Android App on

**NITC CMS**

Class Management System

Software Requirements Specification

Version 1.0

Abhinaba Audhya (M120360CA) Leader

Gagandeep Kaur (M120388CA)

Jay Shankar Yadav(M120387CA)

Sourav Das (M120358CA)

Vikash Kumar (M120354CA)

In partial fulfillment of

Object Oriented Programming Project

Instructor: Mr. Kranti Kumar

2013-2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 20.01.2014 | Version 1 of NITC CMS SRS | Abhinaba Audhya (Leader) | 1st version of the SRS with the detailed plan, basic GUI interfaces and a database outline. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction................................................................................................. 1

1.1 Purpose.............................................................................................................. 1

1.2 Intended Audience .............................................................................................. 1

1.3 Scope …………………………………………………………....................................................... 1

2. General Description..............................................................................................2

2.1 Product Perspective............................................................................................ 2

2.2 Product Functions ................................................................................................ 3

2.3 User Classes and Characteristics ......................................................................... 5

2.4 Operating Environment ....................................................................................... 5

2.5 Design and Implementation Constraints ............................................................. 6

2.6 Assumptions and Dependencies........................................................................... 6

3. System Features.................................................................................................. 7

Core Features .......................................................................................................... 7

3.1 User Login…………………………………........................................................................ 7

3.2 User Welcome ……………………………...................................................................... 7

3.3 Subjects ………....................................................................................................... 8

3.4 Student List .......................................................................................................... 9

3.5 Student Details .................................................................................................... 10

3.6 Attendance ……….................................................................................................. 10

3.7 Settings Menu ……................................................................................................. 11

Additional Features ................................................................................................... 12

3.8 Student Login ………………………….............................................................................12

3.12 Courses ................................................................................................................12

3.13 Student–Course Record .......................................................................................13

4. External Interface Requirements ......................................................................... 14

4.1 User Interface ...................................................................................................... 14

4.2 Hardware Interfaces............................................................................................ 20

4.3 Software Interfaces ……………………………………………………………………………………….. 20

5. Non-Functional Requirements .............................................................................. 21

5.1 Performance Requirements ............................................................................... 21

5.2 Safety Requirements .......................................................................................... 21

5.3 Security Requirements ....................................................................................... 21

5.4 Software Quality Attributes ............................................................................... 21

6. Other Requirements……………………………………….………………………….22

# 1. INTRODUCTION

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the Class Management System application on the Android platform. This document will cover each of the system’s intended features. It will also cover hardware, software and other technical dependencies.

**1.2 Intended Audience**

This document is intended for all individuals participating in and/or supervising the Class Management System. Readers interested in a brief overview of the product should focus on the rest of Part 1 (Introduction), as well as Part 2 of the document (Overall Description), which provide a brief overview of each aspect of the project as a whole.

Readers who wish to explore the features of Class Management System in more detail should read on to Part 3 (System Features), which expands upon the information laid out in the main overview.

Readers interested in the non-technical aspects of the project should read Part 5, which covers performance, safety, security, and various other attributes that will be important to users.

## 1.3 Scope

The Class Management System is an Android application built to help the faculty of NITC to manage the classes in a better way. It is composed of two main components: a client-side application which will run on Android handsets of the faculty, and a server-side application which will support and interact with various client-side features. The system is designed to facilitate the process of viewing student details, taking attendance and many other features right from their mobile devices.

