**Software Project Management Plan (SPMP)  
A Software Subsystem of Learning Management System**Chris Felicitas, Luis Cerillo, Sonny Stuhr, Aaron Gonzalez4/17/2019

Table of Contents

**Introduction1**

1.1 Project Summary1

1.2 Purpose, Scope, and Objective1

1.3 Project Deliverables2

1.4 Evolution of the SPMP2

**Project Organization2**

2.1 Organizational Structure2

2.2 Project Responsibilities2

2.3 Process Model3

**Technical Processes3**

3.1 Development Tools/Environment3

3.2 Diagrams4

3.3 Version Control5

3.5 Data Storage6

3.6 Test Cases6

**INTRODUCTION  
1.1 PROJECT SUMMARY**Learning management systems (LMS) are used to manage and record student data throughout their education career. The LMS is used to provide students, teachers, and administrators around the clock access to important educational data. This project seeks to develop a software engineering process specifically for the student user and admin of the LMS.

**1.2 PURPOSE, SCOPE, AND OBJECTIVE**The purpose of this project is to provide the client with a student information management system. The project is to store and retrieve a student’s name, student’s ID, registered courses in the current semester, each exam’s score in one course, GPA calculation in the current semester. The objective is to provide two different views of the software to the administrator and the user. The administrator will be able to manage the student information management system. Also, the admin will be able to insert update and monitor the whole process. The user, which may be a student or a teacher, will only be able to view the details of the student.**1.3 PROJECT DELIVERABLES**The following will be produced by the project:

* A **Software Project Management Plan** detailing the objective, scope, and process in which the Learning Management System will be designed and implemented.
* Software Prototype
* CRC Cards
* Use Case Diagrams
* Source Code

**1.4 EVOLUTION OF THE SPMP**As the team investigates further into developing the Learning Management System, changes to the SPMP we be made weekly after the team meeting reviewing the project management process. If changes are made, this document will be revised, and the new version will be distributed via GitHub.

**Project Organization  
2.1 Organizational Structure**The internal management structure we chose for this project was the modern hierarchical programming team, with a mix of agile process teams (check each other’s code) and open-source teams (contribute on GitHub). Below is a hierarchal chart the team agreed to at the inception of the project.

