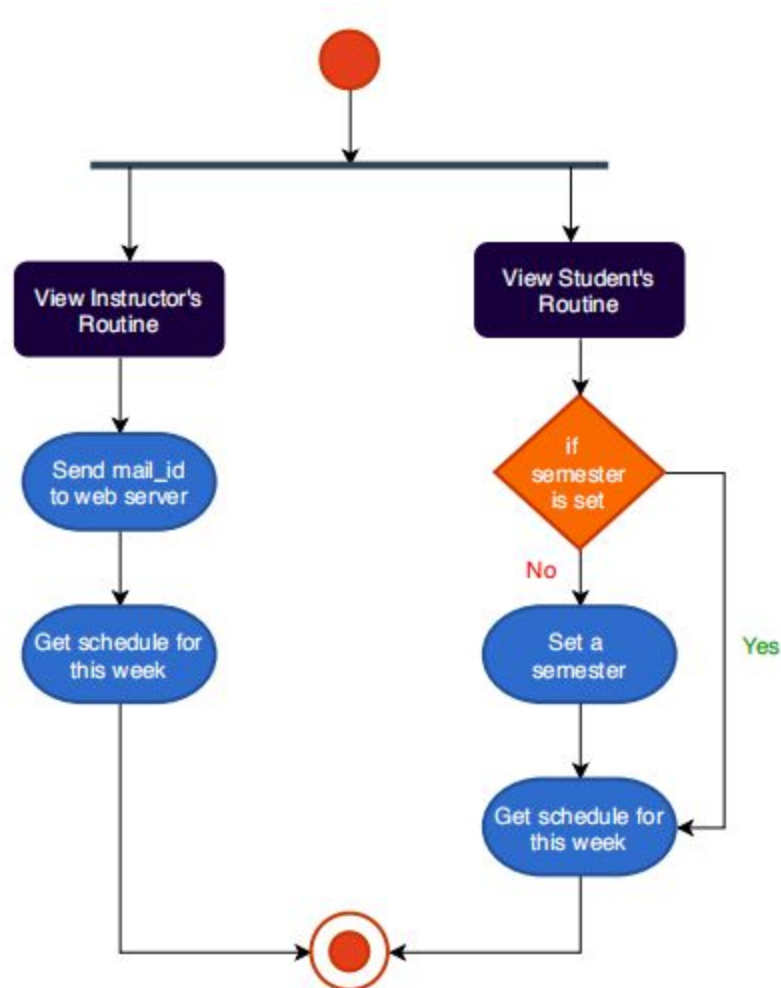
**Name:** Permanent Routine Change



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.3.2, ID: 6. It shows activities for permanent routine change for a semester. We can see only the admin is privileged to make this change.

### Activity Diagram ID: 15

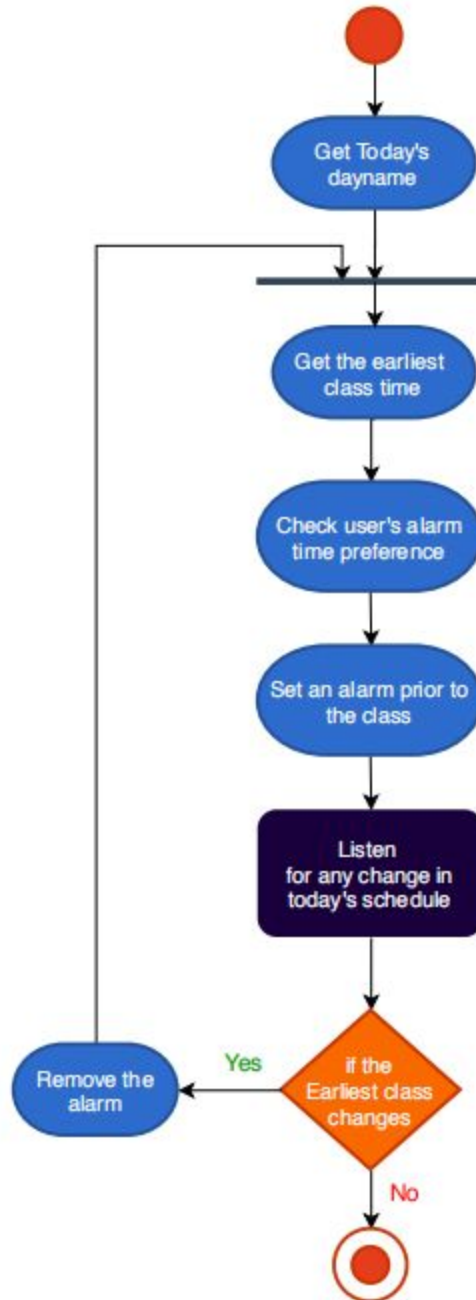
**Name:** View Routine



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.3.3, ID: 6. It shows activities to view the routine of this week as an instructor or as a student.

## Activity Diagram ID: 16

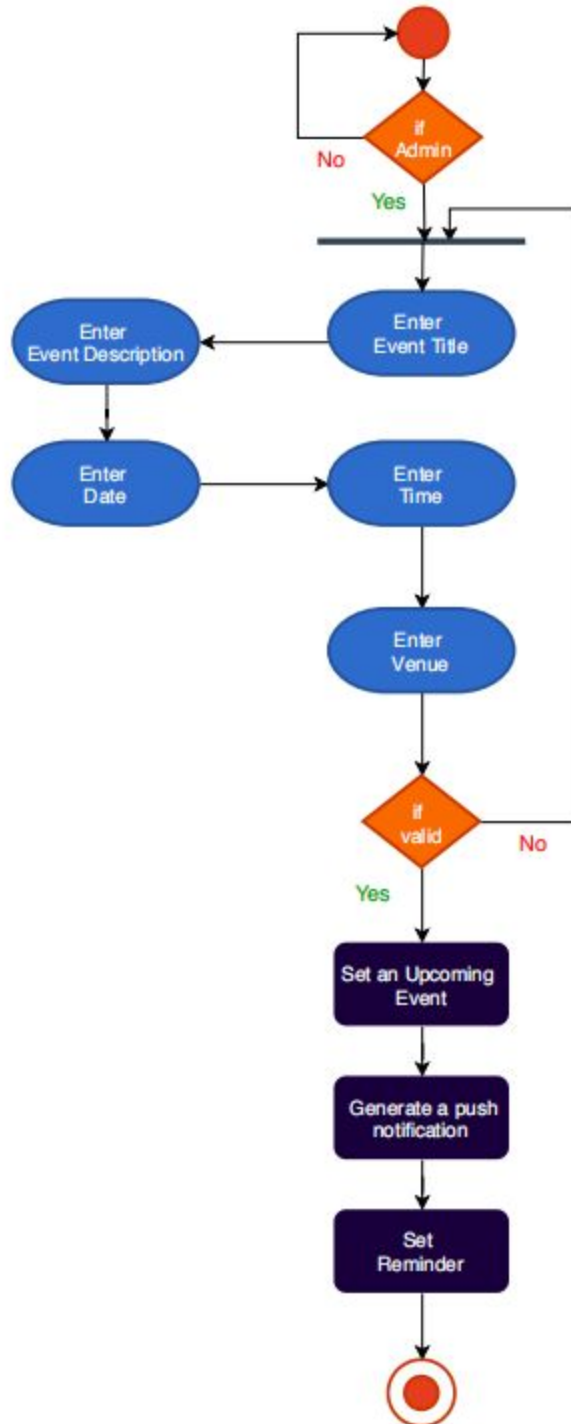
**Name:** Alarm Clock



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.4, ID: 3. It shows the activities for setting an alarm step by step.

## Activity Diagram ID: 17

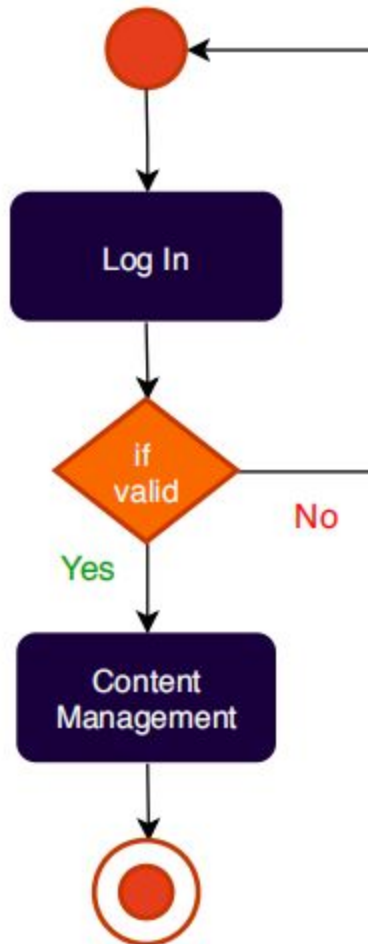
Name: Reminder



**Description:** This is the activity diagram of corresponding use case diagram level 1.1.5, ID: 8. It shows the steps for adding a new event by the Admin & how a reminder for that event will be set by the app.

### Activity Diagram ID: 18

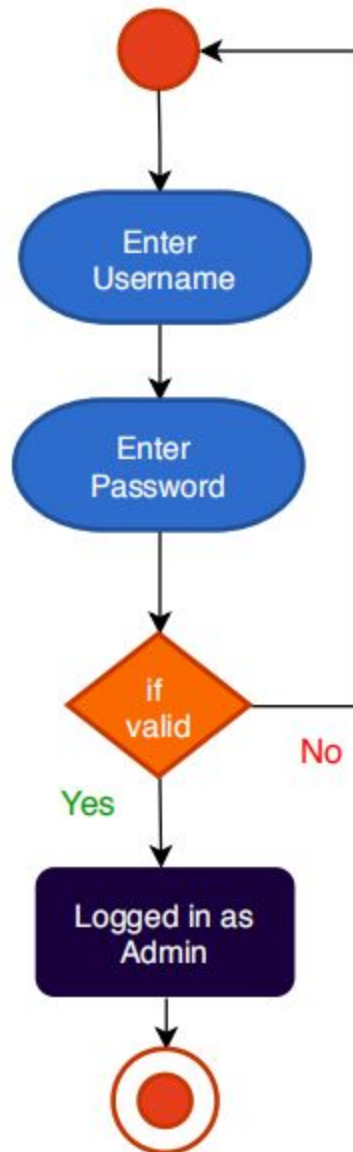
**Name:** Admin Panel



**Description:** This is the activity diagram of corresponding use case diagram level 1.2, ID: 2 It shows the top-level activity for Admin Panel.

**Activity Diagram ID: 19**

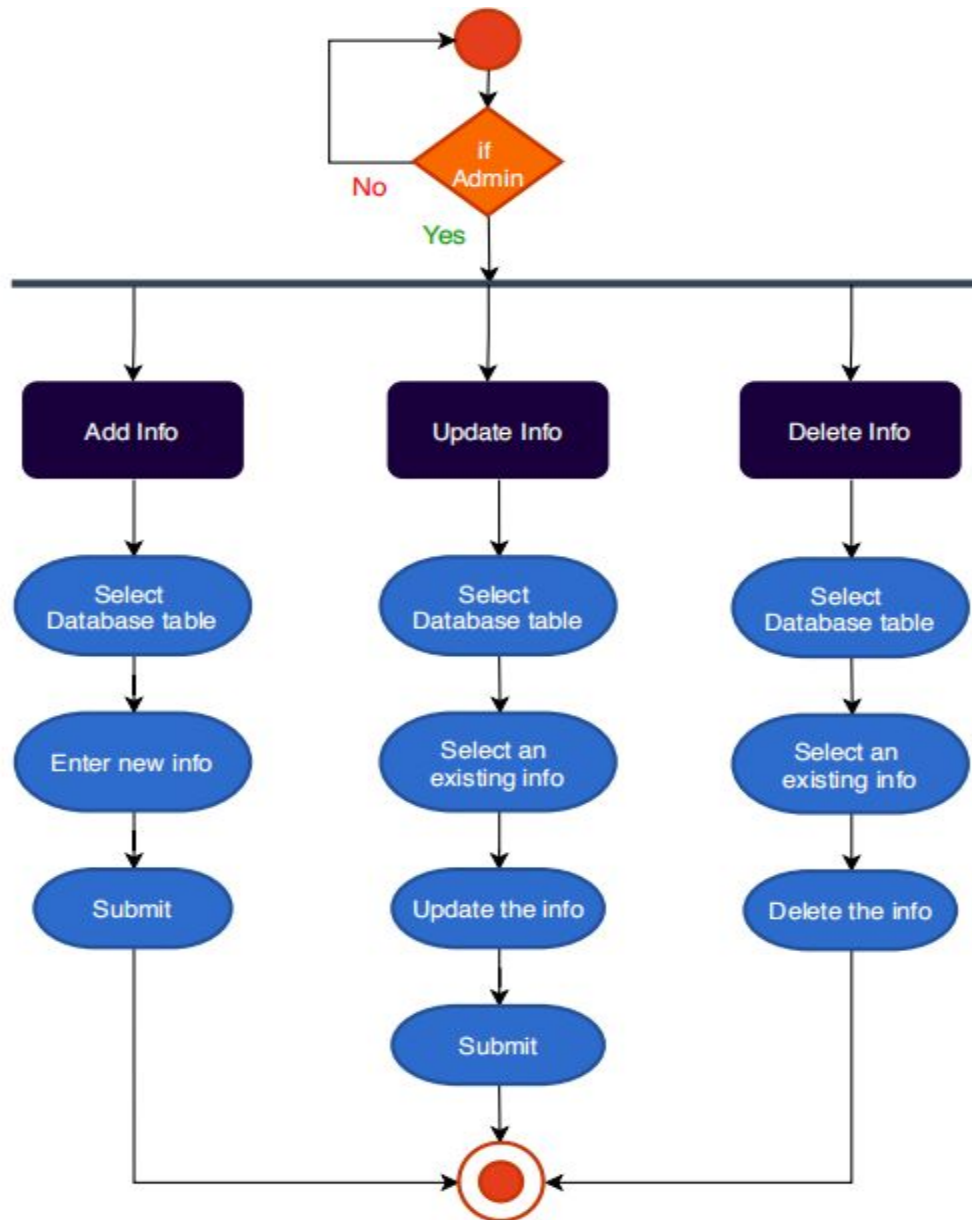
**Name:** Admin Log In



**Description:** This is the activity diagram of corresponding use case diagram level 1.2.2, ID: 9. It shows the steps accessing the Admin Panel.

