Project Management Plan

for

Course Management System

Version 1.0 draft 1

Prepared by Yash Mahendrakumar Wani  
Net ID: yxw230008

University of Texas at Dallas

March 18, 2025

Table of Contents

[1. Overview 1](#_Toc193205334)

[1.1. Project Purpose, Objectives, and Success Criteria 1](#_Toc193205335)

[1.2. Project Deliverables 1](#_Toc193205336)

[1.3. Assumptions, Dependencies, and Constraints 2](#_Toc193205337)

[1.4. References 2](#_Toc193205338)

[1.5. Definitions and Acronyms 2](#_Toc193205339)

[1.6. Evolution of the Plan 3](#_Toc193205340)

[2. Project Organization 3](#_Toc193205341)

[2.1. External Interfaces 3](#_Toc193205342)

[2.2. Internal Structure 3](#_Toc193205343)

[2.3. Roles and Responsibilities 3](#_Toc193205344)

[3. Managerial Process Plans 4](#_Toc193205345)

[3.1. Start-Up Plans 4](#_Toc193205346)

[3.1.1 Estimation Plan 4](#_Toc193205347)

[3.1.2 Staffing Plan 4](#_Toc193205348)

[3.1.3 Staff Training Plan 4](#_Toc193205349)

[3.1.4 Resource Acquisition Plan 5](#_Toc193205350)

[3.1.5 Project Commitments 5](#_Toc193205351)

[3.2. Work Plan 5](#_Toc193205352)

[3.3. Control Plan 6](#_Toc193205353)

[3.3.1 Data Control Plan 6](#_Toc193205354)

[3.3.2 Requirements Control Plan 6](#_Toc193205355)

[3.3.3 Schedule Control Plan 6](#_Toc193205356)

[3.3.4 Budget Control Plan 6](#_Toc193205357)

[3.3.5 Communication, Tracking, and Reporting Plan 6](#_Toc193205358)

[3.3.6 Metrics Collection Plan 6](#_Toc193205359)

[3.4. Risk Management Plan 7](#_Toc193205360)

[3.5. Issue Resolution Plan 7](#_Toc193205361)

[3.6. Project Close-Out Plan 7](#_Toc193205362)

[4. Technical Process Plans 7](#_Toc193205363)

[4.1. Process Model 7](#_Toc193205364)

[4.2. Methods, Tools, and Techniques 8](#_Toc193205365)

[4.3. Configuration Management Plan 8](#_Toc193205366)

[4.4. Quality Assurance Plan 9](#_Toc193205367)

[4.5. Documentation Plan 9](#_Toc193205368)

[4.6. Process Improvement Plan 10](#_Toc193205369)

# Overview

This project develops a web-based system for managing UTD CS course offerings for Spring 2025. It includes JavaScript-based data extraction, a MySQL database for storage, and a responsive PHP interface for viewing and modifying course information. The system features search functionality with auto-complete, and comprehensive CRUD operations to efficiently manage course data through an intuitive user interface.

## Project Purpose, Objectives, and Success Criteria

This project aims to develop a web-based system for managing UTD CS course offerings. The system will allow administrators to view, add, update, and delete course information for Spring 2025 semester offerings through an intuitive user interface.

Objectives:

* Extract courses data from an HTML page using JavaScript.
* Design and implement a relational database to store course information.
* Create a responsive, user-friendly web interface for course management.
* Implement search functionality to easily locate specific courses
* Ensure all data modifications are properly tracked with timestamps

Success Criteria:

* Successful extraction of all required course fields from the source HTML
* Properly functioning database operations (create, read, update, delete)
* Responsive web interface that works across different screen sizes
* Intuitive search functionality with auto-completion features.
* Complete documentation of all components and test cases

## Project Deliverables

* Task 1: JavaScript-based data extraction solution and database schema
  + HTML and JavaScript files for data extraction
  + SQL scripts for database creation and population
  + Database exports in multiple formats (SQL, XML, PHP array, JSON)
* Task 2: PHP-based web application for course management.
  + PHP pages for displaying and managing course information
  + JavaScript functions for test case execution
  + Documentation of test cases with before/after screenshots
* Task 3: Enhanced web interface with styling and advanced features
  + Responsive design with Bootstrap integration
  + CSS stylesheets for visual presentation
  + Search functionality with auto-complete
  + Interacting course detail viewing
* Task 4: Comprehensive documentation
  + Technical documentation with code listings
  + Screenshots of implementation
  + Test case execution results
  + User interface documentation