TEAM LEADER  
CHRIS FELICITAS

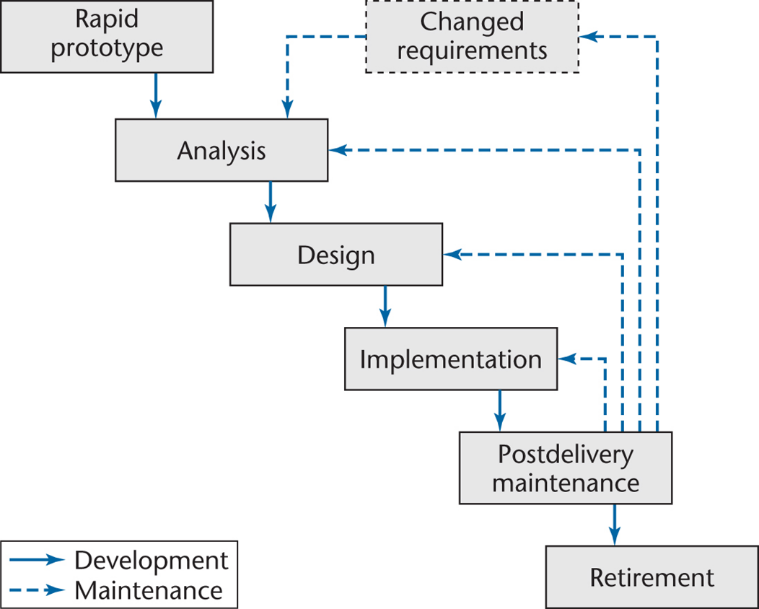
LUIS CERILLO  
PROGRAMMER

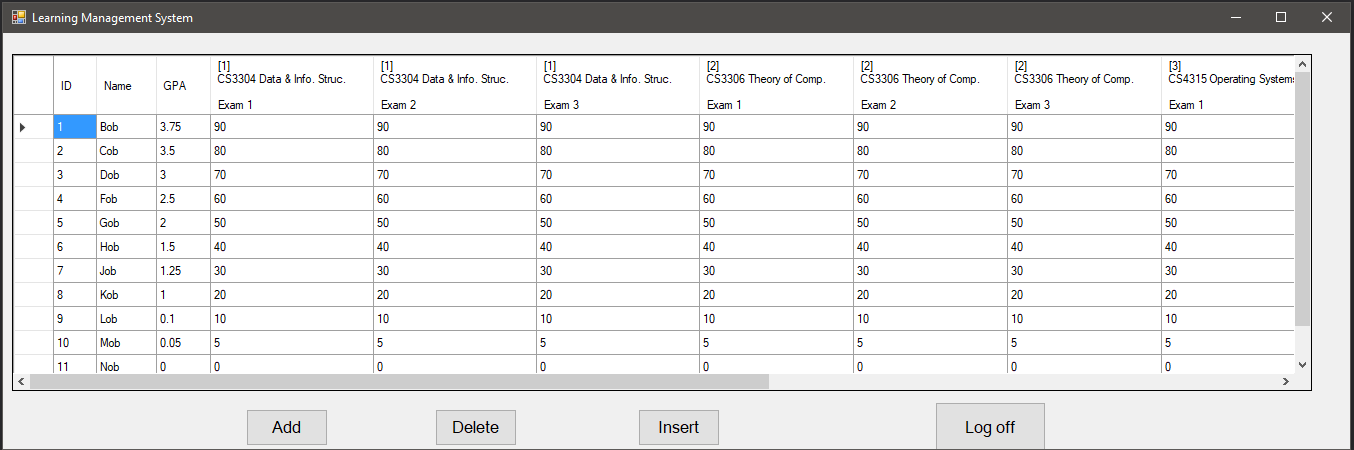
AARON GONZALEZ  
PROGRAMMER

SONNY STUHR  
PROGRAMMER

**2.2 Project Responsibilities**Describes the roles & responsibilities assigned to each team member.

|  |  |
| --- | --- |
| **Name** | **Role & Responsibilities** |
| Chris Felicitas | Team leader, and lead programmer program the rapid prototype. Develop checklist to ensure program specifications. |
| Luis Cerillo | Programmer; produce C# code to implement functions to the software. |
| Sonny Stuhr | Programmer; produce C# code to implement functions to the software. |
| Aaron Gonzalez | Programmer, document progress through SPMP documentation and develop use case and UML classes. |

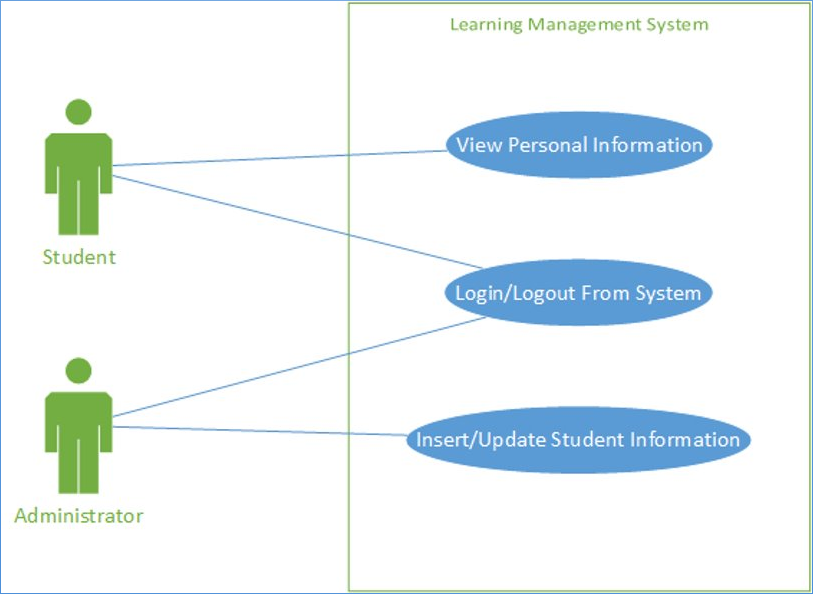
**2.3 Process Model  
**  
For the project we decided to use the rapid-prototype model. In order to see which programming language would be able to construct the software Chris Felicitas developed a rapid prototype which demonstrated that the project could be used C#.



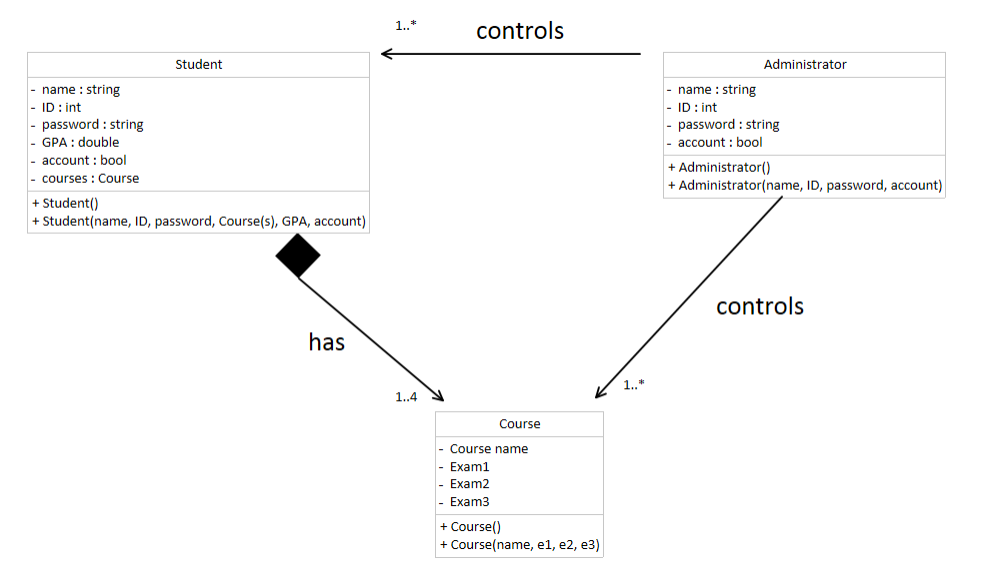
**Technical Processes  
3.1 Development Tools/Development Environment**Based on the team’s previous experience and knowledge the following software development environment (tools, languages, and operating systems) will be used for this project

