Software Project Management Plan

Learning Management System

Version 2.0

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
11/30/19	2.0	Document completed

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 a team of undergraduate students 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 or roughly, 15 weeks.

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 an 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 and 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 15 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 as test cases, and project workflows.

I. Project Plan	An organized document that contains a project's planning in detail.
II. Requirements documents	Documents that outline the requirements for a project given by the client.
III. Source code	Programmable code that will be used to run the software.
IV. UML diagrams	Diagrams showcasing software modeling. Made to visually represent a working system.
V. Test cases	Set of conditions that are tested to determine if a system satisfies a requirement.

1.1.4 Schedule

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

Project milestone	Date
Project created	September 2019
GUI developed	September 2019
Basic backend functions implemented	October 2019
Database established	October 2019
Database connection implemented	October 2019
Project completed	November 2019

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
- IDE – Integrated Development Environment

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 in which where all team members must participate in and showcase progress regarding the project.

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. Additionally, the team will vote on any major adjustments to the project if an alternate method wants to be implemented. Therefore, mutual respect among team members is crucial. Another reason for opting for the democratic approach is for the high productivity, and the team is eager come together and fix any problems during the development stage.

Team Members

     I.   Andrew Tomich
    II.   Julio Quintero
   III.   Ary Hernandez
   IV.   Jacquelyn Johnson
    V.   Andrew Samuel

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
GUI Programmer	- Write source code for the software - Establish back-end and front-end functionality
Database Programmer	Create database to hold records

	- Integrate database to main frame - Establish database tables
Team member	- Provide support to the team wherever necessary - Complete side tasks not covered by main programmers

5   Managerial Process Plan

This section of the SPMP will outline the start-up plan, work plan, control plan, risk-management plan, and project close out plan.

5.1   Start-Up plan

5.1.1   Estimation plan

The estimated time to develop the software will approximately last one full academic semester, or 15 weeks as previously stated. This estimation comes from the date the project was assigned and the due date along with it.

5.1.2   Staffing plan

The team has a fixed staff that was formed at the beginning of the semester. It consists of 5 students, more information regarding the structure of the group can be found in section 4.

5.1.3   Resource acquisition plan

All necessary resources are expected to be available during the entire duration of developing the project. The resources include:

- Hardware resources
  - Each team member has their own PC

- Software resources
  - NetBeans IDE
  - Eclipse IDE
  - mySQL workbench

- Additional resources provided by the university

5.1.4   Project staff training plan

No additional training is required.

5.2   Work plan

5.2.1   Work activities and Schedule

The overall work plan for development will be listed here

Week 1	Project assigned by client; requirements are studied
Week 2 -3	Create Software project management plan, begin planning development
Week 4 - 11	Development of the project begins
Week 12	Class testing and documentation

Week 14	End-of-semester team presentation

5.2.2   Resource allocation

Team members are expected to be working on the project throughout the whole semester. Meetings with the client will be done during regular lecture meeting days to ensure progress. Team members will also meet at the end of lectures to discuss the progress of the project and ensure healthy productivity.

5.3   Control plan

All major changes to the development of the project must be agreed upon by the whole team. The software will be developed iteratively and incrementally throughout its lifespan.

5.4   Risk management plan

The SPMP will specify the risks involved in the project as well as the methods used to prevent them in this sub section.

- Identify, analyze, and prioritize any risk
- Contingency planning
- Extensive unit testing to prevent design issues
- Multiple available machines in case of breakdowns

5.5   Project close out plan

The project must be submitted at the end of the semester including all artifacts which include necessary documents and source code.

- Provide client all documentation and artifacts
- Open GitHub repository publicly for client review
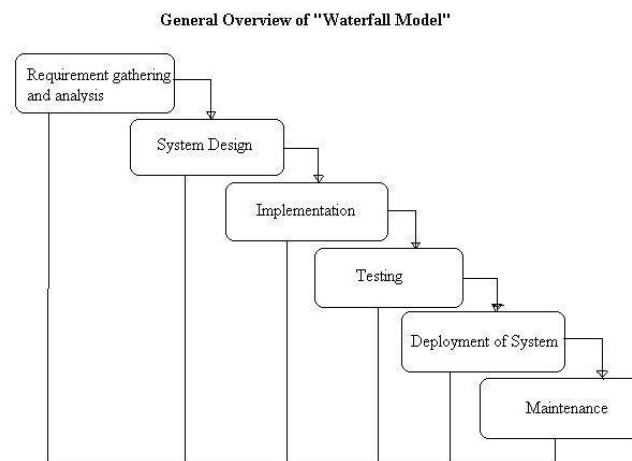- Archive all artifacts for future reference

6 Technical process plans

The SPMP will document the process model used in the project along with methods, tools, and techniques that were used in development. Additionally, an infrastructure and product acceptance plan will be outlined in this section.

6.1 Process plan

The waterfall model will be used for the development of the project. It will work in a set of standard phases. First, the requirements will be analyzed and gathered. The team will familiarize themselves with the requirements along with deliverables. The requirements phase includes preparing the appropriate documents including this SPMP for the client. Second, the team will begin system design. This includes choosing our programming language and acquiring hardware. Third is the implementation phase, this includes building the software from scratch with source code. Next begins testing, in this stage we ensure the project is being built to the specifications set forth by the client. Unit testing was performed extensively throughout development. Finally, after rigorous testing ensuring the product is stable and performing up to standards, the project is then finalized and deployed. This project will not include post-delivery maintenance so the team must ensure everything runs fine by the due date.

The waterfall model was chosen because of its simple design. It was one of the first life cycle models the team learned about in the semester and we wished to adopt it for our software project. Also, waterfall models tend to be best suited for smaller scaled projects where requirements are well defined, which they are in this instance. Lastly, it's very straightforward, each phase must be completed before the next and this allows an easy way to observe milestones.

General Overview of "Waterfall Model"

```
Requirement gathering
and analysis
        System Design
                Implementation
                        Testing
                                Deployment of System
                                        Maintenance
```

6.2 Methods, tools, and techniques

The team will use object-oriented programing concepts for the development of the project. As the semester progresses, the team will also plan to adopt new software engineering practices as they are introduced, if applicable. The source code will be written in the Java programming language and SQL for the database integration.

6.3   Infrastructure plan

The team has access to their own personal computers for working on the project. Additionally, the university provides computers if necessary. The machines include Windows, Linux, and macOS operating systems.

6.4   Product acceptance plan

The acceptance of the product will be ensured by following appropriate software engineering practices. The project description document given by the client will be followed closely to make sure their needs are met.

7 Supporting process plans

This section of the SPMP will include plans for configuration management, testing, documentation, quality assurance, reviews and audits, problem resolution, subcontractor management, and process involvement – if applicable.

7.1 Configuration management plan

No configuration management for artifacts pertains to this project. All artifacts, however, will be held to standard software engineering practices.

7.2 Testing plan

Testing will be conducted throughout the entirety of the project. The most common will be unit testing. During development, various test cases will be conducted to ensure a positive user experience. Some examples would be testing the login module and making sure it works when entering valid information such as the username and password.

7.3 Documentation plan

As previously stated, there will be some documents that will be submitted to client upon the project's completion. More of this can be found in section 1.

7.4 Quality assurance plan

All aspects of the project are subject to quality assurance checks by team members. No formal document outlining software quality assurance exists.

7.5 Problem resolution plan

Any major problems encountered during development will be handled by the team leader. As of the documentation of this management plan, no issues have arisen. In the case of a problem that is unable to be fixed internally, the team will resort to contacting the course professor for assistance.