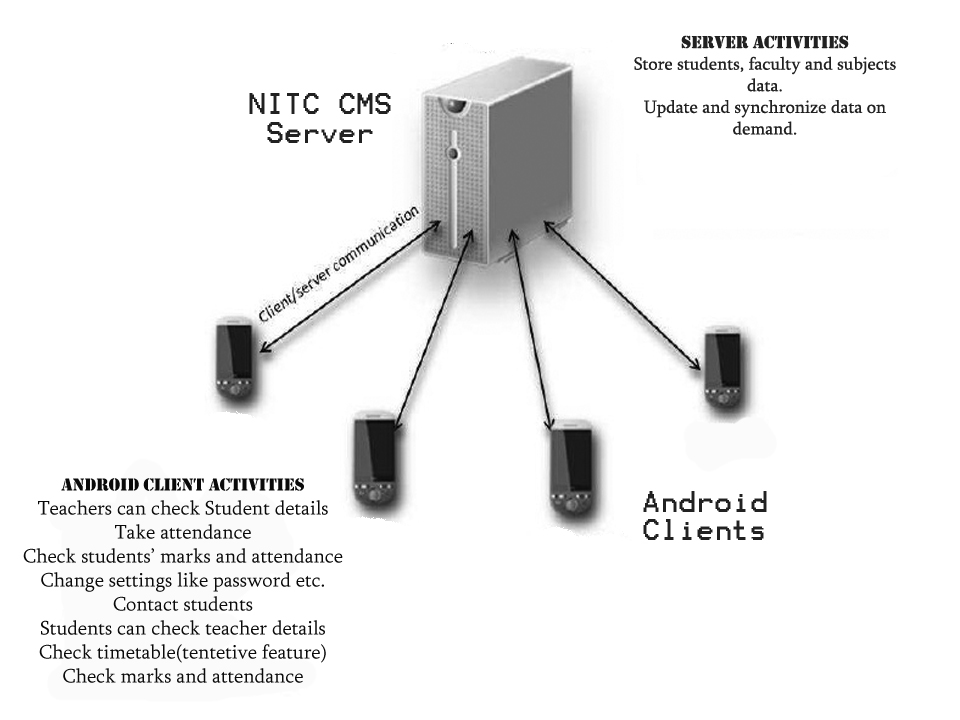
## 

# 2. GENERAL DESCRIPTION

## 2.1 Product Perspective

The Class Management System is a new, self-contained program intended for use on the Android platform. While the mobile application is the main focus of the project, the server side is equally important. The server side is responsible for the database and the synchronizing of the data and the services. The scope of this project encompasses both the server and the client sides, so both the aspects are covered in details in this document.

Below is a diagram of the Class Management System which illustrates the interaction between the server and the client sides.



## 

## 2.2 Product Functions

The following list offers a brief outline and description of the main features and functionalities of the Class Management System. The features are split into two major categories: core features and additional features. Core features are essential to the application’s operation, whereas additional features simply add new functionalities. The latter features will only be implemented as time permits.

**Core Features**

1. User(Faculty) Login :

* Enables the user to login to Class Management System.

1. User Welcome :

* Appears when user login to his/her account.
* Enables the user to choose options such as viewing the list of batches, taking attendance, etc.
* Enables the user to customize his/her account settings.

1. Batches :

* Appears when user clicks on “Check Student Details” button on “Welcome” page.
* Enables the user to keep track of all the batches (classes) taken by him in the current semester.
* Allows the user to choose any batch either to take attendance or to view student details.

1. Student List :

* Enables the user to view the list of all the students in a batch.

1. Student Details :

* Includes the general details of the student such as roll number, mobile number, email, etc.
* Keeps record of marks obtained by student in mid-term exams, quizzes and assignments.
* Enables the user to keep track of classes missed by a student in his/her course.

1. Attendance :

* Allows the user to take attendance from his mobile device.
* Updates number of classes taken by user for a batch till date in the semester.

1. Settings Menu :

* Allows the user to customize his/her preferences.
* Can be accessed at any time using built-in Settings button on Android phone.

**Additional Features**

1. Student Login :

* Allows the student to login to Class Management System.

1. Courses :

* Contains the list of all the courses taken by the user in current semester.

1. Student-Course Details :

* Enables the user to view the marks obtained by him/her in selected course.
* Keeps track of number of classes missed by the user in the course so far.
* Also keeps record of number of classes conducted for that course till date.

1. Settings Menu :

* Allows the user to customize his/her preferences.
* Can be accessed at any time using built-in Settings button on Android phone.

For more detailed information, see Part 3 of the document (System Features).

## 2.3 User Characteristics

The Class Management System Android application is meant to provide the faculty of NIT Calicut a better and easier way to manage the classes from their mobile devices. It will reduce the paperwork and data entry job at the end of each semester. The interface is as simple as possible, and the application itself has no learning curve.

Users can be defined by how they will use the product in a particular situation. The following list categorizes the scenarios in which Class Management System is expected to be utilized by the faculty:

1. Quickly check a student’s details (like mobile number or email) in class or from anywhere.
2. Quickly check the number of days a student has been absent, and warn him if he/she is beyond the 80% rule.
3. Quickly check a student’s marks in that subject, and notify him.
4. Take a note of the students absent in class, and update it directly to the database.

These above scenarios can be supported by the core features of the application. There can be some additional scenarios depending on the implementation of the additional features:

1. A student may check the number of days he has been absent for a particular subject, at any time.
2. A student may check his marks in any subject, whenever he needs to check it.

