*COMP 246 -- [Smart Team: Part A Final Submission Week 4](https://e.centennialcollege.ca/d2l/lms/dropbox/user/folder_submit_files.d2l?db=278043&grpid=480893&isprv=0&bp=0&ou=477093" \o "Submit files to Part A Final Submission Week 4)*

Students’name:

Andrew Trindad

Caique Bispo Ferreira

Harman Sidhu

Robinpreet Kaur

Sejin

**Smart Meeting – Your system of smart attendance control.**

## **Problem Description**

The manual processes impact directly not only on its runtime but also on its control and management. Like any other company, colleges are always looking for solutions to automatize their processes. The student attendance registration’s process has direct impacts on class time, resulting in less time devoted to the teaching-learning task or even in a lack of control especially in larges events with many lectures, workshops, mini-courses, etc. The system of control not only guarantees automation of the presence validation processes, a more efficient and innovative control but also works with the most recent technologies of the market, facilitating the meeting of professionals to act in future maintenance or improvements.

## **System Capabilities**

The new system should be capable of:

* Inserting, modifying and deleting, consulting employees.
* Inserting, modifying and deleting, consulting events by employees.
* Inserting, modifying and deleting, consulting rooms by employees.
* Inserting, modifying and deleting, consulting speakers by employees.
* Inserting, modifying and deleting, consulting lectures by employees.
* Inserting, modifying and deleting, consulting students by employees.
* Inserting, modifying and deleting, consulting semester’s students by employees.
* Inserting, modifying and deleting, consulting students’ attendances by employees.
* Recovering the employees’ password by email.
* Saving a log of deleting the students’ attendances.
* Validating the student’s attendance using QR Code through the smartphone.
* Exporting a report of lectures per events or/and date by employees.
* Exporting a report of students’ attendances per events or/and date by employees.
* Inserting and consulting attendances by students.
* Allowing students to consult the event’s schedule.
* Controlling deadline for the student presence’s registration by the event or the lecture.
* Importing students without losing the historic every semester.
* Allowing that lectures have an event, two speakers, a room, a subject and an observation.
* Generating the tickets per lecture considering the capacity of its room plus 10% in case of selecting the validation by tickets.
* Inserting the name of event, its quantity of days to validate students’ attendance, its date to start the attendance validation and status (enable and disabled).
* Inserting the name of room, its capacity, type (auditorium, classroom, etc.) and status (enable or disable).
* Inserting the name, type (Professor, Instructor, etc.) and status (enable or disable) of speaker.
* Inserting the name, email, date of birth, id, sex and year, term and program of student.
* Inserting the name, email and password of employee.
* Inserting the lecture with its event, room, speaker, type (Workshop, Minicourse, Seminar, etc.), date, duration.
* Providing validation of attendance via smartphone or tickets in every lecture.
* Printing the tickets or QR Codes per lecture of the event.
* Inserting manual the students’ attendance by employees.
* Making backups of its database.
* Saving the QR Codes’ id temporary on smartphone without internet.
* Running on Google Chrome, Internet Explorer and Safari.
* Being available 24 hours per day.
* Alerting the employee in case of empty mandatory fields.

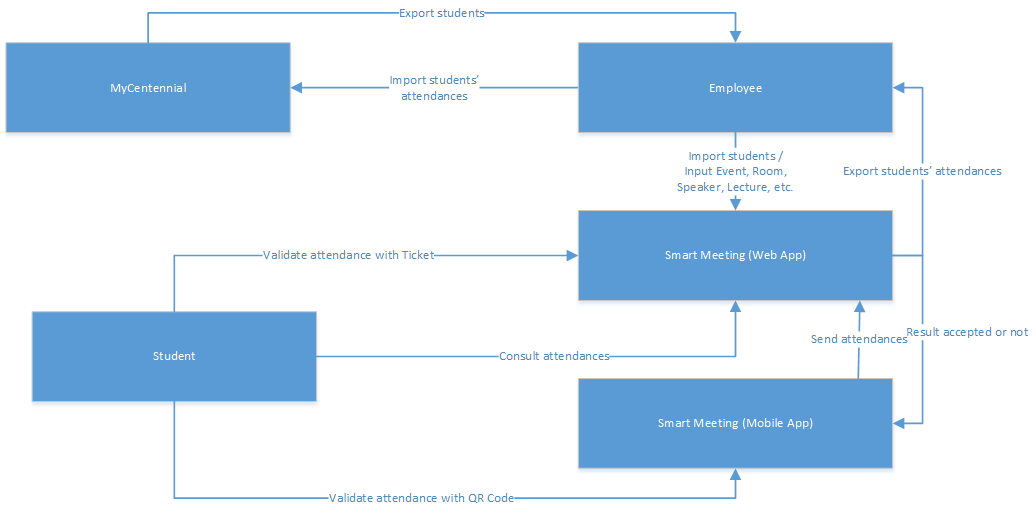
## **Business Benefits**

The deployment of this new system will provide a better control and management of students’ presence with several benefits:

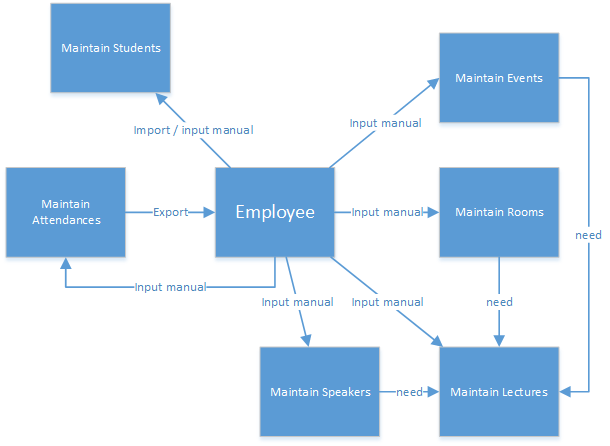
* Allow students to insert and consult their attendances anywhere and anytime using their smartphone or computer.
* Decentralized the control of students’ attendance.
* Allow to identify the most participating students and reward them.
* Optimize the use of lectures’ time.

## **Draw Context Flow Diagrams**

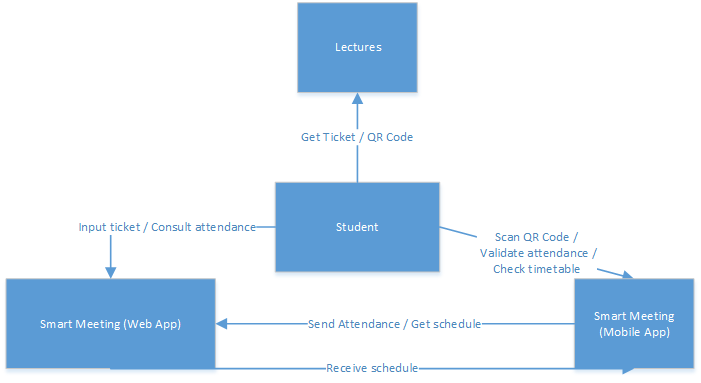
DFD 1 – General Diagram



DFD 2 – Smart Meeting (Web Application)



DFD 3 – Smart Meeting (Mobile Application)



## **List of Subsystems**

* Employees Management Subsystem

It is responsible for controlling the users who access the system.

* Events Management Subsystem

It is responsible for controlling the events.

* Lectures Management Subsystem

It is responsible for controlling the lectures (workshop, mini course, seminar, etc) of events.

* Speakers Management Subsystem

It is responsible for controlling the speakers of lectures.

* Students Management Subsystem

It is responsible for controlling the students per semester.

* Rooms Management Subsystem

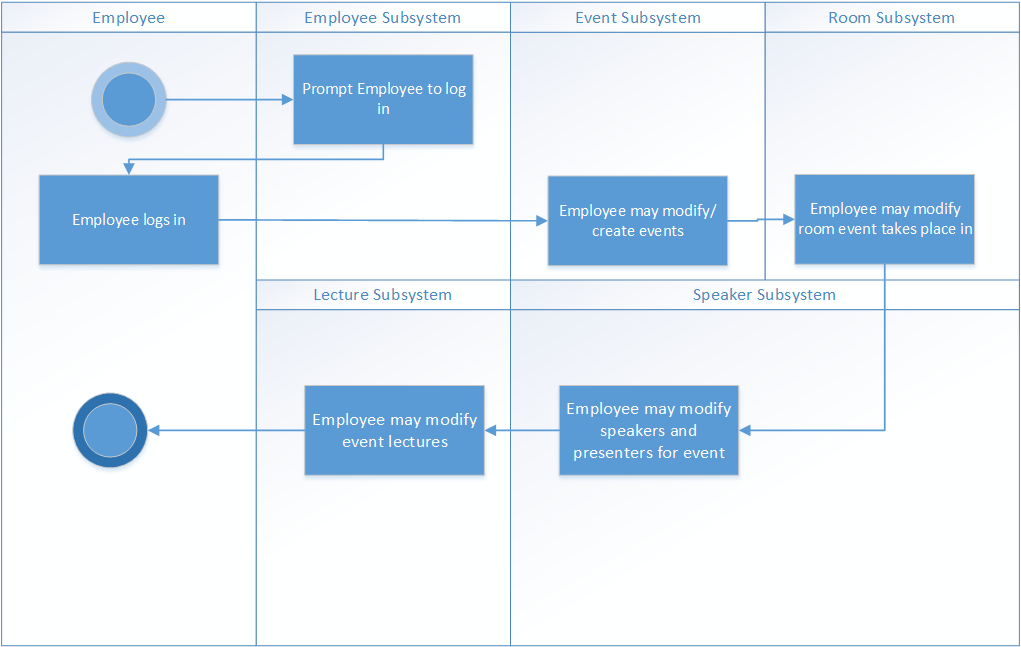
It is responsible for controlling the rooms (auditorium, classroom, etc.) of events.

