

**Project Management Plan Template**

(Group No:09)

**Project Management Plan**

**<** **Project Management System >**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Sazin, A.S.M Rafiuzzaman | 19-41546-3 |
| Rubaiba Hossain | 20-42148-1 |

**Date**

**27-08-2023**

**Table of Contents**

# 1.0 Introduction ……………………………………………………………………………… 3

# 2.0 Project Management Approach ………………………………………………………… 4

# 3.0 Project Title ………………………………………………………………………………. 4

# 4.0 Justification ……………………………………………………………………………….. 4

**5.0 Objectives and Project Scopes …………………………………………………………… 5**

**6.0 Overview of the Project …………………………………………………………………... 6**

**7.0 Stakeholder Analysis ……………………………………………………………………… 9**

**8.0 Milestone List …………………………………………………………………………….. 10**

**9.0 Process to be Followed …………………………………………………………………… 11**

**10.0 Work Breakdown Structure ……………………………………………………………. 12**

**11.0 Estimation ………………………………………………………………………………... 13**

**12.0 Resource Requirements …………………………………………………………………. 16**

**13.0 Project Schedule …………………………………………………………………………. 17**

**14.0 Project Delivery Plan ……………………………………………………………………. 18**

**15.0 Risk Analysis …………………………………………………………………………….. 20**

**16.0 Quality Control Plan ……………………………………………………………………. 20**

**17.0 Budget ……………………………………………………………………………………. 22**

**18.0 Conclusion ……………………………………………………………………………….. 22**

**1.0 Introduction**

A project management system is a software application or platform that assists in the proper planning, execution, and monitoring of projects. It offers a centralized and organized approach to project management, enabling teams to interact, measure progress, and meet project objectives.

# Project Deliverables:

The PMS project will deliver the following key components:

* **User-Friendly Interface:** A web-based interface that is easy to use and that project managers, team members, and other users can access.
* **Project Planning Module:** This is a tool that lets make and see plans for projects, set goals and deadlines, and set stages.
* **Resource Management Module:** The power to assign resources in a way that makes the best use of them.
* **Task Tracking and Collaboration:** Features for giving out projects, keeping track of progress, and making it easy for team members to work together.
* **Real-time monitoring and reporting:** Dashboards and reports give real-time information about the state of a project, which helps people make better decisions.

# Project Benefits:

The implementation of the PMS brings forth a multitude of benefits:

* **Increased efficiency:** When project management processes are streamlined, projects are finished faster and administrative costs go down.
* **Improved Collaboration:** Team members can work together better on a common platform, which also makes it easier for them to talk to each other.
* **Better Use of Resources:** The PMS makes it easier to use resources effectively, stopping them from being overused or underused.
* **Timely choices:** Real-time monitoring and reporting allow stakeholders to make choices quickly that are based on accurate information.
* **Enhanced Accountability:** Accountability is improved when tasks are clearly assigned and the progress of each job is tracked.
* **Risk Mitigation:** Features of integrated risk management help find and deal with possible project risks.

# Project Framework & Purpose:

The PMS will be built on the modern concepts and practices of project management. Its goal is to provide a single, easy-to-use platform that streamlines project management, encourages teamwork, and gives stakeholders access to real-time information. The PMS fits with our company's goals of efficiency, quality, and constant growth.

In the sections that follow more detail about different parts of the project, such as the project management method, scope, stakeholder analysis, milestones, resource needs, schedule, risk analysis, and more. The PMS is going to change the way we handle projects and make it easier for our organization to get projects done.

# 2.0 PROJECT MANAGEMENT APPROACH

The "Project Management System" software application is a strategic initiative aims to design, develop, and deploy a comprehensive solution that enhances project planning, execution, monitoring, and reporting. Through distinct phases, including initiation, planning, execution, monitoring, and closing, the approach prioritizes user-friendly interfaces, real-time collaboration, robust tracking, and customizable reporting. Effective communication, change management, quality assurance, and risk mitigation are central. The project will be guided by a dedicated team, utilize appropriate tools, and emphasize continuous improvement. The goal is to deliver a reliable and secure system that aligns with stakeholder needs while adhering to timelines and elevating project management practices.