**2.4 Operating Environment**

The main component of the Class Management System project is the software application, which will be limited to the Android operating system (specifically Android 2.3 and above). The application is not resource or graphics-intensive, so there are no practical hardware constraints. The app will rely on several functionalities built into Android’s Application Programming Interface (API), so ensuring appropriate usage of the API will be a major concern. Beyond that, the application is a self-contained unit and will not rely on any other Android-related software components.

The application will, however, frequently interact with the Class Management System server, a virtual dedicated server hosted by GoDaddy.com. The server operates on a Linux platform with 1GB of RAM and 100GB of allocated storage space. The Class Management System database will be stored on the server using MySQL and will be interfaced with a wrapper written in PHP 5.

## 2.5 Design and Implementation Constraints

The primary design constraint is the mobile platform. Since the application is designated for mobile handsets and tablets, limited screen size and resolution will be a major design consideration. Creating a user interface which is both effective and easily navigable will pose a difficult challenge. Other constraints such as limited memory and processing power are also worth considering. The Class Management System is meant to be quick and responsive, even when dealing with large groups of data and transactions, so each feature must be designed and implemented with efficiency in mind, reducing the overall response time of the application.

Another main constraint is the availability and speed of data connection on the mobile device. Since the application has to send requests to the server for each and every task, having a good data connection all the time is necessary.

## 2.6 Assumptions and Dependencies

**TIME DEPENDENCIES**

As mentioned previously, the features of Class Management System are divided into two groups: core features and additional features. Core features are crucial to the basic functionality of the application. These features must all be implemented in order for the application to be useful.

Additional (optional) features, however, are *not* critical to the function of the application. They are usability improvements and convenience enhancements that may be added after the core features have been developed. Thus, the implementation of these features is entirely dependent upon the time spent designing and implementing the core features. The final decision on whether or not to implement these features will be made during the later stages of the design phase.

# 3. SYSTEM FEATURES

**3.1 User Login**

After the application is installed, the user can login with the User Id and password provided to each individual by the Institute. Name, User Id and Password of the user, for all the users will already be stored in the database. So, a user doesn’t have to register. The Class Management System server will store an account for each user, enabling all of the application’s synchronization capabilities.

**3.1.1 STIMULUS/RESPONSE SEQUENCES**

**Step 1** Class Management System launched from the Android home screen.

**Step 2**  The user is prompted to enter User Id and Password and then press Login button.

**Step 3** This information is sent to server and verified with the details stored in the database.

**Step 4**  If the details are verified, then user is authenticated and is taken to “Welcome” page.

**3.1.2 USER REQUIREMENTS**

* The user requires valid User Id and Password provided by the Institute to login to Class Management System.
* Internet Connection is required.

**3.1.3 SYSTEM REQUIREMENTS**

**Secure Database System**

The application must ensure that the information stored in database should be safely stored.

**3.2 User Welcome**

The Welcome screen will be the first screen that will appear after the successful login. This screen will display “Welcome <Username>”. Also there will be options to check student details and to take Attendance. This screen will also provide the option to change settings. The user can either choose to view the details of students, or can take attendance for the class.

**3.2.1 STIMULUS/RESPONSE SEQUENCES: “Check Student Details”**

**Step 1** The user selects “Check Student Details” from the Welcome screen.

**Step 2**  A screen appears containing the list of subjects that the particular faculty(user) takes during the current semester.

**3.2.1 STIMULUS/RESPONSE SEQUENCES: “Take Attendance”**

**Step 1** The user selects “Take Attendance” from the Welcome Screen.

**Step 2** A screen appears containing the list of subjects that he/she takes during current semester.

**Step 3** The user can select from the list of subjects, the subject for which he/she wants to take attendance.

**3.2.3 USER REQUIREMENTS**

* The User must have a valid login Id and Password in order to use and obtain full functionality of the application.
* Internet Connection is required.

**3.2.4 SYSTEM REQUIREMENTS**

The application must be able to retrieve data (or lists) from the server within a fraction of seconds.

**3.3 Subjects**

This screen will show the list of subjects taught by the user(faculty) in a given semester. Selecting a subject can perform two different functions, depending upon the previous selected option. If user (faculty) selects “Check Student Details“ then list of subjects taken by that user(faculty) will appear, and after selecting a subject from there, list of students taking that particular subject will appear. Whereas if the user will select “Take Attendance” then again a list of subjects will appear and after selecting a subject from there, list of students appears with the checkboxes in order to mark the list of absent students.