* Attendances Management Subsystem

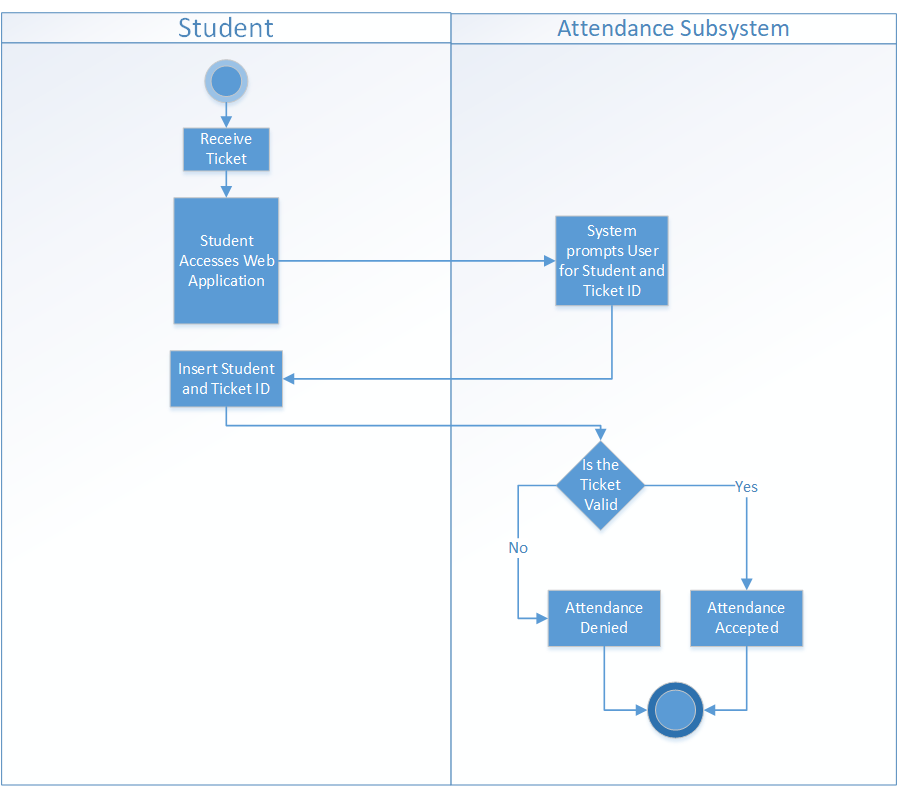
It is responsible for controlling the students' attendances.

