Software Requirements Specification

Version 1.2

8 March 2016

Attendance Management System

Submitted in partial fulfillment

Of the requirements of

CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223). The students who submitted this team projects were :

<<Sangram Gaikwad UG201313010 >>

<<Arnav Chopra UG201310008 >>

<<Arnav Jindal UG201310009 >>

<<Ravindra Saini UG201310028 >>

# Table of Contents

[Table of Contents](#h.1fob9te)

[List of Figures](#h.2et92p0)

[1.0. Introduction](#h.tyjcwt)

[1.1. Purpose](#h.3dy6vkm)

[1.2. Scope of Project](#h.1t3h5sf)

[1.3 Constraints](#h.4d34og8)

[1.4 Assumptions and Dependencies](#h.2s8eyo1)

[1.5. Glossary](#h.17dp8vu)

[1.6. References](#h.3rdcrjn)

[1.7. Overview of Document](#h.26in1rg)

[2.0. Overall Description](#h.lnxbz9)

[2.1 System Environment](#h.35nkun2)

[2.2 Functional Requirements Specification](#h.1ksv4uv)

[2.2.1 Use case 1](#h.44sinio)

[Use case:](#h.2jxsxqh)

[2.3 User Characteristics](#h.z337ya)

[2.4 Non-Functional Requirements](#h.3j2qqm3)

[3.0. Requirements Specification](#h.1y810tw)

[3.1 Functional Requirements](#h.4i7ojhp)

[3.1.1](#h.2xcytpi) Generate QR Code

[3.1.2](#h.1ci93xb) Student: View Statistics

3.1.3 Admin: Generate Report

3.1.4 Input data

3.1.5 Query Student Information

3.1.6 Login

3.1.7 Verify QR Code

3.1.8 Update Attendance

[3.3 Detailed Non-Functional Requirements](#h.3whwml4)

[***3.4*** ***Logical Structure of the Data***](#h.2bn6wsx)

[4.0 Supporting information](#h.qsh70q)

[4.1 Table of contents and index](#h.3as4poj)

[4.2 Appendixes](#h.1pxezwc)

# Document Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 17th January, 2016 | Version 1.0 | Scope, Requirements and Use case Tables were added |
| 27th January, 2016 | Version 1.1 | Use case Diagrams, Class Diagrams, Sequence and Activity diagrams were added |
| 8th March, 2016 | Version 1.2 | Made some changes according to coading |

# List of Figures

|  |  |
| --- | --- |
| **Figure number** | **Description** |
| **2.1** | **System Use Case Diagram** |
| **2.2.1(a)(b)** | **Generate QR code Class Diagram,Sequence Diagram** |
| **2.2.2(a)(b)** | **Login Class Diagram,Sequence Diagram** |
| **2.2.3(a)(b)** | **Input Data Class Diagram,Sequence Diagram** |
| **2.2.4(a)(b)** | **Generate Report Class Diagram,Sequence Diagram** |
| **2.2.5(a)(b)** | **Query Batch Data Class Diagram,Sequence Diagram** |
| **2.2.6(a)(b)** | **Generate Report Class Diagram,Sequence Diagram** |
| **2.2.7(a)(b)** | **Generate QR Class Diagram,Sequence Diagram** |
| **2.2.8(a)(b)** | **Query Course Data Class Diagram,Sequence Diagram** |

# 1.0. Introduction

## 1.1. Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the Attendance management system using QR code.

It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

This document is intended for both the stakeholders and the developers of the system. The stakeholders being the student, course instructor and the admin.

## 1.2. Scope of Project

The designed system is intended to work as attendance management solutions for educational institutions. The system essentially is managed by three kinds of external users namely, admin, course instructor and student. The QR Module verifies requests generated using valid QR codes and modifies the data accordingly.

* Users
* Admin: The admin is the master of the system. The admin has the privilege to manage, create or delete new students, course instructors and courses. The admin can allocate course instructors and a particular batch of students to a course so that the attendance registers are created dynamically.
* Course instructor: The second user is course instructor. The course instructor sets course variables like dates and holidays which are ignored by the admin or can be changed according to the instructor’s liking. This user can also view the overall number of classes, total attended classes and the percentage for students in the course to which the user is assigned.
* Student Module: The final user is the student who can get details that include the profile, and attendance report. The attendance details contain the total classes, attended classes and percentage of attended classes.
* QR Module: The QR module inputs the attendance of the students using QR codes after analysing the requests sent by the student users for a particular session.

Advantages of the proposed system:

* Fast and easy way to mark attendance of students present in a class for a course.
* Minimal effort for the Course instructor thus saving time.
* Reliable and versatile method to mark records and modify attendance.
* Hassle free and secure way to maintain attendance records and retrieve information due to use of unique QR codes for each session.
* Create informative and vivid illustrations by analysing the data and generating trends for the betterment of students and the instructor.
* Create a universal platform for collecting attendance, managing and analysing attendance database for the institute for all courses and students.
* Allow easy code modification and scope for addition of features in the near future.
* Create a scalable legacy software architecture which can provide a base for further editions to build upon without causing significant decrease in performance or increase in system requirements

## 1.3 Constraints:

* A student, who is not present in classroom can submit a proxy, if he gets the QR code from another student during the session timing and submits it to the system.
* No separation of students in batches is present, students are enrolled course-wise by the Admin irrespective of the year of study. Also no different provisions are made for Undergraduate and Postgraduate students.
* No leave management system is present, so no user(Admin, Course Instructor, Student) can modify the attendance database without QR code.
* The system is not designed for capturing QR code and receiving QR code from student side. It is assumed that the QR code are successfully captured and fed to the system.

