

# **ATTENDANCE CHECKER**

## **Use Case Specification**

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo  
Faculty Member  
Department of Computer Science  
College of Engineering  
University of the Philippines, Diliman

Submitted by:

Choa, Jeremy Micah  
Velazquez, Kenneth  
Yao, Faneallrich Li

In partial fulfillment of academic requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> Semester, AY 2016-2017

**Unique Reference:**

This document is stored in the project's requirements engineering folder in GitHub, found [here](#)

**Document Purpose:**

The purpose of this document is to exhibit the use case specification of "1.1 Add Class."

**Target Audience:**

The target audience is the CS 191 Instructor and the fellow team members on the assigned task.

**Revision Control***History Revision:*

<b>Revision Date</b>	<b>Person Responsible</b>	<b>Version Number</b>	<b>Modification</b>
9/28/16	Yao,Faneallrich Li	1.0	Initial Document;
9/29/16	Choa, Jeremy Micah	1.1	Editing for compliance with use-case diagram document

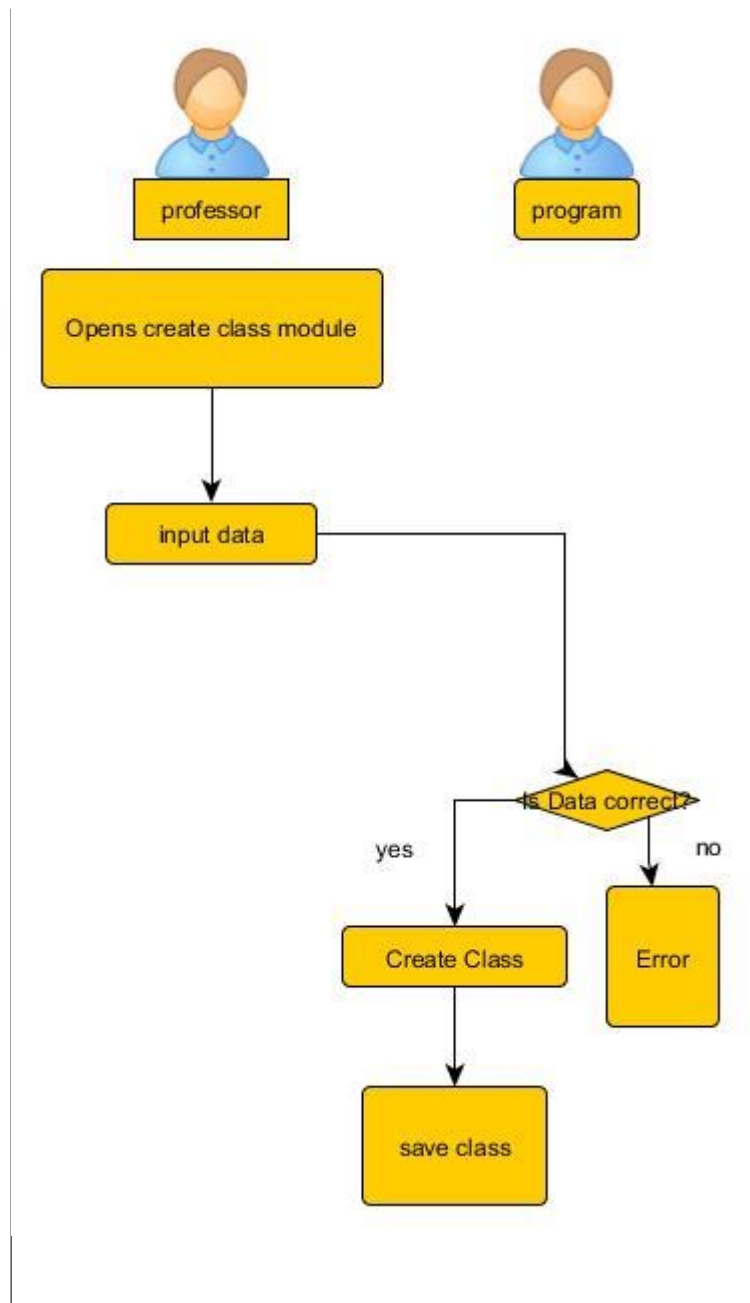
**Use-Case Name:** 1.1 Create Class

**Description:** *The user (henceforth called Prof) can create a class by providing its name and number of sessions. This class is then added to the Prof's total classes. Classes must have unique names.*

**Preconditions:** None

**Flow of Events:**

<b>Scenario Name</b>	<b>Description</b>
Scenario 1 (Basic Flow) Instructor creates a class	<ol style="list-style-type: none"><li>1. The instructor opens the create class module.</li><li>2. The instructor inputs the data necessary for creating the specific class such as the name and number of sessions.</li><li>3. The program checks the data for errors and invalid input such as duplicate names.</li><li>4. The program saves the data input by the instructor.</li><li>5. The class is finally created.</li><li>6. The program saves the created class.</li></ol>
Scenario 2 Instructor fails to create a class	<ol style="list-style-type: none"><li>1. The instructor opens the create class module.</li><li>2. The instructor inputs the data necessary for creating the specific class such as the Name and number of sessions.</li><li>3. The program checks the data for errors and invalid input such as duplicate names.</li><li>4. The program found an error on the data input by the instructor.</li><li>5. The program fails to create a class and sends an error.</li></ol>



*Activity Diagram of the Flow of Events:*

*Postcondition:*            Either a new class is created, or no new class is created.

*Relationships:*           This use-case extends “1.0 Maintain Class.”

*Special Requirements:*  
None