Use Case Description

Use Case ID:	UC001		
Use Case Name:	Take Attendance		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Student (Participating Actor)
Description:	Monitor the students entering the classroom and take their attendance for the lesson
Preconditions:	1. The Professor entered a valid course code and index in the home page
Postconditions:	1. Current Time > (Lesson Start Time + 30 minutes)
Normal Flow:	1. The webcam is turned on and the live video is displayed on the screen 2. The system retrieves the enrolled Students' information from the database and displays their name in red on the left of the screen under the 'Absent' category 3. The Students start to walk in the classroom for their lessons 4. The Students are detected and recognized by the system (UC002) 5. Attendance is taken for Students who are on time (UC003) 6. Late Students are marked accordingly (UC004) 7. The lesson ends
Alternative Flows:	-
Exceptions:	-
Includes:	UC002 - Recognize Face, UC003 - Mark as "Present", UC004 - Mark as "Late"
Extends:	-
Priority:	High
Frequency of Use:	High
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC002		
Use Case Name:	Recognize Face		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Student (Participating Actor)
Description:	Recognize the Students' faces as they walk inside the classroom
Preconditions:	1. Webcam is turned on 2. Student starts to walk in
Postconditions:	1. Student ID is retrieved from the database
Normal Flow:	1. The system detects the faces of the Students 2. The system sends the images of the detected faces to the database to check if the Student has been registered for that class 3. Student ID is known from the database
Alternative Flows:	-
Exceptions:	EX1: If the Student's face is unrecognized, 1. The system goes to UC005 - Face Unrecognized 2. The system then returns to Step 1 for another Student
Includes:	-
Extends:	UC005 - Face Unrecognized
Priority:	High
Frequency of Use:	High
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC003		
Use Case Name:	Mark as ‘Present’		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Student (Participating Actor)
Description:	Mark the Students who enter the classroom on time as ‘Present’
Preconditions:	1. Student ID is retrieved from the database 2. Current Time <= (Lesson Start Time + 15 minutes)
Postconditions:	1. Student is marked as ‘Present’ for the lesson
Normal Flow:	1. Student’s ID is cross-checked with the list in red on the left of the screen under the ‘Absent’ category 2. The respective Student’s name is removed from the ‘Absent’ category 3. The Student’s name is then appended to the list on the right of the screen under the ‘Present’ category in green
Alternative Flows:	AF-S1: If the Student is already marked as ‘Present’ 1. The system ignores the Student 2. The system returns to Step 1 when a new Student ID is retrieved from the database
Exceptions:	-
Includes:	-
Extends:	-
Priority:	High
Frequency of Use:	High
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC004		
Use Case Name:	Mark as ‘Late’		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Student (Participating Actor)
Description:	Mark the Students who enter the classroom late as ‘Late’
Preconditions:	1. Student ID is retrieved from the database 2. Current Time > (Lesson Start Time + 15 minutes)
Postconditions:	1. Student is marked as ‘Late’ for the lesson
Normal Flow:	1. Student’s ID is cross-checked with the list in red on the left of the screen under the ‘Absent’ category 2. The respective Student’s name is removed from the ‘Absent’ category 3. The Student’s name is then appended to the right of the screen under the ‘Present’ category in yellow
Alternative Flows:	AF-S1: If the student is already marked as ‘Present’ 1. The system ignores the Student 2. The system returns to Step 1 when a new Student ID is retrieved from the database
Exceptions:	-
Includes:	-
Extends:	-
Priority:	Medium
Frequency of Use:	Medium
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC005		
Use Case Name:	Face Unrecognized		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Student (Initiating Actor) / Professor (Participating Actor)
Description:	Manually take the attendance of the Student who could not be recognized by the system
Preconditions:	1. Student walks into the classroom 2. Facial recognition system does not recognize him/her
Postconditions:	1. Attendance for the Student is taken, and the face is added in the database
Normal Flow:	1. Student is asked to retake photos using webcam, or enter matriculation number to register new photos in the database 1.1 AF-1: When students' faces are not recognised 2. The students face is recognised and studentID is retrieved from the database 3. The status of the Student is changed according to the time the image was stored in the database
Alternative Flows:	AF-1: 1. Student is chooses to enter their matriculation number to mark their attendance. 2. The system will ask the students to take 10 images while moving their faces 3. New photos are added to their profiles and their attendance is taken accordingly.
Exceptions:	-
Includes:	-
Extends:	-
Priority:	High
Frequency of Use:	Medium
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC006		
Use Case Name:	Summary Report		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Professor / Student (Participating Actors)
Description:	Display the attendance sheet for the class on screen
Preconditions:	1. The lesson has ended
Postconditions:	1. The Professor navigates to another page
Normal Flow:	1. The system collects all the Student's status and update the attendance sheet 2. The 'Late' and 'Absent' Students' names will be highlighted in yellow or red respectively 3. The Professor clicks the 'Send E-Mail' button to automatically generate and send an e-mail to the Students who were 'Absent'
Alternative Flows:	-
Exceptions:	EX1: If the Students has already informed the Professor that he is going to be late for the lesson 1. The Professor clicks the 'Edit' button and changes the status of that particular Student from 'Absent' to 'Valid Reason' 2. The system returns to Step 3, and the Students with a status of 'Valid Reason' will not receive an e-mail
Includes:	
Extends:	-
Priority:	Medium
Frequency of Use:	Low
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC007		
Use Case Name:	Log In		
Created By:	The Team	Last Updated By:	The Team
Date Created:	31 AUG 2019	Date Last Updated:	02 SEP 2019

Actors:	Professor (Initiating Actor)
Description:	Allow the Professor to key in the appropriate course code and course index
Preconditions:	1. Current Time >= (Start Lesson Time - 15 minutes)
Postconditions:	1. The attendance taking system for the lesson has commenced
Normal Flow:	1. The home page displays 2 blank fields to fill in the course code and course index respectively 2. The Professor fills up the text boxes accordingly 3. The Professor clicks “Start” 4. The system checks that the details provided by the Professor is accurate 5. The webcam turns on and the system starts to take the attendance for the lesson
Alternative Flows:	AF-S2: If the course code or index do not exist, 1. The system displays a relevant error message 2. The system returns to Step 1
Exceptions:	-
Includes:	-
Extends:	-
Priority:	Medium
Frequency of Use:	Low
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-