## Activity Diagram ID: 20

**Name:** Content Management



**Description:** This is the activity diagram of corresponding use case diagram level 1.2.1, ID: 10. It shows all the activities in the Admin Panel and the steps of how an Admin can add, update or purge an info.

## 5. DATA-BASED MODELLING OF IIT 360

### 5.1 Data Modelling Concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) identifies all data objects that are processed within the system, the relationships

between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

#### 5.1.1 Data Objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

##### 5.1.1.1 Data Object Identification

We identified all the nouns whether they are in problem space or in solution space from our story:



<b>Serial No.</b>	<b>Noun</b>	<b>Problem (p)/ Solution(s) Space</b>	<b>Attribute</b>
1	User	p	4,11,30
2	IITian	s	
3	Guest	s	
4	IIT mail	p	
5	Authentication	p	
6	Authorization	p	
7	Information	p	
8	Name	s	
9	BSSE	s	
10	MSSE	s	
11	Semester	s	
12	Photo	s	
13	Faculty	s	8, 37, 38, 39
14	Routine	s	25,13,31,11,26,32,33
15	Alarm	s	25, 27
16	Notification	s	
17	Reminder	s	
18	Unauthorized	p	
19	CR	s	
20	Class Cancellation	s	
21	Class Swapping	s	

22	Class Addition	s	
23	Class Rescheduling	s	
24	Hours	s	
25	Date	s	
26	Day	s	
27	Time	s	
28	Event	s	
29	Assignment	s	
30	User Type	s	
31	Course Code	s	
32	Begin Time	s	
33	End Time	s	
34	Assignment Reminder	s	35, 25, 31, 11, 36
35	Title	s	
36	Description	s	
37	Designation	s	
38	Links	s	
39	About Me	s	
40	Research	s	35, 36
41	Project	s	35, 36
42	Publication	s	35, 43
43	Type	s	
44	PDF	s	

45	Document	s	44
46	Image	s	47, 48, 49
47	Details	s	
48	Height	s	
49	Width	s	
50	NewsAndEvent	s	35, 36, 25, 27, 53
51	Achievement	s	35, 36, 25, 53
52	Font	s	54, 55, 56, 57
53	Venue	s	
54	Font_Color	s	
55	Font_Size	s	
56	Font_Style	s	
57	Font_Family	s	
58	Information	s	59
59	Data	s	
60	Notice	s	35, 36, 25, 27

### 5.1.1.2 Final Data Objects

In the following table we finalize the data objects with their attributes. Most of the attributes of the data objects are selected from the usage scenario and some of the attributes are selected to complete the system which are not in the usage scenario but important for the data objects.

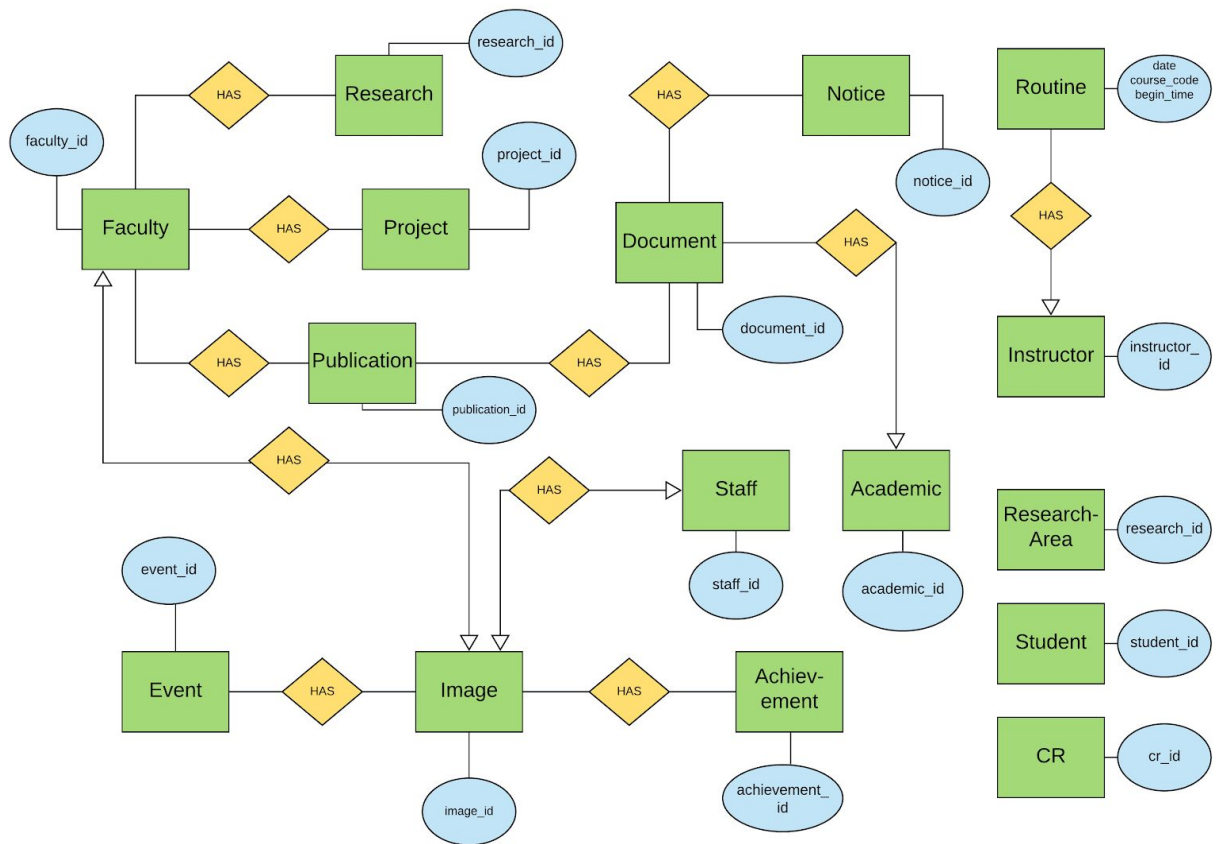
1. User
2. Routine
3. Alarm
4. AssignmentReminder
5. Faculty
6. Research
7. Project
8. Publication
9. Document
10. Image
11. Notice
12. NewsAndEvent
13. Achievement
14. Academic
15. ResearchArea

## 5.2 DATA OBJECT RELATIONSHIPS

Data objects are connected to one another in different ways.

1. A Faculty can do multiple Projects and a Project can be done by multiple Faculties. **(Many to Many)**
2. A Faculty can do multiple Research and a Research can be done by multiple Faculties. **(Many to Many)**
3. A Faculty can have multiple Publications and a Publication can be owned by multiple Faculties. **(Many to Many)**
4. A Publication has multiple Documents and a document belongs to multiple Publications. **(Many to Many)**
5. An Event has multiple Images and an image belongs to multiple Events. **(Many to Many)**
6. An Achievement has multiple Images and an image belongs to multiple Achievements. **(Many to Many)**
7. A Faculty has one Image and an Image belongs to one Faculty **(One to One)**
8. A Staff has one Image and an Image belongs to one Staff **(One to One)**
9. A Notice has multiple Documents and a Document can belong to multiple Notices **(Many to Many)**
10. A Routine has one Instructor and an Instructor belongs to many Routines **(One to Many)**
11. An Academic has many Documents and a Document has one Academic **(One to Many)**

## 5.3 ENTITY RELATIONSHIP DIAGRAM



**Fig:** Entity Relationship Diagram

## 5.4 SCHEMA DIAGRAM

Data Object	Attribute	Type	Size
student	<u>student_mail</u> (PK)	VARCHAR	45
	semester	VARCHAR	5
	name	VARCHAR	45
	connected	TINYINT	
	last_update	DATETIME	
instructor	<u>instructor_id</u> (PK)	VARCHAR	5
	Instructor_mail	VARCHAR	25
	Instructor_name	VARCHAR	45
	connected	TINYINT	
	last_update	DATETIME	
routine	<u>date</u> (PK)	DATE	
	<u>course_code</u> (PK)	VARCHAR	10
	<u>begin_time</u> (PK)	TIME	
	end_time	TIME	
	semester	VARCHAR	5
	day	VARCHAR	45
	fk_instructor_id (FK)	VARCHAR	5
faculty	<u>faculty_id</u> (PK)	VARCHAR	5

	name	VARCHAR	45
	designation	VARCHAR	45
	qualification	VARCHAR	45
	links	VARCHAR	500
	about_me	VARCHAR	10000
	fk_image_id (FK)	INT	
research	<u>research_id</u> (PK)	INT	
	title	VARCHAR	200
	description	VARCHAR	10000
project	<u>project_id</u> (PK)	INT	
	title	VARCHAR	200
	description	VARCHAR	5000
publication	<u>publication_id</u> (PK)	INT	
	title	VARCHAR	1500
	type	VARCHAR	15
document	<u>document_id</u> (PK)	INT	
	pdf	MEDIUMBLOB	
image	<u>image_id</u> (PK)	INT	
	image	MEDIUMBLOB	
	height	INT	
	width	INT	
notice	<u>notice_id</u> (PK)	INT	
	type	VARCHAR	20



	section	VARCHAR	50
	title	VARCHAR	100
	description	VARCHAR	500
	date	DATE	
	time	TIME	
event	<u>event_id</u> (PK)	INT	
	title	VARCHAR	100
	description	VARCHAR	10000
	date	DATE	
	time	TIME	
	venue	VARCHAR	100
achievement	<u>achievement_id</u> (PK)	INT	
	title	VARCHAR	100
	description	VARCHAR	500
	date	DATE	
	venue	VARCHAR	100
academic	<u>academic_id</u> (PK)	INT	
	info	VARCHAR	15000
	program	VARCHAR	100
	admission	VARCHAR	15000
researchArea	<u>research_area_id</u> (PK)	INT	
	research_field	VARCHAR	100
	description	VARCHAR	15000

staff	<u>staff_id</u> (PK)	INT	
	name	VARCHAR	45
	designation	VARCHAR	45
	fk_staff_image (FK)	INT	
cr	<u>cr_mail_id</u> (PK)	VARCHAR	45
event_image	<u>event_image_id</u> (PK)	INT	
	fk_event_evemge (FK)	INT	
	fk_image_evemge (FK)	INT	
achievement_image	<u>achievement_image_id</u> (PK)	INT	
	fk_achievement_achage (FK)	INT	
	fk_image_achage (FK)	INT	
notice_document	<u>notice_document_id</u> (PK)	INT	
	fk_notice_notent (FK)	INT	
	fk_document_notent (FK)	INT	
publication_document	<u>publication_document_id</u> (PK)	INT	
	fk_publication_pubdoc (FK)	INT	
	fk_document_pubdoc (FK)	INT	
faculty_publication	<u>faculty_publication_id</u> (PK)	INT	
	fk_faculty_facpub (FK)	VARCHAR	5
	fk_publication_facpub (FK)	INT	
faculty_research	<u>faculty_research_id</u> (PK)	INT	
	fk_faculty_facrch (FK)	VARCHAR	5

	fk_image_facrch <b>(FK)</b>	INT	
faculty_project	<u>faculty_project_id</u> <b>(PK)</b>	INT	
	fk_faculty_facpro <b>(FK)</b>	VARCHAR	5
	fk_project_facpro <b>(FK)</b>	INT	
academic_document	<u>academic_document_id</u> <b>(PK)</b>	INT	
	fk_academic_acaent <b>(PK)</b>	INT	
	fk_document_acaent <b>(PK)</b>	INT	

## 6. CLASS-BASED MODELING OF IIT 360

This Chapter is intended to describe class based modeling of “IIT 360”.

### 6.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

### 6.2 Noun List for CMS

No	Noun	No	Noun
1	User	31	Course Code
2	IITian	32	Begin Time
3	Guest	33	End Time
4	IIT mail	34	Assignment Reminder
5	Authentication	35	Title
6	Authorization	36	Description
7	Information	37	Designation
8	Name	38	Links
9	BSSE	39	About Me

10	MSSE	40	Research
11	Semester	41	Project
12	Photo	42	Publication
13	Faculty	43	Type
14	Routine	44	PDF
15	Alarm	45	Document
16	Notification	46	Image
17	Reminder	47	Details
18	Unauthorized	48	Height
19	CR	49	Width
20	Class Cancellation	50	NewsAndEvent
21	Class Swapping	51	Achievement
22	Class Addition	52	Font
23	Class Rescheduling	53	Venue
24	Hours	54	Font_Color
25	Date	55	Font_Size
26	Day	56	Font_Style
27	Time	57	Font_Family
28	Event	58	Information
29	Assignment	59	Data
30	User Type	60	Notice

### 6.3 Verb List of CMS

No	Verb	No	Verb
1	Authenticate	12	Remind Events
2	Identify	13	View Routine
3	Sign In	14	Reschedule Routine
4	Sign Up	15	Cancel Class
5	Sign Out	16	Swap Class
6	Set Alarm	17	Add Extra Class
7	Delete Alarm	18	Download Research
8	Snooze Alarm	19	View Project
9	Notify	20	View Publication
10	View Notice	21	Download Publication
11	View News and Events	22	View Image

### 6.4 GENERAL CLASSIFICATION

Candidate classes were then characterized in seven general categories. The seven general characteristics are as follows:

1. **External entities** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.