**3.3.1 STIMULUS/RESPONSE SEQUENCES:”Select any subject”**

**Step 1** The user is presented with a screen containing list of subjects. The user can select the required subject.

**Step 2** After that a screen will appear will depending on the previously selected option.

* The screen containing the list of students taking the selected subject will appear, if last selected option was “Check Student Details”.
* The screen containing the list of students taking the selected subject will appear along with the check boxes to mark the absent students, if last selected option was “Take Attendance”.

**3.3.2 USER REQUIREMENTS:**

Only an active internet connection is required.

**3.3.3 SYSTEM REQUIREMENTS:**

The application must be able to retrieve the list of subjects from the server within fractions of second.

**3.4 Student List**

This screen will contain the list of students of the selected subject. The list can be used either to view the details of a selected student or to take attendance.

**3.4.1 STIMULUS/RESPONSE SEQUENCES: “To check details of student”**

**Step 1:** The user is presented with a screen containing list of students. The user can select a student.

**Step 2:** The details of the selected student appears on the screen.

**3.4.2 STIMULUS/RESPONSE SEQUENCES: “For taking attendance”**

**Step 1:** The user is presented with a screen containing list of students with check box to mark the student if he/she is absent.

**Step 2:** The user can mark the list of absent students on the page and submit.

**3.4.3 USER REQUIREMENTS:**

Only Internet Connection is required.

**3.4.4 SYSTEM REQUIREMENTS:**

The application must be able to retrieve the details of selected student correctly from server within fractions of seconds.

* 1. **Student Details**

After a student is selected to view his/her details, screen appears containing details of the student. The screen will contain Name (of student), Roll No, Email, Mobile No. , marks obtained in mid-term exams and quizzes and number of classes missed in the course taken by that user (faculty).

* + 1. **STIMULUS/RESPONSE SEQUENCES: “Check Student Details”**

**Step 1:** Select a student to view the details.

**Step 2:** Screen appears containing the details of the selected student.

* + 1. **USER REQUIREMENTS:**

Only internet connection is needed.

* + 1. **SYSTEM REQUIREMENTS:**
* The application must be able to retrieve correct information from the server and within fractions of second.
* The server should contain updated marks of all the mid-term exams and quizzes taken so far.
* The number of classes missed by the student in a course should also be up-to-date.
  1. **Attendance**

The application enables the user to take attendance for the classes he/she teaches right from his/her mobile. This is a very important feature that enables user to keep track of the attendance of the students at any time. In order to take attendance the user can mark all the absent students and submit.

* + 1. **STIMULUS/RESPONSE SEQUENCES: “Take Attendance”**

**Step 1:** The user will login with his User ID and Password.

**Step 2:** After successful login, “Welcome” screen appears. Select “Take Attendance” from that screen.

**Step 3:** A screen with list of batches taken by the user will appear. Select the batch for which attendance is to be taken.

**Step 4:** A screen with list of students of the selected batch will appear along with the checkboxes.

**Step 5:** Mark the absent students and press “Submit”.

**Step 6:** Attendance is successfully submitted.

* + 1. **USER REQUIREMENTS:**

Only Internet Connection is required.

* + 1. **SYSTEM REQUIREMENTS:**
* The application should be able to retrieve data from server within fractions of seconds.
* After pressing “Submit”, the total number of classes missed by the students in that course should be successfully updated.
* All the changes made should leave the database in consistent state.
  1. **Settings Menu**

This menu allows the user to modify more advanced settings within the application. The menu is accessed from “Welcome Screen”, a hardware button built into all Android handsets. The application allows the user to change his/her password.

**3.7.1 STIMULUS/RESPONSE SEQUENCES:**

**Step 1** From any menu, the user may press the Settings button

* This is a physical button featured on all Android phones

**Step 2** From the menu, the user may select to modify his/her password.

**Step 3** After making the desired changes, the user may save his/her preferences and exit the menu, or simply exit without saving.

**Step 4** The user is then returned to the screen he/she was on before accessing the Settings menu.

**3.7.2 USER REQUIREMENTS:**

Only Internet Connection is required.

**3.7.3 SYSTEM REQUIREMENTS:**

The changes made should be successfully updated in the database.

