K. J. Somaiya College of Engineering,
Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: A2	**Roll No.: 16010121045**
Experiment / assignment / tutorial No._____	

TITLE: Project Management Plan Document for Mini Project

AIM: To learn and understand the way of developing the software by classical methods of software engg, Planning and monitoring of the project using tools and prepares a document for the same by using the concept of software engineering.

Expected OUTCOME of Experiment:

Books/ Journals/ Websites referred:

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraw Hill, 2010, 6th edition
2. Ian Somerville, Software Engineering, Addison Wesley, 2011,9th edition.
3 http://en.wikipedia.org/wiki/Software_requirements_specification.

Software Project Management Plan

for

CodeCat – Online Compiler

Pargat Singh (16010121045)

Meet Gala (16010121051)

10th October, 2023

Version	Release Date	Responsible Party	Major Changes
1.0	10/10/2023	KJSCE	Initial Document Release for Comment

Table of Contents

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

1. Introduction

1.1 Project Overview: The project, titled "CodeCat - The Online Compiler," is an ambitious undertaking aimed at creating a user-friendly and feature-rich online coding platform. This platform will provide programmers with the capability to write, compile, and execute code in a variety of programming languages. It seeks to address the limitations faced by students during their academic examinations, as well as the general challenges encountered by coders when using existing online compilers. With the project's focus on multi-language support, user-friendly code editing, syntax highlighting, error debugging, and accessibility across various devices, it aspires to become an invaluable tool for learners and developers alike.

1.2 Project Deliverables: The project's primary deliverable is the "CodeCat" online coding platform, designed to support multiple programming languages, offer a user-friendly code editor with advanced features, and provide a robust execution environment. Additional deliverables include comprehensive documentation, a progressive web application (PWA) for mobile access, and a GitHub repository for version control and collaboration.

1.3 Evolution of the SPMP (Software Project Management Plan): This SPMP will serve as a dynamic guide throughout the development of the "CodeCat" project. As the project progresses, the plan may evolve to accommodate any changes in requirements, timelines, or resources. Regular updates will ensure that the SPMP remains aligned with the project's objectives and facilitates effective project management.

1.4 Reference Materials: Reference materials, including academic research papers, articles, and existing online coding platforms, have been crucial in shaping the "CodeCat" project. These materials have provided insights into the latest advancements in the field and have helped in understanding user feedback and expectations. One notable reference is the research paper titled "A Review of Online Code Learning Platforms" by Dara Pirie and Robert McCartney, which explores various online coding platforms and their effectiveness in teaching and learning programming.

1.5 Definitions and Acronyms:

- SPMP: Software Project Management Plan
- PWA: Progressive Web Application
- Figma: Figma is a collaborative browser-based interface design tool, with additional offline features enabled by desktop applications for macOS and Windows.

These definitions and acronyms will be used throughout the project documentation to ensure clarity and consistency in communication.

2. Project Organization

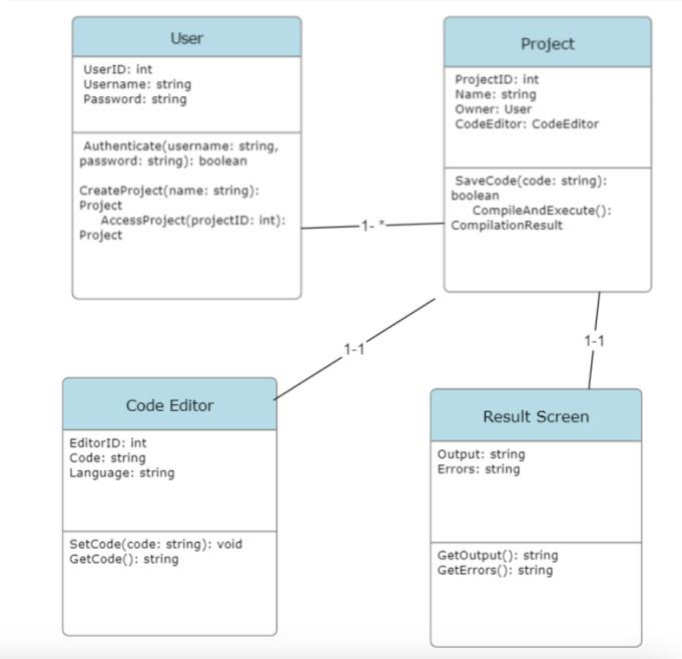
2.1 Process Model:

For the "CodeCat - The Online Compiler" project, we will follow an iterative and incremental software development life cycle model. The process model includes the following phases:

- **Project Initiation:**
  - **Activities:**
    - Define project objectives and scope.
    - Establish initial project schedule and budget.
  - **Entry Criteria:** Approved project proposal and initial funding allocation.
  - **Exit Criteria:** Approved project initiation documentation.
- **Product Development:**
  - **Activities:**
    - Requirement analysis and design.
    - Implementation and testing.
    - Iterative development cycles.
  - **Entry Criteria:** Approved project initiation documentation.
  - **Exit Criteria:** Developed and tested code for each iteration.
- **Product Release:**
  - **Activities:**
    - Final testing and quality assurance.
    - Deployment to the hosting platforms.
  - **Entry Criteria:** Completed product development phase.
  - **Exit Criteria:** Released and available platform for users.
- **Project Termination:**
  - **Activities:**
    - Documentation and project closure activities.
  - **Entry Criteria:** Successful product release.

- **Exit Criteria:** Approved project closure documentation.

Class Diagram :



2.2 Organizational Structure:

The internal management structure of the "CodeCat" project consists of the following key roles:

- **Project Manager:** Leads the project team, oversees project planning and execution, and is responsible for project deliverables.
- **Technical Team Leader(s):** Leads the technical aspects of the project, including code development, architecture, and quality assurance.

The project operates independently within the Department of Computer Engineering at K. J. Somaiya College of Engineering, Mumbai-77, and is part of the educational and research activities within the department.

2.3 Organizational Interfaces:

Organization	Liaison	Contact Information
Customer	Project Manager	officesupplies@gmail.com
Software Quality Assurance	Assurance Architecture	maskify@gmail.com
Software Configuration	Project Manager	maskify@gmail.com

Table F-1. Project Interfaces

The project management function has the following responsibilities:
• Schedule and prepare meetings with clients
• Assign presentations (in-class project meetings, client review, client acceptance test) to project members
• Listening to gripes from the team members
• Resolve conflicts if they cannot be resolved otherwise
Architecture Liaison: The liaison interacts with the liaisons of the other teams and with the project management. Each team has a liaison to the Architecture Team. The responsibilities of the liaison are:
• Responsible for intergroup communication
• Make available public definitions of each subsystem service ("API") to the other teams (ensure consistency, etc.)
• Coordinate tasks that overlap subsystems with the teams
• Responsible for team negotiations, that is, resolve technical issues spanning more than one subsystem

2.4 Project Responsibilities:

The following table outlines the major project functions and activities along with the responsible individuals:

Role	Description	Person
Project Manager	Leads project team; responsible for project deliverables	Pargat Singh
Technical Team Leader(s)	Responsible for the business logic, creating backend and internal functionality	Meet Gala
UI Designer	Responsible for designing and creating webpages and front end	Vishrut Deshmukh

This division of responsibilities ensures that each aspect of the project is effectively managed and overseen by responsible individuals, contributing to the successful development and deployment of the "CodeCat" platform.

3. Managerial Process

3.1 Management Objectives and Priorities: The management objectives and priorities for the "CodeCat - The Online Compiler" project are as follows:

- **Philosophy:** The project management philosophy is to deliver a high-quality online coding platform that addresses the diverse needs of users, prioritizing their experience, and offering features such as multi-language support, syntax highlighting, and accessibility.
- **Goals and Priorities:**
  - **Cost:** Fixed - The project aims to stay within the budget constraints.
  - **Schedule:** Fixed - The project must adhere to the established schedule.
  - **Scope (Functionality):** Flexible - While the core functionality is defined, there is flexibility for enhancements based on user feedback and requirements.