## 1.4 Assumptions and Dependencies

* The Admin knows and fills the course list, assigns professors and students to their respective course correctly.
* The QR code is accurately captured by students’ devices and sent over the network without any loss or modification and arrives safely to the QR code system.
* The database is maintained on the database server in a secure way.
* The Course Instructor will fill in the Session information, course code in a correct manner and display the full QR code to the students.
* QR code will be generated and displayed to the students’ within session time.

## 1.5. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Admin | The master creator of the database. Manages course variables and participants |
| Course | An entity for which student registers and professor controls variables such as dates and sessions |
| Course Instructor | Professor to whom the course is assigned |
| Student | A restricted user who can only view/request to the attendance database |
| QR code | Quick Response Code used to identify every attendance session with a unique ID |
| Report | A final processed result generated after analysing the master database |
| Attendance Database | The master record where all users modify/enter data according to privileges |
| Status | Student is absent or present |
| Participant | A user associated with a course having its own access privilege |
| Session | An activity involving modification of master database as per access privilege |
| Course Variables | Course variables involve variables managed by the Professor such as dates |

## 

## 1.6. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.7. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

## MainUseCaseDiagram.jpg

## Fig. 2.1: System Use Case Diagram

## 2.2 Functional Requirements Specification

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

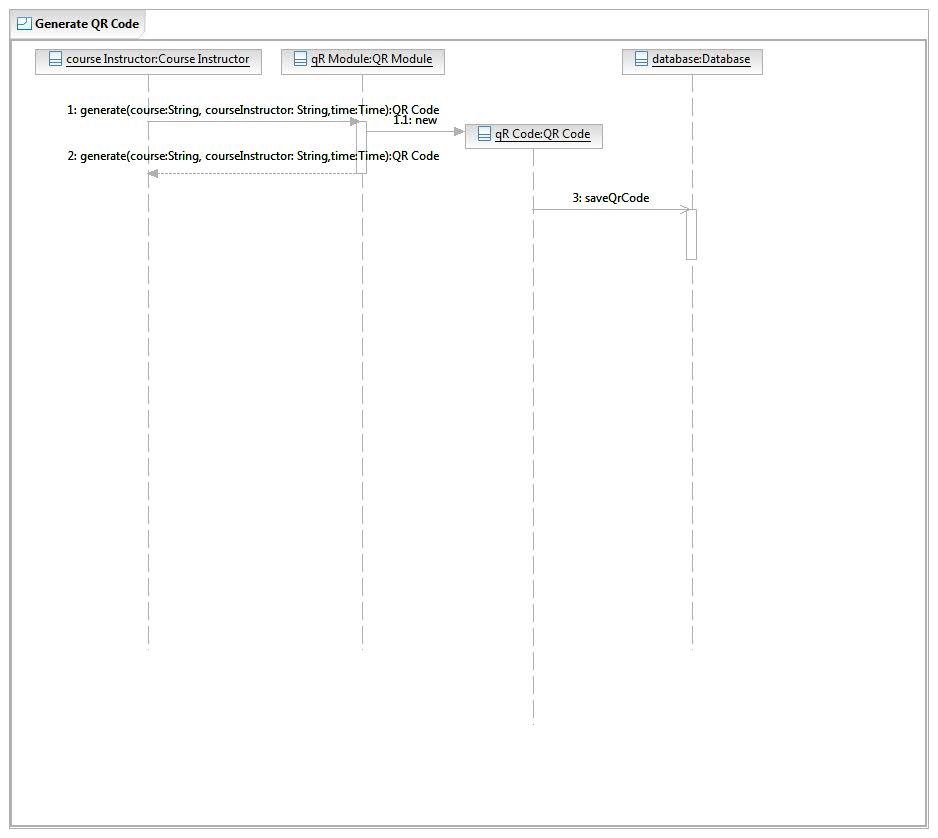
### 

### 2.2.1

### Use case: Generate QR code

### **Class Diagram:**GenerateQRCode.jpg

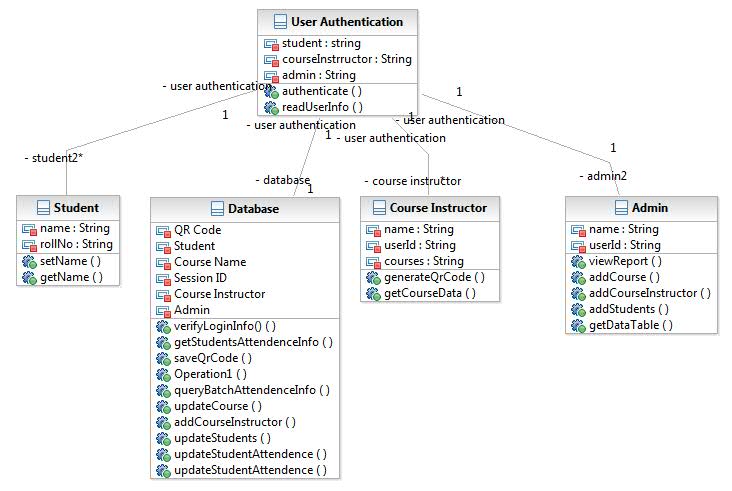
**Fig 2.2.1(a): Class diagram for Generate QR code use case**