## **Activity Diagrams**

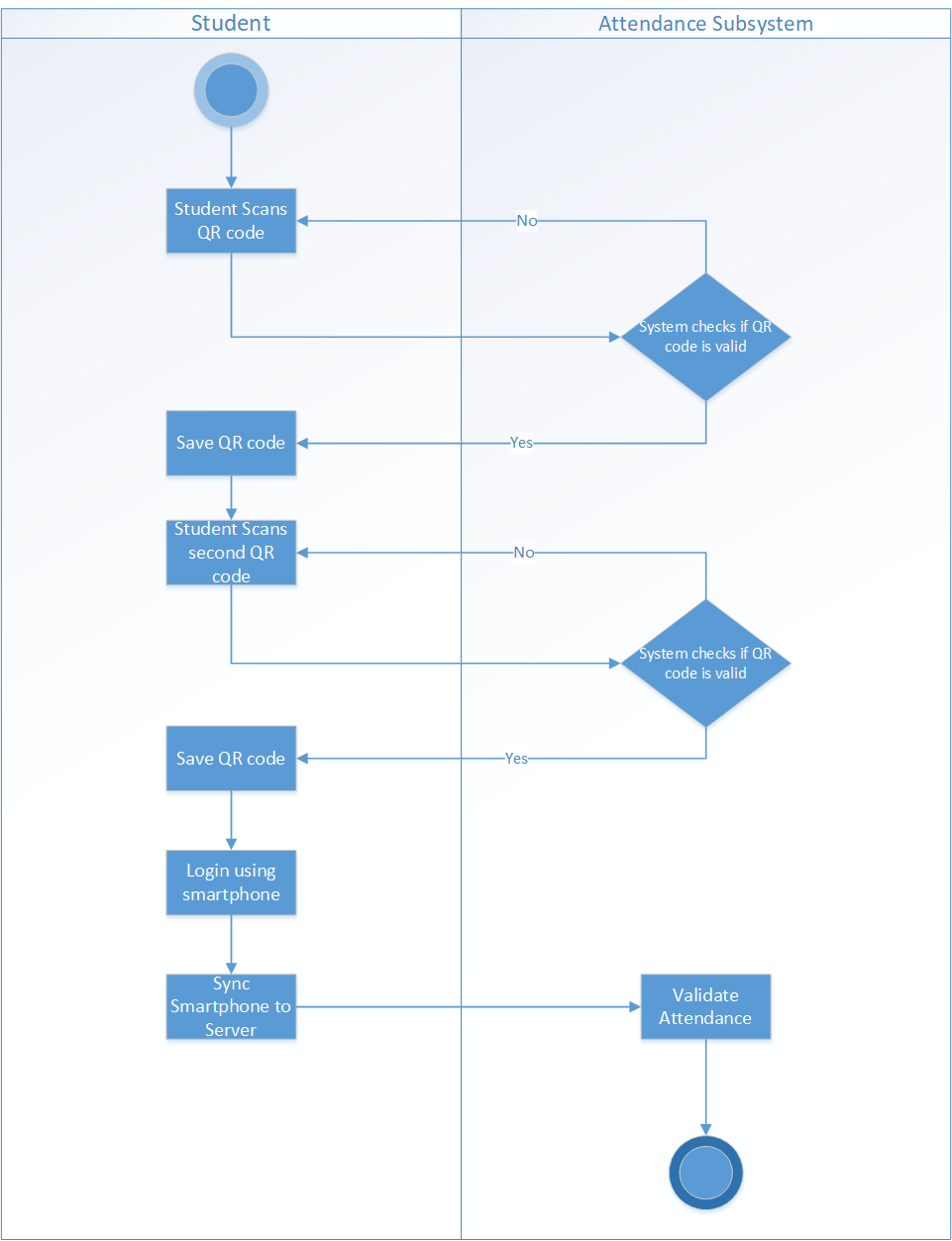
Activity 1 – Smart Meeting (Web Application)



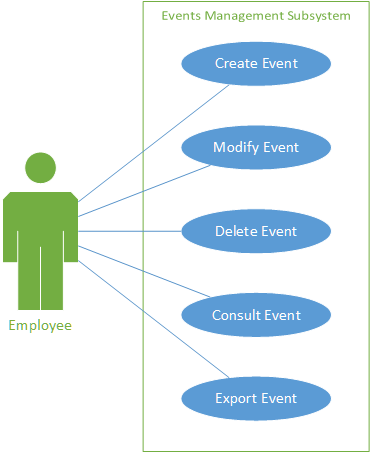
Activity 2 – Smart Meeting (Web Application)



Activity 3 – Smart Meeting (Mobile Application)



## **User Case Diagrams**



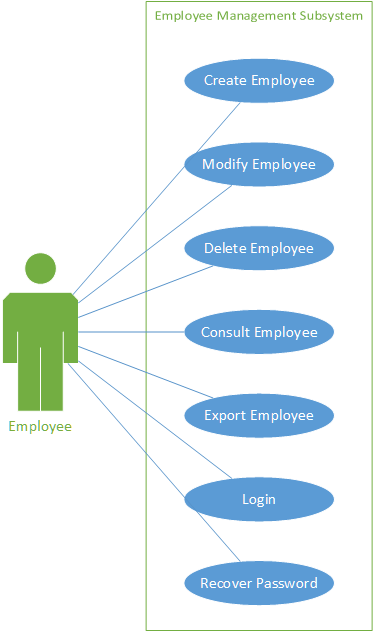
|  |  |
| --- | --- |
| **Event Management Subsystem** | |
| **Use Case:** | Create Event |
| **User Story:** | |
| As a User, I want to create an event. | |
| **Acceptance Criteria:** | |
| 1. The system should take the user to the create event screen. | |
| 1. The user will enter the events basic information such as description and date. | |