2. **Things** (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. **Occurrences or events** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. **Roles** (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. **Organizational units** (e.g., division, group, team) that are relevant to an application.
6. **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. **Structures** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

Following are the specifications of the nouns according to the general Classifications:

**Table:1 General Classification**

<b>NOUN</b>	<b>General Classification</b>
User	4,5,7
IIITian	4,5,7
Guest	4,5,7
Authentication	2,3
Alarm	2
Faculty	4,5,7
Notification	3,7
Routine	2,7

Reminder	3,7
CR	4,5,7
Routine Modification	3
EventReminder	3,7
Assignment	2,7
Assignment Reminder	3
Research	2,7
Project	2,7
Publication	2,7
Document	2,7
Image	2,7
NewsAndEvent	3,7
Achievement	2,7
Notice	2,7
Academic	2,7
ResearchArea	2,3,7



## 6.5 SELECTION CRITERIA

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

**Table:2 Selection Criteria**

<b>NOUN</b>	<b>Specific Classification</b>
User	1,3
ITian	1,3,4,5
Guest	1,3,4,5
Faculty	1,3,4,5
CR	1,3,4,5
Authentication	2
Alarm	1,2,4,5
Notice	1,2,3,4,5
News&Events	1,3,4,5
Reminder	1,2,3,4,5

EventReminder	1,2,3,4,5
Notification	2,3,4,5
Routine	1,2,3,4,5
Research	1,3
Project	1,3
Publication	1,3
Document	1,3
Image	1,3
Achievement	1,3
Academic	1,3
ResearchArea	1,3

## 6.6 ATTRIBUTE AND METHOD IDENTIFICATION

After identifying the classes, we have specified their attributes and methods.

### 6.6.1 For Mobile Application Classes:

**Table:3 Mobile Application Classes**

<b>Class Name</b>	<b>Attribute</b>	<b>Method</b>
IIiTian	<ul style="list-style-type: none"><li>- User_Type (Student/Faculty)</li><li>- Semester</li><li>- mail_address</li><li>- Password</li><li>- Username</li></ul>	<ul style="list-style-type: none"><li>- setIIiTian()</li><li>- changeInfo()</li><li>- addInstructorAccount()</li><li>- addStudentAccount()</li><li>- signIn()</li><li>- instructorValidation()</li><li>- refreshContents()</li><li>- studentValidation()</li><li>- recoverPassword()</li><li>- signOut()</li></ul>
Faculty	<ul style="list-style-type: none"><li>- Faculty_ID</li><li>- Name</li><li>- Designation</li><li>- Qualification</li><li>- Links</li><li>- About_Me</li></ul>	<ul style="list-style-type: none"><li>- setFaculty()</li><li>- getPublication()</li><li>- getResearch()</li><li>- getProjects()</li><li>- getImage()</li></ul>
Staff	<ul style="list-style-type: none"><li>- Staff_id</li><li>- Name</li><li>- Designation</li></ul>	<ul style="list-style-type: none"><li>- setStaff()</li><li>- getImage()</li></ul>
Alarm	<ul style="list-style-type: none"><li>- Alarm_ID</li><li>- set_before</li><li>- Date</li></ul>	<ul style="list-style-type: none"><li>- setAlarm()</li><li>- removeAlarm()</li><li>- snoozeAlarm()</li></ul>

	<ul style="list-style-type: none"> <li>- alarmTime</li> <li>- snoozeTime</li> </ul>	<ul style="list-style-type: none"> <li>- checkSchedule()</li> <li>- Turn off()</li> </ul>
Notice	<ul style="list-style-type: none"> <li>- Notice_ID</li> <li>- Title</li> <li>- Description</li> <li>- Date</li> <li>- Time</li> </ul>	<ul style="list-style-type: none"> <li>- viewNotice()</li> <li>- setNotice()</li> <li>- deleteNotice()</li> <li>- downloadNotice()</li> </ul>
News&Events	<ul style="list-style-type: none"> <li>- Event_ID</li> <li>- Title</li> <li>- Description</li> <li>- Date</li> <li>- Time</li> <li>- Venue</li> </ul>	<ul style="list-style-type: none"> <li>- viewNews&amp;Events()</li> <li>- setNews&amp;Events()</li> </ul>
Event Reminder	<ul style="list-style-type: none"> <li>- Event_Reminder_ID</li> <li>- Title</li> <li>- Description</li> <li>- Date</li> <li>- Time</li> <li>- Venue</li> </ul>	<ul style="list-style-type: none"> <li>- setEventReminder()</li> <li>- deleteEventReminder()</li> <li>- pushNotification()</li> </ul>
Routine	<ul style="list-style-type: none"> <li>- Date</li> <li>- Course_Code</li> <li>- Course_Instructor</li> <li>- Semester</li> <li>- Day</li> <li>- Begin_Time</li> <li>- End_Time</li> </ul>	<ul style="list-style-type: none"> <li>- viewRoutine()</li> <li>- setRoutine()</li> <li>- cancelClass()</li> <li>- addExtraClass()</li> <li>- swapClass()</li> <li>- rescheduleClass()</li> <li>- earliestClassToday()</li> </ul>
Research	<ul style="list-style-type: none"> <li>- Research_ID</li> <li>- Title</li> <li>- Description</li> </ul>	<ul style="list-style-type: none"> <li>- viewResearch()</li> <li>- setResearch()</li> </ul>
Project	<ul style="list-style-type: none"> <li>- Project_ID</li> </ul>	<ul style="list-style-type: none"> <li>- viewProject()</li> </ul>

	<ul style="list-style-type: none"> <li>- Title</li> <li>- Description</li> </ul>	<ul style="list-style-type: none"> <li>- setProject()</li> </ul>
Publication	<ul style="list-style-type: none"> <li>- Publication_ID</li> <li>- Type</li> <li>- Title</li> </ul>	<ul style="list-style-type: none"> <li>- viewPublication()</li> <li>- setPublication()</li> <li>- downloadPublication()</li> </ul>
Document	<ul style="list-style-type: none"> <li>- Document_ID</li> <li>- PDF</li> </ul>	<ul style="list-style-type: none"> <li>- viewDocument()</li> <li>- setDocument()</li> <li>- downloadDocument()</li> </ul>
Image	<ul style="list-style-type: none"> <li>- Image_ID</li> <li>- Image</li> <li>- Details</li> <li>- Height</li> <li>- Width</li> </ul>	<ul style="list-style-type: none"> <li>- viewImage()</li> <li>- setImage()</li> </ul>
Achievement	<ul style="list-style-type: none"> <li>- Achievement_ID</li> <li>- Title</li> <li>- Description</li> <li>- Date</li> <li>- Venue</li> </ul>	<ul style="list-style-type: none"> <li>- setAchievement()</li> <li>- viewAchievement()</li> </ul>
Academic	<ul style="list-style-type: none"> <li>- Academic_id</li> <li>- Info</li> <li>- program</li> <li>- admission</li> </ul>	<ul style="list-style-type: none"> <li>- setAcademic()</li> <li>- viewAcademic()</li> <li>- downloadDocument()</li> </ul>
Research Area	<ul style="list-style-type: none"> <li>- research_area_id</li> <li>- research_field</li> <li>- description</li> </ul>	<ul style="list-style-type: none"> <li>- setResearchArea()</li> <li>- viewResearchArea()</li> </ul>

Analysis: User and Reminder class has been omitted as they are redundant and their children are already present. Guest Class is not needed.

## 6.6.2 For Admin Panel Classes:

**Table:4 Admin Panel Classes**

Class Name	Attribute	Method
Admin	<ul style="list-style-type: none"><li>- Username</li><li>- password</li></ul>	<ul style="list-style-type: none"><li>- login()</li><li>- logout()</li></ul>
Content Manager	<ul style="list-style-type: none"><li>- Table_name</li><li>- data</li></ul>	<ul style="list-style-type: none"><li>- view()</li><li>- add()</li><li>- update()</li><li>- delete()</li></ul>
Semester Routine	<ul style="list-style-type: none"><li>- semester</li><li>- Course_code</li><li>- Instructure_code</li><li>- Day</li><li>- Begin_time</li><li>- End_time</li><li>- Semester_begin</li><li>- semester_end</li></ul>	<ul style="list-style-type: none"><li>- viewSchedule()</li><li>- changeSchedule()</li></ul>

## 6.6.3 For Server-side Classes:

**Table:5 Server Side Classes**

Class	Attribute	Method
Account Manager	<ul style="list-style-type: none"><li>- url</li><li>- mail_address</li><li>- Password</li><li>- connection</li><li>- lastUpdate</li></ul>	<ul style="list-style-type: none"><li>- checkInstructorAccountValidity()</li><li>- checkDuplicateAccountStudent()</li><li>- login()</li><li>- recoverPassword()</li><li>- addStudent()</li></ul>

		<ul style="list-style-type: none"> <li>- addInstructor()</li> <li>- updateInfo()</li> <li>- isConnected()</li> <li>- getLatestUpdateTime()</li> <li>- doRefresh()</li> </ul>
GetContent	<ul style="list-style-type: none"> <li>- url</li> <li>- data</li> </ul>	<ul style="list-style-type: none"> <li>- sendFaculty()</li> <li>- sendStaff()</li> <li>- sendNotice()</li> <li>- sendAchievement()</li> <li>- sendNews&amp;Events()</li> <li>- sendAcademic()</li> <li>- sendResearchArea()</li> </ul>
AddContent	<ul style="list-style-type: none"> <li>- url</li> <li>- data</li> </ul>	<ul style="list-style-type: none"> <li>- addFaculty()</li> <li>- addStaff()</li> <li>- addNotice()</li> <li>- addAchievement()</li> <li>- addNews&amp;Events()</li> <li>- addAcademic()</li> <li>- addResearchArea()</li> </ul>
UpdateContent	<ul style="list-style-type: none"> <li>- url</li> <li>- data</li> </ul>	<ul style="list-style-type: none"> <li>- updateFaculty()</li> <li>- updateStaff()</li> <li>- updateNotice()</li> <li>- updateAchievement()</li> <li>- updateNews&amp;Events()</li> <li>- updateAcademic()</li> <li>- updateResearchArea()</li> </ul>
DeleteContent	<ul style="list-style-type: none"> <li>- url</li> <li>- data</li> </ul>	<ul style="list-style-type: none"> <li>- deleteFaculty()</li> <li>- deleteStaff()</li> <li>- deleteNotice()</li> <li>- deleteAchievement()</li> <li>- deleteNews&amp;Events()</li> <li>- deleteAcademic()</li> <li>- deleteResearchArea()</li> </ul>

Routine Manager	<ul style="list-style-type: none"> <li>- url</li> <li>- date</li> <li>- Mail_id</li> <li>- semester</li> <li>- Course_code</li> <li>- Instructore_code</li> <li>- Day</li> <li>- Begin_time</li> <li>- End_time</li> <li>- semester_begin</li> <li>- semester_end</li> </ul>	<ul style="list-style-type: none"> <li>- checkCR()</li> <li>- sendInstructorWeekRoutine()</li> <li>- sendStudentWeekRoutine()</li> <li>- temporaryChange()</li> <li>- permanentChange()</li> <li>- sendSemesterSchedule()</li> </ul>
-----------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 6.7 Analyzing Classes

User and Reminder class has been omitted as they are redundant and their children are already present. Guest Class is not needed.

## 6.8 CRC Card

Class Name	Responsibility	Collaborator
IITian	<ul style="list-style-type: none"> <li>● Identify a User</li> <li>● Stores information about user</li> <li>● Authenticate users by checking IIT mail</li> <li>● Sign Up</li> </ul>	<ul style="list-style-type: none"> <li>● Account Manager</li> <li>● Routine Manager</li> <li>● Alarm</li> <li>● Event Reminder</li> <li>● Routine</li> </ul>



	<ul style="list-style-type: none"> <li>• Sign In</li> <li>• Sign Out</li> </ul>	
Faculty	<ul style="list-style-type: none"> <li>• Stores all necessary information about a Faculty</li> </ul>	<ul style="list-style-type: none"> <li>• Content</li> <li>• Publication</li> <li>• Research</li> <li>• Project</li> <li>• Image</li> </ul>
Staff	<ul style="list-style-type: none"> <li>• Stores all necessary information about a Staff</li> </ul>	<ul style="list-style-type: none"> <li>• Content</li> <li>• Image</li> </ul>
Alarm	<ul style="list-style-type: none"> <li>• Set alarm</li> <li>• Alert user before first morning class</li> <li>• snooze alarm</li> </ul>	<ul style="list-style-type: none"> <li>• Routine</li> <li>• IITian</li> </ul>
Notice	<ul style="list-style-type: none"> <li>• Hold information of notices</li> <li>• Show notices</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> <li>• Notification</li> </ul>
News&Events	<ul style="list-style-type: none"> <li>• Hold information about News and Events</li> <li>• Show News and Events</li> </ul>	<ul style="list-style-type: none"> <li>• Image</li> </ul>
Event Reminder	<ul style="list-style-type: none"> <li>• Reminds about News and Events</li> </ul>	<ul style="list-style-type: none"> <li>• News&amp;Events</li> <li>• Account Manager</li> <li>• Notification</li> </ul>
Notification	<ul style="list-style-type: none"> <li>• Send Notification</li> </ul>	<ul style="list-style-type: none"> <li>• Event Reminder</li> <li>• Notice</li> </ul>
Routine	<ul style="list-style-type: none"> <li>• Hold information of routine</li> <li>• Stores the earliest class time</li> <li>• Show Routine</li> <li>• Modify Routine</li> </ul>	<ul style="list-style-type: none"> <li>• Alarm</li> <li>• IITian</li> <li>• Routine Manager</li> </ul>