Project Dimension	Fixed	Constrained	Flexible
Cost		X	
Schedule	X		
Scope (functionality)			X

3.2 Assumptions, Dependencies, and Constraints:

- **Assumptions:** The project assumes that the technical team has the required skills and expertise to develop the online coding platform effectively. It also assumes access to necessary hardware and software resources.
- **Dependencies:** The project is dependent on timely access to the Judge0 API for real-time code execution.
- **Constraints:** The project is constrained by budget and time limitations, and the priority is to meet the schedule while maintaining core functionality.

3.3 Risk Management: The project team will actively identify, analyze, and manage risks associated with the project. This includes technical risks, such as API availability and compatibility, as well as potential challenges in meeting user expectations. A risk management plan will be documented separately to address specific risks and contingency plans.

3.4 Monitoring and Controlling Mechanisms: To ensure effective monitoring and control of the project, the following mechanisms will be implemented:

- **Reporting Mechanisms:** Regular status reports will be generated by the project team and submitted to the Project Manager on a weekly basis. Monthly project reviews will be conducted.
- **Review and Audit Mechanisms:** Monthly project reviews will involve a comprehensive review of the project's progress, issues, and risks.
- **Communication Plan:** A communication and reporting plan will be established to ensure that the project team, Project Manager, Software Manager, and other stakeholders are informed at key points in the project.

Information Communicated	From	To	Time Period
Status report	Project Team	Project Manager	Weekly
Status report	Project Manager	Software Manager, Project Team	Weekly
Project Review	Project Team	Software Manager	Monthly

3.5 Staffing Approach:

The project requires a diverse set of skills, including software development, user interface design, and quality assurance. Appropriate personnel will be recruited from the student body, and training will be provided as needed to ensure that team members are equipped to meet project requirements.

- A UI designer is required to create and design the webpages and create a seamless experience.
- A front-end developer who can work with the UI designer to getting the designs
- to life
- A backend developer who can work the internal logic and set up the judge0
- system.

4. Technical Process

4.1 Methods, Tools, and Techniques:

For the "CodeCat - The Online Compiler" project, the following methods, tools, and techniques will be utilized:

VS Code	Code editor
Chrome	Internet browser
Canva	Design Creation
PowerPoint	Slide presentation
Github	Version Control
Figma	Prototype Creation

- **Development Method:** Agile software development with iterative and incremental cycles to enable flexibility and rapid feature enhancements.
- **Programming Languages:** JavaScript (React.js), Python, and other relevant languages for server-side development.
- **Development Tools:** Visual Studio Code, Figma for UI/UX design, and Docker for containerization.
- **Standards and Procedures:** Adherence to industry-standard coding conventions and project management practices.
- **Team Structure:** A cross-functional team with software developers, designers, and quality assurance specialists.
- **Documentation:** Use of a standardized documentation format and style guide to ensure consistency.

4.2 Software Documentation:

The following work products will be developed for the project, along with the types of peer reviews for each product:

- **Software Requirements Specification (SRS):**
  - **Description:** This document will outline the essential requirements of the software, including functions, performance expectations, design constraints, and attributes. It will also define external interfaces.
  - **Review Type:** Peer review.
- **Software Design Description (SDD):**
  - **Description:** The SDD will describe the major components of the software design, including databases and internal interfaces.
  - **Review Type:** Peer review.
- **Software Test Plan:**
  - **Description:** The Software Test Plan will detail the methods and procedures for testing at various development and integration levels, including requirements, design, and code. It will also specify test procedures, test cases, and test results.
  - **Review Type:** Peer review.

4.3 User Documentation:

User documentation will be developed to assist users in effectively using the CodeCat platform. This documentation will include online help, guides, and tutorials. The user documentation plan will be created to outline the process of developing and maintaining this documentation.