## Assumptions, Dependencies, and Constraints

Assumptions:

* The source HTML page structure will remain consistent
* Course information follows a standard format
* XAMPP environment is available and properly configured
* Users have basic knowledge of web interfaces and course management

Dependencies:

* XAMPP web server environment
* MySQL database system
* PHP server-side scripting language
* Modern web browser with JavaScript enabled

Constraints:

* All development must be done using the XAMPP framework
* The project must be completed according to the specified folder structure
* Database design must accommodate the required course information fields
* The interface must be intuitive for users with minimal training
* Time constraints for project completion based on course deadlines

## References

* W3Schools JavaScript tutorials: <https://www.w3schools.com/js/>
* W3Schools PHP tutorials: <https://www.w3schools.com/php/>
* W3Schools Bootstrap tutorial: <https://www.w3schools.com/howto/howto_website_bootstrap.asp>
* Course examples from www10 examples
* MySQL documentation: <https://dev.mysql.com/doc/>
* Bootstrap documentation: <https://getbootstrap.com/docs/>
* UTD branding guidelines (for styling)

## Definitions and Acronyms

* **XAMPP:** Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P)
* **HTML:** HyperText Markup Language
* **CSS:** Cascading Style Sheets
* **JS:** JavaScript
* **PHP:** PHP: Hypertext Preprocessor
* **SQL:** Structured Query Language
* **JSON:** JavaScript Object Notation
* **XML:** eXtensible Markup Language
* **UTD:** University of Texas at Dallas
* **CS:** Computer Science
* **S25:** Spring 2025 semester
* **CRUD:** Create, Read, Update, Delete (basic database operations)
* **DOM:** Document Object Model

## Evolution of the Plan

This plan is subject to revision throughout the project lifecycle. Updates may be necessary based on:

* Feedback from testing
* Discovery of more efficient implementation approaches
* Clarification of requirements
* Technical constraints encountered during development

Any significant changes to the project plan will be documented with:

* Date of revision
* Description of changes
* Reason for changes
* Impact on project timeline or deliverables

The current version represents the initial project plan based on the requirements outlined in the project specification document.

# Project Organization

## External Interfaces

The project interfaces with the XAMPP environment for web server hosting and database management. It utilizes the MySQL database system for data storage and retrieval, PHP for server-side processing, and a web browser for the user interface. The system also interfaces with an original HTML document containing course information that serves as the data source for extraction.

## Internal Structure

The project is organized into three main components:

* Data Extraction Layer: JavaScript scripts that extract course information from HTML
* Data Management Layer: MySQL database and PHP scripts for storing and manipulating course data
* Presentation Layer: HTML, CSS, and JavaScript for user interface and interaction

The codebase is structured into directories as specified in the requirements:

* cs6314/task1: Data extraction and database implementation
* cs6314/task2: PHP interface for course management
* cs6314/task3: Enhanced styling and additional features
* cs6314/task4: Project documentation
* cs6314/sql: SQL scripts and database exports

## Roles and Responsibilities

As this is an individual project, Yash Mahendrakumar Wani is responsible for all aspects of the development:

| **Role** | **Responsibilities** | **Assigned To** |
| --- | --- | --- |
| Database Designer | Design database schema, create tables, manage data structure | Yash Mahendrakumar Wani |
| Backend Developer | Implement PHP scripts, database connectivity, data manipulation operations | Yash Mahendrakumar Wani |
| Frontend Developer | Design user interface, implement CSS styling, create JavaScript functionality | Yash Mahendrakumar Wani |
| Data Engineer | Extract course information, create data transformation scripts | Yash Mahendrakumar Wani |
| Documentation Specialist | Create project documentation, capture screenshots, document test cases | Yash Mahendrakumar Wani |
| Tester | Design and implement test cases, verify functionality | Yash Mahendrakumar Wani |

# Managerial Process Plans

## Start-Up Plans

### Estimation Plan

This course project will require approximately 40 hours of work over a 3-week period. The workload will be distributed as follows:

* Task 1 (Data Extraction): 12 hours
* Task 2 (Web Interface): 10 hours
* Task 3 (Enhanced Features): 10 hours
* Task 4 (Documentation): 8 hours

Regular progress checks will be scheduled after completing each task to ensure the project remains on track.

