Software Project Management Plan for Learning Management System

## Introduction

### Objectives

The objectives of this project are to develop a basic Learning Management System(LMS) as a final project for CS 3321 Introduction to Software Engineering. In practice, a LMS deals with a wide array of information from student details, college details, course details, college curriculum, and much more. It tracks the details and information of a student from the first day they enroll in the college to the day they complete their last course and apply for graduation.

This project, however, will not need to include such a diverse array of details. The LMS we will have to develop for this final project will only store and retrieve partial information regarding the current semester. This will include the student’s name, the student’s identification number, the course the student is registered in, each exam score for the student’s courses, and a way for the student to calculate their GPA for the current semester. Finally, there should be two access modes for this LMS: Administrator and User. Administrators have to manage the system, so they will need to be able to edit the data found within the LMS. Users on the other hand simply need to be able to view this data, and not be able to edit it.

### Major Functions

The major functions of this LMS will be the following: Logging in as either administrator or user, editing data if the individual logging in is an administrator, and viewing data if the individual logging in is a user.

To determine whether or not the individual logging in is either an administrator or a student(user), the username and password the user enters upon login will be cross referenced with the stored username and password information of each individual.

Once they successfully login, a boolean value “isAdmin” will be either true or false. This value will be passed as an argument to the instance of UserFunctions class. The type of access the individual logging in acquires will be determined by this value.

The GUI will display the same way for all users except for administrators. The sole difference on the administrator GUI will be the edit button. This button will allow the administrators to access the addCourse(), deleteCourse(), addGrade(), and deleteGrade() methods of the UserFunctions class.

## Schedule

### Work Breakdown Schedule

As there is only one developer working on this LMS, all of the initial work will be completed by this developer. First, they will draw out basic use case flow diagrams. Then they will draw out basic UML diagrams for all classes involved in the project. This will allow the developer to have a clear understanding of how the LMS will work, and what they need to do once programming begins. After these steps are completed, the developer will begin working on the Software Project Management Plan(SPMP). This is important for the initial developer to have, and will be useful for any future developers who work on the project. A SPMP is essentially a roadmap for a project, and without one the developers will get lost. Finally, the developer will begin to program the project before releasing it for open source development on github.

## Project Resources

### People

For this project, there will only be one individual working on it to start off. Afterwards, depending on the popularity of the project, a community of users may form. This community will offer feedback on what is currently wrong with the project, and what could potentially be added to the LMs in the future. From this community of users, a few of them may want to contribute to the project, and will be able to do so due to the open source nature of the project. Working with the initial developer through github, these users will be able to add their own code to this project. This added code could either be in the form of bug fixes, or new functional additions to the LMS.

### Hardware and Software

This project will be programmed using Java. Java was chosen due to the fact that projects programmed in this language can be run on a wide array of systems. This will give our project flexibility, and allow a wider range of users to utilize this LMS. Java is also an Object Oriented Programming Language. These types of languages allow developers to focus on creating a program that has high cohesion and low coupling between modules. This is important because it allows us to fix bugs without tearing down the whole system, and it also allows us to add additional functions without drastically changing the functions already performed by the LMS.

## Staff Organization

### Team Structure

This project will be completed using an Open Source Team Structure. Initially, this project will have only one team member. This initial developer will do the work discussed in the “Work Breakdown Schedule” section of this SPMP. From there, they will release the source code, and all other relevant documentation on github. This will allow future users to easily suggest bug fixes and additional functions for the LMS, and even implement these suggestions themselves if they so choose.

The Open Source Team Structure gives this project a lot of flexibility, as it allows for a broad range of ideas and perspectives on the project.

### 

### Management Reporting

All forms of management reporting will be carried out through github, as stated in the previous section. The initial developer will act as the project lead. All bug fixes and functionality additions to the LMS will be sent through github for the project lead to review before adding them to the project.

## 

## Risk Management Plan

### Risk Analysis

There is one main risk associated with this project. This risk is the fact that due to the open source nature of the project, it will depend heavily on acquiring an initial user base. Without these users, the project will never gain traction, and nobody will ever contribute to the project. It will simply remain a one developer project.