4.4 Project Support Functions:

For the "CodeCat" project, the following supporting functions will be planned and executed:

- **Configuration Management:** A configuration management plan will be established to manage project artifacts, source code, and changes effectively.
- **Software Quality Assurance:** A quality assurance plan will be implemented to ensure that the software meets defined quality standards, including testing and verification.
- **Verification and Validation:** A plan for verification and validation activities will be developed to ensure that the software meets requirements and is free of defects.

Each of these supporting functions will have defined responsibilities, resource requirements, schedules, and budgets to ensure that they are carried out effectively in line with the project's objectives. The absence of any of these functions will be explicitly justified if applicable.

5. Work Packages, Schedule, and Budget

5.1 Work Packages:

The project activities and tasks for "CodeCat - The Online Compiler" are structured into the following work packages, each uniquely identified. The work packages are organized hierarchically based on their relationships:

1. **Project Initiation (WP-1)**
    - Establish project objectives and team roles.
    - Define project scope and requirements.
    - Develop a project plan.
2. **Requirements Gathering (WP-2)**

- Gather user requirements and expectations.
- Define software functionality and features.
- Document software specifications.

3. **Design and Prototyping (WP-3)**
    - Create the user interface design.
    - Develop software architecture and database design.
    - Build prototypes for user testing.
4. **Development (WP-4)**
    - Write code for the frontend and backend.
    - Implement core features.
    - Integrate third-party services like Judge0 API.
5. **Testing and Quality Assurance (WP-5)**
    - Conduct unit testing, integration testing, and system testing.
    - Implement user feedback for improvements.
    - Ensure quality and reliability.
6. **Documentation and User Guides (WP-6)**
    - Create user documentation and help guides.
    - Develop online help resources.
    - Prepare instructional materials.
7. **Deployment and Release (WP-7)**
    - Deploy the platform on the web.
    - Configure production servers.
    - Release the platform to users.
8. **Monitoring and Support (WP-8)**
    - Monitor platform performance and user feedback.
    - Provide ongoing support and maintenance.
    - Implement updates and enhancements based on user feedback.

5.2 Dependencies:

The work packages in the "CodeCat" project have the following dependencies:

- **WP-1 (Project Initiation)** must be completed before starting WP-2 (Requirements Gathering).

- **WP-2 (Requirements Gathering)** must precede WP-3 (Design and Prototyping).
- **WP-3 (Design and Prototyping)** should be completed before starting WP-4 (Development).
- **WP-4 (Development)** must be finished before WP-5 (Testing and Quality Assurance) begins.
- **WP-5 (Testing and Quality Assurance)** should be completed before initiating WP-6 (Documentation and User Guides).
- **WP-6 (Documentation and User Guides)** should be completed before starting WP-7 (Deployment and Release).
- **WP-7 (Deployment and Release)** must precede WP-8 (Monitoring and Support).

5.3 Resource Requirements:

The estimated resource requirements for the "CodeCat" project include:

- Personnel: A cross-functional team of software developers, designers, quality assurance specialists, and project managers.
- Computer hardware and software: Development and testing environments, servers, and software tools.
- Office and laboratory facilities: Workspace for the project team.
- Maintenance requirements: Ongoing support and maintenance resources.

5.4 Budget and Resource Allocation:

Budget and resource allocation will be determined based on the specific resource requirements for each work package. A detailed budget and resource allocation plan will be established to ensure that each work package receives the necessary resources.

5.5 Schedule:

The project schedule for "CodeCat - The Online Compiler" will be created, taking into account the work package dependencies and milestones. The schedule will be expressed in absolute calendar time, and it will be included as an appendix or reference

document to this plan. The schedule will specify key milestones and deadlines for project completion.

6. Additional Components

6.1 Index:

An index to key terms and acronyms used throughout the Software Project Management Plan (SPMP) is provided for reference and improved usability of the document.