### Staffing Plan

As a student project, I (Yash Mahendrakumar Wani) will handle all development tasks personally. I'll allocate dedicated time blocks each week to ensure consistent progress:

* Weekdays: 2 hours per day
* Weekends: 5 hours per day
* Additional time as needed before deadlines

### Staff Training Plan

To prepare for this project, I'll need to refresh my skills in several areas:

* Review JavaScript DOM manipulation techniques through W3Schools tutorials
* Practice PHP and MySQL database operations using XAMPP
* Study Bootstrap responsive design principles
* Review UTD branding guidelines for styling

I'll allocate 2-3 hours before beginning development to complete these refresher sessions.

### Resource Acquisition Plan

The following resources will be acquired before starting development:

* Install XAMPP on my personal computer
* Download and save the course offering HTML page
* Set up project directory structure as specified
* Prepare a development environment with code editor
* Bookmark relevant documentation resources

All software resources are freely available, and no additional hardware is required beyond my existing computer.

### Project Commitments

I commit to:

* Complete all required tasks according to specifications
* Meet all project deadlines
* Document all aspects of the development process
* Follow best practices for web development
* Create a user-friendly, functional course management system

## Work Plan

The project will be developed in sequential phases following the task structure:

Week 1:

* Set up environment and project structure
* Complete Task 1: Data extraction and database setup
* Begin documentation of Task 1

Week 2:

* Complete Task 2: PHP interface development
* Test basic CRUD functionality
* Document Task 2 implementation and test cases

Week 3:

* Complete Task 3: Enhanced styling and features
* Final testing of all components
* Complete all documentation
* Prepare for submission and demonstration

## Control Plan

### Data Control Plan

* Back up code and documentation daily to cloud storage
* Maintain version control through dated files
* Create periodic database backups after significant changes
* Validate extracted data against source HTML for accuracy

### Requirements Control Plan

* Review project requirements at the start of each task
* Verify that each component meets specifications before moving to the next
* Maintain a checklist of requirements to ensure nothing is missed
* Consult TA if clarification is needed on any requirements

### Schedule Control Plan

* Track progress using a simple Gantt chart or task list
* Set internal deadlines for each task (2-3 days before final deadline)
* Allocate buffer time for troubleshooting unexpected issues
* Adjust the schedule as needed while ensuring final deadline is met

### Budget Control Plan

As this is a student project, no monetary budget is required. Time will be the primary resource to manage:

* Log hours spent on each component
* Identify time-consuming activities for better future estimates
* Balance time allocation between development and documentation

### Communication, Tracking, and Reporting Plan

* Maintain a development log tracking daily progress
* Note any issues encountered and their solutions
* Prepare weekly progress summaries
* Request feedback from peers or TAs when needed

### Metrics Collection Plan

Track the following metrics:

* Time spent on each task and subtask
* Number of database queries implemented
* Code size (lines of code) per component
* Number of test cases executed
* Number of issues identified and resolved

## Risk Management Plan

Potential risks and mitigation strategies:

1. Technical difficulties with XAMPP: Allocate time for troubleshooting; have alternative installation methods ready
2. Data extraction challenges: Prepare multiple approaches for parsing HTML
3. Time management issues: Set internal deadlines with buffer periods
4. Feature complexity: Start with core functionality, then enhance as time permits
5. Knowledge gaps: Identify and address through tutorials before encountering critical path tasks

## Issue Resolution Plan

When issues arise:

1. First attempt to resolve through research and documentation
2. If unresolved within 1 hour, seek help from online resources
3. For persistent issues, consult with classmates or post on course discussion boards
4. For critical blockers, schedule office hours with TA or professor
5. Document all issues and solutions for future reference