Research	<ul style="list-style-type: none"> <li>● Hold information of research</li> <li>● Show Research</li> </ul>	<ul style="list-style-type: none"> <li>● Faculty</li> </ul>
Project	<ul style="list-style-type: none"> <li>● Hold information of research</li> <li>● Show Project</li> </ul>	<ul style="list-style-type: none"> <li>● Faculty</li> </ul>
Publication	<ul style="list-style-type: none"> <li>● Hold information of publication</li> <li>● Show Publication</li> </ul>	<ul style="list-style-type: none"> <li>● Document</li> <li>● Faculty</li> </ul>
Document	<ul style="list-style-type: none"> <li>● Hold information of Document</li> <li>● Show Document</li> </ul>	<ul style="list-style-type: none"> <li>● Notice</li> <li>● Publication</li> </ul>
Image	<ul style="list-style-type: none"> <li>● Hold information of Image</li> <li>● Show Image</li> </ul>	<ul style="list-style-type: none"> <li>● Faculty</li> <li>● Staff</li> <li>● Achievement</li> <li>● News&amp;Events</li> <li>● Content</li> </ul>
Achievement	<ul style="list-style-type: none"> <li>● Hold information about Achievement</li> <li>● Show Achievement</li> </ul>	<ul style="list-style-type: none"> <li>● Image</li> <li>● Content</li> </ul>
Academic	<ul style="list-style-type: none"> <li>● Hold data</li> <li>● Show Information</li> <li>● Facilitate downloading Document</li> </ul>	<ul style="list-style-type: none"> <li>● Document</li> <li>● Content</li> </ul>
Research Area	<ul style="list-style-type: none"> <li>● Hold data</li> <li>● Show Information</li> </ul>	<ul style="list-style-type: none"> <li>● Content</li> </ul>
Admin	<ul style="list-style-type: none"> <li>● Hold Admin credentials</li> </ul>	
Content Manager	<ul style="list-style-type: none"> <li>● Add info</li> <li>● Update info</li> <li>● Delete info</li> </ul>	<ul style="list-style-type: none"> <li>● Content</li> </ul>

Semester Routine	<ul style="list-style-type: none"> <li>• Set schedule for a semester</li> <li>• Make permanent schedule change</li> </ul>	<ul style="list-style-type: none"> <li>• Routine Manager</li> </ul>
Account Manager	<ul style="list-style-type: none"> <li>• Add new account</li> <li>• Update account info</li> <li>• Request for fetch updated info</li> </ul>	<ul style="list-style-type: none"> <li>• IITian</li> <li>• Account</li> <li>• Event Reminder</li> </ul>
Routine Manager	<ul style="list-style-type: none"> <li>• Make temporary change</li> <li>• Make permanent change</li> <li>• Send routine for the current week</li> </ul>	<ul style="list-style-type: none"> <li>• Routine</li> <li>• Semester Routine</li> </ul>
GetContent	<ul style="list-style-type: none"> <li>• Send Faculty</li> <li>• Send Staff</li> <li>• Send Notice</li> <li>• Send Achievement</li> <li>• Send News&amp;Events</li> <li>• Send Academic</li> <li>• Send ResearchArea</li> </ul>	
AddContent	<ul style="list-style-type: none"> <li>• Add Faculty</li> <li>• Add Staff</li> <li>• Add Notice</li> <li>• Add Achievement</li> <li>• Add News&amp;Events</li> <li>• Add Academic</li> <li>• Add ResearchArea</li> </ul>	
UpdateContent	<ul style="list-style-type: none"> <li>• Update Faculty</li> <li>• Update Staff</li> <li>• Update Notice</li> <li>• Update Achievements</li> <li>• Update News&amp;Events</li> </ul>	

	<ul style="list-style-type: none"> <li>• Update Academic</li> <li>• Update Research Area</li> </ul>	
DeleteContent	<ul style="list-style-type: none"> <li>• Delete Faculty</li> <li>• Delete Staff</li> <li>• Delete Notice</li> <li>• Delete Achievement</li> <li>• Delete News&amp;Events</li> <li>• Delete Academic</li> <li>• Delete Research Area</li> </ul>	

## 6.9 Class Card

After identifying our final classes we have generated the following class cards.

### *For Mobile Application Classes:*

Table:6 Class Card for II Tian

II Tian	
Attribute	Method
<ul style="list-style-type: none"> <li>• Mail Address</li> <li>• User_Type</li> </ul>	<ul style="list-style-type: none"> <li>• setII Tian()</li> </ul>

<ul style="list-style-type: none"> <li>• Semester</li> </ul>	
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Identify a User</li> <li>• Stores information about user</li> </ul>	

Table:7 Class Card for Faculty

<b>Faculty</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Faculty_ID</li> <li>• Name</li> <li>• Designation</li> <li>• Links</li> <li>• About_Me</li> </ul>	<ul style="list-style-type: none"> <li>• setFaculty()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ol style="list-style-type: none"> <li>1. Identify Faculty</li> <li>2. Stores all necessary information about a Faculty</li> </ol>	

Table:8 Class Card for Staff

<b>Staff</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Staff_ID</li> <li>• Name</li> </ul>	<ul style="list-style-type: none"> <li>• setStaff()</li> <li>• getImage()</li> </ul>

<ul style="list-style-type: none"> <li>• Designation</li> </ul>	
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Identify Staff</li> <li>• Stores all necessary information about a Staff</li> </ul>	<ul style="list-style-type: none"> <li>• Image</li> </ul>

**Table:9 Class Card for Authentication**

<b>Authentication</b>	
<b>Attribute</b>	<b>Method</b>
	<ul style="list-style-type: none"> <li>- authenticateFaculty()</li> <li>- authenticateCR()</li> <li>- signUp()</li> <li>- signIn()</li> <li>- signOut()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Authenticate users by checking IIT mail</li> <li>• Sign Up</li> <li>• Sign In</li> <li>• Sign Out</li> </ul>	

**Table:10 Class Card for Alarm**

<b>Alarm</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Alarm_ID</li> <li>• Date</li> <li>• Time</li> </ul>	<ul style="list-style-type: none"> <li>• setAlarm()</li> <li>• deleteAlarm()</li> <li>• snoozeAlarm()</li> <li>• checkSchedule()</li> <li>• turnOff()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Alarm goes off</li> <li>• Set alarm before first morning class</li> <li>• Delete alarm</li> </ul>	<ul style="list-style-type: none"> <li>• Routine</li> <li>• Notification</li> </ul>

**Table:11 Class Card for Notice**

<b>Notice</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Notice_ID</li> <li>• Title</li> <li>• Description</li> <li>• Date</li> <li>• Time</li> </ul>	<ul style="list-style-type: none"> <li>• viewNotice()</li> <li>• setNotice()</li> <li>• downloadNotice()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of notices</li> <li>• Show notices</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> <li>• Notification</li> </ul>

**Table:12 Class Card for News&Events**

<b>News&amp;Events</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• Event_ID</li><li>• Title</li><li>• Description</li><li>• Date</li><li>• Time</li><li>• Venue</li></ul>	<ul style="list-style-type: none"><li>• viewNews&amp;Events()</li><li>• setNews&amp;Events()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"><li>• Hold information about News and Events</li><li>• Show News and Events</li></ul>	<ul style="list-style-type: none"><li>• Document</li><li>• Image</li></ul>

**Table:13 Class Card for Notification**

<b>Notification</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• Description</li></ul>	<ul style="list-style-type: none"><li>• sendNotification()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"><li>• Send Notification</li></ul>	



**Table:14 Class Card for EventReminder**

<b>EventReminder</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• Event_Reminder_ID</li><li>• Title</li><li>• Description</li><li>• Date</li><li>• Time</li><li>• Venue</li></ul>	<ul style="list-style-type: none"><li>• setEventReminder()</li><li>• deleteEventReminder()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"><li>• Reminds about News and Events</li></ul>	<ul style="list-style-type: none"><li>• News&amp;Events</li><li>• Notification</li></ul>

**Table:15 Class Card for Routine**

<b>Routine</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• Date</li><li>• Course_Code</li><li>• Faculty(Course_Instructor)</li><li>• Semester</li><li>• Day</li><li>• Begin_Time</li></ul>	<ul style="list-style-type: none"><li>• viewRoutine()</li><li>• setRoutine()</li><li>• cancelClass()</li><li>• addExtraClass()</li><li>• swapClass()</li></ul>

<ul style="list-style-type: none"> <li>• End_Time</li> </ul>	<ul style="list-style-type: none"> <li>• rescheduleClass()</li> <li>• earliestClassToday()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of routine</li> <li>• Show Routine</li> <li>• Modify Routine</li> </ul>	<ul style="list-style-type: none"> <li>• Notification</li> </ul>

**Table:16 Class Card for Research**

<b>Research</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Research_ID</li> <li>• Title</li> <li>• Description</li> </ul>	<ul style="list-style-type: none"> <li>• viewResearch()</li> <li>• setResearch()</li> <li>• downloadResearch()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of research</li> <li>• Show Research</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> </ul>

**Table:17 Class Card for Project**

<b>Project</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Project_ID</li> <li>• Title</li> </ul>	<ul style="list-style-type: none"> <li>• viewProject()</li> <li>• setProject()</li> </ul>

<ul style="list-style-type: none"> <li>• Description</li> </ul>	
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of research</li> <li>• Show Project</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> </ul>

**Table:18 Class Card for Assignment**

<b>Publication</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Publication_ID</li> <li>• Type</li> <li>• Title</li> </ul>	<ul style="list-style-type: none"> <li>• viewPublication()</li> <li>• setPublication()</li> <li>• downloadPublication()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of publication</li> <li>• Show Publication</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> </ul>

**Table:19 Class Card for Document**

<b>Document</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Document_ID</li> <li>• PDF</li> </ul>	<ul style="list-style-type: none"> <li>• viewDocument()</li> <li>• setDocument()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of Document</li> </ul>	

<ul style="list-style-type: none"> <li>• Show Document</li> </ul>	
-------------------------------------------------------------------	--

**Table:20 Class Card for Project**

<b>Image</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Image_ID</li> <li>• Image</li> <li>• Details</li> <li>• Height</li> <li>• Width</li> </ul>	<ul style="list-style-type: none"> <li>• viewImage()</li> <li>• setImage()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information of Image</li> <li>• Show Image</li> </ul>	

**Table:21 Class Card for Achievement**

<b>Achievement</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Achievement_ID</li> <li>• Title</li> <li>• Description</li> <li>• Date</li> <li>• Venue</li> </ul>	<ul style="list-style-type: none"> <li>• setAchievement()</li> <li>• viewAchievement()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information about Achievement</li> <li>• Show Achievement</li> </ul>	<ul style="list-style-type: none"> <li>• Image</li> </ul>

**Table:22 Class Card for Academic**

<b>Academic</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• Academic_id</li> <li>• Info</li> <li>• program</li> <li>• admission</li> </ul>	<ul style="list-style-type: none"> <li>• setAcademic()</li> <li>• viewAcademic()</li> <li>• downloadDocument()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information about Academic</li> <li>• Show Academic</li> <li>• Download Academic Document</li> </ul>	<ul style="list-style-type: none"> <li>• Document</li> </ul>

**Table:23 Class Card for Research Area**

<b>ResearchArea</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• research_area_id</li> <li>• research_field</li> <li>• description</li> </ul>	<ul style="list-style-type: none"> <li>• setResearchArea()</li> <li>• viewResearchArea()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Hold information about Research Area</li> <li>• Show Research Area</li> </ul>	

## For Admin Panel Classes:

Table:24 Class Card for Admin

Admin	
Attribute	Method
<ul style="list-style-type: none"><li>• Username</li><li>• password</li></ul>	<ul style="list-style-type: none"><li>• login()</li><li>• logout()</li></ul>
Responsibilities	Collaborator
<ul style="list-style-type: none"><li>• Hold information about Admin</li><li>• Log In</li><li>• Log Out</li></ul>	

**Table:25 Class Card for Content Manager**

<b>Content Manager</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• Table_name</li><li>• data</li></ul>	<ul style="list-style-type: none"><li>• view()</li><li>• add()</li><li>• update()</li><li>• delete()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"><li>• View</li><li>• Add</li><li>• Update</li><li>• Delete</li></ul>	

**Table:26 Class Card for Semester Routine**

<b>Semester Routine</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"><li>• semester</li><li>• Course_code</li><li>• Instructore_code</li><li>• Day</li><li>• Begin_time</li><li>• End_time</li><li>• Semester_begin</li><li>• semester_end</li></ul>	<ul style="list-style-type: none"><li>• viewSchedule()</li><li>• changeSchedule()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>

<ul style="list-style-type: none"> <li>• View</li> <li>• Add</li> <li>• Update</li> <li>• Delete</li> </ul>	
-------------------------------------------------------------------------------------------------------------	--

## For Server-side Classes:

**Table:27 Class Card for AccountManager**

<b>AccountManager</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• url</li> <li>• mail_address</li> <li>• Password</li> <li>• connection</li> <li>• lastUpdate</li> </ul>	<ul style="list-style-type: none"> <li>• getRequest()</li> <li>• checkInstructorAccountValidity()</li> <li>• checkDuplicateAccountStudent()</li> <li>• login()</li> <li>• recoverPassword()</li> <li>• addStudent()</li> <li>• addInstructor()</li> <li>• updateInfo()</li> <li>• isConnected()</li> <li>• getLatestUpdateTime()</li> <li>• doRefresh()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Check Validity of Instructor Account</li> <li>• Check Duplicate Account Student</li> <li>• Log In</li> <li>• Recover Password</li> <li>• Add Student</li> <li>• Add Instructor</li> </ul>	



<ul style="list-style-type: none"> <li>• Update Information</li> <li>• Check if connected</li> <li>• Get latest update time</li> <li>• Do refresh</li> </ul>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------	--

**Table:28 Class Card for RoutineManager**

<b>RoutineManager</b>	
<b>Attribute</b>	<b>Method</b>
<ul style="list-style-type: none"> <li>• url</li> <li>• date</li> <li>• Mail_id</li> <li>• semester</li> <li>• Course_code</li> <li>• Instructore_code</li> <li>• Day</li> <li>• Begin_time</li> <li>• End_time</li> <li>• semester_begin</li> <li>• semester_end</li> </ul>	<ul style="list-style-type: none"> <li>• getRequest()</li> <li>• checkCR()</li> <li>• sendInstructorWeekRoutine()</li> <li>• sendStudentWeekRoutine()</li> <li>• temporaryChange()</li> <li>• permanentChange()</li> <li>• sendSemesterSchedule()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Get Request</li> <li>• Check CR</li> <li>• Send Instructor Week Routine</li> <li>• Send Student Week Routine</li> <li>• Temporary Change</li> <li>• Permanent Change</li> <li>• Send Semester Schedule</li> </ul>	

