NORTHWEST MISSOURI STATE UNIVERSITY  
Student Attendance Tracking system

July 18, 2017

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| University | Northwest Missouri State University |  | Project Sponsor | Michael Oudshoorn |
| Project Name | Student Attendance Tracking System |  | Project Manager | Michael Oudshoorn |
| Client Name | Dr. Michael Oudshoorn |  | Team Name | Code Green |
| Term | Summer.  Graduate Directed Project. |  | Team Logo |  |
| Start Date | June 12th 2017 |  | Authors | Sirisha Vanamali  Rohith Babu Sadhu  Sri Sai Ram Kumar Mamidala  Vipul Reddy Madadi  Vamsi Devalla  Subba Reddy Pothireddy  Shankar Rao Vallapurapu |
| Last Updated On | 18th July 2017 |  | Version | 1.8 |

Contents

[Table of Figures 3](#_Toc488113272)

[Purpose 4](#_Toc488113273)

[Scope 4](#_Toc488113274)

[Glossary 4](#_Toc488113275)

[Basic Functional Requirements 5](#_Toc488113276)

[Detailed functional requirements list 6](#_Toc488113277)

[Non-Functional Requirements 17](#_Toc488113278)

[Static Requirements 17](#_Toc488113279)

[Dynamic Requirements 17](#_Toc488113280)

[I. Software system attributes: 17](#_Toc488113281)

[II. Design Constraints 18](#_Toc488113282)

[III. Logical Database Design 19](#_Toc488113283)

[System Core Features 20](#_Toc488113284)

[Additional Features 20](#_Toc488113285)

[Assumptions and Dependencies 21](#_Toc488113286)

[Use Case Diagram 21](#_Toc488113287)

[Module1: Admin use case diagram 22](#_Toc488113288)

[I. Admin: Login Successful 24](#_Toc488113289)

[II. Admin: Login Unsuccessful 24](#_Toc488113290)

[III. Admin: Manage Course 24](#_Toc488113291)

[IV. Admin: Manage Student 25](#_Toc488113292)

[V. Admin: Manage Instructor 25](#_Toc488113293)

[VI. Admin: Manage QR 26](#_Toc488113294)

[Module 2: Instructor Use case Diagram 26](#_Toc488113295)

[I. Instructor: Login Successful 27](#_Toc488113296)

[II. Instructor: Login Unsuccessful 28](#_Toc488113297)

[III. Instructor: Display QR code 28](#_Toc488113298)

[IV. Instructor: Manage Students 29](#_Toc488113299)

[V. Instructor: Manage Courses 29](#_Toc488113300)

[Module3: Student Use case Diagram 30](#_Toc488113301)

[I. Student: Login Successful 30](#_Toc488113302)

[II. Student: Login unsuccessful 31](#_Toc488113303)

[III. Student: Scanning QR 31](#_Toc488113304)

[IV. Student: View Course Details 32](#_Toc488113305)

[V. Student: Attendance Percentage and feedback 32](#_Toc488113306)

# 

# Table of Figures

[Figure1: Use case diagram for Admin 23](#_Toc488113240)

[Figure2: Admin Login Use case description 24](#_Toc488113241)

[Figure3: Admin Login Unsuccessful Use case description 24](#_Toc488113242)

[Figure4: Admin Manage Course Use case description 25](#_Toc488113243)

[Figure5: Admin Manage Student Use case description 25](#_Toc488113244)

[Figure6: Admin Manage Instructor Use case description 26](#_Toc488113245)

[Figure7: Admin Manage QR Use case description 26](#_Toc488113246)

[Figure8: Instructor Use case diagram 27](#_Toc488113247)

[Figure9: Instructor Login successful Use case description 28](#_Toc488113248)

[Figure10: Instructor Login Unsuccessful Use case description 28](#_Toc488113249)

[Figure11: Instructor Display QR code Use case description 28](#_Toc488113250)

[Figure12: Instructor Manage Students Use case description 29](#_Toc488113251)

[Figure13: Instructor Manage Courses Use case description 29](#_Toc488113252)

[Figure14: Student Use case Diagram 30](#_Toc488113253)

[Figure15: Student Login Successful Use case Description 31](#_Toc488113254)

[Figure16: Student Login Unsuccessful Use case Description 31](#_Toc488113255)

[Figure17: Student Scan QR Use case Description 32](#_Toc488113256)

[Figure18: Student View Course Details Use case Description 32](#_Toc488113257)

[Figure19: Student Attendance Percentage and Feedback Use case Description 33](#_Toc488113258)

Requirements Document

# Purpose

The purpose of developing attendance tracking system is to computerize the traditional way of taking attendance. Another purpose of this system is to reduce the burden of taking attendance for the instructor and also to eliminate duplicate data entry, errors in time and attendance entries. It improves visibility to track and manage student attendance.

# Scope

The scope of this project is to track the student attendance by generating QR code. To elaborate, we have provided more functionalities to the department head and instructor, they can view the student attendance as well as they can see the number of students present in class on a particular day with representation of bar graph, they can manage courses as per CRN and they can add a course likewise they can remove the course too. Students can view the attendance and we have limited the access for students like one user can only scan QR code with one mac address. This system is limited to the Northwest Missouri State University.

# Glossary

|  |  |
| --- | --- |
| **Key Word** | **Description** |
| QR | Random code generated by the Instructor using which the students scan to mark their attendance |
| Software Requirements Specification. | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| Database | Collection of all the information monitored by this system. |
|  |  |
|  |  |
|  |  |

# Basic Functional Requirements

“A functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.”

