Software Project Management Plan (SPMP)

Learning Management System Project

Cory, Vernon, Alexis, Norma, Rohan

29 April 2019

**Table of Contents**

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

**1.Introduction**   
This project is a learning management system and should be able to work for both a student and an instructor. It is supposed to mimic blackboard in providing both an instructor and student to navigate grades, see classes, and assignments.

**1.1 Project Overview**

This project is supposed to be able to run efficiently and include information such as student’s name, student’s ID, courses enrolled, exam scores, and GPA calculations. Apart from this information it should also be able to add, delete, insert, and modify records.

**1.2 Project Deliverables**

The complete product including all documentation will be delivered by the 25th of April 2019.

**1.3 Evolution of the SPMP**

This plan will be completed by dividing the work between all group members. The chief leader, Vernon, will assign each group member a task to complete. We will all show our progress to the group and submit what we have through GitHub. For any updates the chief leader will look at them and decide what changes need to be made.

**1.4 Reference Materials**

The project will reference a learning management system.

**1.5 Definitions and Acronyms**

LMS- Learning Management System

**2. Project Organization**

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

**2.1 Process Model**

The process model that we decided to go with is the classical chief programmer team. We felt like this was the best fit for the number of team members in the group and the amount of workload we had.

**2.2 Organizational Structure**

The responsibilities are separated by chief programmer, Vernon, for backup programmer, Cory, and for programmers we have Alexis, Norma and Rohan.

**2.3 Organizational Interfaces**  
Describe the administrative and managerial interfaces between the project and the primary entities

with which it interacts.

A table may be a useful way to represent this information.

Organization  
Customer: <name> Subcontractor: <name> Software Quality Assurance Software Configuration Management  
<etc.>

Table F-1. Project Interfaces

Liaison <name>

Contact Information <phone, email, etc.>

2

**2.4 Project Responsibilities**

The project responsibilities are split into the requirements needed. Cory and Vernon are in charge of the Java coding. Alexis and Norma are in charge of HTML for the visuals and SPMS. Rohan is in charge of the CRC cards.

**3. Managerial Process**

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

**3.1 Management Objectives and Priorities**

The ultimate goal of this project is to be able to be a management system for schools to use. It is supposed to have both administration and students be able to check their grades or update them. The main priorities is to have a confidential login system and an interactive one.

**3.2 Assumptions, Dependencies, and Constraints**

The constraints are as followed:

* Deadline has to be met
* Project has to be done in less than five months
* The product must reflect a Learning Management System
* It has to be a closed source system
* The product should be easily usable to users
* The product should have an administrative and student option

Some assumptions would be:

* We can assume that every person in the team will do 20% of the work
* We will have half the work done by week 7

**3.3 Risk Management**

The risk factors associated with the project are as follows.

There is a strict criteria and plan that must be met but no specific instructions to how the project wants to be presented. The biggest risk factor will be getting the back-end program to run with the HTML written by different programmers. The biggest risk we would face is trying to get everything to flow correctly when different parts are written by different programmers. If HTML does not work for the java code written, then we would need to use Java Swing for it. This will cause a problem with time and how long we would need to accomplish and have this done by the due date.

Since this project will not reference or use another code as reference we will also need to test it multiple times along the way to make sure the foundation is strong. It is vital that we conduct different tests as we are writing it to make sure that everything runs smoothly and nothing is off.

**3.4 Monitoring and Controlling Mechanisms**

All the monitoring will be done through the chief programmer. All the information and process that is done will be shown to give during class or will be posted on GitHub for the other group members to look over and give feedback on.

**3.5 Staffing Approach.**

The skills required for this project would be mainly HTML and java.

4. Technical Process

This section specifies the technical methods, tools, and techniques to be used on the project. It also includes identification of the work products and reviews to be held and the plans for the support group activities in user documentation, training, software quality assurance, and configuration management.

4.1 Methods, Tools, and Techniques

Identify the computing system(s), development method(s), standards, policies, procedures, team structure(s), programming language(s), and other notations, tools, techniques, and methods to be used to specify, design, build, test, integrate, document, deliver, modify or maintain the project deliverables

4

4.2 Software Documentation

Specify the work products to be built for this project and the types of peer reviews to be held for those products. It may be useful to include a table that is adapted from the organization's standard collection of work products and reviews. Identify any relevant style guide, naming conventions and documentation formats. In either this documentation plan or the project schedule provide a summary of the schedule and resource requirements for the documentation effort.

To ensure that the implementation of the software satisfies the requirements, the following documentation is required as a minimum:

4.2.1 Software Requirements Specification (SRS)

The SRS clearly and precisely describes each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement is defined such that its achievement is capable of being objectively verified and validated by a prescribed method, for example, inspection, analysis, demonstration, or test.

4.2.2 Software Design Description (SDD)

The SDD describes the major components of the software design including databases and internal interfaces.

4.2.3 Software Test Plan

The Software Test Plan describes the methods to be used for testing at all levels of development and integration: requirements as expressed in the SRS, designs as expressed in the SDD, code as expressed in the implemented product. The test plan also describes the test procedures, test cases, and test results that are created during testing activities.

4.3 User Documentation

Describe how the user documentation will be planned and developed. (This may be just a reference to a plan being built by someone else.) Include work planned for online as well as paper documentation, online help, network accessible files and support facilities.

4.4 Project Support Functions

Provide either directly or by reference, plans for the supporting functions for the software project. These functions may include, but are not limited to, configuration management, software quality assurance, and verification and validation. Plans for project support functions are developed to a level of detail consistent with the other sections of the SPMP. In particular, the responsibilities, resource requirements, schedules and budgets for each supporting function must be specified. The nature and type of support functions required will vary from project to project. The absence of a software quality assurance, configuration management, or verification and validation plan, however, must be explicitly justified in project plans that do not include them.

**5**

**5. Work Packages, Schedule, and Budget**

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**

Outside resources that are required would be Eclipse Java, Atom for HTML coding. Outside class time will be required to complete all the necessary material that is required of each member.

**5.4 Resource Allocation**

For resource allocation the three programmers will do their part and turn in their work to the chief programmer. The chief programmer will ensure that their code and resources are all met before moving on with the project. We will also all divide any writing requirements between each other and submit to chief programmer before anything is submitted.

**5.5 Schedule**

|  |  |
| --- | --- |
| Week 1 (Feb 3-9) | Assign roles to all group members and make sure everyone understands what they need to do |
| Week 2 (Feb 10- 16) | Chief programmer checks in on everyone to make sure they have started on their part and are on the right track |
| Week 3 (Feb 17-23) | Norma will start working on SPMP outline, Rohan on CRC cards, Alexis on research, Vernon will research on what to do for GUI, and Corey will look over the code written so far and give feedback |
| Week 4 (Feb 24- March 2) | We all will show the progress made on the assignments given prior to that |
| Week 5 (March 3- 9) | Continue working on what is required and put any finishing touches on it |
| Week 6 (March 10-16) | Make sure we have the presentation ready to show our progress so far and understand what we need to talk about |
| Week 7 ( March 17-23) | Present to the class what we have so far and what we need to do in the future |
| Week 8 ( March 24-30) | Start working on the GUI for the code (leaning towards HTML) |
| Week 9 (March 31- Apr 6) | Submit all paper work to chief programmer/ edit the other group members work/ continue testing out code |
| Week 10 (April 7-13) | Finish with all the paperwork required for the project |
| Week 11 (April 14-20) | Meet with group to make sure everything is done and complete ready to present and turn in, make any finishing touches if necessary |
| Week 12 (April 21-27) | Put all resources, artifacts, and documentation ready to present |