**3.0 PROJECT TITLE: “**project management system”

# 4.0 JUSTIFICATION:

A project management system is required for successful project execution. It allows for more effective planning, scheduling, and resource allocation, ensuring that projects are finished on schedule and under budget. The technology encourages cooperation, makes communication easier, and serves as a centralized platform for tracking progress and managing assignments. It helps reduce obstacles and guarantees project success with features like as risk and issue management. A project management system boosts productivity, simplifies workflows, and improves decision-making, making it essential for meeting project objectives and delivering effective results.

# 5.0 OBJECTIVES AND PROJECT SCOPES:

# Objectives:

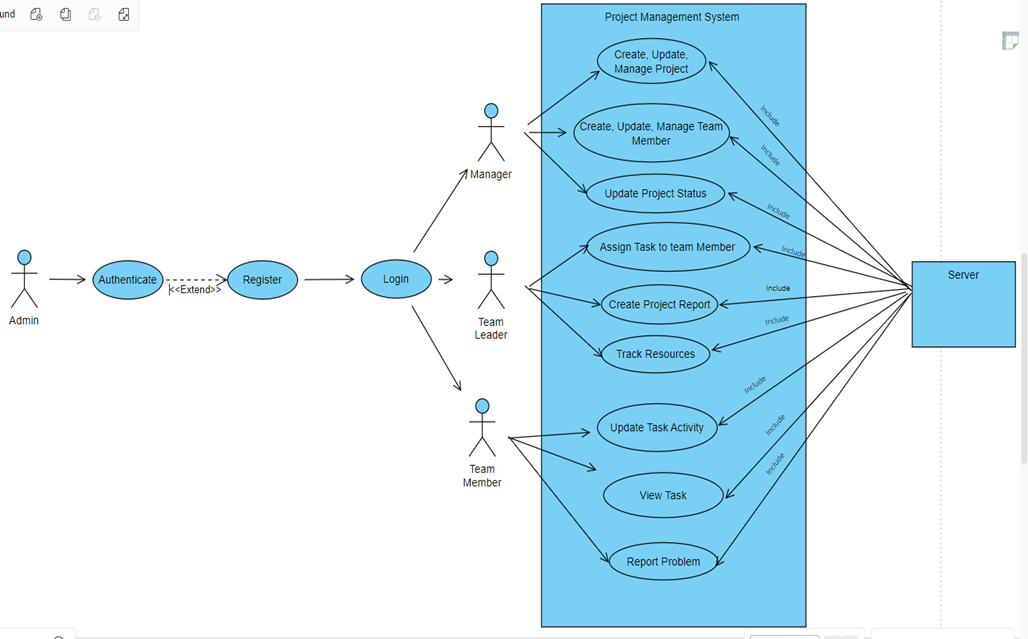
* A project management system software aims to streamline project planning processes, enabling users to create detailed project plans and schedules.
* The software helps allocate resources appropriately, ensuring that the right people are assigned to the right tasks and optimizing resource utilization.
* The objective is to ensure tasks are executed promptly and deadlines are met by providing task management features and progress tracking capabilities.
* The software promotes collaboration among team members by facilitating communication, file sharing, and fostering a cohesive working environment.
* The objective is to provide project managers with better control over project progress, risks, and issues, enabling them to make informed decisions and take necessary actions to keep projects on track.

