

# **i-TURNOUT - AN APPLICATION FOR ONLINE ATTENDANCE MANAGEMENT SYSTEM**

**A PROJECT THESIS**

SUBMITTED BY

**Sarvesh Singh (M150054CA)**

*In partial fulfilment for the award of the Degree of*

**Master of Computer Applications**

Under the guidance of

**Dr. Sudeep K S**

(Assistant Professor, Dept. of Computer Science and Engineering)



**Department of Computer Science and Engineering**

National Institute of Technology Calicut

NIT CAMPUS PO, CALICUT

KERALA, INDIA- 673601

MAY 2018

## ACKNOWLEDGEMENT

It is matter of great pleasure for me to submit this report on “Attendance management System” for the partial fulfilment of the degree of Master of Computer Application from NIT Calicut.

I would like to thank my internal project guide **Dr. Sudeep K S**, Assistant Professor, Dept. of Computer Science and Engineering, NIT Calicut as well as my additional guide, **Ms. Sahla Habeeb M A** for their tremendous guidance, support and excellent direction on project Work.

I would like to give special thanks to my friends and family who always helped me when I got any problem during project.

## **DECLARATION**

I hereby declare that this submission is my own work and to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or another institute of higher learning, except where due acknowledgment has been made in the text.

**Signature**

**Place:** Calicut

**Date:** 9/05/2018

**Name:** Sarvesh Singh

**Reg No:** M150054CA

## **CERTIFICATE**

This is to certify that the report entitled: “An Application for Online Attendance Management System” submitted by Mr Sarvesh Singh having Reg. No: M150054CA to the National Institute of Technology Calicut towards partial fulfilment of the requirements for the award of the Degree of Master of Computer Application is a bona fide record of the work carried out by him under my supervision and guidance during the academic year 2017-2018.

**Dr. Sudeep K S**

Assistant Professor

Dept. of Computer Science and Engineering

NIT Calicut

Place: Calicut

Date: 09-05-2018

Signature of the Head of the Department

## CONTENTS

Topic	PageNo
Acknowledgement.....	2
Declaration.....	3
Certificate.....	4
Contents.....	5
Abstract.....	8
List of Figures.....	9
Chapter 1: Introduction.....	10
1.1: Purpose.....	10
1.2: Problem Definition.....	10
1.2.1: Input to the System.....	10
1.2.2: Output from the System.....	10
1.3: Background.....	11
1.4: Motivation.....	11
Chapter 2: Literature Survey.....	12
Chapter 3: Design and Implementation.....	13
3.1: Requirement Specification.....	13
3.1.1: Functional Requirement.....	13
3.1.1.1: Moderator.....	13
3.1.1.1.1: Login.....	13
3.1.1.1.2: Moderator Roles.....	13
3.1.1.1.3: Add Teacher Details.....	13
3.1.1.1.4: Add Student Details.....	14
3.1.1.1.5: Assign course to the teacher.....	14
3.1.1.1.6: Enroll student.....	14
3.1.1.1.7: Update Enrollment of student.....	14
3.1.1.1.8: Delete Records.....	15
3.1.1.1.8.1: Delete Records of semester.....	15
3.1.1.1.8.2: Delete Records of Student.....	15
3.1.1.1.8.2: Delete Records of Teacher.....	15
3.1.1.1.9: Log out.....	16
3.1.1.2: Teacher.....	16
3.1.1.2.1: Login.....	16

3.1.1.2.2: Teacher Roles.....	16
3.1.1.2.3: Query.....	17
3.1.1.2.4: Check Attendance.....	17
3.1.1.2.5: View Courses and Enrollment.....	17
3.1.1.2.6: Take Attendance.....	17
3.1.1.2.7: Change Password.....	18
3.1.1.2.8: Log out.....	18
3.1.1.3: Student.....	18
3.1.1.3.1: Login.....	18
3.1.1.3.2: Student Roles.....	19
3.1.1.3.3: Query.....	19
3.1.1.3.4: Check Attendance.....	19
3.1.1.3.5: View Courses.....	20
3.1.1.3.6: Notification.....	20
3.1.1.3.7: Change Password.....	20
3.1.1.3.8: Log out.....	20
3.1.2: Non Functional Requirement.....	21
3.1.2.1: Performance Requirement.....	21
3.1.2.1.1: Response Time.....	21
3.1.2.1.2: Reliability.....	21
3.1.2.2: Security.....	21
3.1.2.3: Software Quality Attribute.....	21
3.1.2.3.1: Usability.....	21
3.1.2.3.2: Reliability.....	21
3.1.2.3.3: Performance.....	21
3.2: Design.....	22
3.2.1: Use Case Diagram.....	22
3.2.1.1: Use case for Moderator.....	22
3.2.1.2: Use case for Teacher.....	23
3.2.1.3: Use case for Student.....	24
3.2.2: Entity Relationship Diagram.....	25
3.2.3: Class Diagram.....	26
3.2.4: Sequence Diagram.....	27
3.2.4.1: Sequence diagram for login.....	27
3.2.4.2: Sequence diagram for make attendance.....	28
3.2.4.3: Sequence diagram for notification.....	29
3.2.4.4: Sequence diagram for add teacher student details.....	30
3.2.4.5: Sequence diagram for add update course.....	31
3.2.4.6: Sequence diagram for enrollment.....	32

3.2.4.7: Sequence diagram for query.....	33
3.2.4.8: Sequence diagram for check attendance.....	34
3.3 Operating and Development Environment.....	35
3.4 Practical Implementation.....	36
3.4.1 Moderator.....	36
3.4.2 Teacher.....	37
3.4.3 Student.....	38
Chapter 4: Conclusion.....	39
Chapter 5: Glossary.....	40
Chapter 6: References.....	41

## ABSTRACT

Attendance management is one of the most general task that need to be done in any academic institute. In every semester department assigns courses to the teachers and teacher have to take the attendance in that course.

Taking attendance in each class, maintaining it till the end of the semester, informing to the students for short attendance and responding to their queries is a very tedious task for Teachers. It waste lots time and creates inaccuracy when it is done manually using pen and paper.

Students should also be aware of their attendance. They should get notified if their attendance get shorted. There should be an ease of communication between teachers and students regarding the attendance.

These task can be done very easily when we use technology for that. Attendance Management System (AMS) application - “i-Turnout” is an android application that can perform these task very easily. It just need an android phone and internet connection. There are three kind of users involved Moderator , Teacher and Student. Moderator adds and manages all the details of Students and Teachers. Teacher takes and views attendance and responds to Students queries. Student can view his/her attendance, get notification for short attendance and can ask queries if he/she found any issue in his/her attendance.

i-Turnout performs these activities very easily and saves lots of time, increases accuracy and reduces attendance issues. Using i-Turnout is also environment friendly, it saves lots of paper wastage.



## LIST OF FIGURES

Figure 3.2.1 .....	22
Figure 3.2.2 .....	23
Figure 3.2.3 .....	24
Figure 3.2.4 .....	25
Figure 3.2.5 .....	26
Figure 3.2.6 .....	27
Figure 3.2.7 .....	28
Figure 3.2.8 .....	29
Figure 3.2.9 .....	30
Figure 3.2.10 .....	31
Figure 3.2.11 .....	32
Figure 3.2.12 .....	33
Figure 3.2.13 .....	34

## **CHAPTER 1: INTRODUCTION**

### **1.1: PURPOSE**

The purpose of this Android application is to manage attendance related activity. This application will help to both, Student and Teachers in all attendance related activities .

### **1.2: PROBLEM DEFINITION**

“i-Turnout”, An android application for taking students attendance and managing the associated activities. It facilitates to access the attendance information by Student, Teacher in a particular subject. This system will also help in evaluating attendance and it will generate notification based on short attendance. Student can query/communicate to the teachers regarding short attendance.

#### **1.2.1: Input to the system**

- Details of Student added by Moderator.
- Details of Teacher added by Moderator.
- Courses assigned to the teachers.
- Course enrollment details of the students.
- Attendance of students taken in each subject.
- Queries and replies made by student and teachers.

#### **1.2.2: Output from the system**

- Student can view his/her enrolled courses.
- Student can view his/her attendance details of each subject.
- Notifications for short attendance.
- Teacher can view the courses that are assigned to him.
- Attendance details of student will be made available to teachers.

### 1.3 BACKGROUND

Attendance management is one of the most general task that need to be done in any academic institute. In every semester department assigns courses to the teachers and teacher have to take the attendance in that course.

Taking attendance in each class, maintaining it till the end of the semester, informing to the students for short attendance and responding to their queries is a very tedious task for Teachers. It waste lots time and creates inaccuracy when it is done manually using pen and paper.

Students should also be aware of their attendance. They should get notified if their attendance get shorted. There should be an ease of communication between teachers and students regarding the attendance.

For handling these task there should be an application that performs these task easily and efficiently.

### 1.4 MOTIVATIONS

i-Turnout is a standalone application. It will help to modernize our old attendance system, which primarily uses a pen and papers. Our application provides every teacher with a digital or electronic register which can be their very own android based smart phone. The application provides with simple and easy to use UI for the teachers so that they can easily and conveniently access the application.

#### **Teacher :**

- Easy and efficient taking attendance.
- Easy to respond student query related subject.
- Provides an easier way to maintain the recorded attendance details.
- It will make the attendance maintenance error free.

#### **Student :**

- Application will provide an easy way to know the attendance details.
- Application will provide query related subject
- Application will automatically give notifications to the student.

#### **Social Aspects :**

- It will save papers and help to our environment.
- If class is big then manual attendance breaks the discipline of class and students start chatting.
- Reduces the issues of attendance mistakes between students and teachers.

## CHAPTER 2: LITERATURE SURVEY

This section will give a brief description of the literature review/survey needed for The early phase of this project. It will discuss the information from the website, and research that has been done.

**Manual taking attendance :** In the most of colleges and schools they are taking attendance by pen and papers and maintains registers. It waste lots of extra time and wastage of paper. It also increases the workload.

**Biometric identification /Smart card /Voice recognition :** Several high technologies are also used for managing attendance like fingerprint, face detection and voice recognition. But these technologies are very expensive and less user friendly.

**Other Applications :** There are many other applications like AMS by agcTools available on internet, that help in managing attendance, but most of them don't provides notification and query facilities.

In compare of these systems, managing attendance by Android application is more easier and effective. It just needs an android phone and internet connection for these activities.

## CHAPTER 3: DESIGN AND IMPLEMENTATION

### 3.1: REQUIREMENT SPECIFICATION

#### 3.1.1: Functional Requirement

##### 3.1.1.1: Moderator

##### 3.1.1.1.1: Login

Moderator will enter the credentials and can log in with the right credentials.

**Pre-condition:**

Moderator should not already log in to the application.

Moderator must have working internet connection.

Moderator must have valid id and password.

**Post-condition:**

Moderator will be logged in after the successful authentication and it will redirect to the moderator interface.

##### 3.1.1.1.2: Moderator Roles

Moderator can add details of Teacher and student. He can also delete the records related to teacher and student.

**Pre-condition:**

Moderator must have logged in.

Moderator must have a working internet connection.

**Post-condition:**

The role will be added to the application.

##### 3.1.1.1.3: Add Teacher Details

Moderator will upload details of teacher.

**Pre-condition:**

Moderator must have logged in.

Moderator must have a working internet connection.

Teacher should not be existing already.

**Post-condition:**

Teacher will be added to the application.

#### **3.1.1.1.4: Add Student Details**

Moderator will upload details of student.

##### **Pre-condition:**

Moderator must have logged in.

Moderator must have a working internet connection.

Student should not be existing already.

##### **Post-condition:**

Student will be added to the application.

#### **3.1.1.1.5: Assign course to the teacher**

Moderator will assign courses to the teacher

##### **Pre-condition:**

Moderator must have logged in.

Moderator must have a working internet connection.

Course should not be existing already.

Teacher should be existing already.

##### **Post-condition:**

Course will be added to the application.

#### **3.1.1.1.6: Enroll student**

Moderator will enroll the students for course.

##### **Pre-condition:**

Moderator must have logged in.

Moderator must have a working internet connection.

Course should be existing already.

Student should be existing already.

##### **Post-condition:**

Student will enroll to the application.

#### **3.1.1.1.7: Update Enrollment of student**

Moderator can update enrollment of student.

**Pre-condition:**

Moderator must have logged in.  
Moderator must have a working internet connection.  
Course should be existing already.  
Student should be existing already.

**Post-condition:**

Student enrollment will update for the application.

**3.1.1.1.8: Delete Records**

Moderator can delete records according to Teacher, Student and Semester wise.

**3.1.1.1.8.1: Delete Records of semester**

Moderator can delete records of each semester.

**Pre-condition:**

Moderator must have logged in.  
Moderator must have a working internet connection.

**Post-condition:**

Records will be deleted from application.

**3.1.1.1.8.2: Delete Records of Student**

Moderator can delete records of particular student.

**Pre-condition:**

Moderator must have logged in.  
Moderator must have a working internet connection.  
Moderator must have to either select or select after searching to student.

**Post-condition:**

Student Records will be deleted from application.

**3.1.1.1.8.3: Delete Records of Teacher**

Moderator can delete records of particular teacher.

**Pre-condition:**

Moderator must have logged in.  
Moderator must have a working internet connection.  
Moderator must have to either select or select after searching to teacher.

**Post-condition:**

Teacher Records will delete from application.

#### **3.1.1.1.9: Log out**

Moderator logouts and ends the session when he no longer wants to do anything in the application.

**Pre-condition:**

Moderator must have logged in.

**Post-condition:**

Moderator will be logged out, and the session will be destroyed.

#### **3.1.1.2: Teacher**

##### **3.1.1.2.1: Login**

Teacher will enter the credentials and can log in with the right credentials.

**Pre-condition:**

Teacher should not already log in to the application.

Teacher must have working internet connection.

Teacher must have valid id and password.

**Post-condition:**

Teacher will be logged in after the successful authentication and it will redirect to the teacher interface.

##### **3.1.1.2.2:Teacher Roles**

Teacher can view the courses that were assigned by the moderator to him. He can also see the list of students enrolled in his courses. Teacher can take attendance of students for each instructional day and can view the attendance of each students. He will also be able to respond students queries and asked query to the student.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

**Post-condition:**

The role will be added to the application.



### 3.1.1.2.3: Query

Teacher will get message from student related to attendance and he can respond on that message.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

**Post-condition:**

Teacher can view list of messages.

Teacher can view details of message.

### 3.1.1.2.4: Check Attendance

Teacher can check attendance of each student in particular subject that are assigned to him/her. He can also view the students with short attendance.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

Teacher must have that course which subject he want to view attendance.

**Post-condition:**

Teacher can view easily who have short attendance. The name will be colorful of student who have short attendance.

Teacher can view overall attendance of the student (Total Class, Presence, absence and percentage).

### 3.1.1.2.5: View Courses and Enrollment

Teacher can view courses which are assigned by moderator and he can view the student who enroll in that course.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

**Post-condition:**

Teacher will get list of course.

Teacher will get list of student who enroll in that course.

### 3.1.1.2.6: Take Attendance

Teacher will take attendance of each subject, that are assigned to him/her.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

Teacher must have assigned that subject in which he want to take attendance.

**Post-condition:**

Teacher will get list of course.

Teacher will get list of student with checked attendance.

Teacher will submit attendance of particular class by unchecking the attendance of absent student.

Teacher can uncheck the the attendance by searching the specific student.

### **3.1.1.2.7: Change Password**

Teacher can change his/her password.

**Pre-condition:**

Teacher must have logged in.

Teacher must have a working internet connection.

Teacher must have id and password.

**Post-condition:**

Teacher password will be changed for application.

### **3.1.1.2.8: Log out**

Teacher logouts and ends the session when he no longer wants to do anything in the application.

**Pre-condition:**

Teacher must have logged in.

**Post-condition:**

Teacher will be logged out, and the session will be destroyed.

### **3.1.1.3: Student**

#### **3.1.1.3.1: Login**

Student will enter the credentials and can log in with the right credentials.

**Pre-condition:**

Student should not already log in to the application.

Student must have working internet connection.

Student must have valid id and password.

**Post-condition:**

Student will be logged in after the successful authentication and it will redirect to the student interface.

### **3.1.1.3.2: Student Roles**

Student can view courses and view enrollment in those courses. He can check his attendance(calculated). He will be able to ask queries related to attendance. He will get notification regarding short attendance.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

**Post-condition:**

The role will be added to the application.

### **3.1.1.3.3: Query**

Student can query to the teachers those are teaching him. He will get response of that query.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

Student must have enroll in that course

**Post-condition:**

Student can view list of message.

Student can view details of message.

### **3.1.1.3.4: Check Attendance**

Student can check attendance of each subject in which he/she is enrolled.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

Student must have that course which subject attendance he wants to view.

**Post-condition:**

Student can view overall attendance of the subject (Total Class, Presence, absence and percentage).

### **3.1.1.3.5: View Courses**

Student can view his enrolled courses.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

Student must have enrolled in that course.

**Post-condition:**

Student will get list of courses.

### **3.1.1.3.6: Notification**

Student will get notification based on each courses.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

Student must have enrolled in that course.

**Post-condition:**

Student will get list of notification.

Student can view details of those notification.

### **3.1.1.2.7: Change Password**

Student can change his password.

**Pre-condition:**

Student must have logged in.

Student must have a working internet connection.

Student must have id and password.

**Post-condition:**

Student password will change for application.

### **3.1.1.3.8: Log out**

Student logouts and ends the session when he no longer wants to do anything in the application.

**Pre-condition:**

Student must have logged in.

**Post-condition:**

Student will be logged out, and the session will be destroyed.

**3.1.2: NON FUNCTIONAL REQUIREMENT****3.1.2.1: Performance Requirement****3.1.2.1.1: Response Time**

The application shall be user-friendly, and for every action of the system, there is an immediate response.

**3.1.2.1.2: Reliability**

The application shall be highly reliable and efficient. It will operate 24\*7 without any failure for a separate period or a specified number of users.

**3.1.2.2: Security**

All the user information are kept under supervision. No one can log onto others account without completing the authentication process. Personal details of each user must be kept secretly in our database.

**3.1.2.3: Software Quality Attribute****3.1.2.3.1: Usability**

The application will be presented and organised in such a manner that it is both, visually appealing and easy for the user to navigate.

**3.1.2.3.2: Reliability**

Highly reliable as there will be close to zero tolerance for errors in the algorithm that computes.

**3.1.2.3.3: Performance**

The performance shall depend upon hardware components of the client/customer.

## 3.2: DESIGN

### 3.2.1: USE CASE DIAGRAM

#### 3.2.1.1: Use case for Moderator

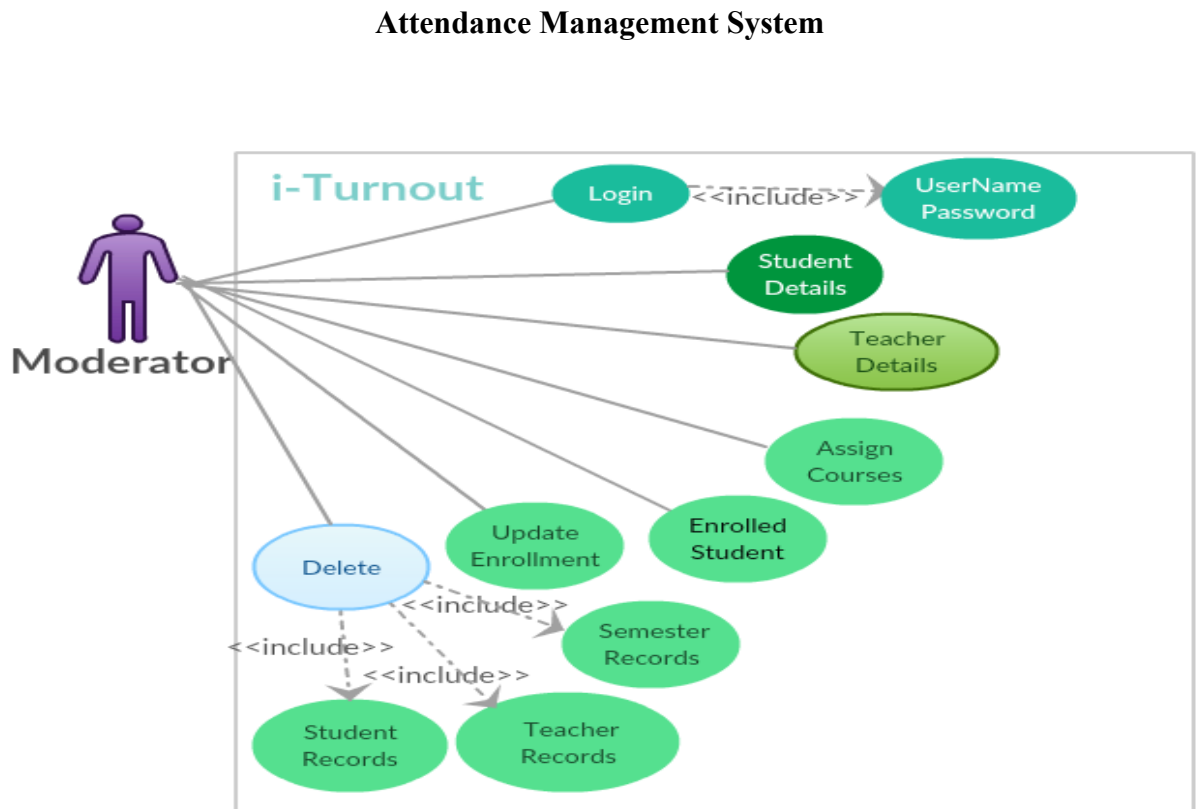


Figure 3.2.1

### 3.2.1.2: Use case for Teacher

#### Attendance Management System

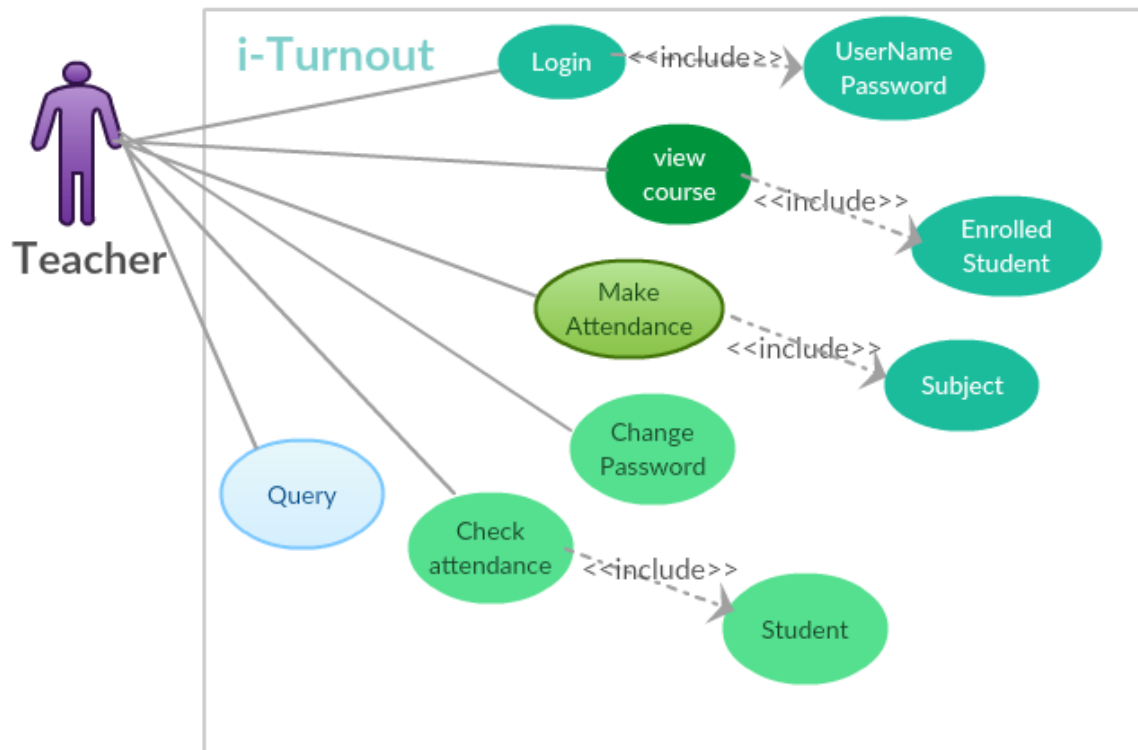


Figure 3.2.2

### 3.2.1.3: Use case for Student

#### Attendance Management System

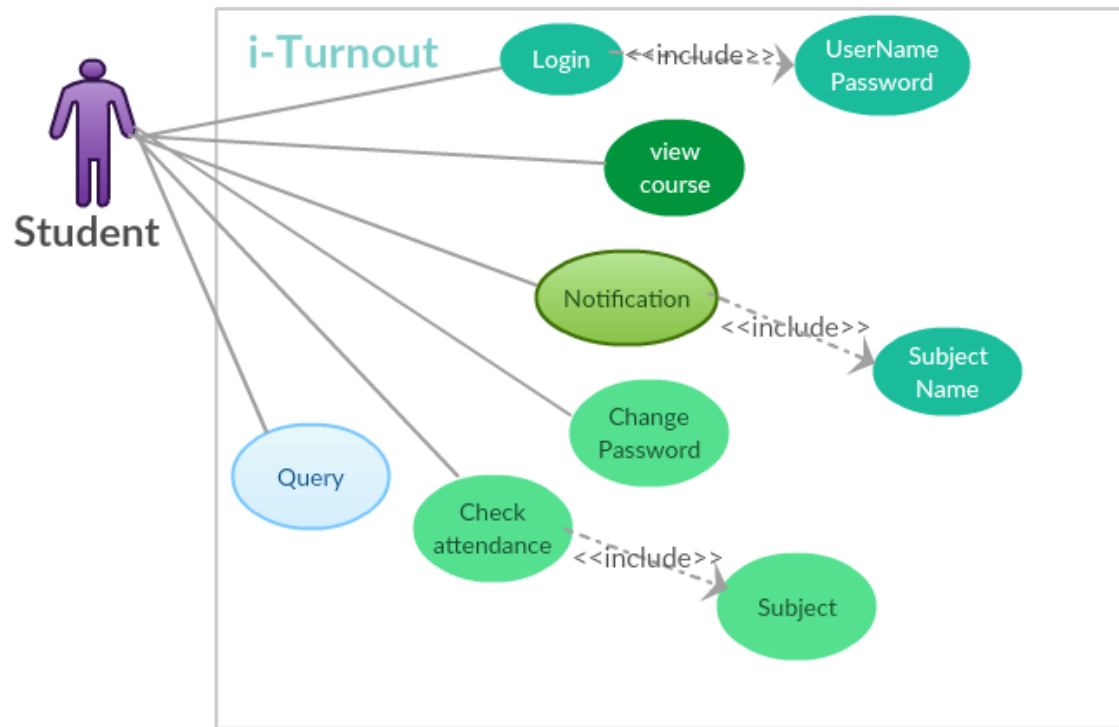


Figure 3.2.3



### 3.2.2: ENTITY RELATIONSHIP DIAGRAM

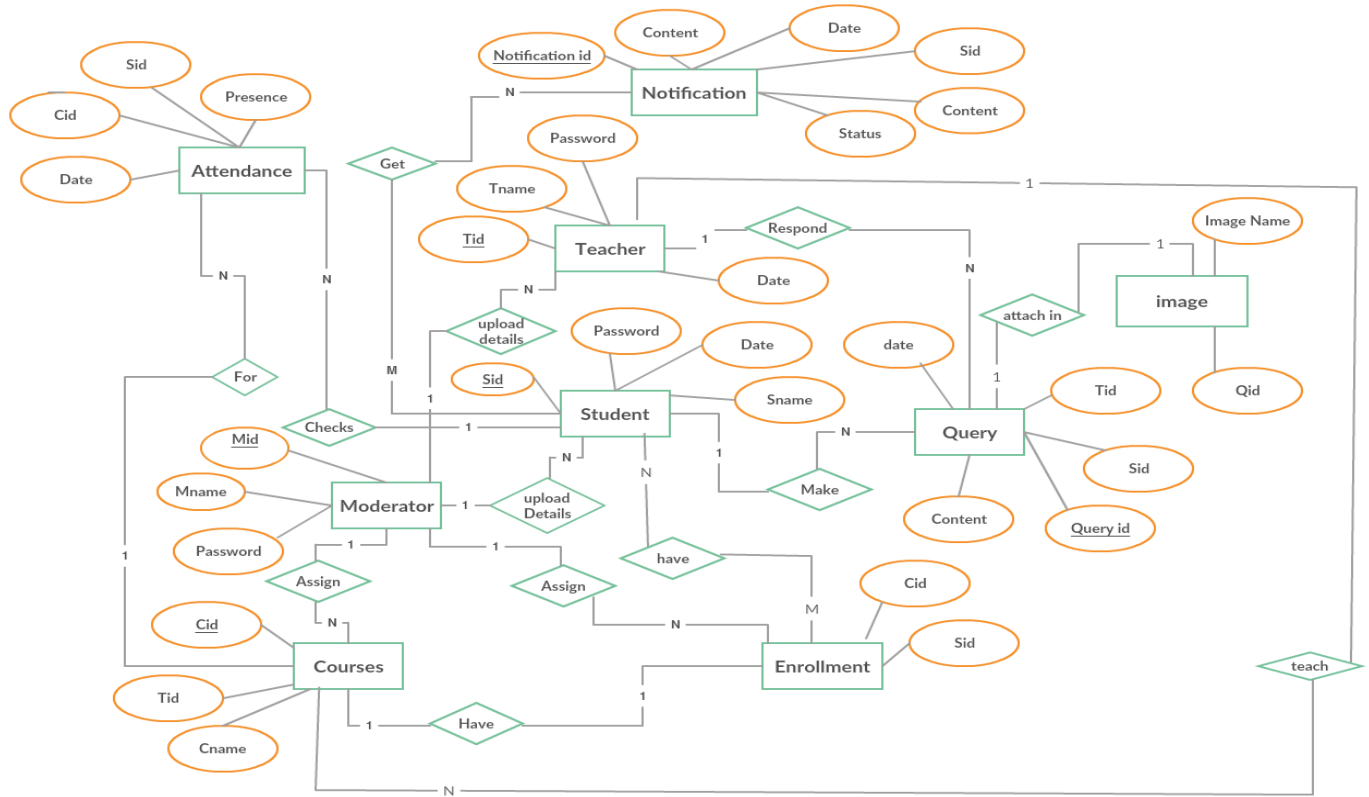


Figure 3.2.4

### 3.2.3: CLASS DIAGRAM

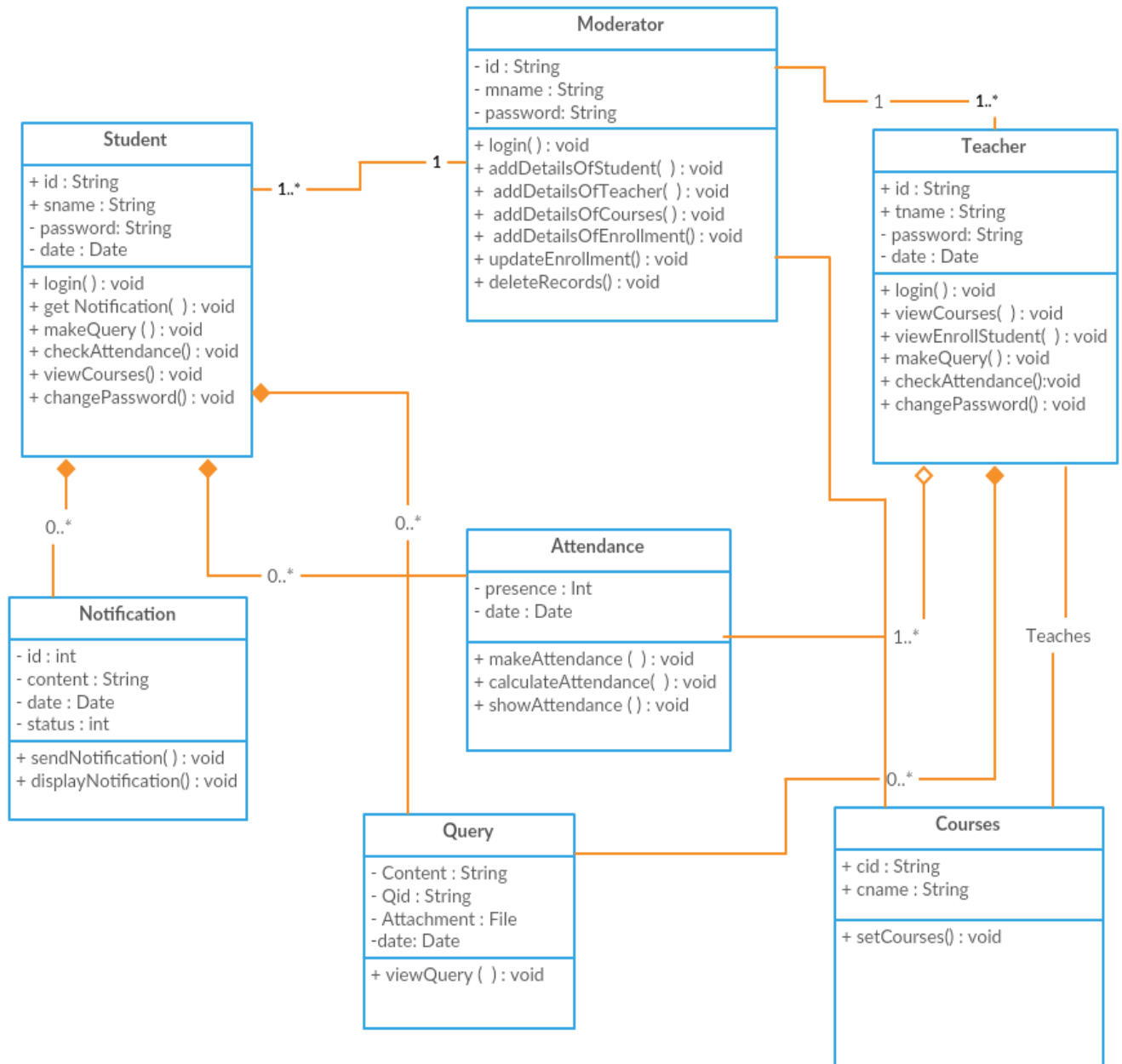


Figure 3.2.5

### 3.2.4: SEQUENCE DIAGRAM

#### 3.2.4.1: Sequence diagram for login

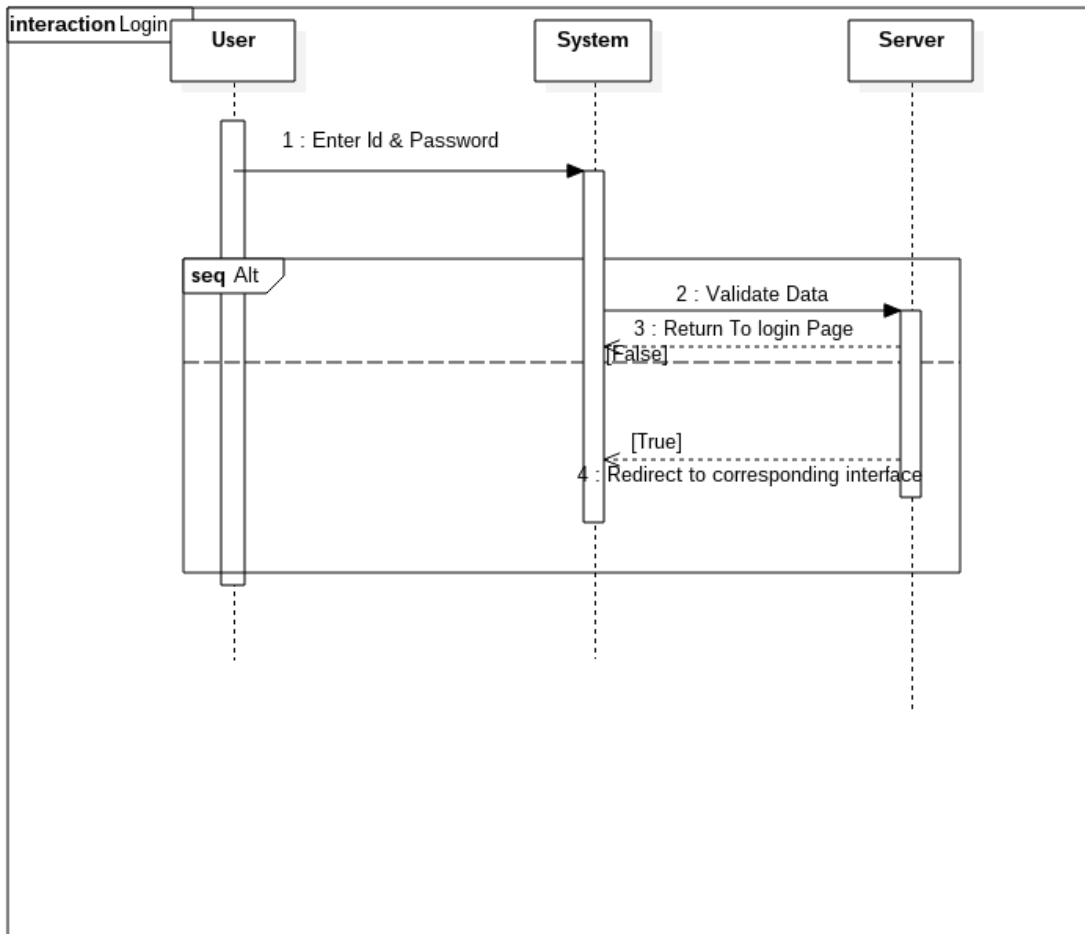


Figure 3.2.6

### 3.2.4.2: Sequence diagram for make attendance

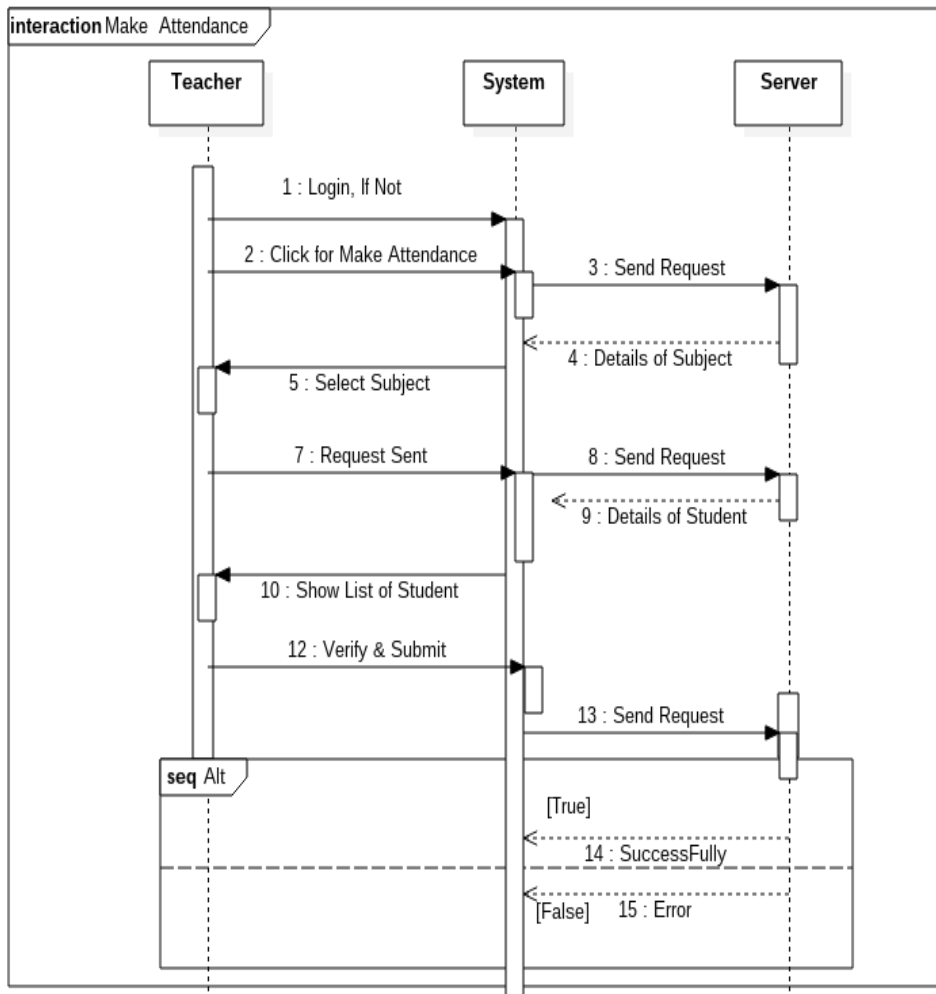
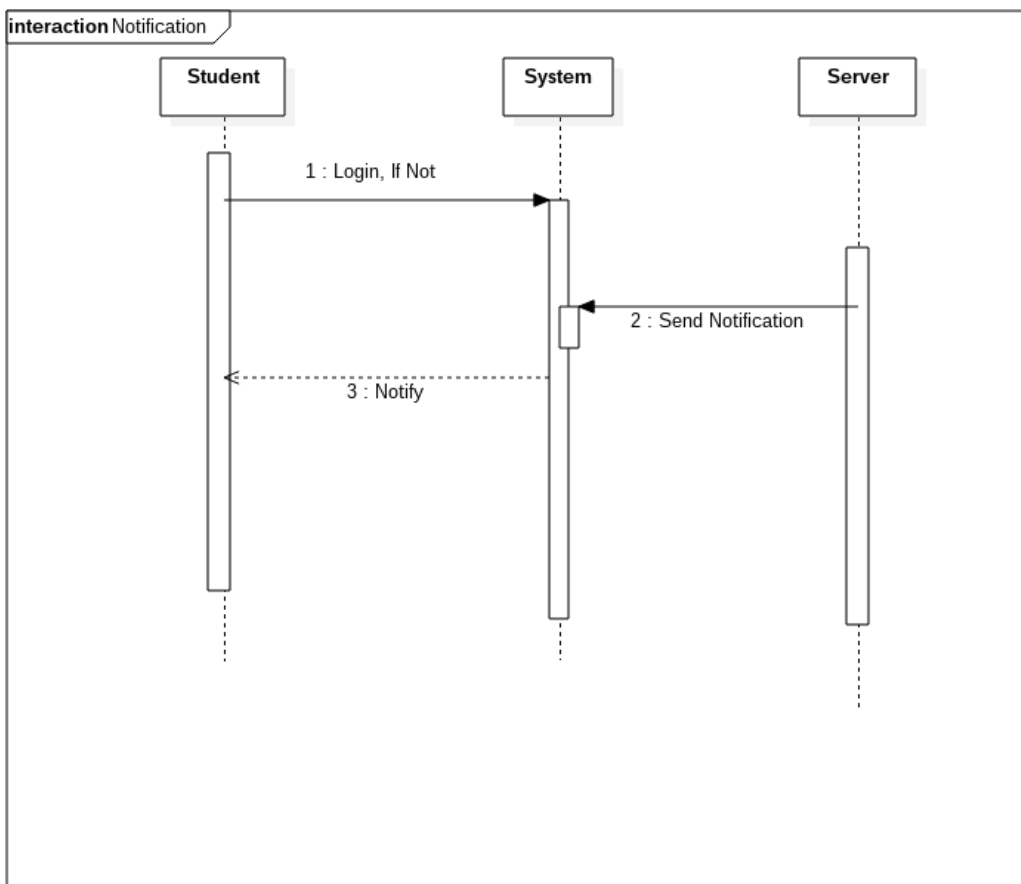