* OS – Windows
* Development
  + Language – C#
  + Development IDE – Microsoft Visual Studio
* Google Hangout
* Testing
  + Unit Testing

**3.2 Diagrams**Our team will use standard UML diagrams to represent data, relationships, and requirements. Below is an example of a use case diagram that we have provided.

**Use Case Diagram**  


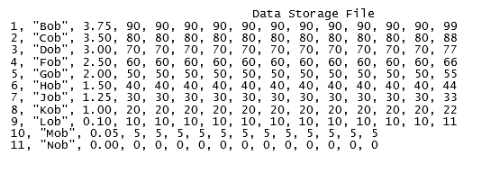
**Class Diagram**

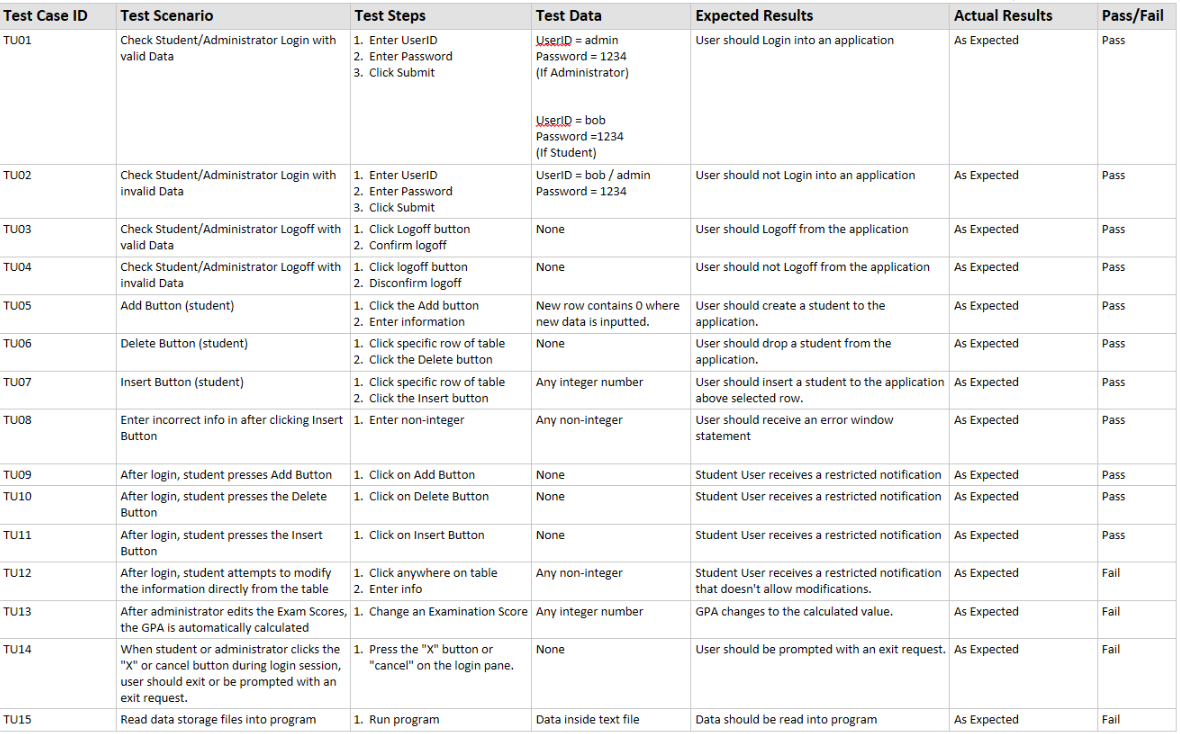
****

**3.3 Version Control**

|  |  |
| --- | --- |
| **Date** | **Changes** |
| 02/27/19 | A prototype of the Learning Management System was created. |
| 04/3/19 | Student.cs, Administrator.cs, Course.cs, Student.CS, Create Course.cs updated. |
| 04/13/19 | Added Student and Course classes to LMS prototype. |
| 04/16/19 | Added “add” and “delete” buttons, renamed Administrator.cs variable . |
| 04/17/19 | Added Insert button temporary. |
| 04/17/19 | Added Log Off button, removed Modify, Administrator/Student Login |
| 04/24/19 | Added student login (for one person only:bob) |
| 04/24/19 | Added side scroll on table, spaced out columns. |
| 04/28/19 | Refined code, added Data Storage File.txt |

**3.5 Data Storage**

****

**3.6 Test Cases**