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
1st 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 "1.3 Edit Class."

Target Audience:

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

Revision Control

History Revision:

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

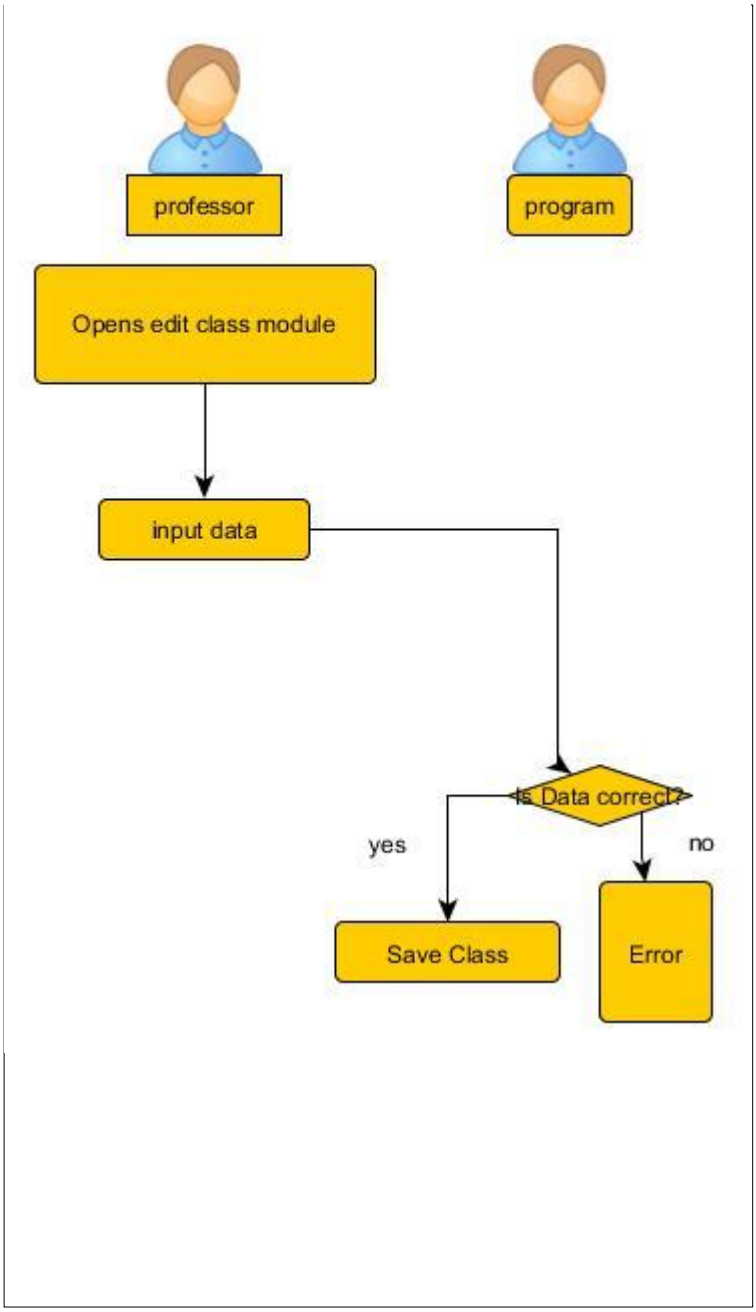
Use-Case Name: 1.3 Edit Class

Description: *The professor can edit a class by changing its name, number of sessions and students in the class. A new class is not created from this action. The edited name cannot be a duplicate of an already existing class*

Preconditions: A class **must exist** in order to edit one.

Flow of Events:

Scenario Name	***Description***
Scenario 1 (Basic Flow) Instructor Creates a class	1. The instructor opens the create class module. 2. The instructor edits the data necessary for the specific class such as the name and number of sessions. 3. The program checks the data for errors and invalid input such as duplicate names. 4. The program saves the data input by the instructor. 5. The class is finally edited with the inputted data 6. The program saves the edited class.
Scenario 2 Instructor fails to create a class	1. The instructor opens the create class module. 2. The instructor inputs the data necessary for creating the specific class such as the name and number of sessions. 3. The program checks the data for errors and invalid input such as duplicate names. 4. The program found such an error on the data input by the instructor. 5. The program fails to edit the class and sends an error.



Activity Diagram of the Flow of Events:

Postcondition: The edited class has its name or number of sessions changed with the new data inputted by the Prof.

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

Special Requirements:

None