|  |  |
| --- | --- |
| **Event Management Subsystem** | |
| **Use Case:** | Modify Event |
| **User Story:** | |
| As a User, I want to modify an event. | |
| **Acceptance Criteria:** | |
| 1. The system should take the user to the event information screen. | |
| 1. The user should be able to enter various information about an event (Rooms, lectures, etc). | |

|  |  |
| --- | --- |
| **Event Management Subsystem** | |
| **Use Case:** | Delete Event |
| **User Story:** | |
| As a User, I want to delete an event. | |
| **Acceptance Criteria:** | |
| 1. The system should take the user to remove an event from the event list. | |

|  |  |
| --- | --- |
| **Event Management Subsystem** | |
| **Use Case:** | Consult Event |
| **User Story:** | |
| As a User, I want to consult an event. | |
| **Acceptance Criteria:** | |
|  | |

|  |  |
| --- | --- |
| **Event Management Subsystem** | |
| **Use Case:** | Export Event |
| **User Story:** | |
| As a User, I want to export an event. | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to export information about the event. | |
| 1. The system should allow different file types to make ease of information access better. | |



|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Create Employee |
| **User Story:** | |
| As a User, I want to create an Employee (Employee account). | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to create an employee account | |
| 1. The user should be prompted to input an Employee ID, email and password | |
| 1. The system should be able to check if Employee ID and password are valid | |
| 1. The system should send an email informing the user they have created their account. | |

|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Modify Employee |
| **User Story:** | |
| As a User, I want to modify an Employee | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to modify an employee account (email and password) | |

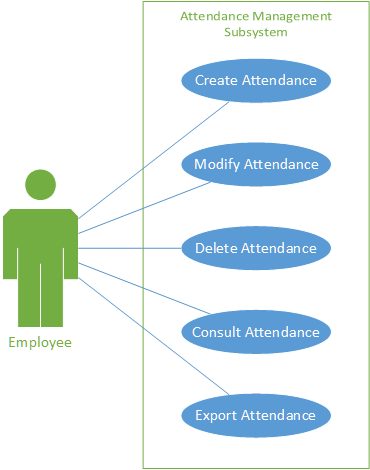
|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Delete Employee |
| **User Story:** | |
| As a User, I want to delete an Employee (Employee account). | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to delete an employee account. | |

|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Consult Employee |
| **User Story:** | |
| As a User, I want to consult an Employee. | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to message another Employee. | |

|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Export Employee |
| **User Story:** | |
| As a User, I want to export an Employee. | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to export Employee information | |
| 1. The system should allow the user to export in different file types. | |

|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Login |
| **User Story:** | |
| As a User, I want to login to my Employee account. | |
| **Acceptance Criteria:** | |
| 1. The system should prompt the user to input Employee ID/Email, and Password. | |
| 1. The system should then check if the login information is valid. | |
| 1. The user should be then taken to the application management screen. | |

|  |  |
| --- | --- |
| **Employee Management Subsystem** | |
| **Use Case:** | Recover Password |
| **User Story:** | |
| As a User, I want to recover my password to my account. | |
| **Acceptance Criteria:** | |
| 1. The system should send an email to the users email address. | |
| 1. The user should click on the email and click on the link in said email. | |
| 1. The user then should be prompted to enter a new password and to confirm that password | |



|  |  |
| --- | --- |
| **Attendance Management Subsystem** | |
| **Use Case:** | Create Attendance |
| **User Story:** | |
| As a User, I want to create an Attendance list for an event. | |
| **Acceptance Criteria:** | |
| 1. The option to create a list in this system should be intuitive and simple to the user. | |
| 1. The system should be able to check if an attendance list exists for an event. | |

|  |  |
| --- | --- |
| **Attendance Management Subsystem** | |
| **Use Case:** | Modify Attendance |
| **User Story:** | |
| As a User, I want to modify an Attendance list for an event. | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to add or delete a student from the attendance list. | |
| 1. The user should be prompted to input a Students name and Student ID. | |
| 1. The system should be able to check if a student exists on the attendance list. | |

