

# SEUHDSpring2020

Software Project Management Plan

Rosendo Chavez ;Duy Truong ;Herson Lazo ;Michael Ly;  
Jonathan Landry  
5-3-2020

## **Table of Contents**

<b>1 Introduction.....</b>	<b>2</b>
1.1 Project Overview.....	2
1.2 Project Deliverables .....	2
1.3 Evolution of the SPMP.....	2
1.4 Reference Materials .....	2
1.5 Definition & Acronyms.....	2
<b>2 Project Organization.....</b>	<b>3</b>
2.1 Process Model.....	3
2.2 Organizational Structure and Interfaces.....	4
2.3 Project Responsibilities.....	4
<b>3 Project Management and control .....</b>	<b>5</b>
3.1 Project Management Objectives and Priorities .....	6
3.2 Assumptions, Dependencies and Constraints .....	6
3.3 Risk Managements .....	6
3.4 Issue Resolution .....	7
<b>4 Technical Process.....</b>	<b>8</b>
4.1 Methods, Tools, and Techniques .....	8
4.2 Software Documentation .....	8
4.3 Documents.....	9
4.4 Schedule .....	9

## **1. Introduction**

This project will be done by our team to create a BlackBoard 2.0. The team consists of undergraduate students at the University of Houston-Downtown. This project is meant to be completed as a semester project to teach cooperation and to further enforce and build upon coding strategies.

### **1.1 Project Overview**

This project will consist of creating a learning management system to record, update, and delete any of the student's basic information. The software system only stores and retrieves students' partial information in the current semester and other basic information including 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.

### **1.2 Project Deliverables**

The software project management plan is under version control. Proposed changes and new versions of the plan will be released on GitHub and an announcement will be sent through text messages to be made available to all project members.

### **1.3 Evolution of the SPMP**

Since the foreseeable future is never clear, any updated or new information gathered will force the updating of this document. Ensuring efficiency and accuracy said team members will update, report, and distribute to the team during the weekly meeting.

### **1.4 Reference Materials**

"How To Calculate Semester GPA." GPA Calculator, [gpacalculator.net/how-to-calculate-gpa/semester-gpa/](http://gpacalculator.net/how-to-calculate-gpa/semester-gpa/).

Gilhooly, Kym. "Making E-Learning Effective." Computerworld, Computerworld, 16 July 2001, [www.computerworld.com/article/2582540/making-e-learning-effective.html](http://www.computerworld.com/article/2582540/making-e-learning-effective.html).

Meerts, John. Course Management Systems . EDUCAUSE Evolving Technologies Committee, 20 Oct. 2003. PDF File.

## 1.5 Definitions and Acronyms

Actor – Models a type of role played by an entity that interacts with subjects (e.g, by exchanging signals and data), but which is external to the subject.

eLearning – learning conducted via electronic media, typically on the internet.

GitHub - A web-based version-control and collaboration platform for software developers.

Graphic User Interface (GUI) - A category of interfaces that allows a user to visually interact with software (usually via mouse or touch screen), as opposed to traditional text-based interfaces.

Hypertext Markup Language (HTML) - The standard markup language for documents on the web. HTML forms the building blocks for the content of a webpage.

JavaScript- A programming language typically tied into a client's web browser. JavaScript is typically used to make rendered pages more interactive.

Learning Management System (LMS) – an LMS is a software program that helps you create, manage, and deliver eLearning courses.

Universal Modeling Language (UML)- A collection of tools for abstractly modeling software systems

