# Functional Specifications

The overall system looks like so having three partitions – three subsystems so to say.

1. Mobile application to work on iOS devices.

2. Web application to work on desktop.

3. Data Manager to control and secure the interactions with the database.

Web Application

Instructor and Admin

Data Manager

Mobile Application

Student

Database

Figure 1: Partition based system.

Three kinds of users work on this System:

1. Students,
2. Instructors,
3. Administrator.

Each of these three types of users has different use of the system so each of them has their own requirements. There is an overlap of permissions and level of access rather than different set of access levels for each user.

**User Class 1 - Student  
Functional requirement 1.1  
ID: R24**TITLE: Student registration - iOS mobile application  
DESC: After the student downloads the application, he/she is able to register through the iOS mobile application. The user must provide user-name as email address and password. The user can choose to provide a regularly used phone number.

* **public void setEmail(String email);**
* **public String getEmail();**
* **public void setPhoneNo(String phone);**
* **public String getPhoneNo();**

**User Class 1 - Student  
Functional requirement 1.3  
ID: R25**TITLE: Student Persistent Login – iOS mobile application

DESC: After the student logs in to the mobile application with his/her email ID and password, the login should persist such that the user need not log in repeatedly.

**User Class 1 - Student  
Functional requirement 1.4  
ID: R30**TITLE: Scan and Capture QR – iOS mobile application

DESC: The student should be able to capture the QR code displayed by the instructor on screen and capture it as soon as he scans it.

* **public double generateQR(int courseNo, String accessCode);**

**User Class 1 - Student  
Functional requirement 1.6  
ID: R33**TITLE: View Courses – iOS mobile application

DESC: The mobile application shall let the student view the courses he/she has registered to in the department. This view will be available when user clicks on 'View Courses' menu. The view populates with data from the course table and student table.

* **public String getCourseDetails(double Student ID);**

**User Class 1 - Student  
Functional requirement 1.7  
ID: R34**TITLE: View Attendance Percentage – iOS mobile application

DESC: The mobile application shall let the student view his/her current attendance percentage. This view will be visible only when the student clicks on View 'Attendance percentage' menu. The view populates with data from the attendance table and student table.

* **public double getAttendancePercentage(int StudentID, int courseNo);**

**User Class 2 - Instructor  
Functional requirement 2.1  
ID: R36**TITLE: Login – Web application

DESC: The web application shall let the Instructor login to the system with given user id and password.

1. User ID Requirements - Instructor must have a unique user ID of min 4 character length. Should exclude special characters and should not begin with a digit.

2. Password requirements - Must be 8-character length, should allow special characters like!@#$ only and must have a Capital letter, a special character and a digit.

3. Encryption - All communications with external systems have to be encrypted using a hash function.

4. All data Username and password combination will be stored into a database for future reference.

* **public void loginInstructor(String email, String password);**

**User Class 2 - Instructor  
Functional requirement 2.2  
ID: R37**TITLE: Login successful– Web application

DESC: On successful login, the timestamp is captured and stored in the database, the Instructor should be redirected to a 'Instructor View' page, where admin will be able to view courses, students assigned to that course and will be able to generate QR code if necessary.

**User Class 2 - Instructor  
Functional requirement 2.3  
ID: R39**TITLE: Add Course– Web application

DESC: The stand-alone application shall let the Instructor to add courses across departments. The instructor should be able to select and add courses from the 'Instructor view' page and the changes will be saved and preserved in the instructor database.

* **public void addCourse(String CRN);**

**User Class 2 - Instructor  
Functional requirement 2.3  
ID: R43**TITLE: Display QR– Web application

DESC: The stand-alone application shall let the Instructor display that unique QR code to the students in the class. This 'displayQR' page will have a button saying 'Generate another QR code’, which will redirect the instructor to 'generateqr' page.