## Project Close-Out Plan

* Verifying all requirements are met using a checklist
* Conduct final testing of all features
* Complete and review all documentation
* Organize code and files according to submission requirements
* Create final backups of all project components
* Submit deliverables according to instructions
* Prepare for demonstration with rehearsed examples
* Document lessons learned for future projects

# Technical Process Plans

## Process Model

For this project, I will follow an incremental development process model with iterative elements. This approach allows me to:

* Develop the system in distinct, manageable tasks as outlined in the project requirements
* Test and validate each component before moving to the next
* Refine implementations as needed based on testing feedback

The development flow will follow this sequence:

1. Data extraction and database creation (Task 1)
2. Basic web interface with CRUD operations (Task 2)
3. Enhanced interface with styling and advanced features (Task 3)
4. Comprehensive documentation (Task 4)

Each phase will include planning, implementation, testing, and documentation activities before proceeding to the next phase.

## Methods, Tools, and Techniques

Development Environment:

XAMPP (Apache, MySQL, PHP)

Code editor: Visual Studio Code with extensions for PHP, JavaScript, and CSS

Web browsers: Chrome and Firefox for testing

Programming Languages and Technologies:

HTML/CSS for structure and styling

JavaScript for client-side functionality and data extraction

PHP for server-side processing

SQL for database operations

Bootstrap framework for responsive design

Development Techniques:

DOM manipulation for data extraction

AJAX for asynchronous interactions

Prepared statements for database security

Responsive web design principles

Form validation for data integrity

Testing Methods:

Manual testing with predefined test cases

Cross-browser compatibility testing

Database integrity verification

User interface functionality validation

## Configuration Management Plan

File Organization:

* Follow the specified directory structure (cs6314/task1, cs6314/task2, etc.)
* Use consistent file naming conventions as specified in requirements
* Store SQL scripts in the designated sql subdirectory

Version Control:

* Maintain dated backups of key files
* Use descriptive file naming with version indicators for significant changes
* Store backups in cloud storage (Google Drive or similar)

Change Management:

* Document any deviations from initial requirements
* Track changes to code and database schema
* Maintain a change log for significant modifications

Backup Strategy:

* Daily backups of all code files
* Export database after each significant change
* Store backup copies in multiple locations

## Quality Assurance Plan

Code Quality Standards:

* Include appropriate comments in all code files
* Follow consistent indentation and formatting
* Use meaningful variable and function names
* Apply separation of concerns (data, logic, presentation)

Testing Approach:

* Test each function individually as it's developed
* Verify database operations with test data
* Perform integration testing between components
* Conduct user interface testing for usability

Validation Criteria:

* All required functionality works as specified
* Database operations maintain data integrity
* User interface is responsive and intuitive
* Error handling is implemented for edge cases

Bug Tracking:

* Document any issues encountered during development
* Record the solution applied for each issue
* Verify fixed issues don't reoccur in later testing

## Documentation Plan

Code Documentation:

* Include header comments in each file with:
  + File purpose
  + Author information
  + Creation/modification date
  + Relation to project components
* Document functions with:
  + Purpose description
  + Parameter descriptions
  + Return value information

User Documentation:

* Include screenshots of the interface
* Provide step-by-step instructions for common operations
* Document error messages and their meanings

Technical Documentation:

* Database schema description
* API/function reference
* System architecture overview
* Installation and setup instructions

Test Documentation:

* Document test cases with:
  + Purpose of test
  + Steps to reproduce
  + Expected results
  + Actual results
  + Before/after screenshots

## Process Improvement Plan

Feedback Collection:

* Note difficulties encountered during development
* Identify bottlenecks in the development process
* Record areas where requirements were unclear

Analysis:

* Review development process after project completion
* Identify areas for improvement in future projects
* Assess effectiveness of tools and techniques used

Learning Integration:

* Document lessons learned for future reference
* Research alternative approaches for problematic areas
* Plan specific skill improvements based on challenges faced

Documentation Improvement:

* Update documentation based on user feedback
* Enhance code comments where clarification was needed
* Refine test cases based on discovered edge cases

By following these technical process plans, I aim to ensure a structured, efficient development process that produces a high-quality course management system meeting all requirements*.*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
| Yash Mahendrakumar Wani | March 18, 2025 | First Draft | 1.0 draft 1 |