# Scopes:

* Planning and Scheduling: A project management system software will allow users to create project plans, define tasks, set deadlines, and allocate resources electronically.
* Task Management: It will help in organizing and tracking tasks, assigning responsibilities, and monitoring progress to ensure timely completion of project deliverables.
* Collaboration and Communication: The software will provide team collaboration by providing features for sharing information, discussing project-related matters, and ensuring effective communication among team members.
* Reporting and Analytics: A project management system software will generate reports and provides insights on project progress, resource utilization, and key performance indicators, aiding in decision-making and project evaluation.

**6.0 OVERVIEW OF THE PROJECT:**

**Use Case Diagram:**

****

**Component Diagram:**

**System Component Diagram:**

**7.0 STAKEHOLDERS ANALYSIS:**

There are two types of stakeholders.

1. **Primary Stakeholders:** Primary stakeholders are people or groups who have a clear and important interest in how the project turns out. Their participation can have a big effect on how well the project does. They are directly affected, either happily or negatively, by the project and its results.
2. **Secondary Stakeholders:** People or groups with a detached or less important interest in the project are secondary stakeholders. Most of the time, they have less of an effect on the project's success than main stakeholders.

In our project, Primary and secondary stakeholders are:

**Primary Stakeholder:**

* Administration
* Team Member
* Team Leader

**Secondary Stakeholder:**

* Client
* Vendor

# 8.0 MILESTONE LIST:

|  |  |  |
| --- | --- | --- |
| Milestone | Description | Date |
| Complete SRS | The first milestone involves gathering comprehensive system requirements and finalizing the System Requirements Specification (SRS). This document lays the foundation for the entire project, outlining what the system should accomplish. | 01.09.2023-14.09.2023  (2 weeks) |
| Design | This milestone signifies the completion of the design phase, where the system's architecture, components, and user interfaces are mapped out. Clear design documentation ensures a unified vision for development. | 15.09.2023- 06.10.2023  (3 weeks) |
| Complete Coding | At this point, the coding process is over and all of the parts of the Project Management System have been built. Coding standards and best practices are used to make sure the system is strong and easy to manage. | 07.10.2023-03.11.2023  (4 weeks) |
| Complete Testing and Debugging | This milestone means that all of the hard work testing and fixing bugs is done. The Project Management System goes through a lot of testing to find and fix any bugs or problems. This makes sure that the final output is high-quality and reliable. | 11.11.2023-18.11.2023  (1 Weeks) |
| Documents – User Guides and Installation | The last step is to make user guides and directions for installing the software. These papers give users clear instructions on how to understand and use the system well. An easy-to-use experience is a top priority. | 19.11.2023-29.11.2023  (11 Days) |

**9.0 Process Model to be followed:**

The chosen process model for the project is the Agile model. This selection is justified based on several factors that align with the project's goals, requirements, and characteristics.

Agile Process Model Justification:

* **Iterative and Incremental Development:** The Agile methodology encourages iterative and incremental development, which lets the project release functional parts at regular intervals. This fits well with the goal of the project, which is to make a complete system for managing projects and keep making its features better.
* **Flexibility and Adaptability:** Agile highlights being able to adjust to changes in needs and goals. Because project management tools are always changing and people's needs are always changing, Agile lets the team make changes and incorporate them well.
* **Stakeholder Collaboration:** Agile methods motivate development teams, stakeholders, and final users to work together as closely as possible. For a project management system to work well, it's important to get feedback from possible users all the time.
* **User-Centric Approach:** Agile is all about giving people what they want quickly. By having users take part in each version, the project management system can be fine-tuned based on how users actually use it, which makes users happier.
* **Risk Mitigation:** The agile delivery method used by agile lets problems and risks be found early on. This helps deal with problems ahead of time, which lessens the damage they could do to the whole job.
* **Continuous Improvement:** Agile practices encourage an attitude of always getting better. As the project goes on, the team can use feedback to improve features, functions, and the user experience, making sure that the final result is of high quality.
* **Visibility and Transparency:** Agile methods focus on being open and honest by showing working features on a regular basis. This makes sure that the stakeholders and the project team are on the same page about how the project is going and what they can expect.
* **Short Development Cycles:** Agile divides work into short cycles (called "sprints") that usually last a few weeks. This pattern fits well with the need for progress reports to be given on time in a course project setting.

**10.0 WORK BREAKDOWN STRUCTURE:**

Fig: Work Breakdown Structure

**11.0: ESTIMATION:**

For coding (4 weeks necessary), for design part (3 weeks necessary), and for testing and installation (1 weeks necessary).

Table as a historical data (Previous Projects):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Projects | External Input type | External Output type | Internal file type | SLOC |
| A | 8 | 11 | 12 | 16500 |
| B | 6 | 8 | 9 | 10400 |

**New Project Data**

**4 File or records**

1. Project\_Record

Data Types: (7 types of data) Pname, PId, ProjectType, ProjectStat, Projectbuyer, StartingDate, EndingDate,

2. Project\_Manager\_Record

Data Types: (7 types of data) PMname, PMId, PhoneNo, Gender, PmType, JoiningDate, Email.

3. Team\_Record

Data Types: (7 types of data) TMname, TMId, PhoneNo, Gender, TMdesignation, JoiningDate, Email.

4. Resource\_Record

Data Types: (3 types of data) CurrentResource, UsedResources, AvailableResources.

Here,

**External Input Types: (3 type)(total 21 data types)**

Project\_Record, Project\_Manager\_Record, Resource\_Record

**External Output Types: (2 type) (total 20 data types)**

Project Report, Provided Resource Report.

from Albrecht's Function Point Analysis,

File type Complexity:



External Input type Complexity:

|  |  |
| --- | --- |
| Number of file Types | Number of data types |
| 3 | 21 (High) |

External Output type Complexity:

|  |  |
| --- | --- |
| Number of Record Types | Number of data types |
| 2 | 20(High) |

From Albrecht Complexity Multipliers,

|  |  |
| --- | --- |
| External User Type | Multiplier |
| External Input type | High (6) |
| External Output type | High (7) |
| Internal file type | Average (10) |

**Euclidean distance** from the source and the target project,

From Project A,

Square Root of ((10-12) ^2 +(6-8) ^2 + (7-11) ^2)

= 4.8

From Project B,

Square Root of ((10-9) ^2 +(6-8) ^2 + (7-8) ^2)

= 2.4

Project B has a Closer analogy then project A.

As we are following Top Down approach and Project A has 6010 line of code, so by taking SLOC=6010, from COCOMO MODEL, (ORGANIC TYPE SOFTWARE)