**ADDITIONAL FEATURES**

* 1. **Student Login**

After the application is installed, the student can login with the ID(that will be Roll No) and password provided to each individual by the Institute. Name, Roll No and Password of the Students will be already stored in the database. So, student doesn’t has to register. The Class Management System server will store an account for each student, enabling all of the application’s synchronization capabilities.

**3.8.1 STIMULUS/RESPONSE SEQUENCES**

**Step 1** Class Management System launched from the Android home screen.

**Step 2**  The student is prompted to enter ID(Roll No) and Password and then press Login button.

**Step 3** This information is sent to server and verified with the details stored in the database.

**Step 4**  If the details are verified, then user is authenticated and is taken to “Course” page.

**3.8.2 USER REQUIREMENTS**

* The student requires valid User Id and Password provided by the Institute to login to Class Management System.
* Internet Connection is required.

**3.8.3 SYSTEM REQUIREMENTS**

**Secure Database System**

The application must ensure that the information stored in database should be safely stored.

* 1. **Courses**

The screen containing the list of courses that student has registered in the current semester will be the first screen that will appear after the successful login. After selecting a particular course student can view details regarding that course such as marks obtained in mid-term exams, quizzes and no. of classes missed so far in that course.

**3.9.1 STIMULUS/RESPONSE SEQUENCES:**

**Step 1** The student will login.

**Step 2**  Screen appears containing the list of courses, for which student has registered in the current semester.

**Step 3** The student can select a course to view his record in that course.

**3.9.3 USER REQUIREMENTS**

* The student login in order to obtain full functionality of the application.
* Internet Connection is required.

**3.9.4 SYSTEM REQUIREMENTS**

The application must be able to retrieve data (or lists) from the server within fractions of seconds.

* 1. **Student-Course Record**

After the student selects to view his/her record in a course, screen appears containing record of the student in that course. The screen will contain Course code, marks obtained in mid-term exams and quizzes and number of classes missed so far in that course.

* + 1. **STIMULUS/RESPONSE SEQUENCES: “Check Student-Course Record”**

**Step 1:** Select a course to view his/her record in that course.

**Step 2:** Screen appears containing the record of the selected in that course.

* + 1. **USER REQUIREMENTS:**

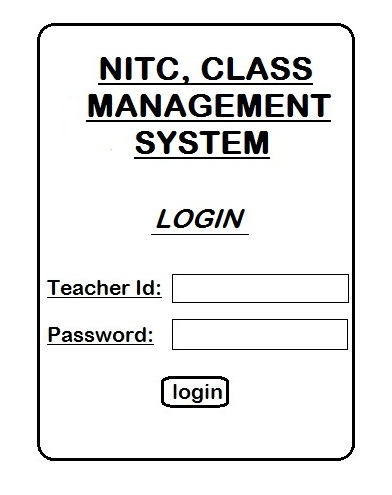
Only internet connection is needed.

* + 1. **SYSTEM REQUIREMENTS:**
* The application must be able to retrieve correct information from the server and within fractions of second.
* The server should contain updated marks of all the mid-term exams and quizzes taken so far.
* The number of classes missed by the student in a course should also be up-to-date.

**4. EXTERNAL INTERFACE REQUIREMENTS**

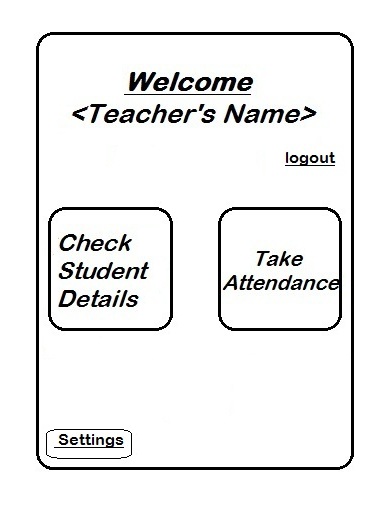
## 4.1 User Interface (Tentative design of the core features)

**4.1.1 Login Screen**



* Used by the user (faculty) to login to the system.
* Prompts the user to provide his/her valid login credentials and validates it against the data stored on the server.
* On successful login, will lead to the “Welcome” screen.

**4.1.2 Welcome Screen**

****

* Displays a welcome message along with the user’s name.
* Displays the options available to the user -> “Check Student Details” or “take Attendance”.
* Has a “logout” button in case the user wants to logout.
* Has a settings button by which the user can access the different settings like changing the password etc.