**Sequence Diagram:**

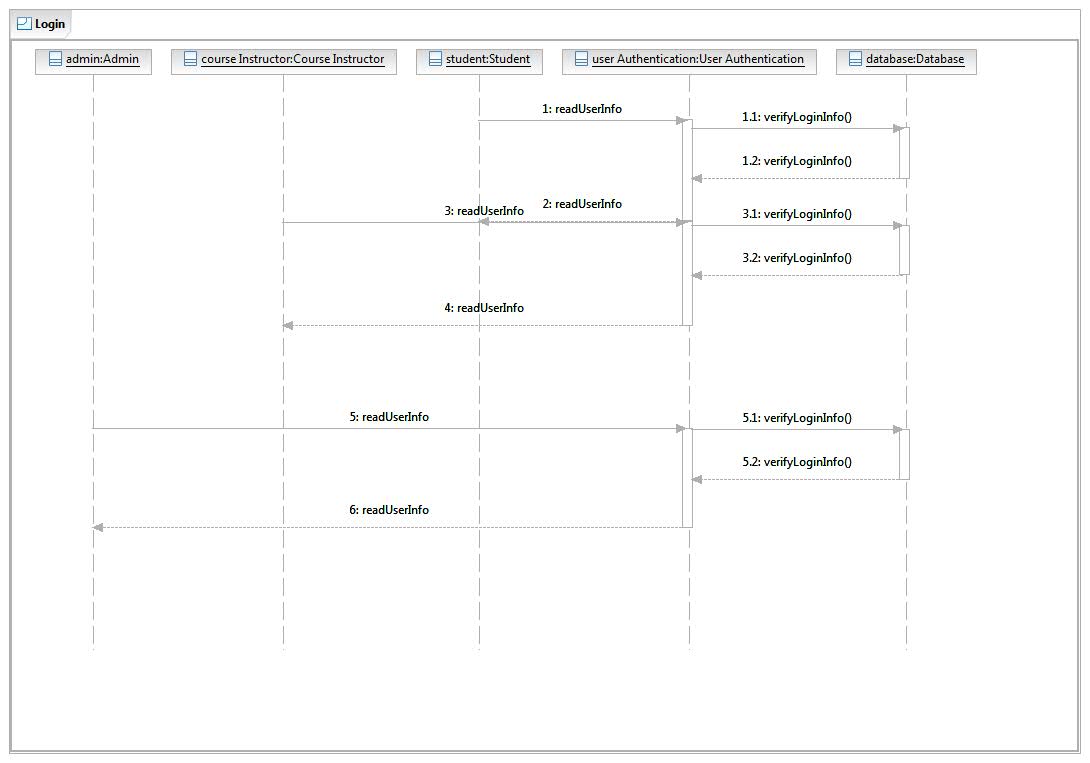
**Fig: 2.2.1(b) Sequence diagram for Generate QR code use**

### 2.2.2

#### Use case: Login



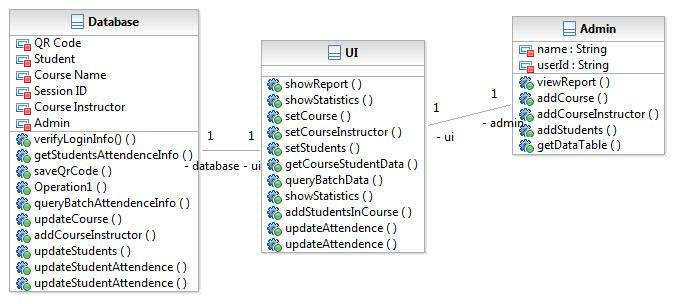
**Fig: 2.2.2(a) Class diagram for Login use case**



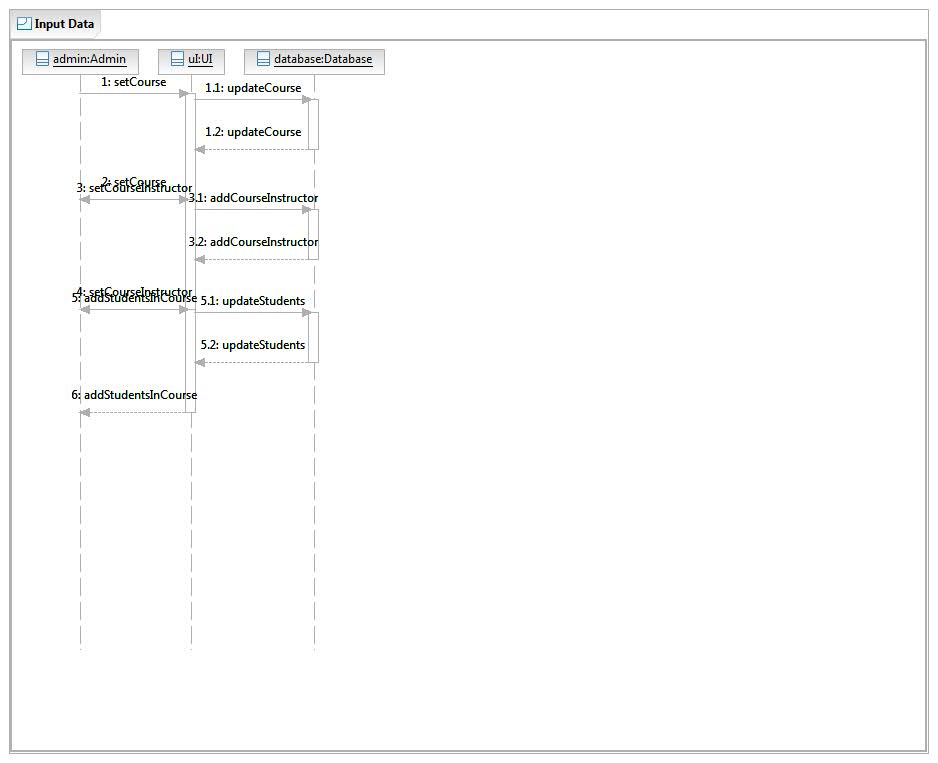
**Fig: 2.2.2(b) Sequence diagram for Login use case**

### 2.2.3

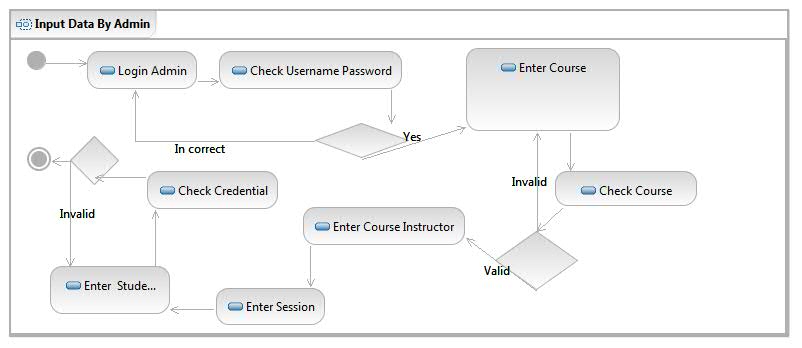
#### Use case: Input Data



**Fig: 2.2.3(a) Class diagram for Input Data use case**



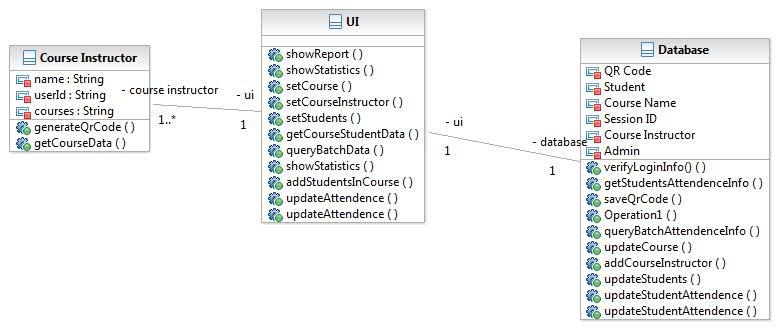
**Fig: 2.2.3(b) Sequence diagram for Input Data use case**



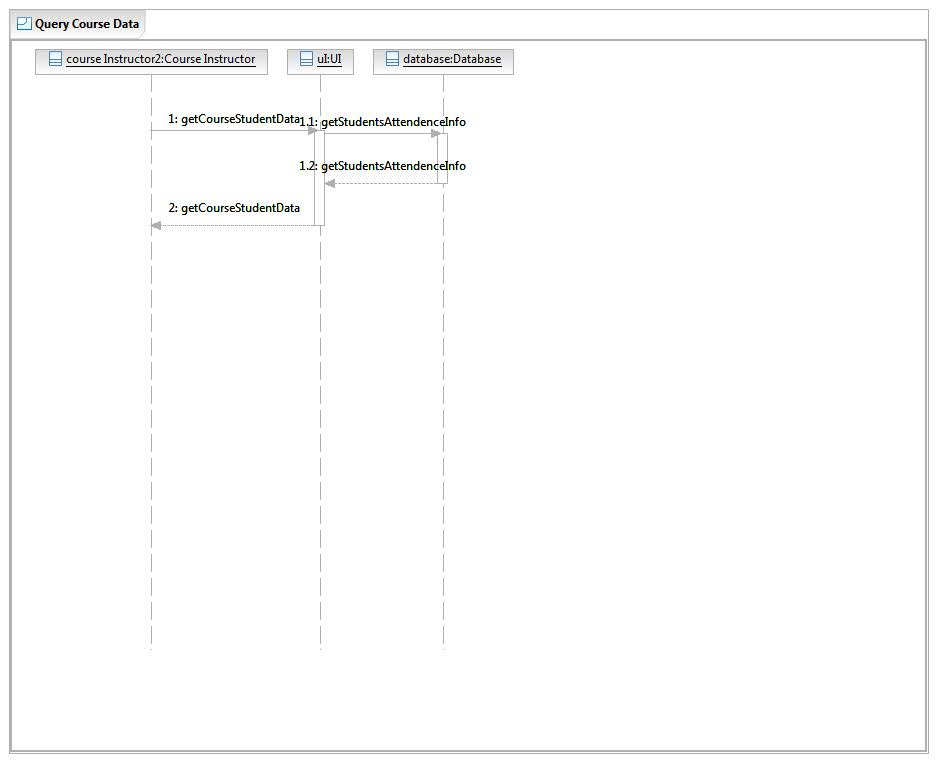
**Fig: 2.2.3(c) Activity diagram for Input Data use case**

**2.2.4**

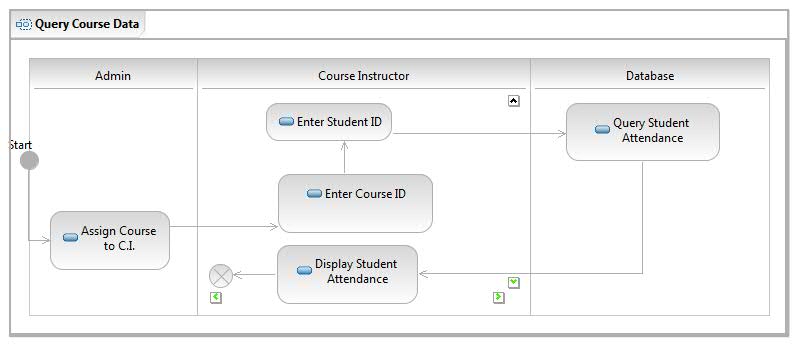
**Use Case : Query Course Data**



**Fig: 2.2.4(a) Class diagram for Query Course Data use case**

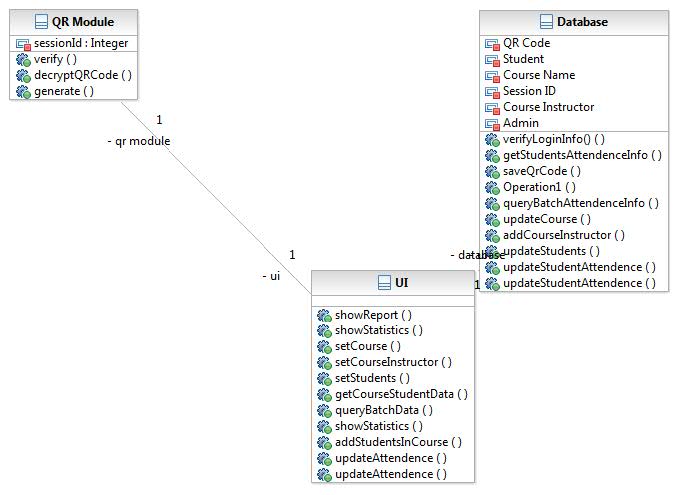


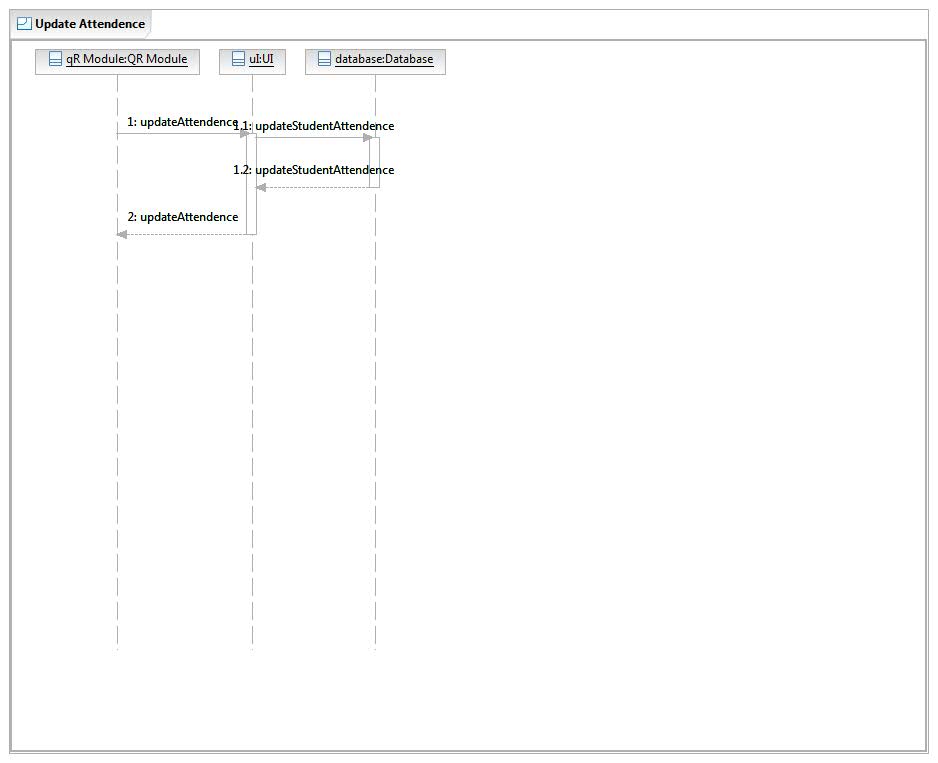
**Fig: 2.2.4(b) Sequence diagram for Query Course Data use case**



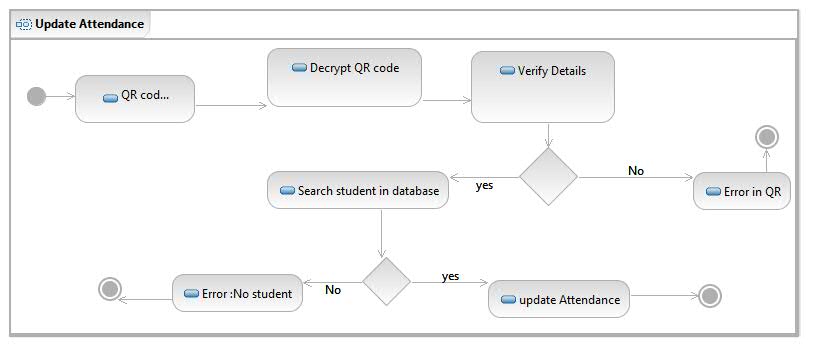
**Fig: 2.2.4(c) Activity diagram for Query Course Data use case**

**2.2.7**

**Use Case: Update Attendance** 

**Fig: 2.2.7(a) Class diagram for Update Attendance use case**

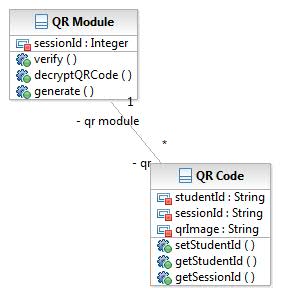
**Fig: 2.2.7(b) Sequence diagram for Update Attendance use case**



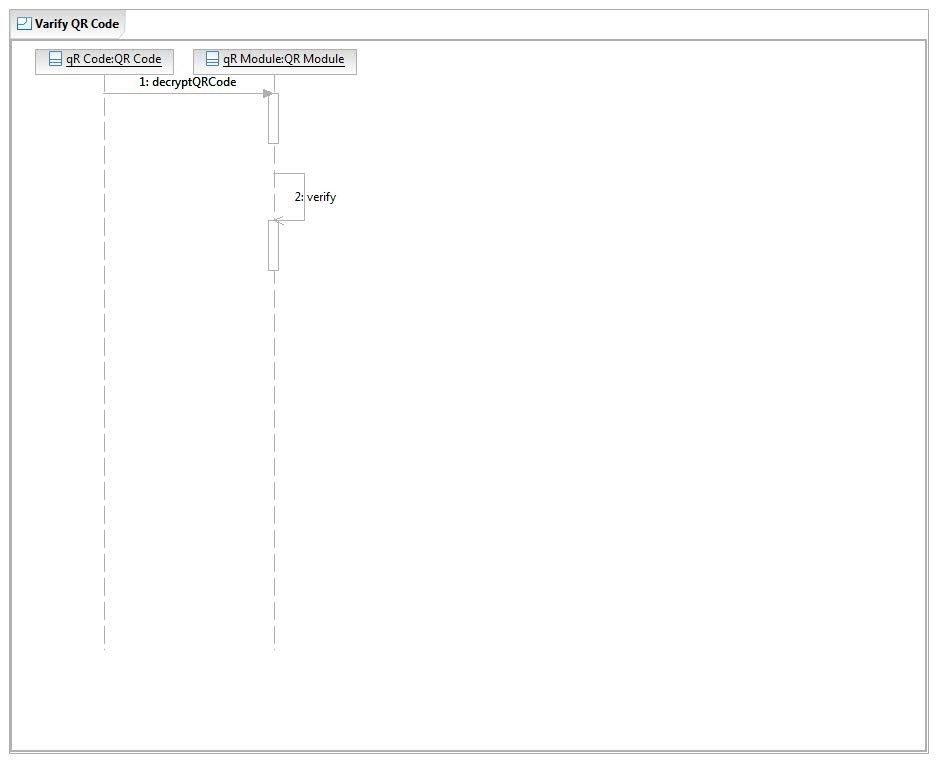
**Fig: 2.2.7(c) Activity diagram for Update Attendance use case**

**2.2.8**

**Use Case: Verify QR Code**



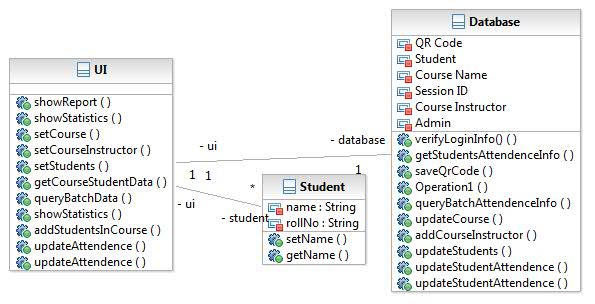
**Fig: 2.2.8(a) Class diagram for Verify QR Code use case**

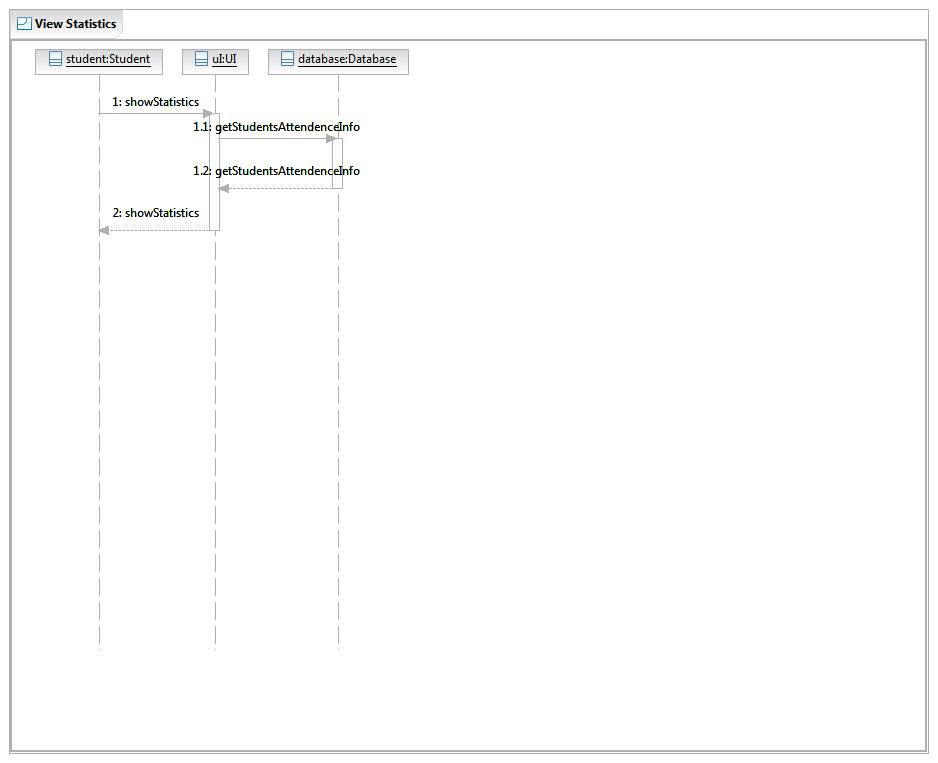


**Fig: 2.2.8(b) Sequence diagram for Verify QR Code use case**

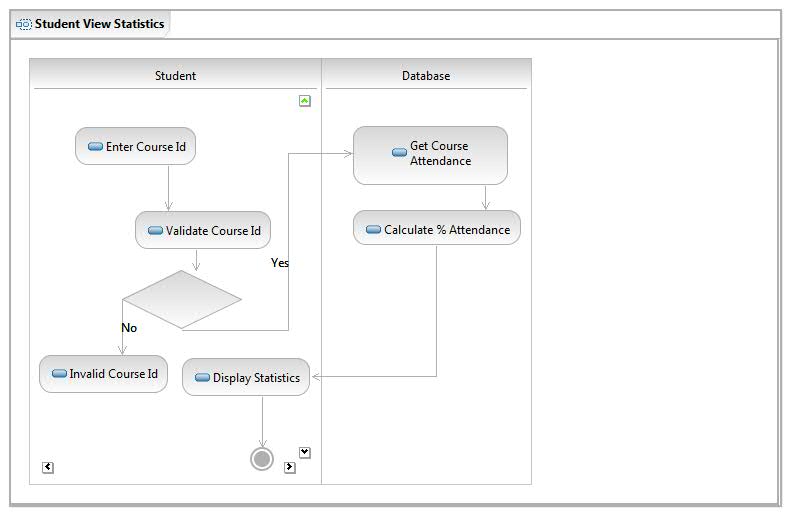
**2.2.9**

**Use Case : View Statistics**



**Fig: 2.2.9(a) Class diagram for View Statistics use case**

**Fig: 2.2.9(b) Sequence diagram for View Statistics use case**



**Fig: 2.2.9(c) Activity diagram for View Statistics use case**

## 2.3 User Characteristics

* The Course Instructor and Admin are expected to be Internet literate and be able to use a graphical software interface with menu, buttons and input information in a textbox. The course Instructor must have a laptop during the attendance taking phase - in order to generate QR code for the session. The Course Instructor must be able to query using GUI provided using student name, and course information.
* The admin must be able to input courses, enroll students and assign course instructor. The admin’s desktop must be connected to the database server.
* The Student is expected to use mobile device for capturing QR code and send it to the system over a network (LAN) to the system. He/She must be Windows literate and be able to use GUI provided by the system for tracking his/her attendance for a particular course.

## 2.4 Non-Functional Requirements:

**2.4.1 Performance**

Collection of attendance, its maintenance and analysis will be done. The performance characteristics are shown below. Two types of requirements

A. Hardware Requirements

As the application is hosted on web. The Web server must provide sufficient storage space for for the application. The

B. Dynamic Requirements

Response time of the system depends on the speed of the network and through which the system will access the database. Also, time dependency depends on the number and type of queries sent to the system. The Response time for query will thus be set according.

**2.4.2 Reliability**

In case of network failure (institute LAN) the database cannot be accessed and hence will not be updated. Also, students will not be be able to send the QR code if the network failure. In this case the system will notify the users.

**2.4.3 Availability**

The software will be made available to the students in departmental computers and to the authorized course instructors and college admin.

**2.4.4 Security**

The departmental computers ,the admin’s and instructors desktop will have the software installed. Only the administrator has right to assign permission like creating new courses, assign . Only authorized users can access the system with username and password.