**Effort** = PM = Coefficient<EffortFactor>\*(SLOC/1000) ^P

EFFORT = 2.4\*(10400/1000) ^1.05

=28.06

**Development Time** = DM = 2.50\*(PM)^T

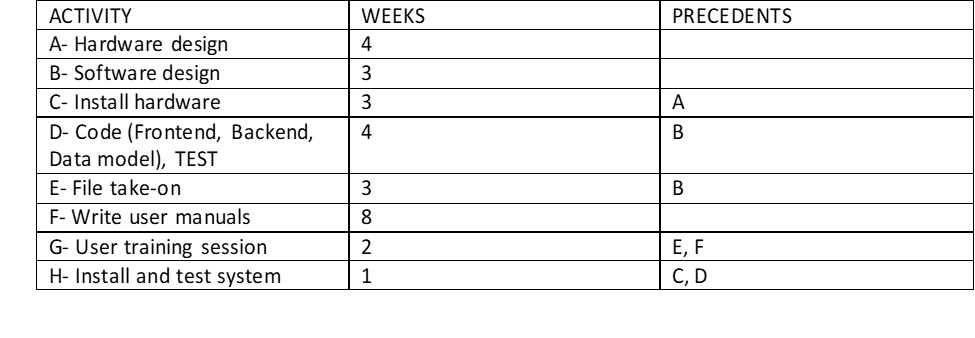
=2.50\*(15.77) ^0.38

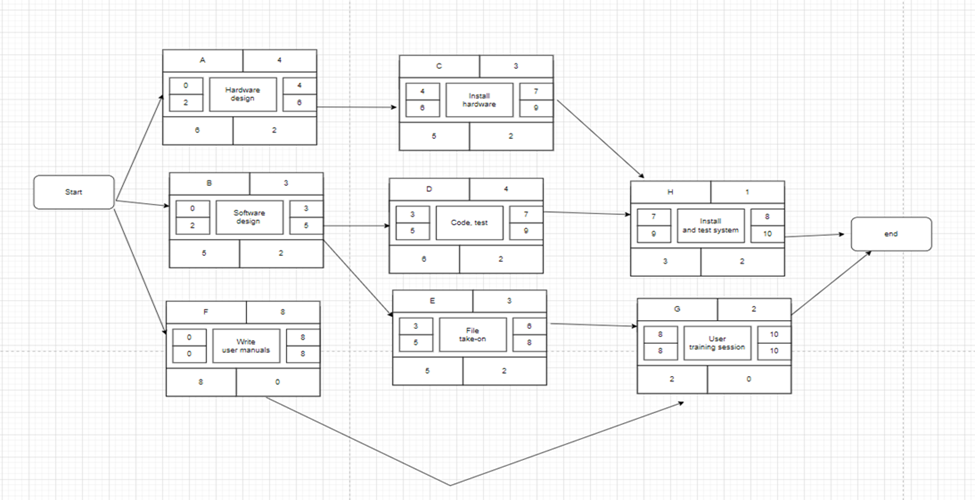
=7.1MONTH (around 7month)

Required Number of people = ST = Effort (PM)/Development

Time (DM) = 28.06/7 =3.9==4 people

**Activity Diagram:**

****

****

Here f + g=10. And the float is (ls-es) =0 in those case. So, this is the critical path (F, G) and the project time is 10.

**12.0 RESOURCE REQUIREMENTS**

**12.1 SOFTWARE REQUIREMENTS:**

**• Integrated Development Environments (IDEs):** Visual Studio

**• Programming language:** C#

**• Database:** MySQL

**• UI/UX design tool:** Figma

**• Version Control Systems (VCS):** GitHub

**• Bug Tracking Tool: Bugzilla**

**• Continuous Integration (CI) Tool:** GitLab

**• Testing Tool:** Selenium

**• Documentation Tool:** MS Words

**• Continuous Monitoring and Logging Tool:** Splunk

**• Project Management Tool:** Clickup

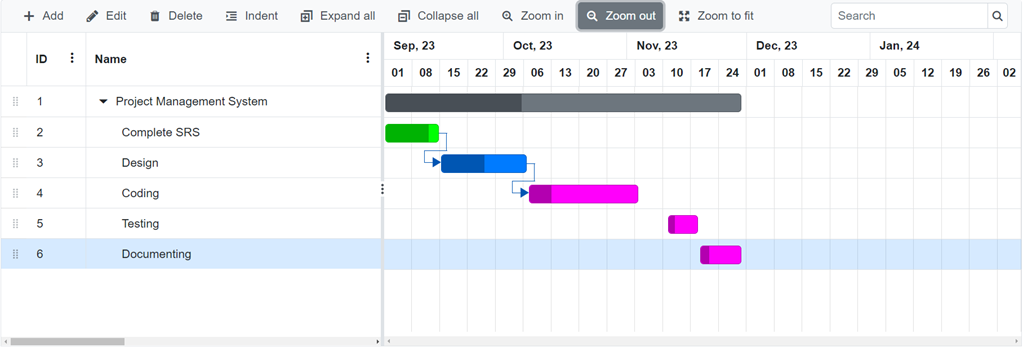
**12.2 HARDWARE REQUIREMENTS:**

* + 1. **Ram:** 8gb
    2. **Hard Disk:** 120gb SSD
    3. **Graphics Card:** 4gb
    4. **Desktop Computer/ Laptop:** Modern Processor like Intel Core i5 or Ryzen 5.

**12.3 HUMAN RESOURCE REQUIREMENTS:**

* **Project Manager:** A skilled project manager will be responsible for overall project coordination, stakeholder communication, and ensuring project objectives are met.
* **Software Developers:** Backend and frontend developers proficient in the chosen programming languages and frameworks will work on system development.
* **Database Specialist:** An expert in database management will design and manage the database system.
* **Quality Assurance Analysts:** QA analysts will ensure the system functions correctly by conducting testing and identifying and rectifying issues.
* **UI/UX Designers:** Designers will be responsible for creating an intuitive user interface and optimizing user experience.
* **Documentation Specialist:** A technical writer will create comprehensive documentation including user manuals, technical specifications, and system architecture documentation.
* **Support and Maintenance Personnel:** After deployment, support staff will be available to address user queries, troubleshoot issues, and perform system updates as needed.

**13.0: PROJECT SCHEDULE:**

****

**14.0 DELIVERY PLAN:**

1. Project Initiation:

- Define project vision, objectives, and scope.

- Identify initial user roles and functionalities.

- Set up communication tools (Slack, email) and collaboration platform (Jira, Trello).

- Assemble Agile team and define roles.

2. Release Planning:

- Create a backlog of user stories and features.

- Prioritize backlog based on business value and complexity.

- Break down user stories into tasks.

- Determine the sprint duration (e.g., 2 weeks).

3. Sprint 1: Basic User Management (2 weeks)

- User stories: User registration, login, password reset.

- Develop user management features.

- Test and validate user management functionalities.

- Conduct sprint review and retrospective.

4. Sprint 2: Project Creation and Management (2 weeks)

- User stories: Create project, assign members, set deadlines.

- Develop project management features.

- Test project creation, member assignment, and deadline setting.

- Conduct sprint review and retrospective.

5. Sprint 3: Task Tracking (2 weeks)

- User stories: Create tasks, assign tasks, track progress.

- Develop task tracking features.

- Test task creation, assignment, and progress tracking.

- Conduct sprint review and retrospective.

6. Sprint 4: Document Management (2 weeks)

- User stories: Upload documents, link to tasks/projects.

- Develop document management features.

- Test document upload and linking functionalities.

- Conduct sprint review and retrospective.

7. Sprint 5: Reporting and Analytics (2 weeks)

- User stories: Generate project/task reports, view analytics.

- Develop reporting and analytics features.

- Test report generation and analytics views.

- Conduct sprint review and retrospective.

8. Sprint 7: Usability Enhancements (2 weeks)

- User stories: Improve UI/UX based on user feedback.

- Implement user interface enhancements.

- Test UI improvements and gather user feedback.

- Conduct sprint review and retrospective.

9. Sprint 8: Final Testing and Refinement (2 weeks)

- Comprehensive system testing and bug fixes.

- Address any outstanding issues.

- Prepare for production deployment.

- Conduct sprint review and retrospective.

10. Deployment and User Training:

- Deploy the application to the production environment.

- Provide user training and documentation.

- Monitor initial usage and address user inquiries.

11. Ongoing Iterations:

- Continue with regular sprints to add new features, refine existing ones, and address user feedback.

- Adapt backlog based on changing priorities and emerging requirements.

12. Project Closure:

- Conduct a project review to assess the achievement of goals.

- Document lessons learned and areas for improvement.

- Celebrate the successful delivery of the PMS.

**15.0 RISK ANALYSIS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Risks** | **Probability** | **Impact** | **RMMM** |
| **1** | System failure | 20% | 9 | Make sure the units or components pass the required test cases before integrating the system |
| **2** | Late delivery | 80% | 7 | Make sure the project progress is on track, other take immediate action. |
| **3** | Technology will not meet expectations | 5% | 2 | Check whether the technologies are acquirable. |
| **4** | Changes in requirements | 60% | 5 | Check if the changed Requirements are feasible, try to make the requirement change before starting the development phase. Deliver the product in short increments time-boxes so that the user gets less time to finalize the requirements and change their mind. |
| **5** | Poor Productivity | 30% | 3 | Set achievable time frames and a sustainable pace during project estimations to avoid burn-out of staff. |
| **6** | Inadequate Management | 40% | 7 | Including risk in estimations. Find out the root cause and attempt for risk reduction procedures. |
| **7** | Poor comments in code | 20% | 5 | Train the programmers. |
| **8** | Unrealistic Schedules and Budgets | 40% | 7 | Using historical data and using multiple models for estimation |

**16.0 QUALITY CONTROL PLAN:**

Quality control is an important part of making sure that the "Project Management System" is built and used well. This section describes the full quality control plan that will be used to keep standards high and give a reliable and effective project management solution.

**Quality Objectives:**

* The main goals of the plan for quality control are toMake sure the "Project Management System" meets the functional standards that have been set.
* Find problems and fix them early on in the creation process.
* Make sure that the final product meets the best practices and standards of the business.

**Quality Control Activities:**

* **What's needed Validation:** A careful look at and check of the system requirements to make sure they are correct, clear, and full.
* **Design Reviews:** Design reviews are done on a regular basis to make sure that the building and design meet the project goals and industry standards.
* **Code Reviews:** Code should be reviewed often to find and fix any coding mistakes, keep things consistent, and make sure that coding standards are followed.
* **Testing:** Full testing, which includes unit testing, integration testing, and system testing, to make sure the system works and is reliable.
* **User Acceptance Testing (UAT):** It is when the system's actual users check to see if it meets their wants and expectations.
* **Documentation Review:** Carefully check the accuracy and clarity of all project documentation, such as user guides and installation instructions.

**Quality Metrics:**

* **Defect Density:** Tracking the number of defects identified per unit of code to assess code quality.
* **Code Coverage:** Measuring the proportion of code that has been tested to ensure comprehensive testing coverage.
* **Requirements Traceability:** Ensuring that each requirement is addressed in the design, development, and testing phases.
* **User Satisfaction:** Gathering feedback from users during UAT to gauge their satisfaction with the system.

**Process Improvement:**

Keeping quality is all about making improvements all the time. There will be regular retrospectives and post-project reviews to find ways to improve the process and use lessons learned in future projects.

**Responsibilities:**

* The project manager is in charge of making sure the quality control plan is carried out, coordinating review activities, and keeping quality standards.
* Development Team: Is in charge of putting quality control activities, like code reviews and tests, into place in their own areas.

**Escalation Process:**

In the case of quality problems or differences that can't be fixed, the project manager will bring the issue to the authority

for help.

**17.0 BUDGET:**

|  |  |
| --- | --- |
| Project Management and Implementation | Taka 1,50,000 |
| Software Quality Testing Cost | Taka 1,00,000 |
| Maintenance Cost | Taka 50,000 |
| Hardware Cost | Taka 4,50,000 |
| Utility | Taka 1,00,000 |
| Total Cost | Taka 8,50,000 |

**18.0 CONCLUSION:**

In conclusion, the Project Management System (PMS) for the "Software Development & Project Management" course is expected to improve how a company manages projects. It will improve project performance by streamlining processes, making it easier for people to work together, and giving real-time insights. The plan has many parts, such as an introduction, a management method, a scope, a list of stakeholders, a list of resources, and an analysis of risks.

The PMS will be a success not only when it is finished, but also by how it changes the way project management is done. The project team promises to stick to the management method that has been set up, deal with risks in a proactive way, and make sure there is good communication with stakeholders. The goal of the PMS is to create a system that is easy to use and fits with the goals of the company.