The main purpose of this project is to build an easy and a faster way to gain and track attendance of the students. The student has to scan the QR code for his attendance, the details such as student ID number, student name, student course and semester details needs to be captured.

The requirements are listed below:

* The system must allow the user to scan the QR code
* The system should be able to recognize a QR code
* The system must decode the QR code that is scanned
* The system must allow the user to capture a picture of the QR code
* The system must allow the user to send the captured QR code to the instructor
* The system should be able to capture the details of the system such as the MAC ID.
* The system should be able to connect to the database for sending and retrieving the student information
* The system should be able to show the student the registered courses
* The system should be able to allow the instructor to change the percentages of the student
* The system must allow the user to Login into the account
* The system should allow the user to register for a new account.
* The system should send a captcha every time the user Logins into the system.
* The system should allow the instructor to generate the QR code any number of times
* The system should allow the instructor to check the number of students in the class
* The system should allow the instructor to set timer for the QR codes.
* The system should allow the user to check his attendance percentage.
* The system should allow the user to check the feedback of the attendance percentage
* The system must allow the admin to manage the instructor details
* The system must allow the instructor to generate the QR code using the CRN of the course.

# Detailed functional requirements list

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Module** | **Functional Requirement** |
| **Admin** | | |
| R1 | Admin Login Precondition: Admin is not logged in. | The stand-alone application shall let the admin to login with a correct admin ID and correct password.  1. User ID Requirements - Admin 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 should be encrypted using a hash function.  4. All data Username and password combination will be stored into a database for future reference. |
|
|
|
| R2 | Admin Login Successful | On successful login, the timestamp is captured and stored in the database, the admin should be redirected to a page, where admin will be able to view courses, and students assigned to that course and will be able to generate QR code if necessary. |
| R3 | Login Failure | The admin should be let a maximum of three times to login and on further failure, a notification should popup saying 'Login Failure: Please reset your password.' with ok and cancel buttons. Once the user clicks ok redirect to the 'reset password' page. |
| R4 | Admin Logout Precondition: Admin is logged in. | The stand-alone application shall not let the admin logout when the admin clicks on logout-button. 1. When the admin logs out, the timestamp and the session time are stored in the database. |
| R5 | Add Course | The stand-alone application shall let the admin to add a course to a department. This should add a new course row in the course table and the changes should be preserved. |
| R6 | Delete Course | The stand-alone application shall let the admin to delete a course to a department. This should delete the particular row in course database and the column reference of course column in the admin database and preserve the changes. |
| R7 | Modify Course | The stand-alone application shall allow the admin to modify the course details of a department. The changes should be saved on to the course database as soon as the changes are made by the admin. The changes should reflect in other related databases. |
| R8 | Add Instructor | The stand-alone application shall allow the admin to add instructor to a course. This should make necessary changes to the database by adding reference to the new Instructor column in the Instructor database and preserve the changes in the admin database. |
| R9 | Remove Instructor | The stand-alone application shall allow the admin to remove instructor of a course. This action should make necessary changes to the admin database by deleting instructor column reference assigned to the particular course in the database and preserve the changes in admin database. |
| R10 | Modify Instructor Details | The stand-alone application shall let the admin to modify the instructor details of a course of a department. These changes should reflect in the database and the changes should be preserved after every modification in the Instructor database. |
| R11 | Add Students | The stand-alone application shall let the admin to add students taking a particular course of a department. This should add student’s column reference under one course to the admin database and preserve changes. |
| R12 | Remove Students | The stand-alone application shall let the admin to remove students taking a particular course. This should remove students ID column reference under one course in the admin database and preserve changes. |
| R13 | Modify Student Details | The stand-alone application shall let the admin to modify the student’s details taking a particular course of a department. These changes should reflect in the Student database and the changes should be preserved. |
| R14 | Manage QR code | The stand-alone application shall let the admin generate a unique QR code based on the CRN and the timer. The CRN will be stored in the CRN database and care should be taken that not two CRN's match. |
| R15 | Add Multiple Instructors | The stand-alone application shall let the admin to add multiple instructors to a particular course of a department. This action should make necessary changes to the Instructor database in such that it can refer to multiple courses at once. |
| R16 | Add Multiple Students | The stand-alone application shall let the admin to add multiple students to a particular course of a department. Each course in the course table can have multiple no of students. |
| R17 | Add Multiple Courses | The stand-alone application shall let the admin to add multiple courses of a department. This should add multiple rows in the course database at once and accordingly add references to the admin database and changes should be preserved. |
| R18 | View Student Attendance | The admin shall be able to view student attendance over the whole semester of a department. The admin should be able to access student attendance percentage from the attendance table. |
| R19 | Attendance Analysis | The admin shall be able to view the analysis diagram of the student’s attendance data for a semester data of a department. This should let the admin access the attendance percentage from the attendance table. |
| R20 | Feedback | The admin shall receive an immediate and appropriate feedback on performing any action using the system. This would be event based and will be triggered on every important action initiated by the admin like above. |
| R21 |  |  |
| Student | | |
| R22 | Student Login | The mobile application shall let the admin to login with a correct admin ID and correct password.  1. User ID Requirements - Student 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 should be encrypted using a hash function.  4. All data Username and password combination will be stored into a database for future reference. |
| R23 | Login Successful | On successful login, the timestamp should be captured and stored into a database and the student should be redirected to 'student view' page where the student should be able to capture the QR code and view attendance and courses the student is registered for during that semester. |
| R24 | Login Failure | The mobile application should let the student login with a correct username and password following the above mentioned rules. If the student fails to login for several times, then a popup should notify the student saying "Please take help from the Help page / Please consult the professor" for attendance. |
| R25 | Persistent Login | The mobile application shall save the students login and shan’ts ask for login again and again when the user opens the application. In short, shall let a persistent login. |