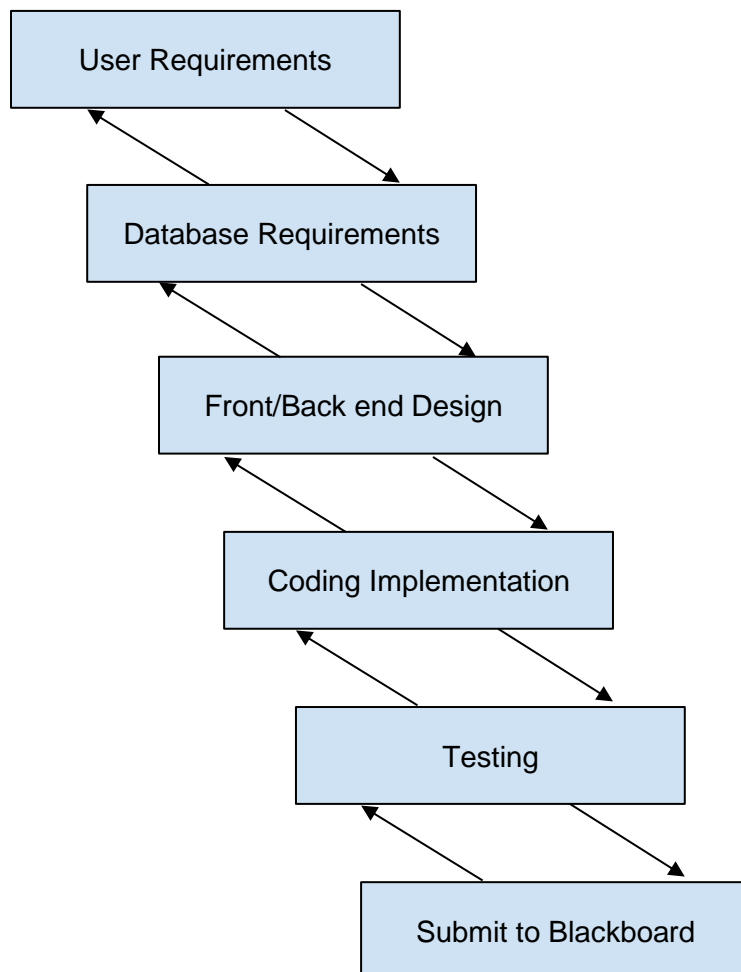
Use case- An algorithmic description of a user's interactions with a system.

## 2 Project Organization

This section contains details pertaining to the projects architecture which includes the process model, organizational structure, interfaces, and responsibilities.

### 2.1 Process Model

The project is initiated on January 14, 2020 and will be terminated by the end of the semester May 3, 2020.



## 2.2 Organizational Structure and Interfaces

Members within BlackBoard 2.0

NAME	Role & Responsibilities
Duy Truong	<b><u>Lead Developer</u></b> 1.Implementation of software changes 2.Updates software
Michael Ly	<b><u>Project Manager</u></b> 1.Plans, organizes and coordinates the activities of the group. 2.Helps the team to produce, assess, updating and reporting of the software.
Herson Lazo	<b><u>System Analyst</u></b> 1.Write and maintain system documentation. 2.Prepare flow charts and diagrams for systems capabilities and processes.
Jonathan Landry	<b><u>Quality Control</u></b> 1.Compares software to documents 2.Test changes made by Developer
Rosendo Chavez	<b><u>System Analyst</u></b> 1.Write and maintain system documentation. 2.Prepare flow charts and diagrams for systems capabilities and processes.

## 2.3 Project Responsibilities

Describes each phase of the project and the member of the team in charge.

Tasks	Leader
Initial planning process/Scheduling	Full team
Documentation	Rosendo Chavez
UML Diagram creation	Herson Lazo
Data collection	Michael Ly
Database design	Duy Truong
Testing	Jonathan Landry
SPMP creation	Rosendo Chavez

## **3 Project Management and Control**

This section contains various objectives, assumptions, and risks related to the project.

### **3.1 Project Management Objectives and Priorities**

BlackBoard 2.0's objective is to replicate and improve the service provided by any typical LMS (BlackBoard or Moodle) for educational institutions. The key solution for this objective will be to (insert when I have an idea to finish this sentence).

#### **Team Meetings:**

We have tried to have face-to-face meetings before the outbreak of COVID-19 but having been met with mandatory stay-home orders with exemptions of essential workers that became improbable. With that said we had resorted to have an online meeting at least once a week announced and agreed upon during a group message. During these meetings, the team will discuss the status of work, any problems that may arise, and what future steps to continue.

### **3.2 Assumptions, Dependencies and Constraints**

This section consists of assumptions, dependencies, and constraints based on the project.

#### **3.2.1 Assumptions**

- It is assumed that users will use this database to register, maintain and create classes for school.
- It is assumed each student/faculty will have their own username and password to log into the database.
- It is assumed that the administrator will be able to monitor the database while in use.

#### **3.2.2 Dependencies**

- Web based project created in MySQL
- Project completed before May 3, 2020
- Must have separate views for Student, Faculty and Administrators

#### **3.2.3 Constraints**

- There will be time constraints based on the length of the semester and from any team members commitments.
- There will be constraints on getting everyone familiar with Github and MySQL.
- There will be health constraints because COVID-19 may affect the team.