**Table:29 Class Card for GetContent**

<b>GetContent</b>	
<b>Attribute</b>	<b>Method</b>
	<ul style="list-style-type: none"><li>• Send Faculty()</li><li>• Send Staff()</li><li>• Send Notice()</li><li>• Send Achievement()</li><li>• Send News&amp;Events()</li><li>• Send Academic()</li><li>• Send ResearchArea()</li></ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"><li>• Send Faculty</li><li>• Send Staff</li><li>• Send Notice</li><li>• Send Achievement</li><li>• Send News and Events</li><li>• Send Academic</li><li>• Send Research Area</li></ul>	

**Table:30 Class Card for AddContent**

<b>AddContent</b>	
<b>Attribute</b>	<b>Method</b>
	<ul style="list-style-type: none"><li>• Add Faculty ()</li><li>• Add Staff ()</li><li>• Add Notice ()</li><li>• Add Achievement ()</li></ul>

	<ul style="list-style-type: none"> <li>• Add News&amp;Events ()</li> <li>• Add Academic ()</li> <li>• Add ResearchArea()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Add Faculty</li> <li>• Add Staff</li> <li>• Add Notice</li> <li>• Add Achievement</li> <li>• Add News&amp;Events</li> <li>• Add Academic</li> <li>• Add ResearchArea</li> </ul>	

**Table:31 Class Card for UpdateContent**

<b>UpdateContent</b>	
<b>Attribute</b>	<b>Method</b>
	<ul style="list-style-type: none"> <li>• Update Faculty ()</li> <li>• Update Staff ()</li> <li>• Update Notice ()</li> <li>• Update Achievements ()</li> <li>• Update News&amp;Events ()</li> <li>• Update Academic ()</li> <li>• Update Research Area ()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Update Faculty</li> <li>• Update Staff</li> <li>• Update Notice</li> <li>• Update Achievements</li> <li>• Update News&amp;Events</li> <li>• Update Academic</li> </ul>	

<ul style="list-style-type: none"> <li>• Update Research Area</li> </ul>	
--------------------------------------------------------------------------	--

**Table:32 Class Card for DeleteContent**

<b>DeleteContent</b>	
<b>Attribute</b>	<b>Method</b>
	<ul style="list-style-type: none"> <li>• Delete Faculty()</li> <li>• Delete Staff ()</li> <li>• Delete Notice ()</li> <li>• Delete Achievement ()</li> <li>• Delete News&amp;Events ()</li> <li>• Delete Academic ()</li> <li>• Delete Research Area ()</li> </ul>
<b>Responsibilities</b>	<b>Collaborator</b>
<ul style="list-style-type: none"> <li>• Delete Faculty</li> <li>• Delete Staff</li> <li>• Delete Notice</li> <li>• Delete Achievement</li> <li>• Delete News&amp;Events</li> <li>• Delete Academic</li> <li>• Delete Research Area</li> </ul>	

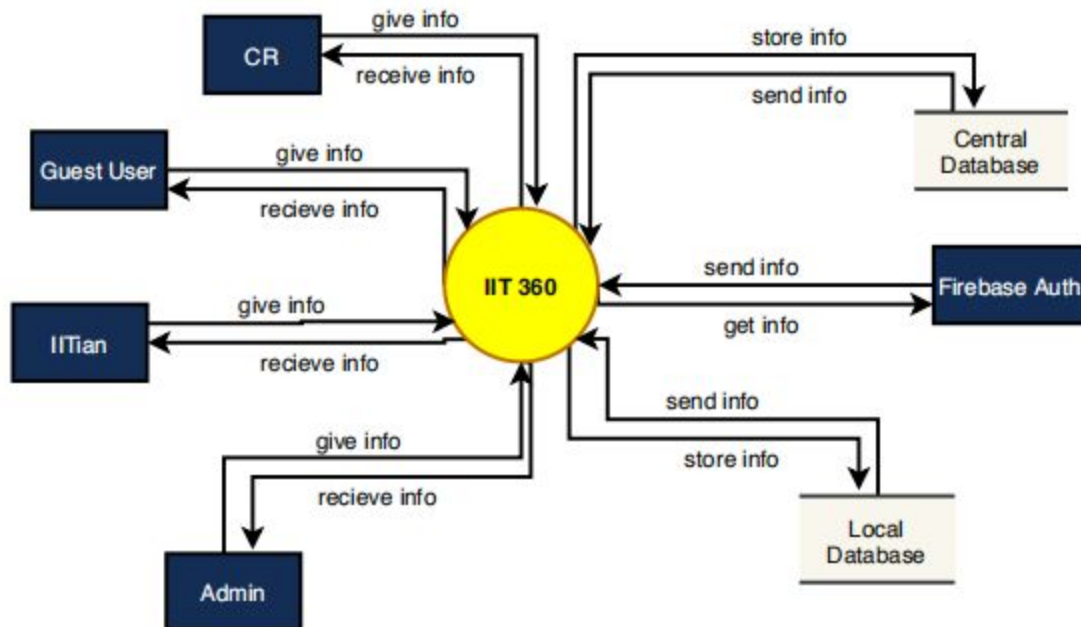
## 7. FLOW ORIENTED MODEL OF IIT 360

### DATA FLOW DIAGRAMS:

#### DFD level: 0

DFD ID: 01

Name: Bird's Eye View

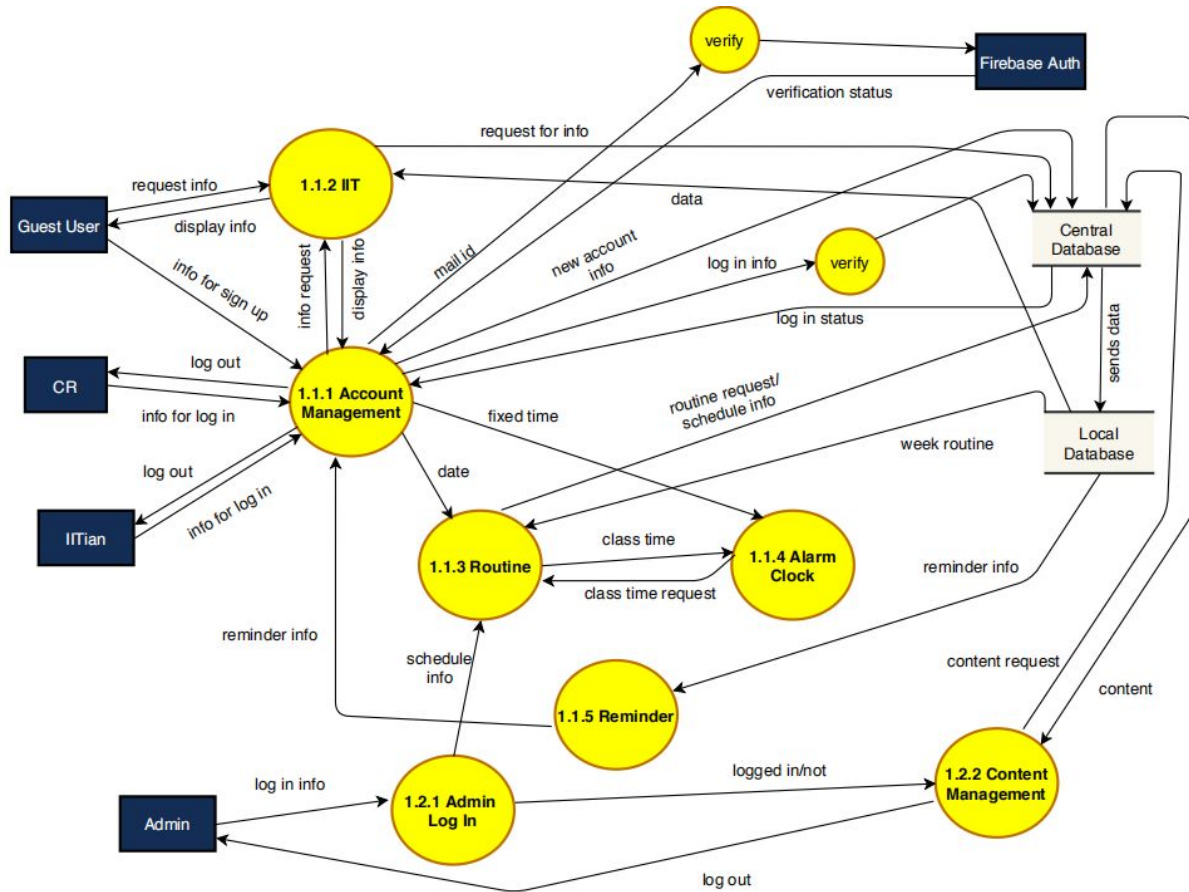


**Description:** This is the level 0 DFD of the system. This will be further divided into smaller sub-systems in DFD level 1.

## DFD level: 1

DFD ID: 02

Name: IIT 360

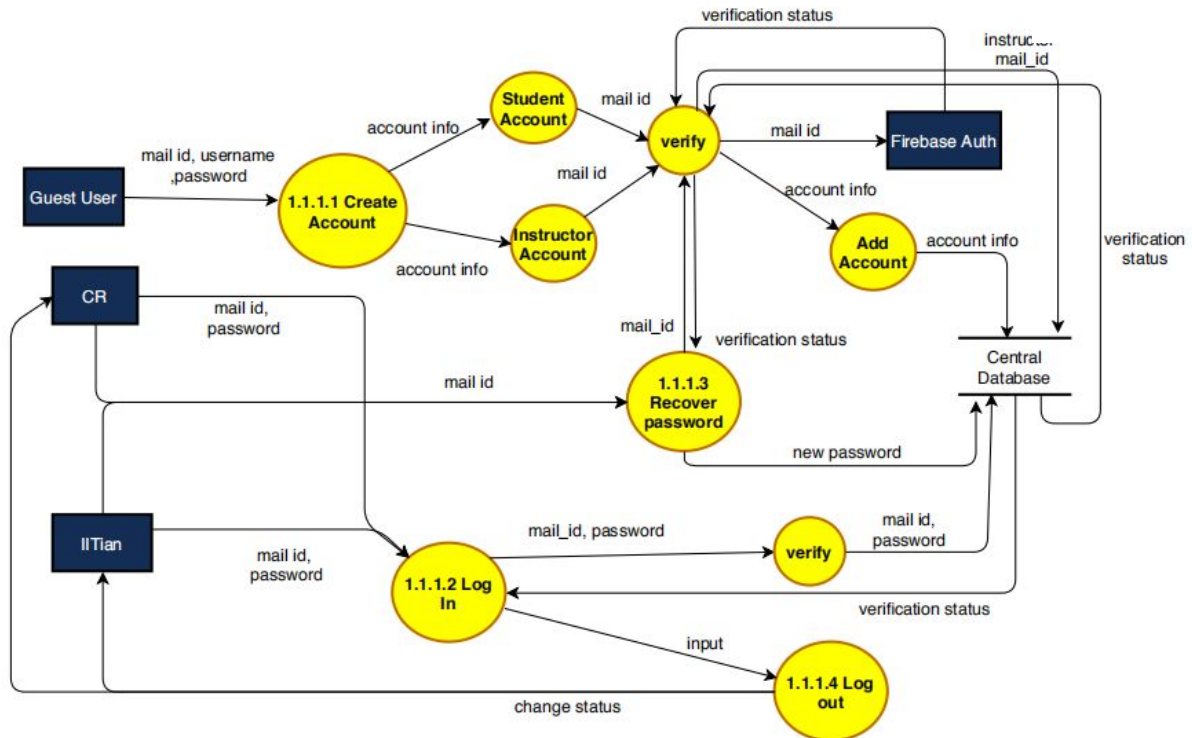


**Description:** This is the level 1 DFD of IIT 360. We can see here the major subsystems of it & how data flows in between them. To make the data flow more reasonable, the major subsystems will be divided into smaller processes from level 02.

## DFD level: 2

DFD ID: 03

Name: Account Management

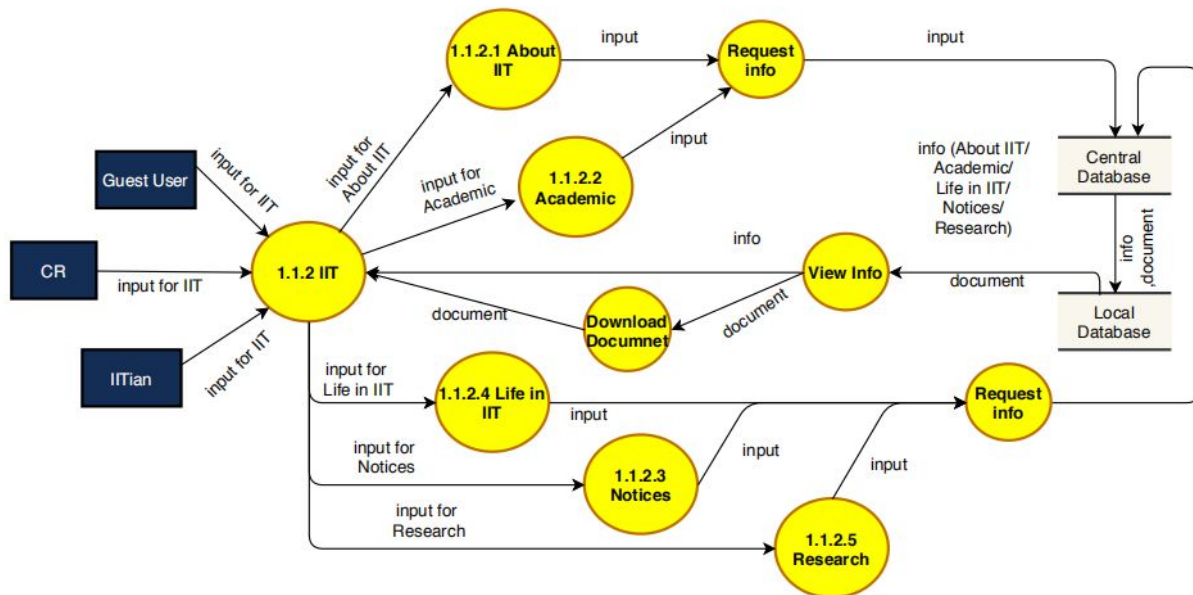


**Description:** This is the level 2 DFD of Account Management sub system. We can see here what and how data flows through sub-systems and processes in this level of Account Management sub-system.