| R26 | Scan QR | The mobile application shall let the user scan the unique QR code displayed by the Instructor. If the user scans the right QR then  1. The application should take him to a 'student view'' page. Student should be able to capture QR code, view courses and view attendance percentage from this page. 2. The application should identify the student and save further details based on unique student ID. 2. Timestamp of the scan is captured and saved onto student table. Or else  1. The application should ask the student to scan the QR code again. |
| R27 | Scan QR Failure | The mobile application shall let the student scan the QR code for a maximum of three times. If the scan continues to fail, the student should get a notification saying "Please consult your Instructor" |
| R28 | Capture QR | The mobile application shall let the user capture the unique QR code displayed by the Instructor. This captured QR will be saved in the QR database. |
| R29 | Capture Failure | The mobile application shall let the user capture the unique QR code displayed by the Instructor. Should the capture fail, the student should be let a maximum of three times to recapture it again. |
| R30 | Attendance Recorded Feedback | Once the QR code is scan and capture go through, The mobile application shall send a proper feedback to the user saying 'QR code scanned and attendance recorded'. |
| R31 | View Courses | The mobile application shall let the student view the courses he/she has registered to in the department. This view should be made available when user clicks on 'View Courses' menu. The view is populated with data from the course table and student table. |
| R32 | View Attendance Percentage | The mobile application shall let the student view his/her current attendance percentage. This view should be made visible only when the student clicks on View 'Attendance percentage' menu. The view is populated with data from the attendance table and student table. |
| R33 | Student Logout | The mobile application shall let the student logout of the application. A warning should popup saying 'You can't login within the next 24 hours' whenever the student clicks on the logout button. Should the student logs out: 1. A logout screen should appear showing 'Logged out successfully and you can now not login for another 24hours/ See the instructor immediately for any further help.' 2. Timestamp of the logout activity should be captured into the students table. 3. Login functionality for the next 24 hours should be disabled for this student. |
| R34 |  |  |
| Instructor | | |
| R35 | Instructor Login Precondition: Instructor is logged out of the application. | The stand-alone 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 should be encrypted using a hash function. 4. All data Username and password combination will be stored into a database for future reference. |
| R36 | Login Successful | On successful login, the timestamp is captured and stored in the database, the Instructor should be redirected to '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. |
| R37 | Login Failure | Should the Instructor fail to login, the instructor should be let a maximum of three times to login. If the instructor fails even then, then the instructor should see a popup saying 'Please take help from the help page or consult the admin' |
| R38 | Add Course | 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 should be saved and preserved in the instructor database. |
| R39 | Add sections (Feature to be thought about and implemented in the final sprint) | The stand-alone application shall let the Instructor to add multiple sections for a course. Whenever the instructor selects a course using a CRN, multiple sections should appear and instructor should be able to add the sections to his view. This should add necessary changes to instructor database and changes have to be preserved. |
| R40 | Invite another instructor (Feature to be thought about and implemented in the final sprint) | The stand-alone application shall let the Instructor invite another instructor. There are instances where the instructor might invite another professor to give a lecture. In this case, provision should be provided for the Instructor to invite another instructor. This should:  1.Notify the students with an email notification saying ‘This so and so instructor has been invited for the lecture tomorrow.’ 2. Notify the invited professor with an email notification. |
| R41 | Generate QR | The stand-alone application shall let the Instructor generate a unique QR code when the instructor clicks on 'Generate QR button' on the Instructor view. The instructor is redirected to 'displayQR Page'. The display QR page has the provision to select the subject from the dropdown, no of QR’s required dropdown, QR time active dropdown and time interval dropdown. The QR will be generated only when the instructor fills all these dropdowns. The dropdowns default to a number to make instructors job easy. And once the instructor clicks on generate QR button, these selection details are saved on to the QR database and The timestamp at which the QR is generated should be saved onto QR table and should be active for not more than selected 'timer' by the instructor. |
| R42 | Display QR | 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. |
| R43 | Select Timer | The stand-alone application shall let the Instructor choose the time interval from the 'Time Interval between QR' dropdown. Will default to 20minutes. Unique QR code will be generated and displayed to the students in the class after the instructor choice of time Interval. These dropdown details are saved onto the QR database. |
| R44 | Select the no of QR's | The stand-alone application shall let the Instructor choose the number of QR's to be generated by the application in the 'generateqr' page. The stand-alone application shall let the Instructor generate as many as unique QR codes according to Instructors choice. These details should be saved onto the QR table and the QR codes should be generated after the chosen time interval. |
| R45 | QR Active Time | The stand-alone application should let the instructor choose the QR active time from the 'generate qr' page. This should be a dropdown defaulting to 2 mins. Once the instructor clicks on 'generate qr', this field should be saved onto the QR table. |
| R46 | Home Page button | The stand-alone application shall let the Instructor redirect to the 'instructorview' page when the instructor clicks on home page button on 'generate qr' page. |
| R47 | Modify Attendance | The stand-alone application shall let the Instructor modify student attendance records. On the 'instructorview' page, there should be a button named 'Student' which should redirect the instructor to student view page where the instructor can mark attendance of particular student and finally changes must be saved onto attendance database. |
| R48 | Delete Attendance | The stand-alone application shall let the Instructor delete student attendance records. On the 'instructorview' page, there should be a button named 'Student' which should redirect the instructor to student view page where the instructor can unmark attendance of particular student and finally changes must be saved onto attendance database. |
| R49 | Analyze Attendance | The stand-alone application shall let the Instructor analyze student attendance records. There should be a 'Student' button on 'instructorview' page. When the instructor clicks on it, the instructor is redirector to 'studentattendancetracking' page where an analysis of the student’s attendance over the semester is displayed. This analysis is presented as a bar chart over time and attendance percentage. |
| R50 | No of students | The stand-alone application shall let the Instructor view the number of students in each course. The instructor can view the no of students registered for the course when the instructor clicks on 'course' button on the 'instructorview' page. This action will retrieve no of students registered for a course from the course table and display that on screen. |
| R51 | View Attendance Count | The stand-alone application shall provide the appropriate count of the students in the class. The instructor can view the no of students who could capture the QR successfully when he clicks on 'student' button on the 'instructor view' page and then click on 'view present' button. This will retrieve the no of QR captures from the QR database and displays on the screen. |
| R52 | Feedback | The stand-alone application shall provide appropriate feedback to the Instructor after modifying any student records. The Instructor is notified saying 'Attendance records modified' whenever the student attendance records are modified successfully. |