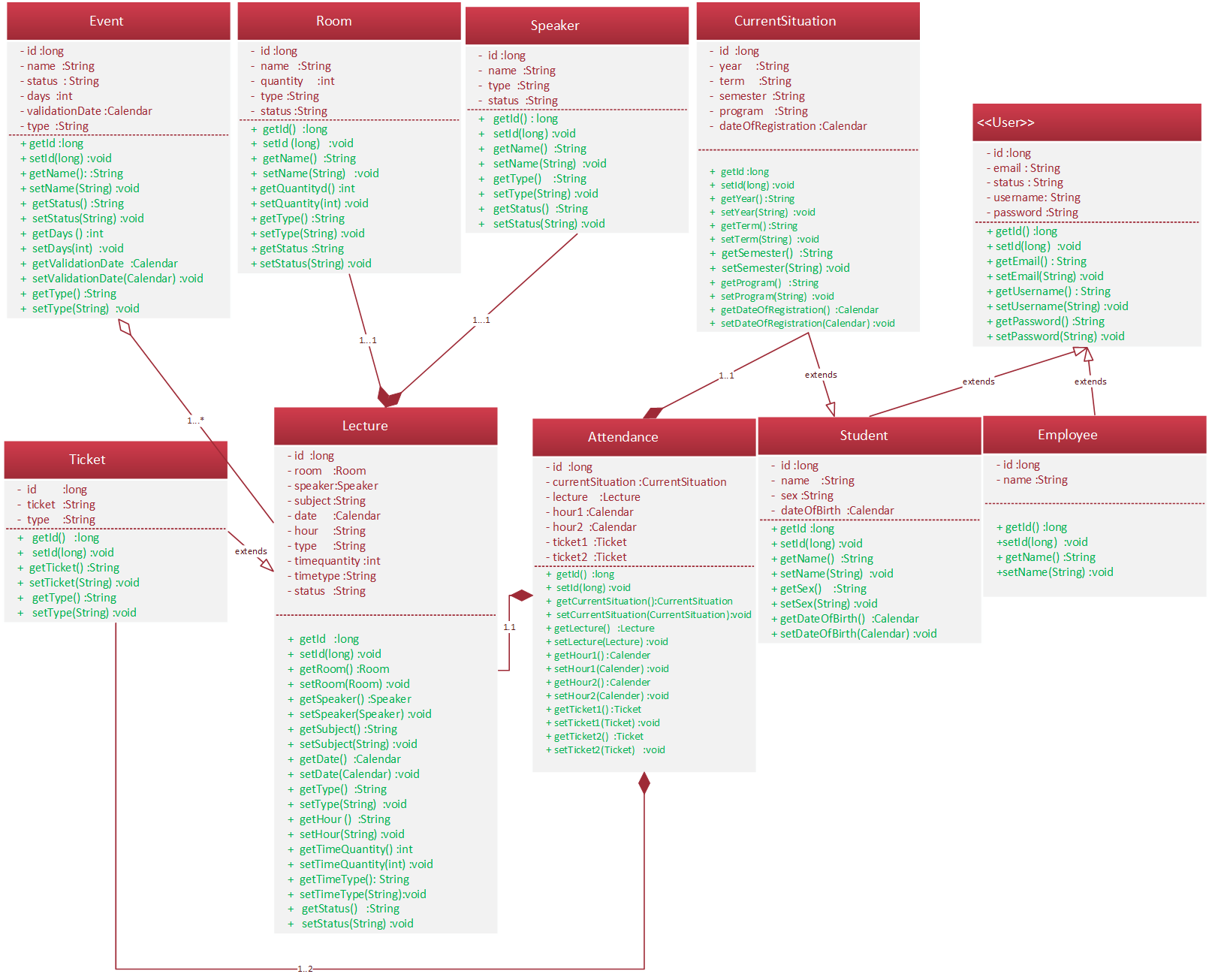
|  |  |
| --- | --- |
| **Attendance Management Subsystem** | |
| **Use Case:** | Delete Attendance |
| **User Story:** | |
| As a User, I want to modify an Attendance list for an event. | |
| **Acceptance Criteria:** | |
| 1. The system should allow the user to delete an Attendance list from an event. | |

|  |  |
| --- | --- |
| **Attendance Management Subsystem** | |
| **Use Case:** | Consult Attendance |
| **User Story:** | |
| As a User, I want to consult Attendance for an event. | |
| **Acceptance Criteria:** | |
|  | |

|  |  |
| --- | --- |
| **Attendance Management Subsystem** | |
| **Use Case:** | Export Attendance |
| **User Story:** | |
| As a User, I want to export an Attendance list for an event. | |
| **Acceptance Criteria:** | |
| 1. The option to export an attendance list should be very simple. | |
| 1. The system should support different file types. | |

