

# Software Requirements Specification (SRS) Document

<Academic Administration Application, R.Vamshi Krishna and Sunil Kumar>

## Brief problem statement

Taking the student's attendance using pen, paper and entered into the systems is a boring and routine activity for faculty and admin people. To resolve this problem, we are going to make an android app for faculty, admin, students and also a portal for faculty. In addition to above mentioned problem, we can add some more features like announcements, displaying grades, time-table to the students by the faculty and admin.

## System requirements

1. PHP
2. Apache Server
3. MySQL Database

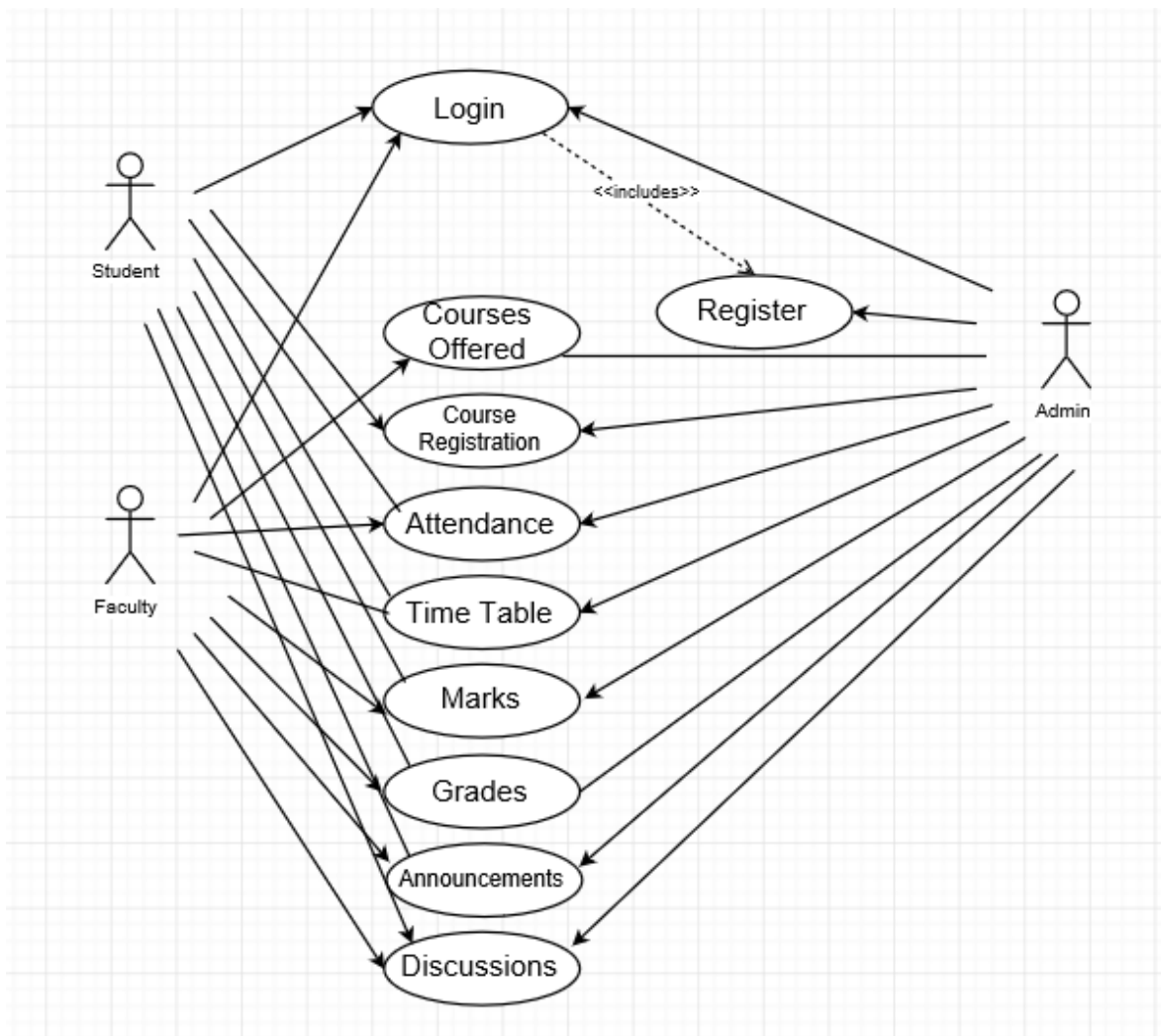
## Users profile

1. **Students** who are registered under 'Student Mode' can be used.
2. **Faculty** who are registered under 'Faculty Mode' can be used.
3. **Admin People** who are registered under 'Admin Mode' can be used.

## Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	User Registration	User must be registered as a student or faculty or admin with his/her true information.	R1
2.	User Login	After Registration, User can login to portal or into app	R1
3.	Courses Offered	Faculty can give the name of offered courses to admin	R1
4.	Course Registration	Admin can offer course registration to students. Students can be registered courses.	R1
5.	Attendance	Admin or Faculty can take attendance of students using android app. Students can see their attendance by registered courses wise.	R2
6.	Time Table	Admin updates the time table and students, faculty can see it.	R2
7.	Marks Entering	Admin can be entered students marks into server.	R2
8.	Grades	Faculty can give grade to each student. Admin and students can see them	R3
9.	Announcements	Faculty, Admin can make announcements like class postpone or prepone, Deadlines..... to students. Students can get such announcements as notifications.	R3
10.	Discussions	Discussions can be made between faculty and student, faculty and admin.	R3
11.			
12.			

## Use case diagram



\*Arrow indicates writable; Line indicates readable.

## Use case description

<b>Use Case Number:</b>	UC-01
<b>Use Case Name:</b>	Registration
<b>Overview:</b>	User need to register to login into the app or portal
<b>Actors:</b>	Faculty, Student, Admin
<b>Pre condition:</b>	User must be have valid information like valid emailed etc.
<b>Flow:</b>	Main (success) Flow: User presses the Submit button <ul style="list-style-type: none"><li>• User conformation through email authentication</li></ul>
<b>Post Condition:</b>	User get logged.

<b>Use Case Number:</b>	UC-02
<b>Use Case Name:</b>	Login
<b>Overview:</b>	After Registration, User can login to portal or into app
<b>Actors:</b>	Faculty, Student, Admin
<b>Pre condition:</b>	User must be registered before login
<b>Flow:</b>	Main (success) Flow: Entered into App or Portal
<b>Post Condition:</b>	User get logged.

<b>Use Case Number:</b>	UC-03
<b>Use Case Name:</b>	Courses Offered
<b>Overview:</b>	Faculty offers the list of courses which are going to teach.
<b>Actors:</b>	Faculty, Admin
<b>Pre condition:</b>	Faculty needed to be registered

<b>Flow:</b>	Main (success) Flow: Faculty presses the “SEND” button after giving the courses. <ul style="list-style-type: none"><li>• Admin got the list of courses which are offered by faculty</li></ul>
<b>Post Condition:</b>	Faculty getting offered courses.

<b>Use Case Number:</b>	UC-04
<b>Use Case Name:</b>	Course Registration
<b>Overview:</b>	Admin can offer course registration to students. Students can be registered courses.
<b>Actors:</b>	Student, Admin
<b>Pre condition:</b>	Student must be registered and Admin must be got the list of offered courses by faculty.
<b>Flow:</b>	Main (success) Flow: Student need to tick the courses in which he/she desires. <ul style="list-style-type: none"><li>• Press Submit button after chosen the courses</li><li>• Admin got the list of courses registered by student wise.</li></ul>
<b>Post Condition:</b>	Student get registered into courses.

<b>Use Case Number:</b>	UC-05
<b>Use Case Name:</b>	Attendance
<b>Overview:</b>	Admin or Faculty can take attendance of students using android app. Students can see their attendance by registered courses wise.
<b>Actors:</b>	Student, Admin, Faculty
<b>Pre condition:</b>	Faculty can take only the students who are registered his/her offered courses. Admin can take all courses students.
<b>Flow:</b>	Main (success) Flow: Admin/Faculty needs to tick the students who are attended the class. <ul style="list-style-type: none"><li>• Press Submit button after taken the attendance.</li><li>• Admin can send the attendance of each student by month wise.</li></ul>
<b>Post Condition:</b>	Attendance got counted by month wise.

<b>Use Case Number:</b>	UC-06
<b>Use Case Name:</b>	Time Table
<b>Overview:</b>	Admin can notify the time table to students and faculty.
<b>Actors:</b>	Student, Admin, Faculty
<b>Pre condition:</b>	Admin only can update the time table.
<b>Flow:</b>	<p>Main (success) Flow: After browsing the pdf of time table admin can be uploaded.</p> <ul style="list-style-type: none"> <li>• Students and faculty get notification on time table modification.</li> <li>• Students and Faculty can see the time table</li> </ul>
<b>Post Condition:</b>	Admin notified the time table modification

<b>Use Case Number:</b>	UC-07
<b>Use Case Name:</b>	Marks Entering
<b>Overview:</b>	Admin can be entered students marks into server and students, faculty can see the marks.
<b>Actors:</b>	Student, Admin, Faculty
<b>Pre condition:</b>	Admin, faculty only can update the marks
<b>Flow:</b>	<p>Main (success) Flow: Admin can enter the marks of students course wise.</p> <ul style="list-style-type: none"> <li>• Student and Faculty can see the students marks.</li> <li>• Faculty and Admin can update the marks.</li> </ul>
<b>Post Condition:</b>	Student marks can be entered course wise.