# Non-Functional Requirements

**“**A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.”

**Performance**: Easy tracking of attendance can be done by the instructor effortlessly.

Static Requirements**:** These requirements do not impose any constraints on the execution characteristics of the system. They are:

1. Number of terminals: This system makes complete use of a database, while the front end will be available in the form of an application of the student and a standalone application for the instructor.
2. Number of users: The number of users for this system are mainly three they are the student, instructor and the admin who can also be referred as the department head.

Dynamic Requirements: These specify some of the requirements which may be changed based on the constraints or the limitations of that particular requirement. They typically include response time and throughput of the system.

1. Software system attributes:
2. **Availability**: This system is available for the users of any educational institutions such as the colleges and high schools for marking the attendance of the students in an easy and flexible way.
3. **Security:** This system is highly secure since it allows the instructor to generate multiple QR codes and also ask for an image of the QR code scanned by the student to ensure that the student is in the class and also reduces the abuse of the system.
4. **Portability**: This system is an application which can be installed in any system satisfying the system requirements of the software.
5. **Reliability:** The system will not crash on invalid data. If data is not recognized, the system will ask the user to scan the QR code in the correct format or the system will produce an error. The data will be stored on a nonvolatile storage device such as a hard drive, so that the data can be retained when the system is shut off. If there is an error the system will ask the user to scan the code again or seek assistance. The user can generate different codes any number of time changing the time of generation of the QR.
6. **Maintainability:** Backups for the database can be done for future use.
7. Design Constraints

**User Interface and Human Factors**

More than one user can use this system simultaneously since there are many students present in class, all should scan the attendance with in the time frame. The usage of this system is easy to use and easy to handle. We are going to have all the constraints so that student cannot abuse the system.

**Hardware considerations:**

We would recommend to use the hardware system with the latest configuration for the fast and easy access of the applications and for the students to scan the QR code we recommend to use an iOS applications with the latest update.

**Performance Characteristics:**

There are neither throughput, speed nor the response constraints for the proposed systems. There are no size or capacity constraints for the system.

**Error handling and extreme conditions:**

If there are any input error in the systems then the system provides feedback to the user specifying the error. If there are any extreme conditions then the system would provide notification to the user.

1. Logical Database Design

There are five main data entities. The following classification is in no way a suggestion of design but rather a logical classification of those data entities as well as their attributes.

|  |  |  |
| --- | --- | --- |
| **Data** | **Attributes** | **Use** |
| Student input data | * Student ID * Student Name * Contact Number * Student Attendance * Student QR capture time | This data is used to track the student attendance an also able to view his attendance percentage. |
| Instructor data | * Instructor Name * Instructor ID * Instructor Contact * QR Active Time * QR Time Interval | This data is used to determine the Instructor details and also to know the time interval between the QR codes. |
| Admin Data | * Admin ID * Admin Name * Admin Contact | This data can be able to provide details for both student and instructor if there is any need of contact. |
| Department | * Department ID * Department Name | This data is used to provide details of the department. |
| Course | * Course ID * Course Name * Course Description * Course Year * Course Semester * Course Schedule | This data is used to provide details and timings of the course for a particular department. |

# System Core Features

1. **User Sign In and Welcome:** Allow the user to sign in with the application. With the acceptable credentials and with a valid password with all the constraints specified.
2. **Scan QR**: The user should be capable of scanning the QR code which is generated by the system. Student should also be able to capture the QR code as soon as the student scans the QR.
3. **Manage QR:** The instructor and the admin has the flexibility of generation or deletion of the QR code as per the constraints specified such as the time at which the class meets special code for that particular section.
4. **Manage Student:** The instructor and the admin has the flexibility of managing the student details such as checking the student details such as name, id and other credentials of the student.
5. **Mark Attendance:** Instructor and the admin has the flexibility to mark the attendance of the student as per the request of the student if he is not able to scan the QR code for any specified reasons.
6. **Manage Courses:** Admin and the Instructor has the feature for managing the courses such as they are capable of adding new courses to the system which they are likely to teach in that particular semester.
7. **Manage Instructor:** Being an admin he has all the functionalities which overlap with that of the instructor yet has an extra functionality of adding or deleting the new instructor’s in that department, he is also capable of managing them by viewing their details, and also viewing their courses and the timings their classes meet.