## **3.3 Risk Managements**

Risk management is used to identify and lessen potential sources of expense or delay. A few risks will be common to every project phase, and other risks are based on particular phases.

**Team member unavailability:**

**Probability: Highly**

**Impact: Low-High**

**Prevention:** Team members should alert team of any potential absences if possible, and then coordinate with other members of the team to cover their responsibilities for the duration.

**Correction:** Excessive or unannounced availability will trigger a team discussion. Continued unavailability will need to be dealt with professor and may lead to removal from the team.

**Miscommunication & or lack of:**

**Probability: High**

**Impact: Medium**

**Prevention:** Prevention methods are documentation and verbal communication verification.

**Correction:** Incase miscommunication persists then the team will need to hold meeting to reassess documentation and verification process.

**Requirements Elicitation and Analysis Risks**

**Forgotten Stakeholder:**

**Probability: High**

**Impact: Medium**

**Prevention:** Team must make sure it knows its intended users and their intended needs are.

**Correction:** Team will need to implement and correct areas that the forgotten stakeholders need by reanalyzing and changing any aspects needed.

**Software Implementation Risks**



### **Lack of Experience with Relevant Technologies:**

**Probability: High**

**Impact: High**

**Prevention:** Members will be assigned to task that suit their technologic experience. If a member is face with an aspect, they do not understand they should ask for help from members that understand said aspect.

**Correction:** Multiple failures to keep up with assignments due to inexperience can lead to reshuffling of assignments between members.

### **Software Testing Risks:**

#### **Non-representative Tests:**

**Probability: High**

**Impact: Low**

**Prevention:** BlackBoard 2.0 will need to make sure that their testing processes and methods are up to date and applicable.

**Correction:** During refining, new test should be made to give a wider range to uncover defects.

## **3.4 Issue Resolution**

Explains how issues are resolved between teammates.

Issues will arise as the project evolves. Most issues can be handled through discussions either in person or through messaging. For any complex or conflicting issue, the team will need to document or ask the professor for issue resolution help.

## **4 Technical Process**

The technical process model describes the methods that the team will use in representing the technical details of the project.

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

This section contains an outline of any specific plans, methods or tools used during the duration of said project.

**Diagrams:** BlackBoard 2.0 will use standard UML diagrams to represent data, relationships, and requirements.

Phase	Description	Model/Diagram
Elicitation	Development team will meet to identify the actors of the system and how they interact with the system.	Use Cases
Specification	Those developing the system will analyze the various classes involved in creating the system and how they interact with each other.	Class iterations

#### **Programming Languages & Tools:**

BlackBoard 2.0 will utilize multiple programming languages and tools. For the web-based aspect of this project, said languages used are HTML, Javascript, Python, and other tools that are determined, as necessary.

## **4.2 Software Documentation**

**First Draft:** First draft, assigned Team Members will outline major sections and subsections. Any questions or concerns raised will be dealt with during the next possible meeting.

**Second Draft:** Second draft will be expanded upon and made more specific. Any questions raised will be answered. Any changes needed to be made from the first draft will be updated and should be reevaluated and reviewed again.

**Final Copy:** Final Copy will have all updates and revisions. This will be document stakeholders and clients will see.

**Additional Changes:** If any changes need to be made after the final copy has been turned in, the team will need to agree and request to send a request to update said documentation towards their professor.

## **4.3 Documents**

This section contains any extra documents associated with this project with a brief description.

Document Title	Brief Description
----------------	-------------------

SPMP	Document used as a guide for execution and control of said project.
Use Case & Class Iterations	A document containing interactions between actors, and also how that actor interacts with classes.
Collaboration Diagram	A document containing relationships and interactions among software objects in the UML.
Code	Source code created to allow the project to run.
Database Files	Information that allows our database to run without errors.

## 4.4 Schedule

Contains a list of phases, deliverables, and estimated date of completion

Tasks	Projected Schedule	Actual Schedule
Phase	01/31/2020	On time
Planning Phase	02/15/2020	On time
Development Phase	02/29/2020	On time
SPMP	04/05/2020	On time
Code	04/05/2020	On time
Deliverable	04/19/2020	On time
Delivery Phase	04/30/2020	On time
Est. Finish	05/03/2020	On time