## DFD level: 2

DFD ID: 04

Name: IIT



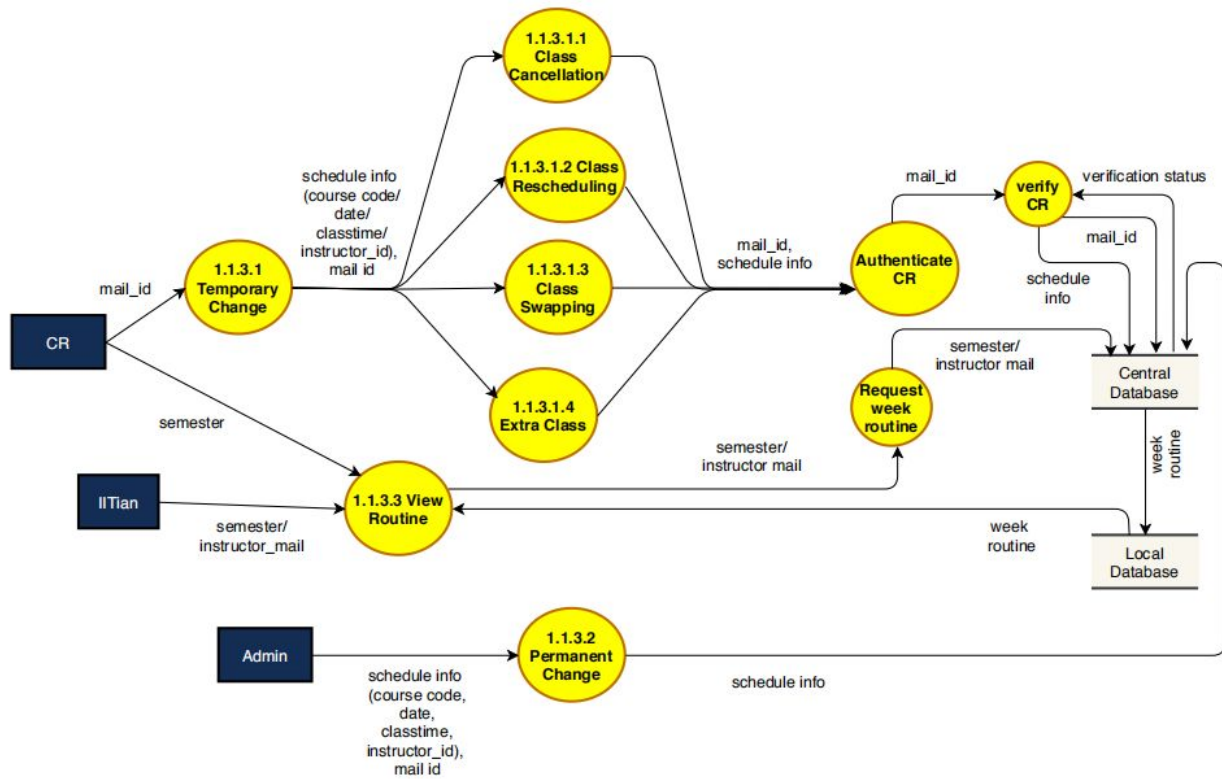
**Description:** This is the level 2 DFD of IIT sub system. We can see here what and how data flows through sub-systems and processes in this level of IIT sub system,



## DFD level: 2

DFD ID: 05

Name: Routine

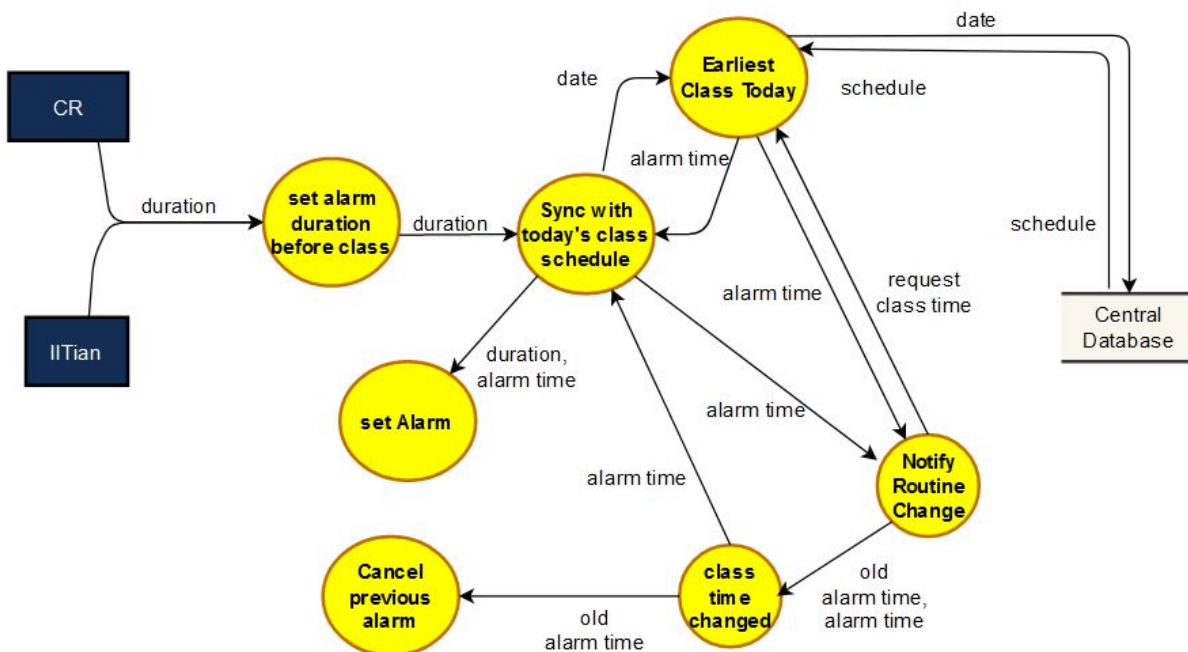


**Description:** This is the level 2 DFD of the Routine sub system. We can see here what and how data flows through sub-systems and processes in this level of Routine sub-systems.

## DFD level: 2

DFD ID: 06

Name: Alarm Clock

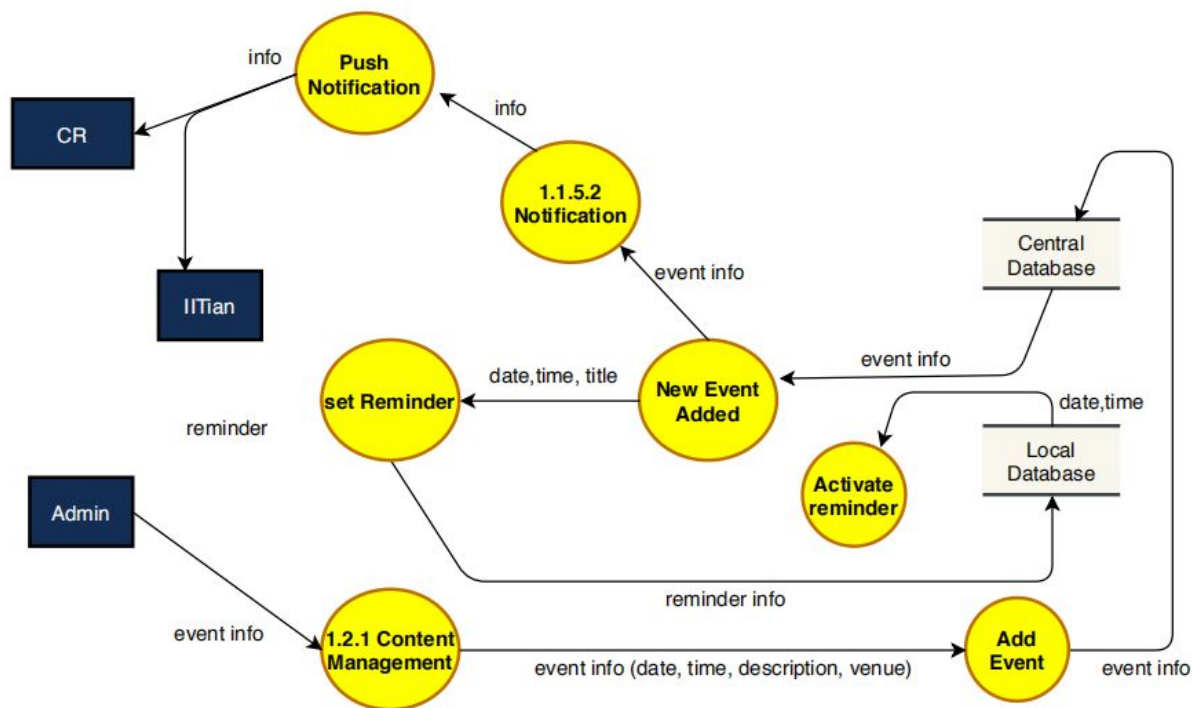


**Description:** This is the level 2 DFD of Alarm Clock sub system. We can see here what and how data flows through sub-systems and processes in this level of Alarm Clock sub-system.

## DFD level: 2

DFD ID: 07

Name: Reminder

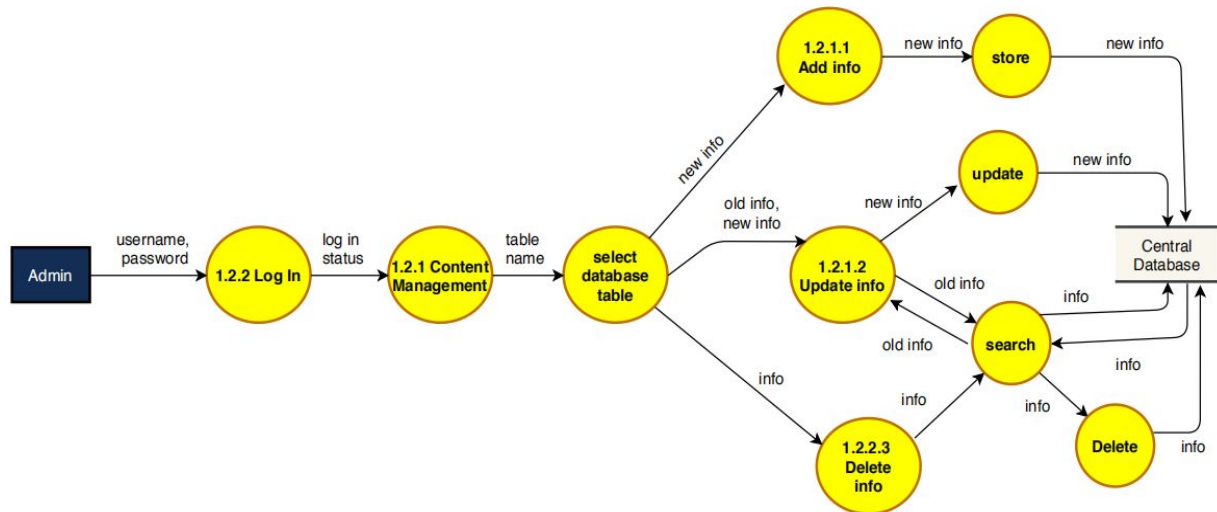


**Description:** This is the level 2 DFD of Alarm Clock sub system. We can see here what and how data flows through sub-systems and processes in this level of Reminder sub-system.

## DFD level: 2

DFD ID: 09

Name: Admin Panel



**Description:** This is the level 2 DFD of the Admin Panel sub system. We can see here what and how data flows through sub-systems and processes in this level of Admin Panel sub-system.

## 8. BEHAVIORAL MODELING OF IIT 360

### 8.1 Event Identification Table

No.	Event	State Name	Initiator	Collaborator
1	Sign Up	Providing info	IITian	Account Manager
2	Account Validation	Validating account info	Account Manager	IITian
3	Mail Verification	Verifying mail address	IITian	
4	Duplicate Account	Existing account found	Account Manager	IITian
5	Create Account	Creating Account	Account Manager	IITian
6	Authentication	Authenticating log in info	Account Manager	IITian
7	Invalid Account	No account found	Account Manager	IITian
8	Login Successful	Account found	Account Manager	IIT

<b>9</b>	Log In	Providing info	IITian	Account Manager
<b>10</b>	Recover Password	Requesting account validation	IITian	Account Manager
<b>11</b>	Set New Password	Setting new password	IITian	Account Manager
<b>12</b>	Confirm New Password	Changing password	Account Manager	
<b>13</b>	Update Account Info	Providing new info	IITian	Account Manager
<b>14</b>	Refresh App Contents	Updating app content	IITian	Account Manager
<b>15</b>	View Faculty	Requesting faculty info	Faculty	Image
<b>16</b>	View Publications	Requesting faculty publications	Faculty	Publication Document Content
<b>17</b>	Fetch Publications	Fetching content from database	Publication	Document GetContent
<b>18</b>	View Projects	Requesting faculty projects	Faculty	Project GetContent

<b>19</b>	Fetch Projects	Fetching content from database	Project	GetContent
<b>20</b>	Download Publication	Downloading Publication	Faculty	Document Publication
<b>21</b>	View Researches	Requesting faculty researches	Faculty	Research
<b>22</b>	Fetch Researches	Fetching content from database	Research	GetContent
<b>23</b>	View Officers and Staffs	Requesting staffs info	Staff	Image GetContent
<b>24</b>	View Notices	Fetching notices from database	Notice	GetContent Document
<b>25</b>	Download Notice	Downloading Notice	Notice	Document
<b>26</b>	View Academic info	Fetching academic info	Academic	GetContent Document
<b>27</b>	Download Academic Document	Downloading Document	Academic	Document