# Additional Features

1. **Email Notification:** The user logging in to the system for the first time or if he failed to remember the credentials of his account then he has an option of sending the notifications to the email for password changing and for password conformation.
2. **Feedback:** Whatever may be the system the user using the system would love to see if his action has been working properly or not, for such feature we provided the feedback option to the users acknowledging his/her actions.
3. **Reset Password**: User has an option of resetting the password if he has difficulty in remembering the password but he can only do that using the email notifications.
4. **View Attendance Percentage**: One of the user i.e. Student has an extra option of viewing the attendance percentage of all the registered courses for that semester, showing the statistics on the daily basics.
5. **View number of students capturing the QR code**: Instructor has an option of checking the total number of student who have scanned the QR code for that particular session comparing the total number of student of that class.

# Assumptions and Dependencies

1. The main Assumption of the system is that we consider that all the students use mobile phone with iOS compatibility.
2. All the instructor’s bring their laptop for creating their standalone application.
3. Student capturing the QR code is present in the class.
4. No student can carry more than one mobile device.
5. The student attending the class has registered for that class
6. The mobile device is in the working condition for capturing the QR code
7. Instructor has the proper knowledge to generate the QR code.
8. Having a valid internet connection.
9. The instructor is able to project the QR code on the screen.

# Use Case Diagram

The use case diagram is usually a graphical description of interactions between the elements of a system. This is also a methodology which is used in system analysis to identify, organize and clarify the requirements of the system.

##### Purpose:

The main functionality of the use case diagram is to show in what way a user can communicate with the system, this may help in developing a prototype of the system and identifying specific requirements for that particular task. A use case diagram is similar to that of a flow chart. A use case diagram mainly consists of four basic components, they are:

**Actor:** Individuals who are involved in the system, defined as per their roles.

**Use Cases:** these are said to be the specific roles played by the actors within the system or around the system.

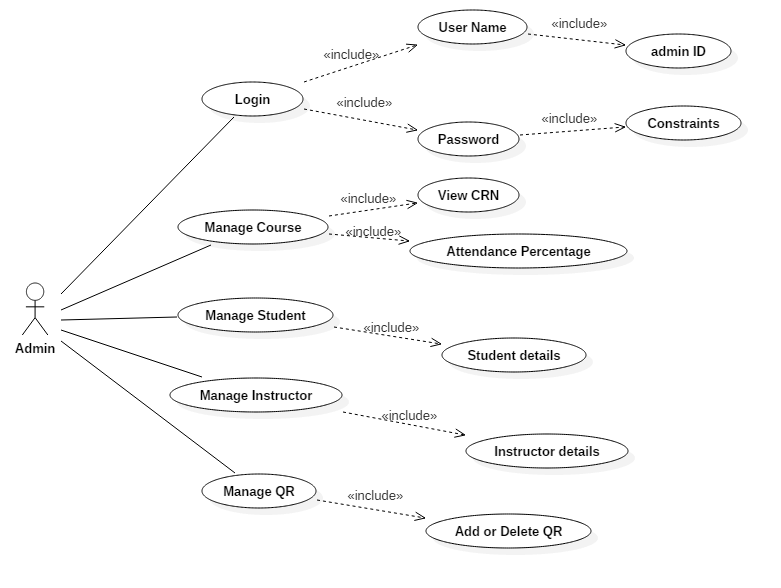
**Boundary:** This defines the system of interest in relation with the world which is around that particular system.

**Relationships:** These are the specific relation between the actors and the use cases of the system.

A use case diagram generally consists of an actor and all the task which can be performed by the actor which are represented in the form of an oval called use cases. In the proposed system there are mainly three actors around which the complete system is based up on they are Student, Instructor and the Admin.

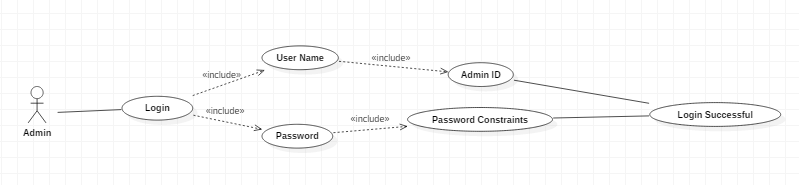
## Module1: Admin use case diagram

Admin plays a kind of actor who is responsible for all the roles, he can also be defined as the head of the department. He is capable of login into the system and view different courses present in the system, capable of managing students such as check student details, students attendance, capable of managing the instructor such as checking the details of the instructor and also capable of add or deleting or the QR code for a particular course.



**Figure1: Use case diagram for Admin**

1. Admin: Login Successful



**Figure2: Admin Login** Use case description

In the initial stage the admin must login with his user name and password. User name of the admin is his Admin ID and password has few constraints. Once logged in the Admin will get a popup as login successful. In the admin view once he logged in he can see four modules like Manage Course, Manage Student, Manage Instructor and Manage QR.

1. Admin: Login Unsuccessful

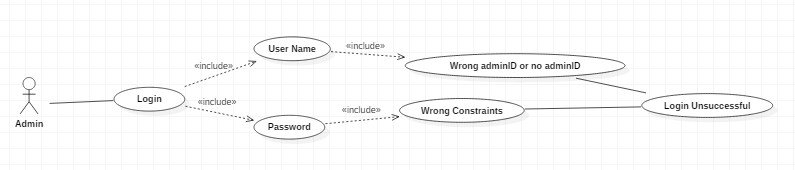


Figure3: Admin Login Unsuccessful Use case description

In the initial stage if the Admin enters wrong username or wrong password or if they have not entered any username or the password they will get a popup as login unsuccessful.