## 

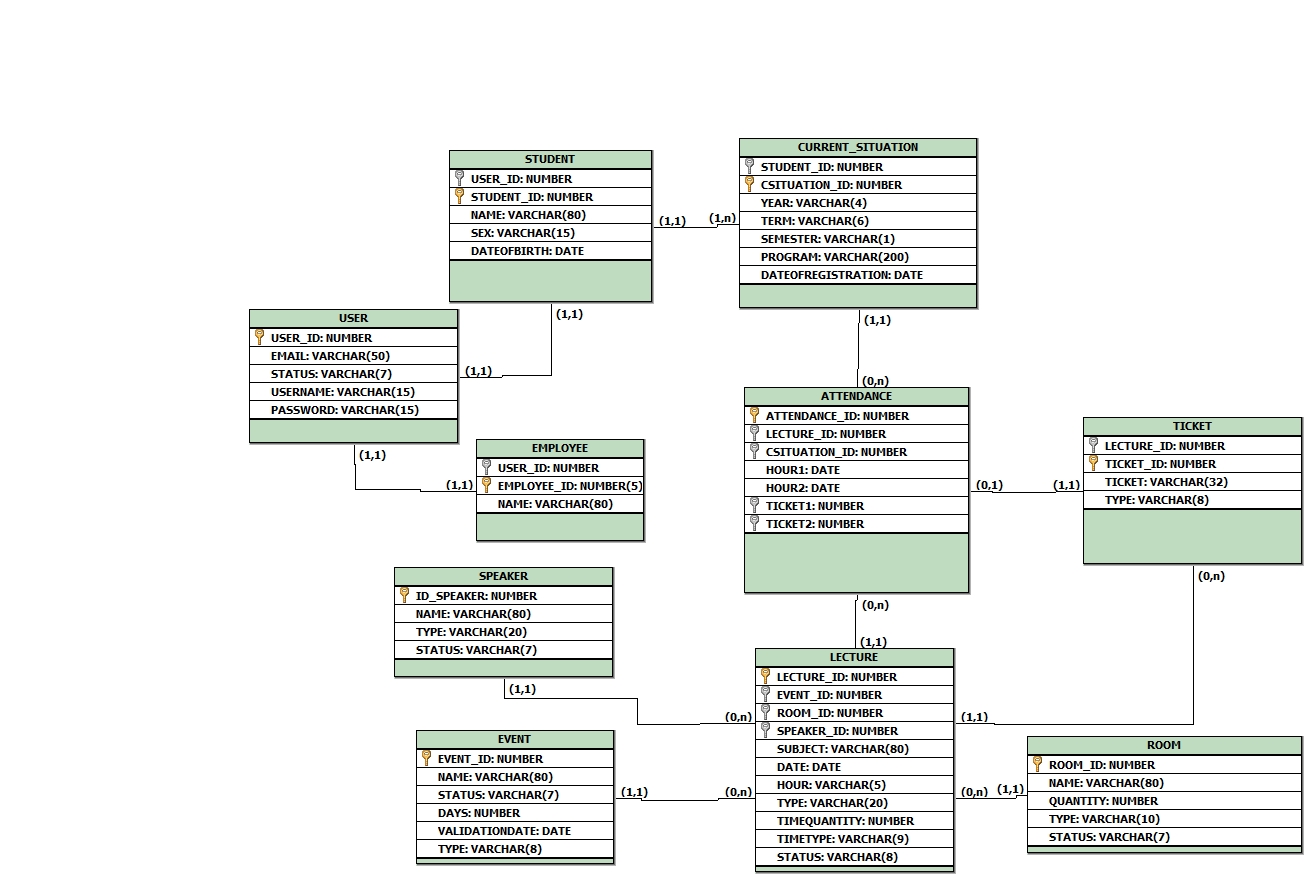
## **Class Diagram**



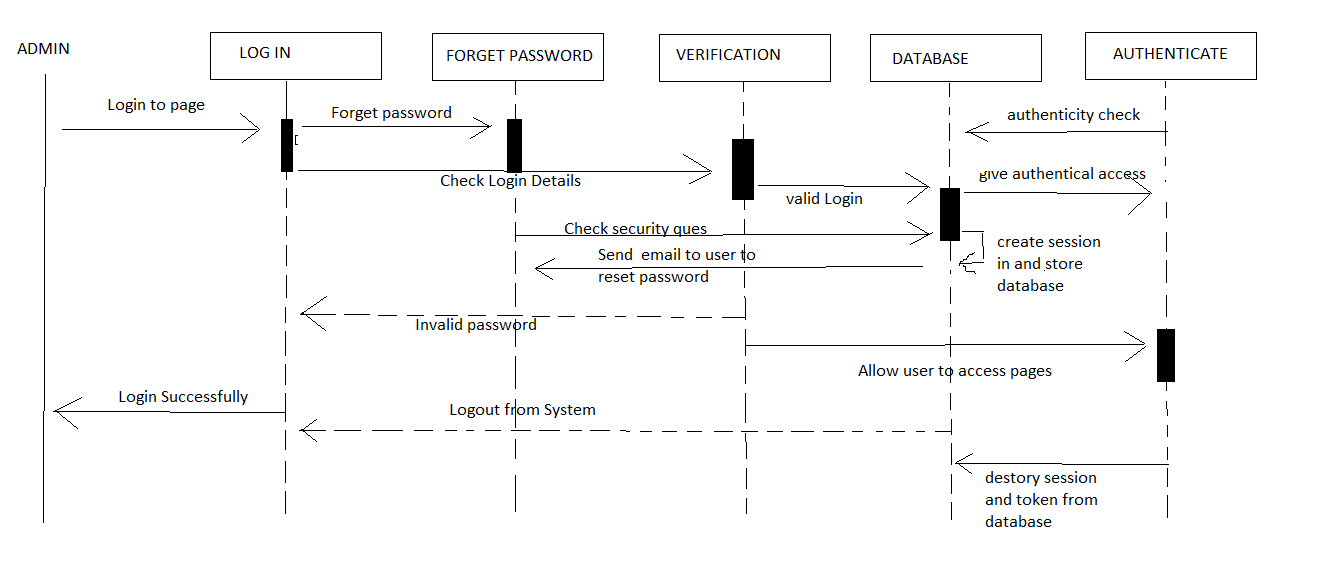
## **CRC document table**

|  |  |
| --- | --- |
| **Class: Event** | |
| **Description:** | |
| Entity class represent the event | |
| **Responsibility** | **Collaborator** |
| Create event | Employee |
| Modify event |
| Delete event |
| Consult event |
| Export event |
|  |  |
| **Class: Room** | |
| **Description:** | |
| Entity class represent the room | |
| **Responsibility** | **Collaborator** |
| Create room | Employee |
| Modify room |
| Delete room |
| Consult room |
|  |  |
| **Class: Speaker** | |
| **Description:** | |
| Entity class represent the speaker | |
| **Responsibility** | **Collaborator** |
| Create speaker | Employee |
| Modify speaker |
| Delete speaker |
| Consult speaker |
|  |  |
| **Class: Lecture** | |
| **Description:** | |
| Entity class represent the lecture | |
| **Responsibility** | **Collaborator** |
| Create lecture | Employee |
| Modify lecture |
| Delete lecture |
| Consult lecture |
| Export lecture |
|  |  |
| **Class: Ticket** | |
| **Description:** | |
| Entity class represent the ticket | |
| **Responsibility** | **Collaborator** |
| Create ticket | Employee |
| Create ticket QR Code |
| Delete ticket |
| Consult ticket |
|  |  |
| **Class: Attendance** | |
| **Description:** | |
| Entity class represent the attendance | |
| **Responsibility** | **Collaborator** |
| Create attendance | Employee / Student |