<b>28</b>	View IIT Research Area	Requesting IIT research area	Research Area	GetContent
<b>29</b>	View Routine	Fetching week routing	Routine	Routine Manager IITian
<b>30</b>	Set Morning Class	Setting earliest class time	Routine	
<b>31</b>	View News and Events	Requesting news and events	News&Events	GetContent Image
<b>32</b>	View Achievements	Requesting achievements	Achievement	GetContent Image
<b>33</b>	CR Authentication	Authenticating CR	Routine	Routine Manager
<b>34</b>	Temporary Routine Change	Requesting routine change	Routine	Routine Manager
<b>35</b>	Permanent Routine Change	Changing semester routine	Semester Routine	Routine Manager
<b>36</b>	Set Alarm	Checking earliest class time	Alarm	Routine
<b>37</b>	Alarm Goes Off	Alarm time reached	Alarm	



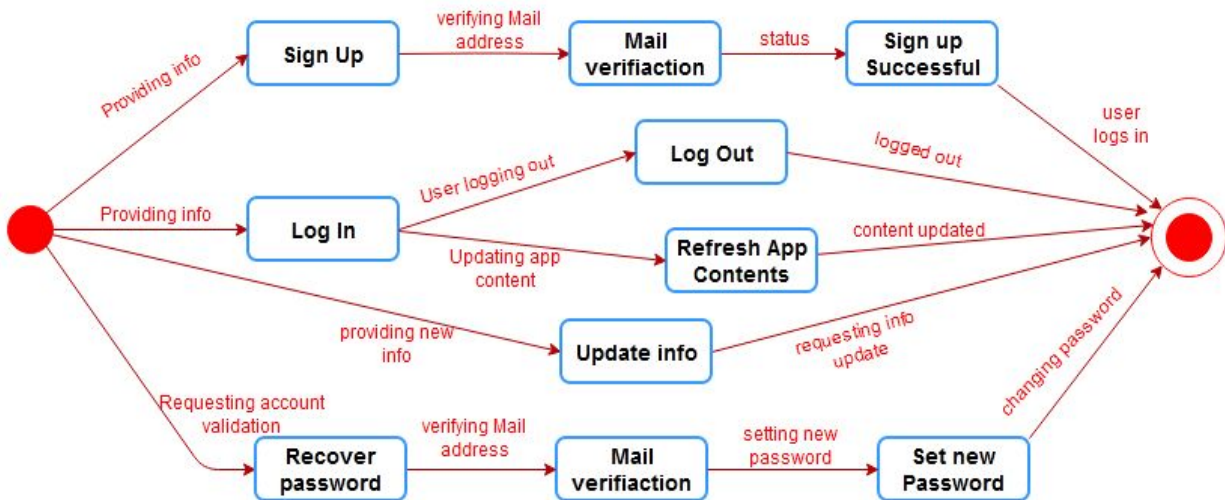
<b>38</b>	Reset Alarm	Resetting alarm time	Alarm	Routine
<b>39</b>	Turn off Alarm	Keep alarm turned off	Alarm	ITian
<b>40</b>	Snooze Alarm	Alarm goes off	Alarm	ITian
<b>41</b>	Set Event Reminder	Fetching event info	Content Manger	Event Reminder
<b>42</b>	Push Notification	showing reminder info	Reminder	
<b>43</b>	Cancel Event Reminder	Removing reminder	Event Reminder	ITian
<b>44</b>	Admin Log in	Logging in	Admin	
<b>45</b>	Add Content	Providing new content	Content Manager	AddContent
<b>46</b>	Search Content	Searching Content	Content Manager	GetContent
<b>47</b>	Update Content	Providing Updated Content	Content Manager	UpdateContent
<b>48</b>	Delete Content	Deleting Content	Content Manager	DeleteContent

<b>49</b>	View Semester Routine	Request semester routine	Semester Routine	Routine Manager
<b>50</b>	Temporary Schedule	Updating temporary schedule	Routine manager	Routine
<b>51</b>	Send Week Routine	Sending week routine	Routine manager	Routine
<b>52</b>	Send Semester Routine	Sending semester routine	Routine manager	Semester Routine
<b>53</b>	Permanent Schedule	Updating semester schedule	Routine manager	Semester Routine
<b>54</b>	User Log Out	User logging out	IITian	
<b>55</b>	Admin Log Out	Admin logging out	Admin	

## 8.2 STATE TRANSITION DIAGRAMS

**Class Name:** IITian

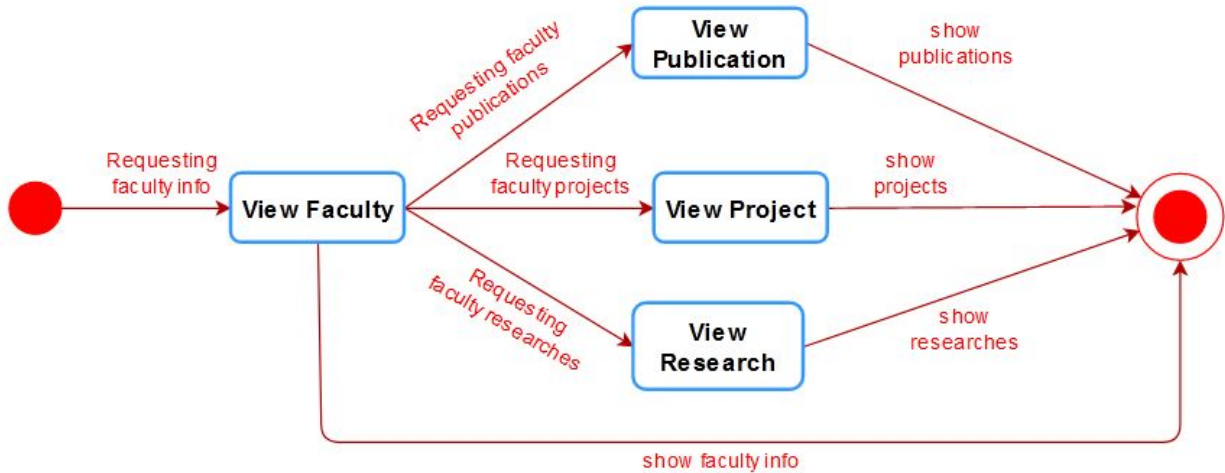
**ID:** C01



**Description:** The state transition diagram of class IITian , ID: C01, shows the number of different states that this class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to managing an IITian account. Transition is mostly caused by the user's active involvement.

**Class Name:** Faculty

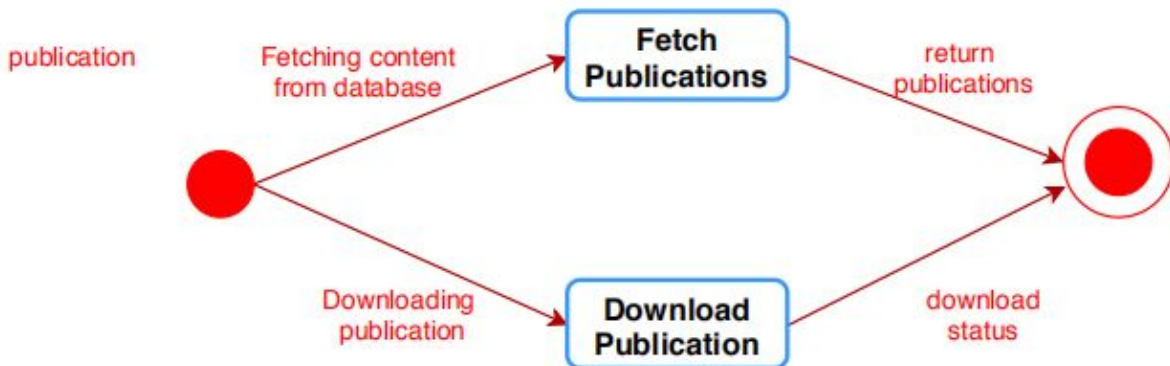
**ID:** C02



Description: Faculty info can be viewed by requesting for the info. Further, the publication, project and research of the faculty can be viewed from there.

**Class Name:** Publication

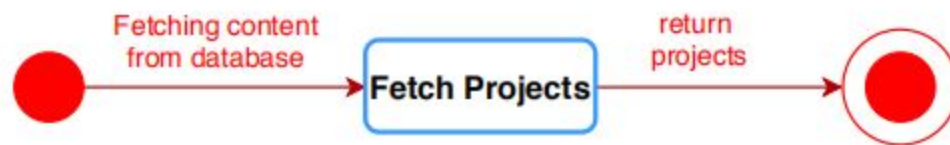
**ID:** C03



**Description:** Publications can be fetched or downloaded depending on the user request

**Class Name:** Project

**ID:** C04



**Description:** Projects can be fetched from the database on user's request

**Class Name:** Research

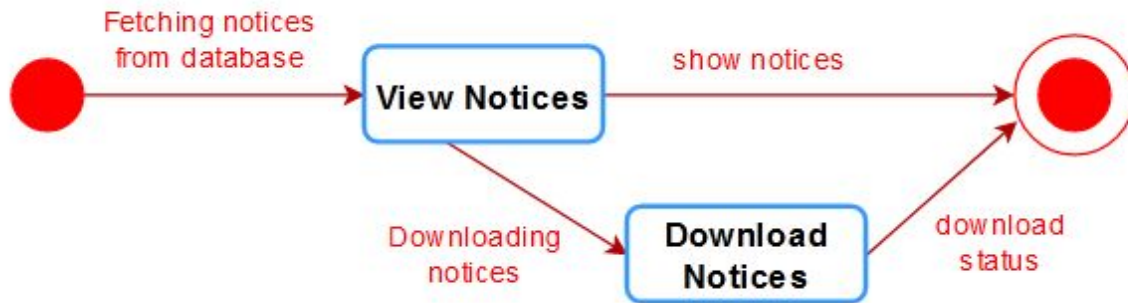
**ID:** C05



**Description:** Researches can be fetched from the database on user's request.

**Class Name:** Notice

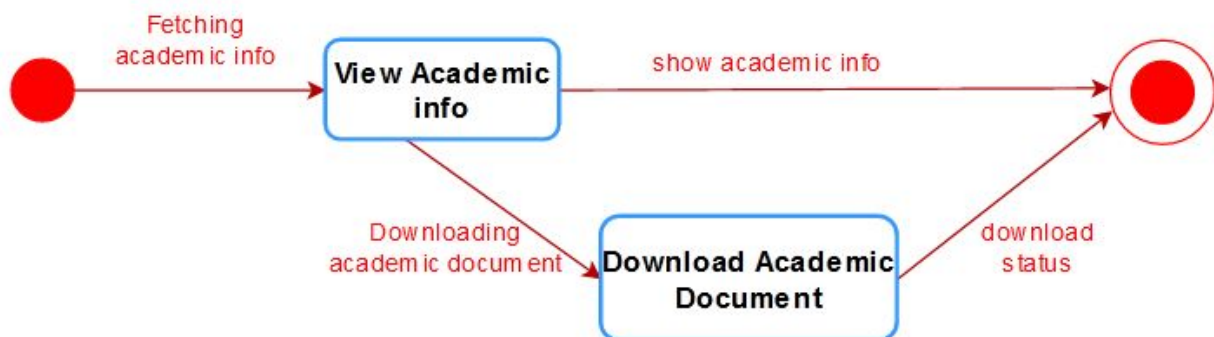
**ID:** C06



**Description:** Notices can be viewed on user's request and can be downloaded too.

**Class Name:** Academic

**ID:** C07



**Description:** Academic info can be fetched and academic documents can be downloaded further.

**Class Name:** Achievement

**ID:** C08



**Description:** Achievements can be viewed on user request.

**Class Name:** News&Events

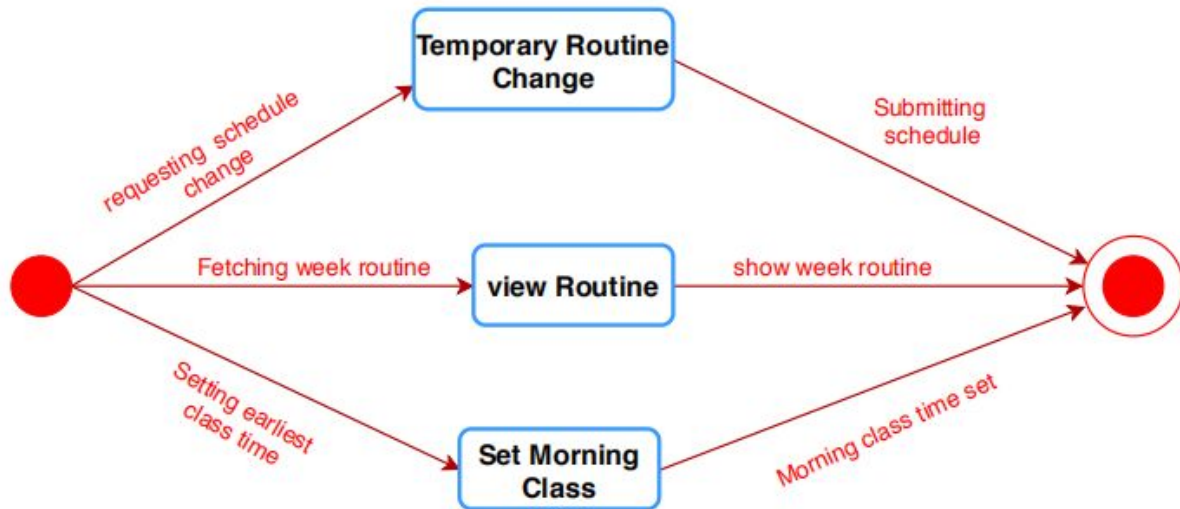
**ID:** C09



**Description:** News and Events can be viewed on user request.

**Class Name:** Routine

**ID:** C10

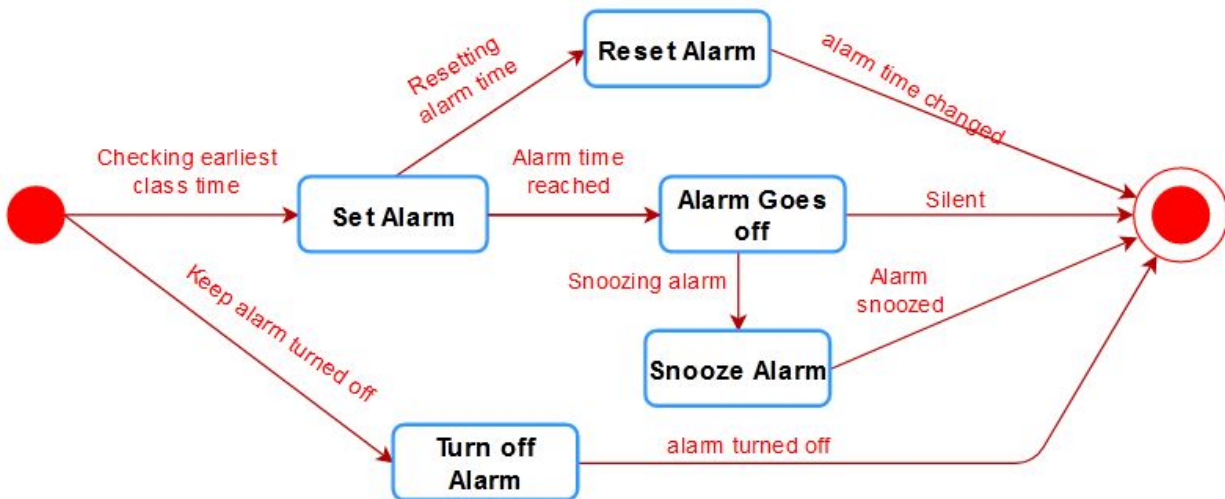


**Description:** Temporary changes can be made to the routine. Routine can be viewed and First Morning Class will be automatically set on each day.