1. Admin: Manage Course

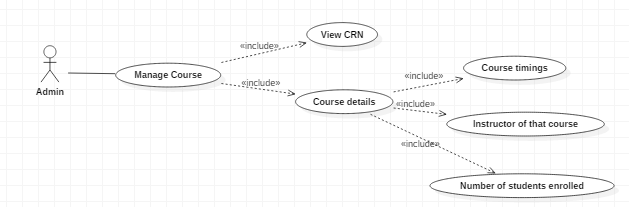


Figure4: Admin Manage Course Use case description

In the manage course module the admin can view CRN and course details. In the course details the admin can check the course timings, instructors of that course and number of students enrolled for that course.

1. Admin: Manage Student

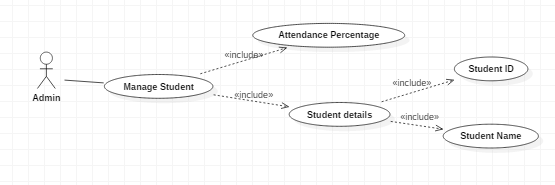


Figure5: Admin Manage Student Use case description

In the manage student module the admin can check the attendance percentage of each student and also he can see the student details like student ID and student name.

1. Admin: Manage Instructor

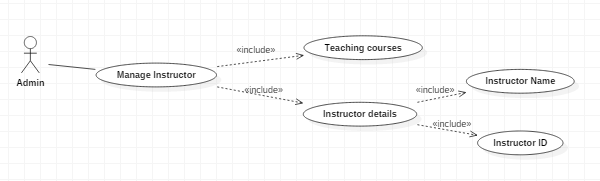


Figure6: Admin Manage Instructor Use case description

In the Manage Instructor module the admin can view what all courses the instructor is assigned with during that semester and also he can check the instructor details like instructor name and instructor ID.

1. Admin: Manage QR

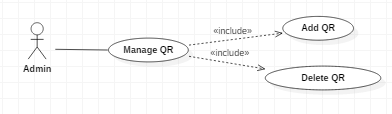


Figure7: Admin Manage QR Use case description

In the Manage QR module as soon as the admin login in to his account he has an option to display the QR code for the students to scan for marking the attendance and he also has the capability to add or delete the QR codes.

## Module 2: Instructor Use case Diagram

This actor is capable of managing the student and his attendance. He is also capable of updating the attendance details of the student. He has many vital roles such as managing the courses, managing the students, and displaying the QR codes. In the process of managing the student instructor is capable of viewing the details of the students and also check the percentage of the student and update it. In the process of managing the courses he is able to add or delete the courses with his own sections. He is also allowed to display the QR codes according to the time at which the class meets and the CRN number and he also has an access to generate multiple QR codes based on his requirements for a specific class at any point of time.