Figure 3.2.7

### 3.2.4.3: Sequence diagram for notification



**Figure 3.2.8**

### 3.2.4.4: Sequence diagram for add teacher student details

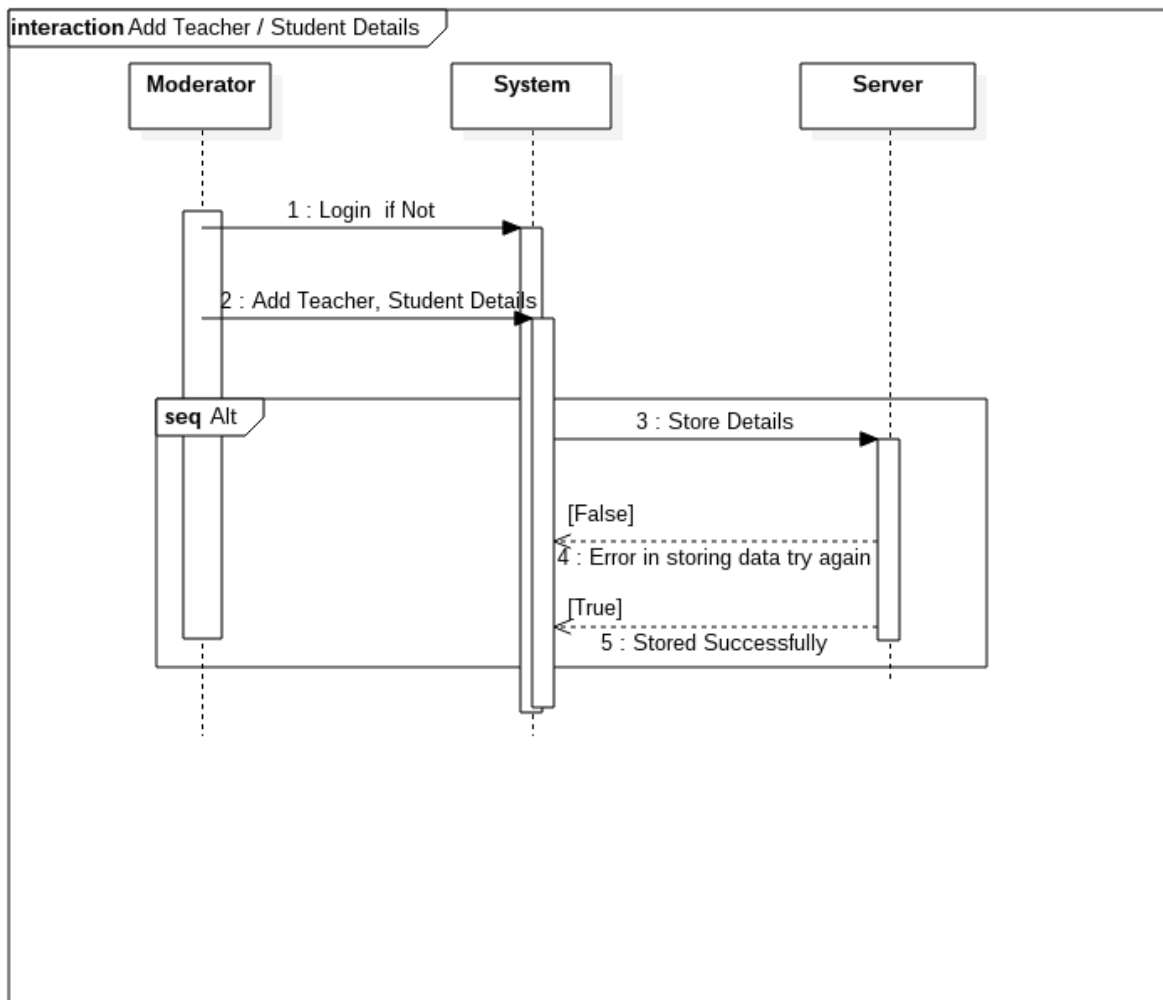
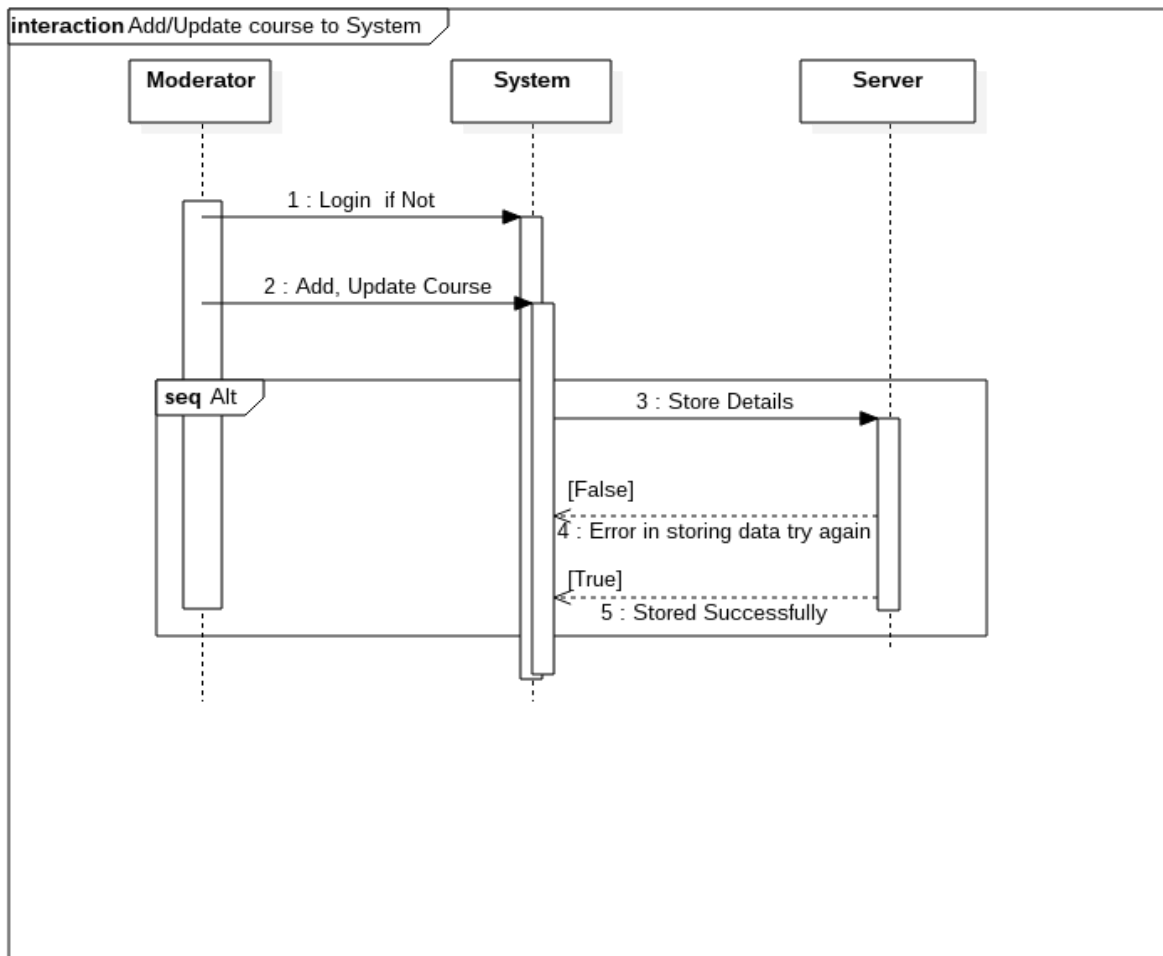


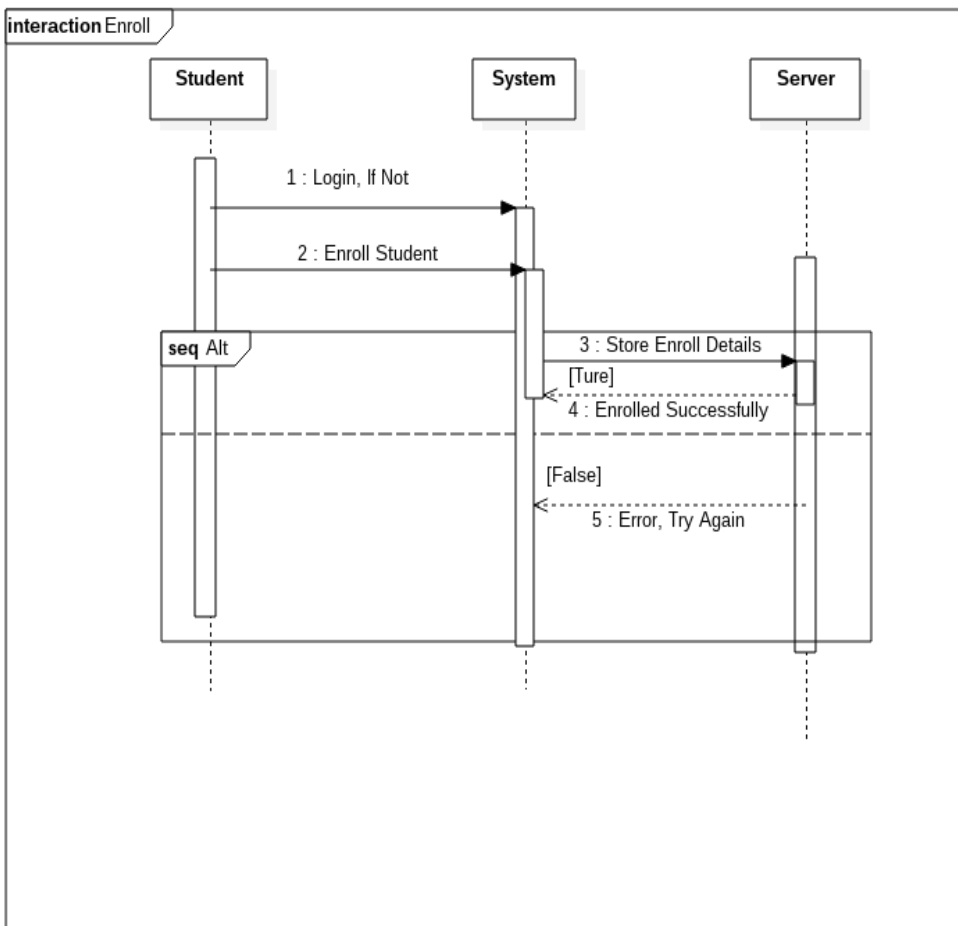
Figure 3.2.9

### 3.2.4.5: Sequence diagram for add update course



**Figure 3.2.10**

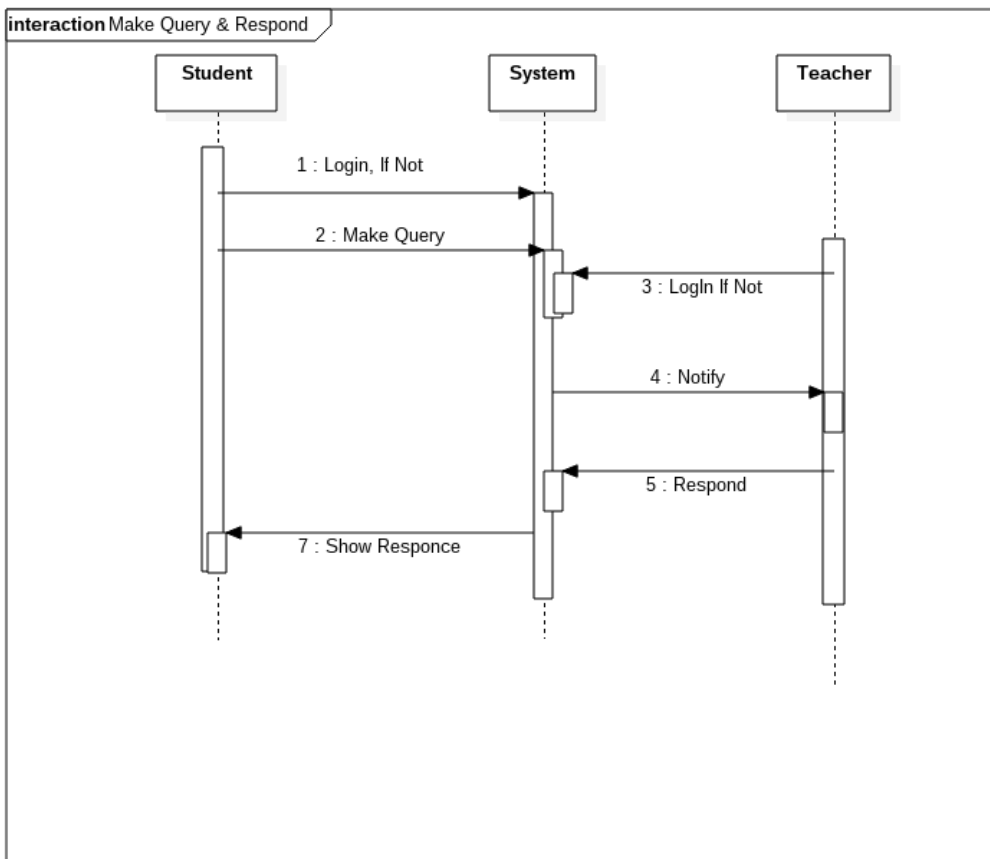
### 3.2.4.6: Sequence diagram for enrollment



**Figure 3.2.11**



### 3.2.4.7: Sequence diagram for query



**Figure 3.2.12**

### 3.2.4.8: Sequence diagram for check attendance

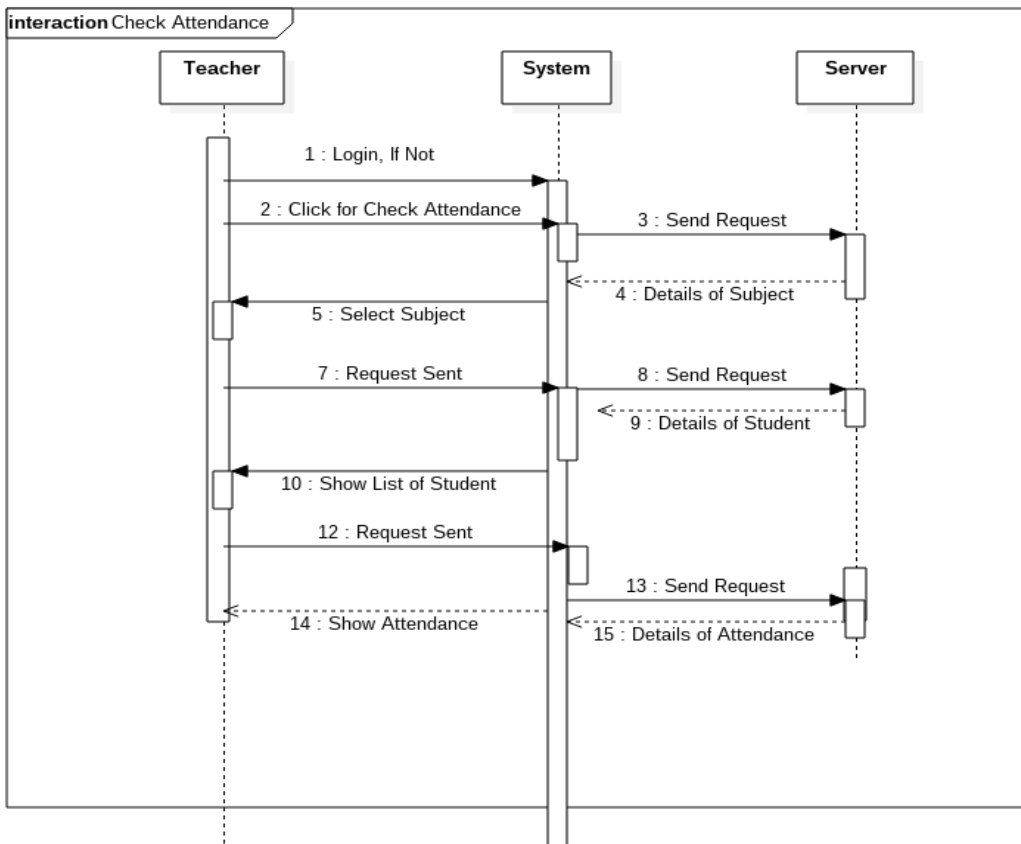


Figure 3.2.13

## 3.3 OPERATING AND DEVELOPMENT ENVIRONMENT

### Client Side Requirement

1. Android OS (android 4.0 and above)
2. Connectivity: Good Internet connection

### Server Side Technologies used

1. Language : PHP
2. Database : MySQL

### Development Environment used

1. Android Studio and JDK
2. Languages : Java and XML
3. Windows/Linux System

### Github

Github is used for managing the source code.

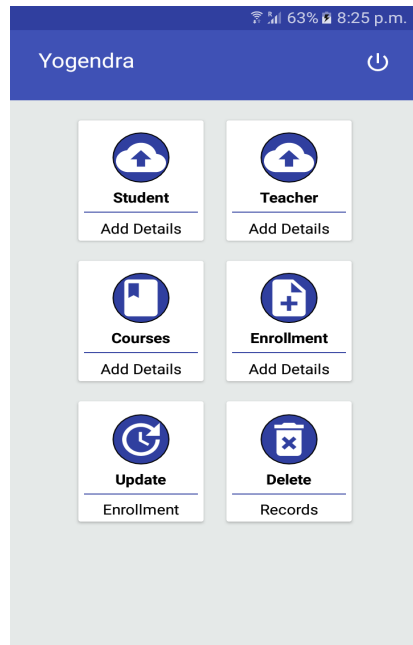
Repository access : <https://github.com/Singh007sarvesh/iTurnout>