**2.4.5 Maintainability** Backups for database are available.

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### 3.1.1 Generate QR Code

|  |  |
| --- | --- |
| **Use Case Name** | Generate QR Code |
| **Trigger** | Professor sends qr code generate event |
| **Precondition** | 1. Professor must logged in the system  2. QR Code should not be already generated |
| **Basic Path** | 1. Log in the system and go to respective course section  2. Fill in the course ID and date  3. The system generates a QR code based on the information.  4. System saves the course ID and date.  4. System displays the QR code on the Screen (connected to the projecting device. |
| **Alternative Paths** | No |
| **Postcondition** | QR Code generated and displayed on screen |
| **Exception Paths** | Course Instructor may abandon the session at any time. |
| **Other** | QR code contains Course code, date, lecture no. and other data in encrypted format. |

### 3.1.2 Student: View Statistics

|  |  |
| --- | --- |
| **Use Case Name** | Student:View Statistics |
| **Trigger** | Student sends a query for view statistics |
| **Precondition** | 1. Student must logged in the system  2. Student should be registered for the courses |
| **Basic Path** | 1. Log in the system  2. Now student can view percentage of attendance. If student wants complete detail he/she has to select course. |
| **Alternative Paths** | No |
| **Postcondition** | Student can see attendance details on the screen in tabular form and in the form of bar chart |
| **Exception Paths** | Student may abandon the session at any time. |
| **Other** | - |

### 3.1.3 Admin: Generate Report

|  |  |
| --- | --- |
| **Use Case Name** | Admin: Generate Report |
| **Trigger** | Admin sends a query for view statistics |
| **Precondition** | 1. Admin must logged in the system |
| **Basic Path** | 1. Log in the system  2. Go to statistics section  3. Select course for which admin wants to see statistics |
| **Alternative Paths** | No |
| **Postcondition** | 1. Admin can see attendance details on the screen.  2. |
| **Exception Paths** | Admin may quit the operation at any time. |
| **Other** | - |

### 

### 3.1.4 Input Data

|  |  |
| --- | --- |
| **Use Case Name** | Admin:Input Data |
| **Trigger** | Admin inputs data for adding courses, students into courses, and lectures hours. |
| **Precondition** | 1. Admin must logged in the system.  2. Name of the offered courses should be available  3. Details of registered students should be available for every course |
| **Basic Path** | 1. Log in the system  2. Go to Edit information section and selection add courses for adding course. |
| **Alternative Paths** | In step 2 If admin selects to add student  3. If admin wants to add students to course then he has to select course and add student. |
| **Postcondition** | Student can see attendance details on the screen. |
| **Exception Paths** | Admin may abandon the operation at any time, before it is submitted to the database. |
| **Other** | - |

### 

### 3.1.5 Query Student Information

|  |  |
| --- | --- |
| **Use Case Name** | Query Student Information |
| **Trigger** | Course Instructor or Admin explicitly asks for information for a student |
| **Precondition** | 1. Course instructor or Admin is querying for a course he is assigned to. 2. The student is enrolled for the particular course |
| **Basic Path** | 1. Course instructor or Admin logs into his account 2. Selects the desired course 3. Selects view student information option 4. Specifies the particular student for the report |
| **Alternative Paths** | 1. Course instructor or Admin logs into his account 2. Selects the desired course 3. Selects view students option 4. Selects view information option next to the student name |
| **Postcondition** | 1. A line/bar graph is displayed showing attendance for Attendance vs course for the student 2. Data is displayed in a tuple with number of records equal to the number of days of the class |
| **Exception Paths** | Network Connection Error or Network Failure, Server Faillure |
| **Other** | - |

### 

### 3.1.6 Login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Trigger** | User clicks login button |
| **Precondition** | 1. System is on  2. User inserted username and password. |
| **Basic Path** | 1. Input username and password  2. Select Role  3. Click login button |
| **Alternative Paths** | - |
| **Postcondition** | User logged into system |
| **Exception Paths** | User may abandon the operation at any time. |
| **Other** | - |

### 

### 3.1.7 Verify QR Code

|  |  |
| --- | --- |
| **Use Case Name** | Verify QR Code |
| **Trigger** | Student sends QR Code |
| **Precondition** | 1. System is on  2. QR Code received successfully  3. QR code is present in system |
| **Basic Path** | 1. Compare QR Code received from student to QR Code generated by Course Instructor |
| **Alternative Paths** | - |
| **Postcondition** | If QR Code matches then forward status as present to the attendance update module with student and course details.If QR Code does not match then forward status as absent to the attendance update module with student and course details. |
| **Exception Paths** | User may abandon the operation at any time. |
| **Other** | - |

### 

### 3.1.8 Update Attendance

|  |  |
| --- | --- |
| **Use Case Name** | Update Attendance |
| **Trigger** | Receives Attendance status from Verify QR Code module |
| **Precondition** | 1. System is on  2. Attendance status received successfully |
| **Basic Path** | 1. Go to respective course database.  2. Find Student in database  3. Update Attendance in database |
| **Alternative Paths** | - |
| **Postcondition** | Attendance is marked in database |
| **Exception Paths** | - |
| **Other** | - |

### 

## 3.3 Detailed Non-Functional Requirements

### ***3.4 Logical Structure of the Data***

<< Keep this blank for the time being>>

# 4.0 Supporting information

## 4.1 Table of contents and index

## 4.2 Appendixes