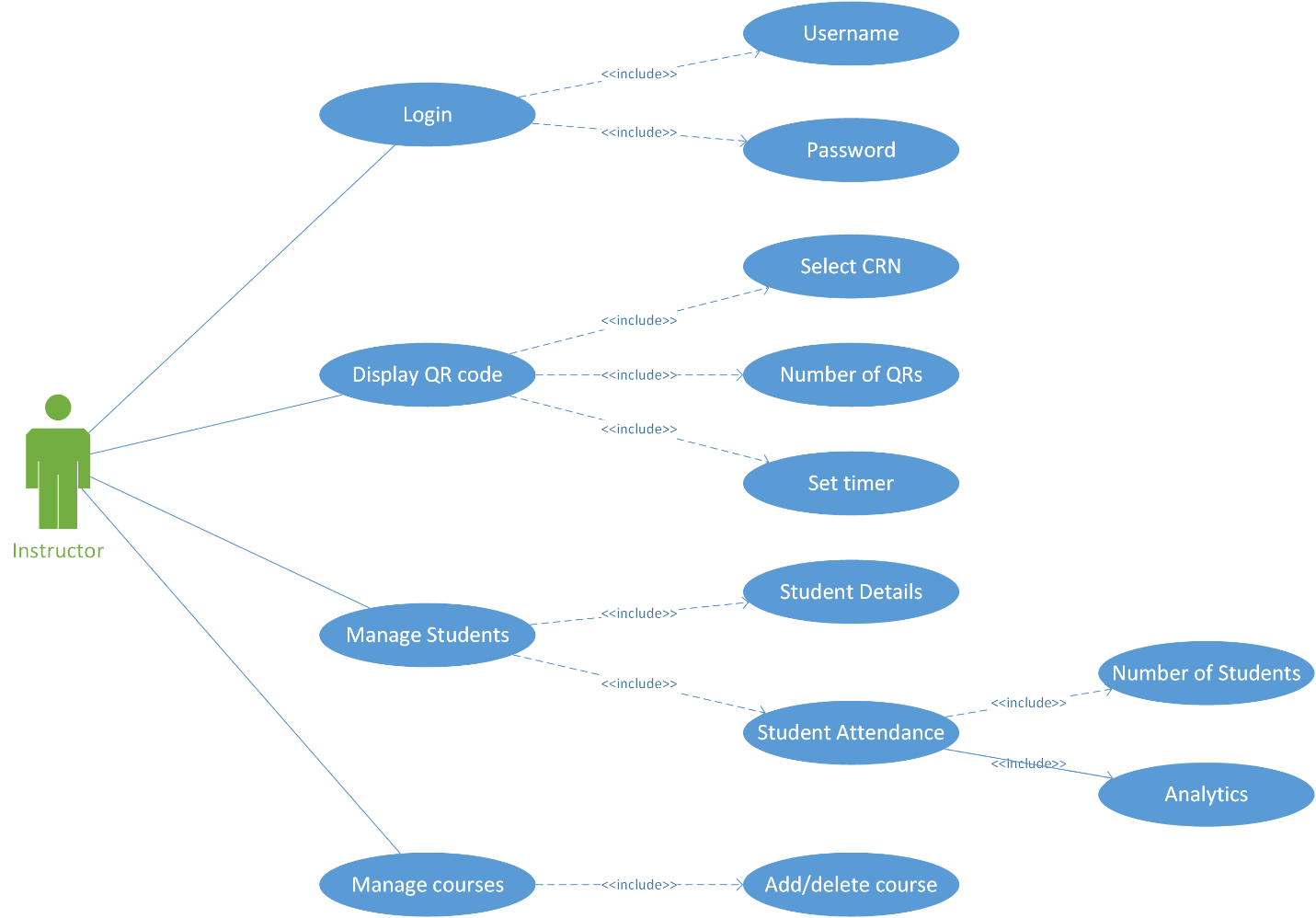


Figure8: Instructor Use case diagram

1. Instructor: Login Successful

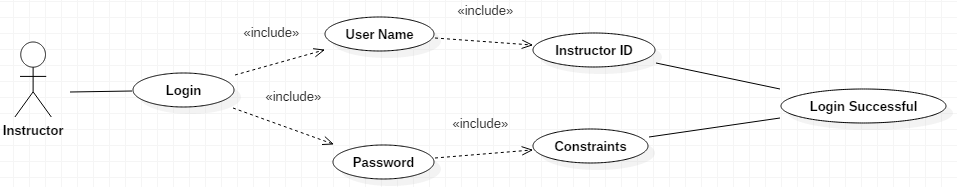


Figure9: Instructor Login successful Use case description

In the initial stage the Instructor must login with his User Name and Password. User Name of the admin is his Instructor ID and Password has few constraints. Once logged in the Instructor will get a popup as login Successful. In the Instructor view once he logged in he can see four modules like Manage Course, Manage Student and Display QR.

1. Instructor: Login Unsuccessful

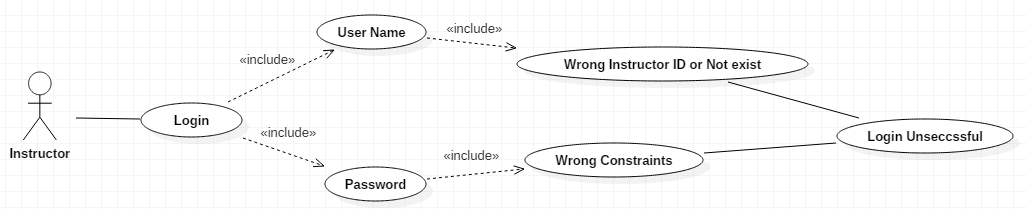


Figure10: Instructor Login Unsuccessful Use case description

In the initial stage if the Instructor enters wrong username or wrong password or if they have not entered any username or the password they will get a popup as login unsuccessful

1. Instructor: Display QR code

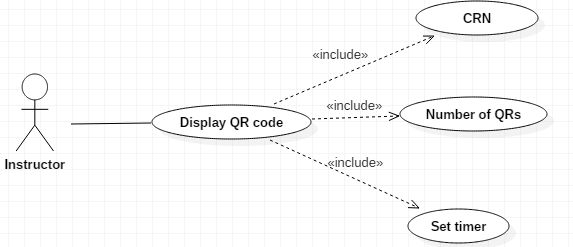


Figure11: Instructor Display QR code Use case description

In the Display QR code module, the instructor can display a unique QR code based on CRN and this QR code will be active for the time that he/she had chosen in the ‘set timer field’

1. Instructor: Manage Students

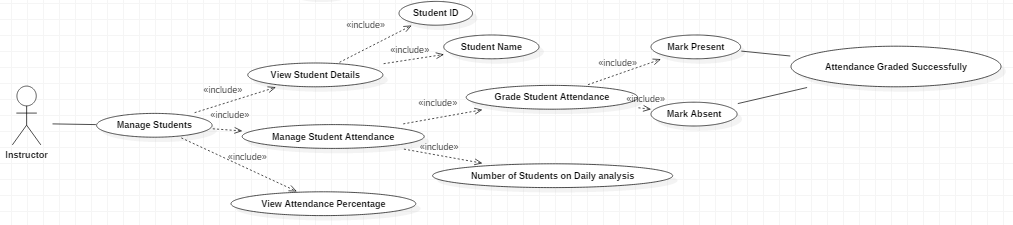


Figure12: Instructor Manage Students Use case description

In the Manage Students Module the instructor can view student details like student ID and student name, Manage student Attendance and View Attendance percentage of each student. In Manage student attendance the instructor can grade student attendance and he can view number of students present on daily analysis.

1. Instructor: Manage Courses

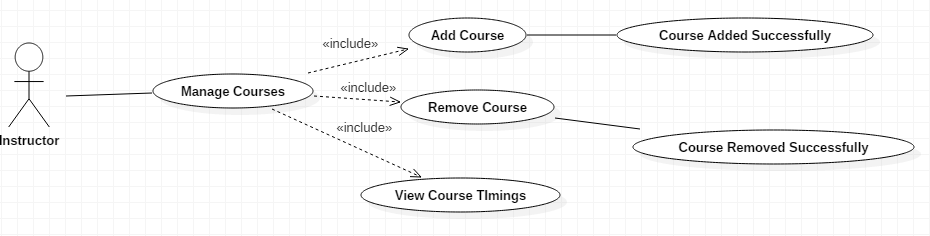


Figure13: Instructor Manage Courses Use case description

In the Manage Courses module Instructor can add course, remove course and view course timings. If the instructor add course he will get a popup as course added successfully and vice-versa.