Commits : <https://github.com/Singh007sarvesh/iTurnout/commits/master>

## 3.4 PRACTICAL IMPLEMENTATION

### 3.4.1: Moderator

Moderator Dashboard



Add Details of Student

Add Details Of Teacher

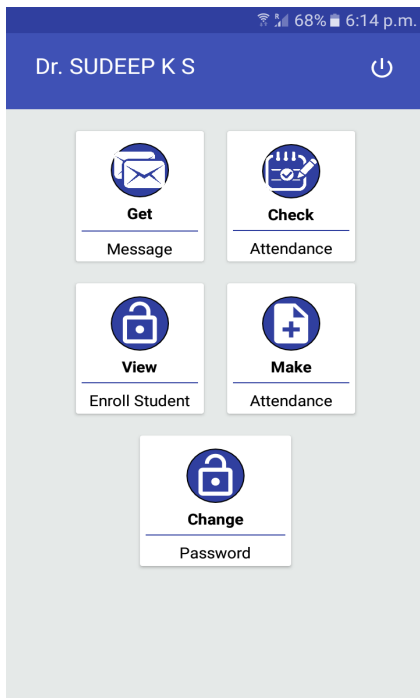
Add Course Details

Update Course Enrollment

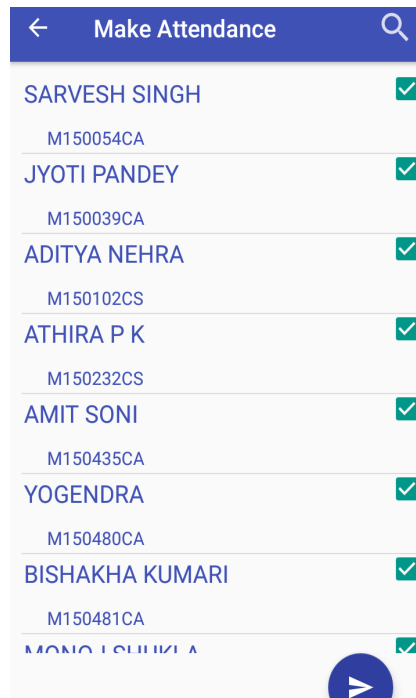
Delete Records

### 3.4.2: Teacher

Teacher DashBoard



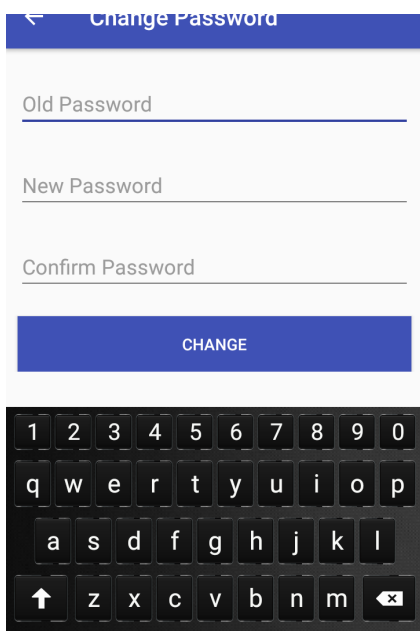
Take Attendance



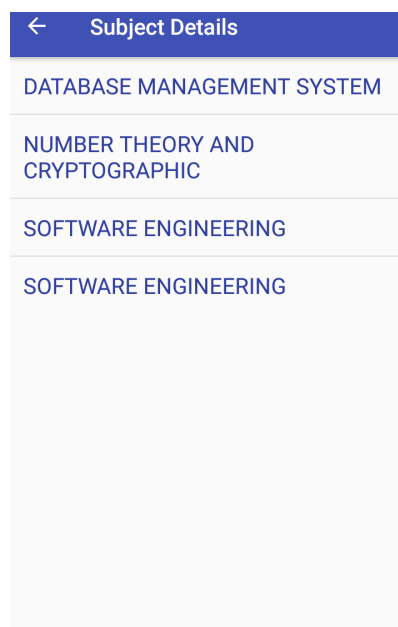
Check Attendance



Change Password



View Courses

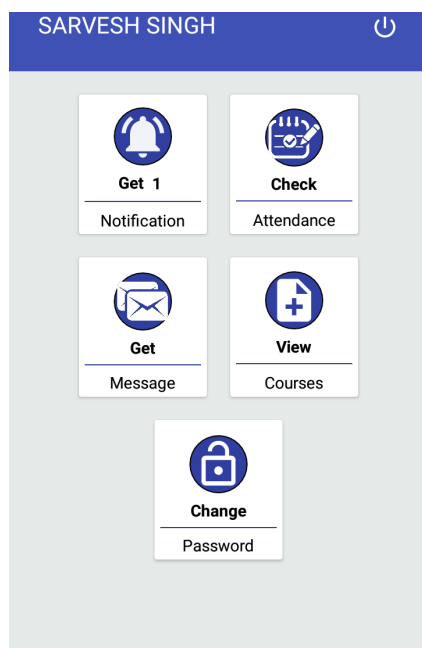


Attendance Details

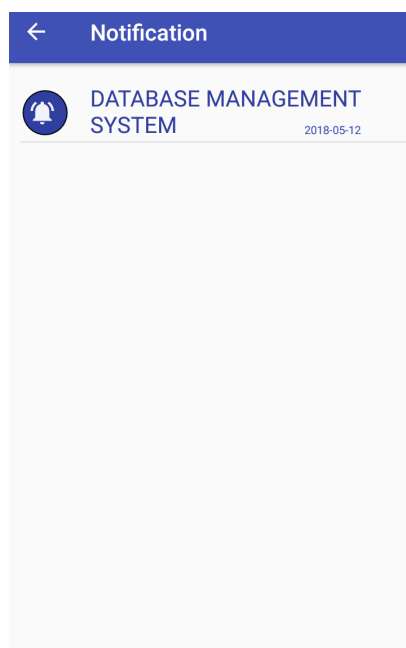
Attendance Details	
Total Class	4
Presence Class	2
Absent Class	2
Percentage	50 %

### 3.4.3: Student

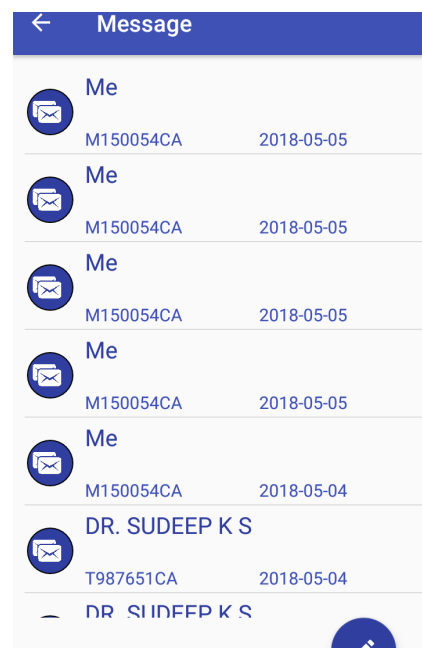
Student DashBoard



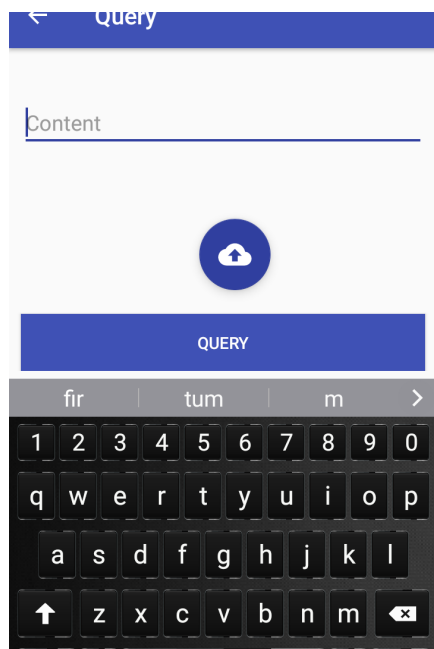
Notification



Message List



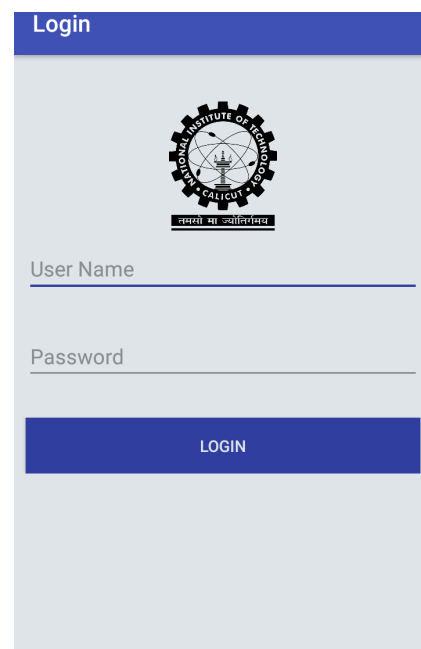
Query



Splash



User Login



## **CHAPTER 4: CONCLUSION**

Attendance Management System (AMS) application i-Turnout has been developed for taking attendance and managing the associated activities. It helps Teacher in taking attendance, managing attendance related records and replying to students queries. Student can also be aware of their attendance , get notifications for short attendance and ask queries for any attendance issues. So it makes the task of both Teacher and Student easier , efficient and more accurate.

This android application makes the attendance management inexpensive and environment friendly too by saving the lots of paper wastage.

## **CHAPTER 5: GLOSSARY**

JDK- Java Development Kit

OS- Operating System

XML- Extensible markup language

PHP- Hypertext Preprocessor

UI- User Interface

AMS- Attendance Management System



## CHAPTER 6: REFERENCES

- [1] <https://www.udemy.com/complete-android-n-developer-course/> ‘Working of Android’  
(Accessed on : 05/05/2018)
- [2] <https://developer.android.com/studio/> (Accessed on : 07/05/2018)
- [3] [https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp) (Accessed on : 08/05/2018)
- [4] Ramez Elmasri, Shamkant B. Navathe: Database Systems: models, languages, design and application programming, Pearson, Edition 6, 2013.
- [5] <https://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language>  
‘XML reference’
- [6] [https://www.youtube.com/results?search\\_query=android+studio+by+prabeesh](https://www.youtube.com/results?search_query=android+studio+by+prabeesh) (Accessed on : 01/05/2018)
- [7] <https://in.udacity.com/course/android-developer-nanodegree-by-google--nd801> (Accessed on : 14/04/2018)
- [8] <https://creatly.com/app/#> (Accessed on : 18/05/2018)
- [9] <http://staruml.io/download> (Accessed on : 12/05/2018)