| Export attendance | Employee |
| Delete attendance |
| Consult attendance |
|  |  |
| **Class: Student** | |
| **Description:** | |
| Entity class represent the student | |
| **Responsibility** | **Collaborator** |
| Create student | Employee |
| Modify student |
| Delete student |
| Consult student |
| Import student |
|  |  |
| **Class: CurrentSituation** | |
| **Description:** | |
| Entity class represent the student's current situation | |
| **Responsibility** | **Collaborator** |
| Create student's current situation | Employee |
| Modify student's current situation |
| Delete student's current situation |
| Consult student's current situation |
| Import student's current situation |
|  |  |
| **Class: User** | |
| **Description:** | |
| Entity class represent the user's system | |
| **Responsibility** | **Collaborator** |
| Create user | Employee |
| Modify user |
| Delete user |
| Consult user |
| Recover user's password |
|  |  |
| **Class: Employee** | |
| **Description:** | |
| Entity class represent the employee | |
| **Responsibility** | **Collaborator** |
| Create employee | Employee |
| Modify employee |
| Delete employee |
| Consult employee |

## **ER Diagram**



## **Sequence Diagrams**



## **Technologies**

|  |  |
| --- | --- |
| **Web Application** | |
| Database | Oracle 12c |
| Back-end | Java 8 |
| Front-end | Bootstrap 4 |
|  |  |
| **Mobile Application** | |
| Database | SQL Lite |
| Back-end | Cordova |
| Front-end | Bootstrap 4 |

## **Gantt**