Index

- CodeCat - The Online Compiler
- SRS - Software Requirements Specification
- SDD - Software Design Description
- QA - Quality Assurance
- PWA - Progressive Web Application
- API - Application Programming Interface
- UI/UX - User Interface/User Experience

6.2 Appendices:

Appendices may be included as supporting details and can be referenced as needed to provide additional information. Suggested appendices for the "CodeCat - The Online Compiler" project include:

Appendix A: Current Top 10 Risk Chart

- This appendix will outline the current top 10 risks identified for the project, their likelihood, impact, and mitigation strategies.

Appendix B: Current Project Work Breakdown Structure

- The Work Breakdown Structure (WBS) will provide a detailed breakdown of project activities, tasks, and their dependencies.

Appendix C: Current Detailed Project Schedule

- This appendix will present a detailed project schedule, including milestones, deadlines, and resource allocation for each work package.

These appendices will provide supplementary information and details that support the main body of the SPMP without overwhelming it with excessive content.

Conclusion: Through this experiment we understand the how to organize a project and ensure it meets deadlines. We create a project management plan for our mini project – CodeCat – Online Compiler.

Post Lab Descriptive Questions

1. **State various Scheduling principles and explain them in detail.**

Scheduling principles are essential in project management as they help in the effective allocation of resources, time management, and the successful completion of tasks. Here are various scheduling principles and their explanations in detail:

1. **Define Clear Objectives:**
    - **Explanation:** The first step in scheduling is to define clear and specific project objectives. These objectives help in determining the scope of work, expected outcomes, and deliverables. Without well-defined objectives, scheduling becomes challenging as there is no clear direction for the project.
2. **Task Breakdown:**
    - **Explanation:** Break down the project into smaller, manageable tasks or work packages. This division helps in organizing the project and understanding the interdependencies between tasks. A Work Breakdown Structure (WBS) is often used to categorize and structure the tasks.
3. **Sequencing:**
    - **Explanation:** Once tasks are identified, they need to be sequenced in the order they should be executed. Sequencing ensures that tasks are performed in a logical and efficient sequence, taking into account dependencies between them.
4. **Task Duration Estimation:**

- **Explanation:** Accurately estimate the time required to complete each task. This estimation helps in creating a realistic schedule. Various techniques, such as expert judgment, historical data, and analogous estimation, can be used for task duration estimation.

5. **Resource Allocation:**

   - **Explanation:** Allocate the necessary resources, including personnel, equipment, and materials, to each task. Resource allocation ensures that tasks have the required resources available when needed, preventing bottlenecks and resource constraints.

6. **Critical Path Analysis:**

   - **Explanation:** Identify the critical path, which is the sequence of tasks that determines the project's overall duration. Tasks on the critical path have no slack, meaning any delay in these tasks will delay the project's completion. This analysis is crucial for time-sensitive projects.

7. **Buffer Management:**

   - **Explanation:** Incorporate buffers or contingency time in the schedule to account for uncertainties and unexpected delays. These buffers, often called slack or float, provide flexibility and help mitigate risks that could impact the project timeline.

8. **Task Dependencies:**

   - **Explanation:** Understand and define the dependencies between tasks, which can be categorized as finish-to-start, start-to-start, finish-to-finish, or start-to-finish. Recognizing dependencies is crucial for determining the order in which tasks must be executed.

9. **Resource Leveling:**

   - **Explanation:** Ensure that resources are efficiently utilized by resolving resource conflicts and over-allocations. Resource leveling helps in optimizing resource usage and preventing resource burnout.

10. **Monitoring and Control:**

    - **Explanation:** Continuously monitor the project's progress against the schedule. Any deviations from the plan should be identified and addressed

promptly. Adjustments may need to be made to accommodate changes or unforeseen issues.

11. **Regular Updates:**

- **Explanation:** Schedules should be living documents that are regularly updated to reflect the project's current status. As the project progresses, tasks may be completed earlier or later than planned, and updates ensure that the schedule remains accurate.

12. **Communication and Stakeholder Involvement:**

- **Explanation:** Effective communication with project stakeholders is essential. Keep stakeholders informed about the project's schedule, progress, and any changes. Involving stakeholders in scheduling decisions can lead to better collaboration and buy-in.