**Class Name:** Alarm

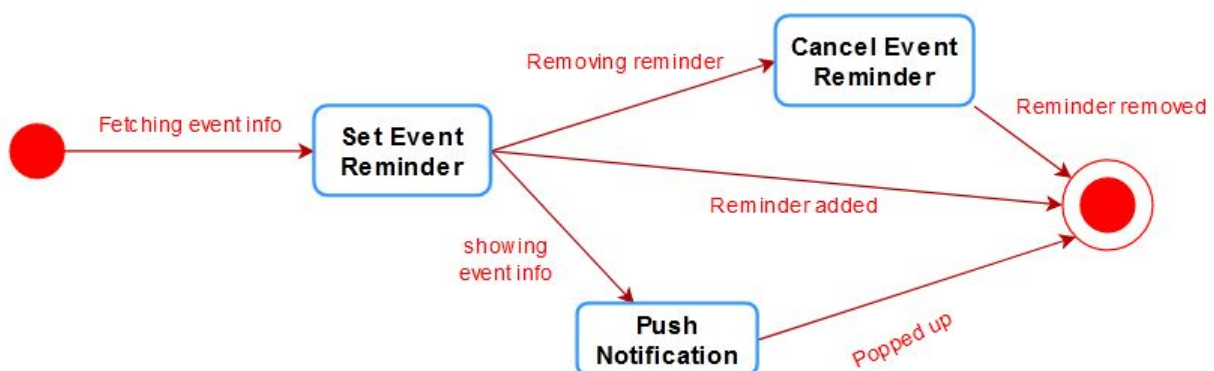
**ID:** C11



**Description:** Alarm can be set. Alarm can be turned off and snoozed. Alarm will go off. Alarm can be reset.

**Class Name:** Event Reminder

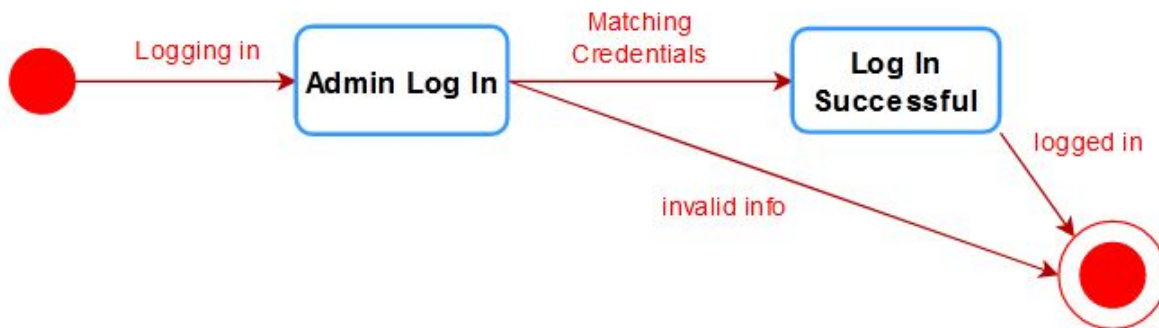
**ID:** 12



**Description:** Event Reminder will be set based on event data. Event Reminder can be cancelled anytime. Push notification will be shown.

**Class Name:** Admin

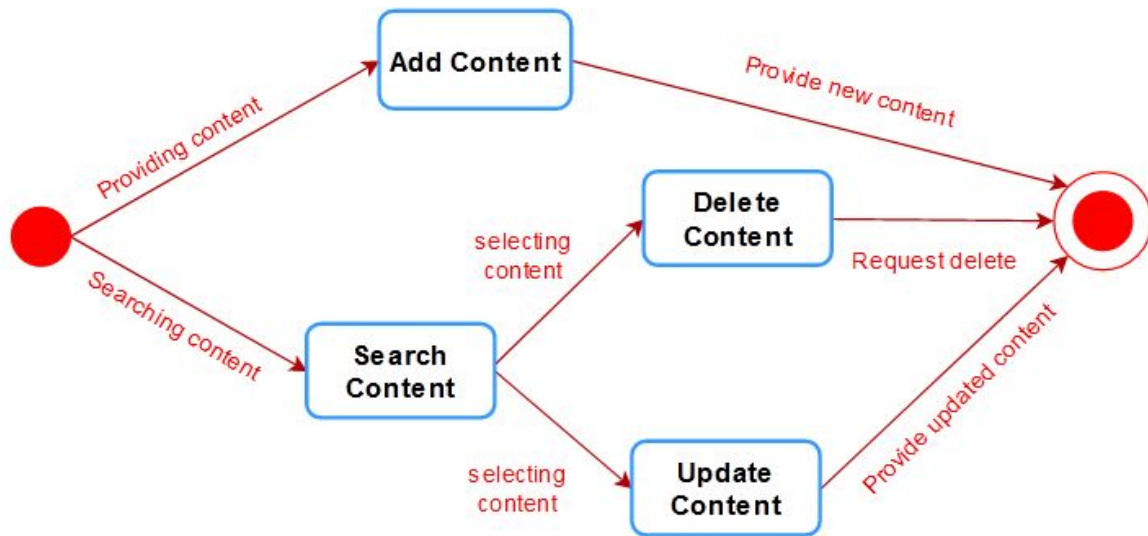
**ID:** C13



**Description:** Admin can log in and validity check will take place.

**Class Name:** Content Manager

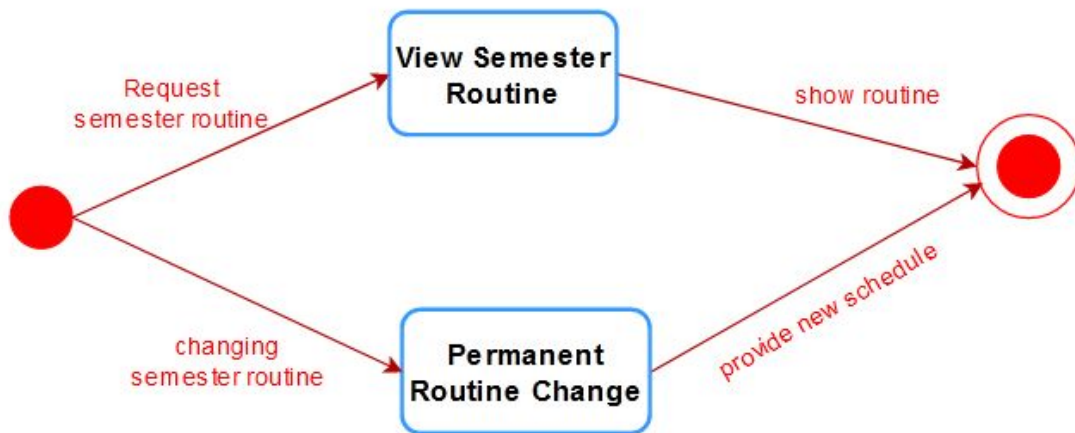
**ID:** C14



**Description:** Contents can be searched, added, deleted, updated as per user request.

**Class Name:** Semester Routine

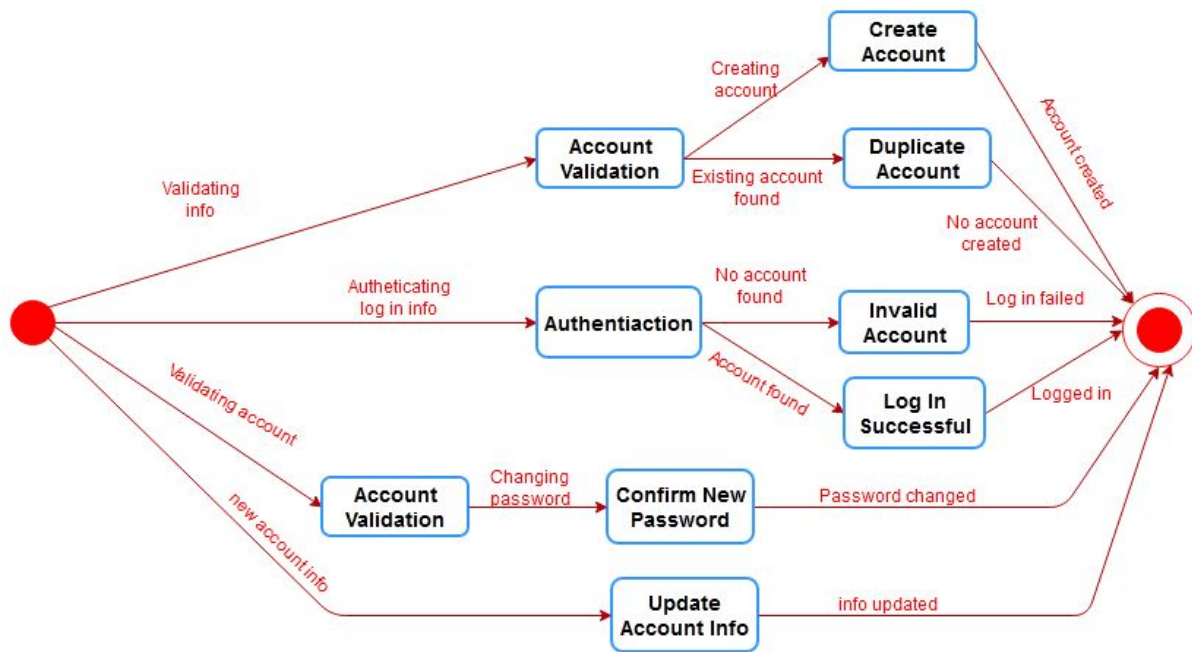
**ID:** C15



**Description:** Semester Routine can be viewed and permanent changes can be made.

**Class Name:** Account Manager

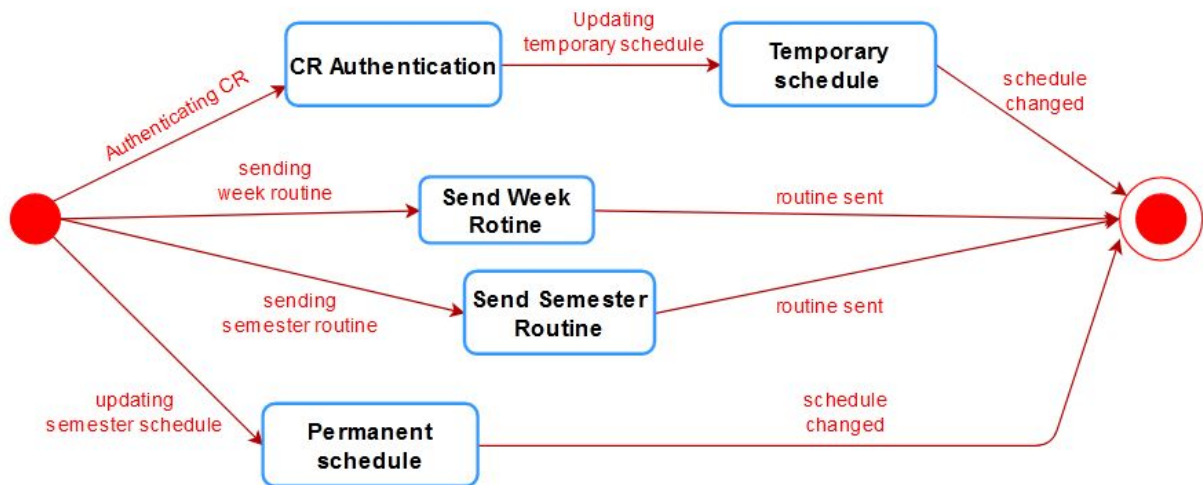
**ID:** C16



**Description:** Account validation can be done. Account can be created and duplicacy will be checked. Authentication can be done. Passwords can be changed. Account info can be updated.

**Class Name:** Routine Manager

**ID:** C17



**Description:** CR can be authenticated. Temporary schedule can be updated. Week Routine can be sent. Semester Routine can be sent. Permanent change to schedule can be made

**Class Name:** GetContent

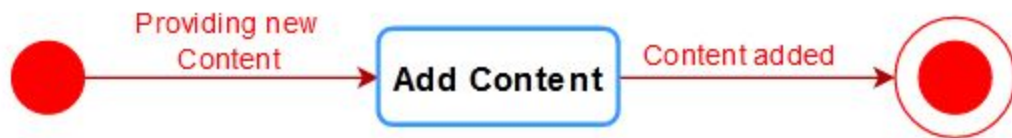
**ID:** C18



**Description:** Contents can be searched.

**Class Name:** AddContent

**ID:** C19



**Description:** Contents can be added.

**Class Name:** UpdateContent

**ID:** C20



**Description:** Contents can be updated.

**Class Name:** DeleteContent

**ID:** C21



**Description:** Contents can be deleted.



### 8.3 SEQUENCE DIAGRAM:

