**SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)**

The basic template to be used is derived from IEEE Std. 1058-1998, IEEE Standard for Software Project Management Plans. The following is a template for the SPMP. Provides a detailed overview of the procedures which go towards the completion of the software project. It surfaces the managerial and technical processes and showcases a general organization of a software product.

Software Project Management Plan for

<Project Title: Learning Management System>

<Contributors: Jesus Banda, Eric Martinez, Lim Nguyen, Simon Phan Tyler Thornton >

<Date: November 26, 2019>

Version Release Responsible Party Major Changes Date

0.1 Initial Document Release for Comment

**TABLE OF CONTENTS**

Build the table of contents here. Insert it when you finish your document.

1. **Introduction / Overview**

This collaborative project seeks to create a foundational *platform* application for management of student accounts and gradebooks with multi-use for either k12 or universities.

* 1. **Project Overview**

This platform allows for the management of registered students and their respective campuses administrators. Comparable to already established third party applications like *Blackboard.com.* The collaborators will work together to make this an easily accessible and free product; thus, the budget will be limited.

* 1. **Project Deliverables**

Primary deliverables for the client are to provide a product which will efficiently store student data with high consideration for security of the data.

* 1. **Evolution of the SPMP**

The programming team will go about developing the product through means of an iteration and incrementation life-cycle model and will use cloud repositories to collaborate on the product accordingly.

**PROJECT ORGANIZATION**

This section specifies the process model for the project and its organizational structure.

2.**1 Process Model**

We decided to follow the iteration and incrementation life-cycle model.

2.2 **Organizational Structure**

We are using the democratic team organization approach.

**2.3 Project Responsibilities**

Identify and state the nature of each major project function and activity and identify the individuals who are responsible for those functions and activities. Tables of functions and activities may be used to depict project responsibilities.

Programmer – Jesus Banda, Tyler Thornton, Lim Nguyen, Eric Martinez, Simon Phan

Secretary – Eric Martinez, Lim Nguyen, Tyler Thornton

**MANAGERIAL PROCESS**

This section of the SPMP specifies the management process for this project.

**3.1 Management Objectives and Priorities**

Complete the assignment as assigned in the rubric.

Make the finished product aesthetically pleasing.

**3.2 Assumptions, Dependencies, and Constraints**

The project is dependent on members being able to work independently and be able to combine what we have toward the project. Each member will have to look up any information or algorithm that is needed for the program.

## **3.3 Monitoring and Controlling Mechanisms**

The team communicates through GroupMe, GitHub, and personal meetings.

We share information at our own personal discretion.

3.4 **Staffing Approach**

Members were chosen in class.

**TECHNICAL PROCESS**

4.1 **Methods**, **Tools**, **and** **Techniques**

For this project we came to the agreement to use Visual Studio, since most of the team is familiar with C++. For the GUI, Visual Studio has a program already included so we are going to use that. For the Database we used text files. The programmers can use any technique and method if it is comprehendible to the others or the programmer can easily explain it to the others. The programmer is to finish a version of their program and hand it to another programmer for testing and to help updated it.

4.2 **Software** **Documentation**

For documentation we will have the crc cards, uml, use cases, to guideline our project, and be uploaded to GitHub.

**WORK PACKAGES, SCHEDULE & BUDGET**

Specify the work packages, dependency relationships, resource requirements, allocation of budget and resources to work packages, and a project schedule. Much of the content may be in appendices that are living documents, updated as the work proceeds.

5.1 **Work Packages**

Specify the work packages for the activities and tasks that must be completed in order to satisfy the project agreement. Each work package is uniquely identified. A diagram depicting the breakdown of project activities and tasks (a work breakdown structure) may be used to depict hierarchical relationships among work packages.

5.2 **Dependencies**

Specify the ordering relations among work packages to account for interdependencies among them and dependencies on external events.

Techniques such as dependency lists, activity networks, and the critical path method may be used to depict dependencies among work packages.

5.3 **Resource** **Requirements**

The program should run without issue on any personal computer produced in the last decade.

5.4 **Budget and Resource Allocation**

No Budget

5.5 **Schedule**

We operate via rough guidelines, trusting that all members will complete their chosen sections in a reasonably timely manner.

END OF DOCUMENT