**Software Project Management Plan**

Learning Management System

Version 1.9

Team: Jacquelyn Johnson, Andrew Samuel, Julio Quintero, Andrew Tomich, Ary Hernandez

Due Date: December 1, 2019

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Description** |
| 9/5/19 | 1.0 | Document created |
|  | 2.0 | Document completed |

**Table of Contents**

Project Overview………………………………………………………………………………. 3

Project Organization ……………………………………………………………………………3

**1 Introduction**

This Software Project Management Plan (SPMP) will serve as a guide to the project and will include descriptions of all deliverables including deadlines. From reading this document, the client, a computer science professor, will see the software engineering planning practices undertaken by the team to deliver a full working LMS.

Today, the integration of education with the use of technology has become commonplace. The need for an LMS is now required especially in higher education. A Leaning Management System is a type of software used to provide documentation, reporting and tracking when delivering educational courses. Blackboard, Moodle, and Canvas are all examples of widely used learning management systems. The team is tasked to develop an LMS in similar scale in a time period of about a full academic semester.

**1.1 Project Summary**

This section of the Software Project Management Plan (SPMP) will give an overview of the purpose and objectives of the project. All estimated timelines are subject to change as the progress of the project will be updated in this document.

**1.1.1 Purpose, Scope and Objective**

The purpose of this project is to develop a LMS that is used to store important student data such as grades, student ID, registered courses, and GPA. It will contain two access modes, one for administrators and one for students. The project will be developed using Java and SQL programming languages. More specifically, Java will be used to develop the GUI and backend functionality while SQL will be used for database integration, the storing of records.

**1.1.2 Assumptions and Constraints**

The several assumptions and constraints will be outlined here. All of these serve importance to the team.

**1.1.2.1 Assumptions**

* The team has previous experience working with each other in previous semesters
* The team must and will work together in a productive manner
* The client will respond to all questions regarding the project’s development during the semester and provide feedback

**1.1.2.2 Constraints**

* Financial resources are not available. The team is not being paid or compensated in any way to develop this project
* As the assigning of the project, the team has about 14 weeks to complete it
* The team must use an online GitHub repository for version control
* The team will develop the project using Windows, Linux and macOS machines

**1.1.3 Project Deliverables**

The team will produce a working LMS that will be delivered by the due date of December 1, 2019 and satisfies all requirements listed in the project description. The project must have a functioning GUI and make use of a UML diagram during the development phase. Along with the complete product, an SPMP documenting the development of the project will be provided as well other necessary documents such test cases, and project workflows.

1. Project Plan 12.01.19
2. Requirements documents 12.01.19
3. Source code 12.01.19
4. UML diagrams 12.01.19
5. Test cases 12.01.19

**1.1.4 Schedule**

A schedule listing out project phases will be provided along with their date below:

|  |  |
| --- | --- |
| **Project milestone** | **Date** |
| Project created |  |
| GUI developed |  |
| Basic backend functions implemented |  |
| Database established |  |
| Database connection implemented |  |
| Project completed |  |

**1.2 Evolution of the SPMP**

All changes to the project will be documented in this SPMP. The Project management plan will be updated as development of the project continues. The document was created early September and is expected to be completed late November.

**2 Reference Materials**

* *IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans, IEEE 1998*
* Schach, Stephen R. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, 2011.

**3 Definitions and Acronyms**

* UML – Unified Modeling Language
* LMS – Learning Management System
* SPMP – Software Project Management Plan
* GUI – Graphical User Interface
* Java – A general purpose, class based, and object-oriented programming language
* SQL – “sequel”, a programming language used for database systems and storing data
* GitHub – online repository and version control
* SIMS – Student Information Management System

**4 Project Organization**

This section of the SPMP will outline the organizational structure of the project including roles and responsibilities of each team member.

**4.1 External Structure**

The client for this project is software engineering professor, Dr. Yuchou Chang. A document detailing the requirements for the project was provided to the team before actual development began. As well as this, the team will occasionally discuss the progress of the project with the client throughout the regular semester. Presentations are also planned throughout the semester where all team members must participate in.

**4.2 Internal Structure**

The project will make use a democratic team approach. Every team member is responsible for the project, and decisions will made in a democratic fashion. Java was chosen as the main programming language before the code was implemented.

**4.2 Organizational Structure**

Team Members

I. Andrew Tomich

II. Julio Quintero

III. Ary Hernandez

IV. Jacqulyn Johnson

V. Andrew Samuel

**4.3 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.

**4.3 Roles and Responsibilities**

All team members are responsible for the project whether by direct development of the software or by other methods. A table including roles and duties by each member is provided below.

|  |  |
| --- | --- |
| **Role** | **List of Responsibilities** |
| Team Leader | * Facilitate team discussion and ensure all members are actively working on the project * Help allocate team resources * Maintain team prosperity |
| Developer | * Develop source code for the software |
|  |  |
|  |  |

**5 Managerial Process**

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

**5.1 Management Objectives and Priorities**

Describe the philosophy, goals, and priorities for managing this project. A flexibility matrix might be helpful in communicating what dimensions of the project are fixed, constrained and flexible. Each degree of flexibility column can contain only one "X".

**5.2 Assumptions, Dependencies, and Constraints**

State the assumptions on which the project is based, any external events the project is dependent upon, and the constraints under which the project is to be conducted. Include an explicit statement of the relative priorities among meeting functionality, schedule, and budget for thi project.

**5.3 Risk Management**

Describe the process to be used to identify, analyze, and manage the risk factors associated with the project. Describe mechanisms for tracking the various risk factors and implementing contingency plans. Risk factors that should be considered include contractual risks, technological risks, risks due to size and complexity of the product, risks in personnel acquisition and retention, and risks in achieving customer acceptance of the product.

The specific risks for this project and the methods for managing them may be documented here or in another document included as an appendix or by reference.

**5.4 Monitoring and Controlling Mechanisms**

Define the reporting mechanisms, report formats, review and audit mechanisms, and other tools and techniques to be used in monitoring and controlling adherence to the SPMP. Project monitoring should occur at the level of work packages. Include monitoring and controlling mechanisms for the project support functions (quality assurance, configuration management, documentation and training).

A table may be used to show the reporting and communication plan for the project. The communication table can show the regular reports and communication expected of the project, such as weekly status reports, regular reviews, or as-needed communication. The exact types of communication vary between groups, but it is useful to identify the planned means at the start of the project.

Information From To Time Period

Communicated

Status report Project Team Project Manager Weekly

Status report Project Manger Software Manager, Project Weekly

Team

Project Review Project Team Software Manager Monthly <etc>

Table F-4: Communication and Reporting Plan

**5.5 Staffing Approach.**

Describe the types of skills required for the project, how appropriate personnel will be recruited, and any training required for project team members.

**6. 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.

**6.1 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

**6.2 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:

**6.2.1 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.

**6.2.2 Software Design Description (SDD)**

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

**6.2.3 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.

**6.3 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.

**6.4 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.

**7. Work Packages, Schedule, and 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.

**7.1 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.

**7.2 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.

**7.3 Resource Requirements**

Provide, as a function of time, estimates of the total resources required to complete the project. Numbers and types of personnel, computer time, support software, computer hardware, office and laboratory facilities, travel, and maintenance requirements for the project resources are typical resources that should be specified.

**7.4 Budget and Resource Allocation**

Specify the allocation of budget and resources to the various project functions, activities, and tasks.

**7.5 Schedule**

Provide the schedule for the various project functions, activities, and tasks, taking into account the precedence relations and the required milestone dates. Schedules may be expressed in absolute calendar time or in increments relative to a key project milestone.