## 4.1.3 Subjects List Screen

## D:\college\s4\oops\project\oops srs\figure3.jpg

## Displays the list of subjects taken by the logged in user (faculty).

* This screen is displayed when the user selects either of the two options -> “Check Student Details” or “Take Attendance”.
* Depending on the previously selected option, this screen will lead to different screens on selecting a particular subject.
* If the previously selected option was “Check Student Details”, then this screen will lead to the “Student Details” screen.
* Else if the previously selected option was “Take Attendance”, then this screen will lead to the “Take Attendance” screen.

## 4.1.4 Student Details Screen

## D:\college\s4\oops\project\oops srs\figure4.jpg

## Displays the list of students taking the selected subject taught by the logged in user (faculty).

* Selecting a particular student will lead to the “Student Details” screen displaying the details of that student.

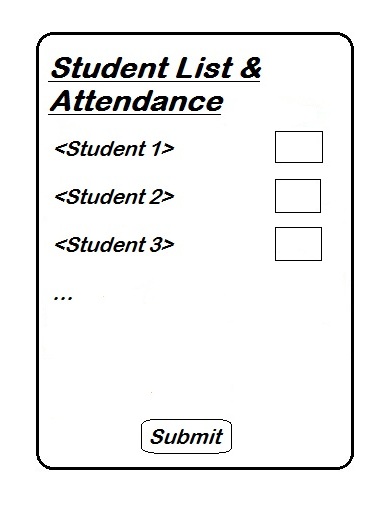
## 4.1.5 Student Details Screen

## D:\college\s4\oops\project\oops srs\figure5.jpg

## Displays the details of the selected student.

* Details include name, roll number, email id, mobile number, number of days absent, marks in different exams.

**4.1.6 Take Attendance Screen**

****

* Displays the list of students taking that subject, along with a check box beside their name.
* The user (faculty) can check each student who is absent.
* On submitting, the data will be updated on the server.

### 

### 4.2 Hardware Interfaces

CMS is intended as a mobile application for the Android platform and hence is solely

supported on Android-powered devices. Updates and data synced between Android devices are transmitted to and handled by the central server. CMS is being developed specifically for Android 2.3 (Gingerbread) and all versions released after it.

The Android platform supports push messages that will be used to synchronize data between the local application and the main application server. Information will be sent using TCP/IP and the HTTP protocol.

The Android platform provides abstractions for all network communication interfaces and thus the hardware as well.

**4.3 Software Interfaces**

The CMS is to be developed under the Android OS using the Java JDK (Java Development Kit) and the Android SDK (Software Development Kit).

### 5. NON-FUNCTIONAL REQUIREMENTS

### 5.1 Performance Requirements

Performance should not be an issue because all of our server queries involve small pieces of data. Changing screens will require very little computation and thus will occur very quickly. Server updates should only take a few seconds as long as the phone can maintain a steady signal. The time taken to view list of batches, list of students, to view details of student or to take attendance will also be very less.

### 5.2 Safety Requirements

Class Management System will not affect data stored outside of its servers nor will it affect any other applications installed on the user’s phone. It cannot cause any damage to the phone or its internal components.

**5.3 Security Requirements**

This application assumes that only the user or whoever he/she allows will have access to

his/her Android handset. With that being said, only User Id given by the institute is required to verify the identity of the user upon opening the app. It is well protected with password. There is no method to login without knowing the User Id and password.

**5.4 Software Quality Attributes**

The graphical user interface of CMS is to be designed with usability as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate. There will be notifications to inform users of updates (if any) from time to time.

Special care has been taken to ensure reliability and correctness. With CMS being ported solely for the Android platform, this software application has the advantage of being portable and convenient to use whenever and wherever.

### 6. OTHER REQUIREMENTS

A database for CMS calls for a server side implementation that holds information about the students, teachers and subjects as well as all the relationships involved. The database will be using MySQL. The following provides an example of information that may be stored in the database:

* **Student** : Roll, Firstname, Lastname, Email id, Mobile Number, Branch, Year
* **Teacher** : T\_ID, T\_Firstname, T\_Lastname, T\_Password, T\_Email, T\_Mobile
* **Subject** : Sub\_Code, Sub\_Name
* **Stud\_Sub** : Roll, Sub\_Code, Days\_Absent, T1\_Marks, T2\_Marks, Assign\_Marks, Endsem\_Marks
* **Teach\_Sub** : T\_ID, Sub\_Code

The server will be configured on a Linux platform, and through use of PHP will allow interaction and processing in conjunction with the database. Processes to be done on the server include: pushing, pulling and updating data.