## Module3: Student Use case Diagram

This actor plays a major role in the system. He is responsible for login in to the system using his Student ID as his user name and using his unique password with all the constraints. He then scans the QR and captures an image of the QR and then sends it to the instructor for grading his attendance. He can also view the overall courses he is registered in to and also check the individual percentage in each of his course. He receives a feedback as soon as the code is captured and is sent to the instructor for grades.

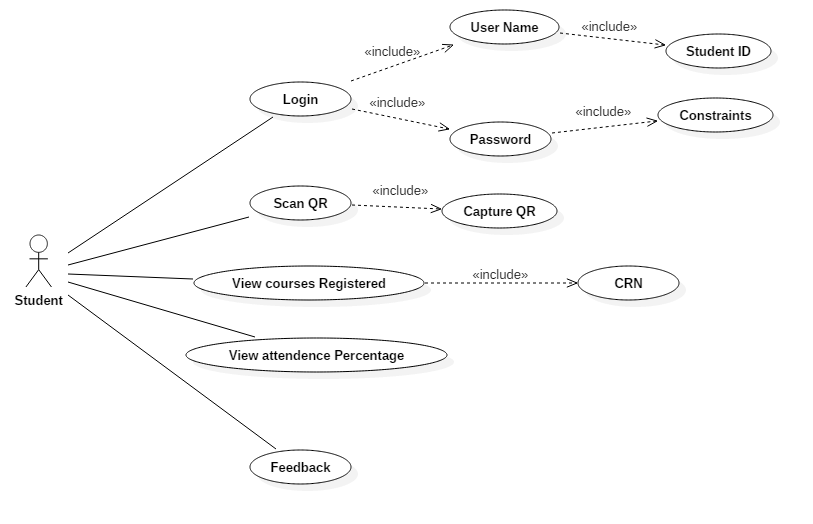


Figure14: Student Use case Diagram

1. Student: Login Successful

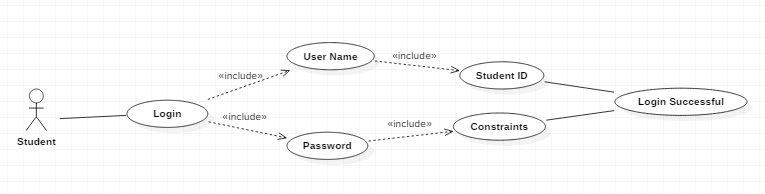


Figure15: Student Login Successful Use case Description

In the initial stage the student must login with his User Name and Password. Username of the student is his SID and password has few constraints. Once logged in the student will get a popup as login successful. In the student view once they logged in they can see three modules like Scan QR, View courses registered and view attendance percentage.

1. Student: Login unsuccessful

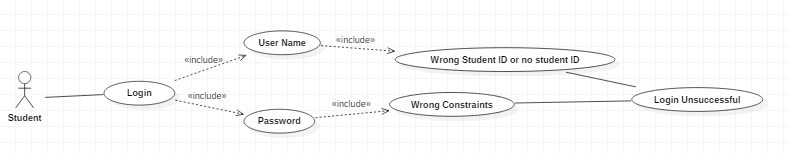


Figure16: Student Login Unsuccessful Use case Description

In the initial stage if the student enters wrong username or wrong password or if they have not entered any username or the password they will get a popup as login unsuccessful.

1. Student: Scanning QR

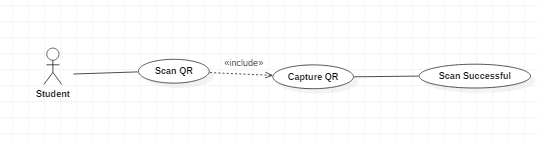


Figure17: Student Scan QR Use case Description

In the scan QR module, student can scan QR code with his mobile application and later he must capture the QR code for security purpose. After capturing QR the student gets a popup as scan successful.

1. Student: View Course Details

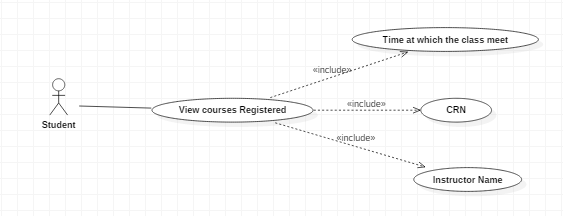


Figure18: Student View Course Details Use case Description

In the view courses registered module students can view their courses which they have been registered for that semester. Within that module they can view timings of the class, CRN and instructor name of that course.

1. Student: Attendance Percentage and feedback

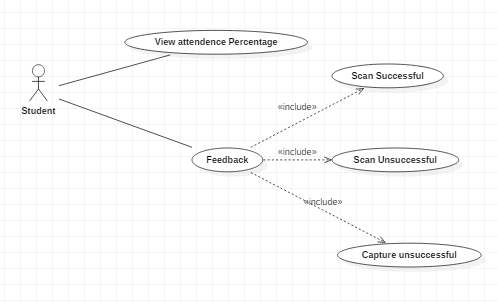


Figure19: Student Attendance Percentage and Feedback Use case Description

In the view attendance percentage module students can view their attendance percentage and students will receive feedback after scanning and capturing the QR like scan successful, scan unsuccessful and capture unsuccessful.