

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Laboratory Management System (LMS) | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Muhammad Talha (Lead)Abdul RehmanZohaib ZafarDate:05-March-2025Course title:Software ConstructionTeacher’s name:Dr. Onaiza Maqbool |  | | | |
|  | | |  | | |  | | |

**Member 1 :**

Name : Muhammad Talha

Signature

**Member 2 :**

Name : Abdul Rehman

Signature \_\_\_\_\_\_\_

**Member 3 :**

Name : Zohaib Zafar

Signature \_\_\_\_\_\_\_\_

**Supervisor :**

Name : Dr. Onaiza Maqbool

Signature

Change History :

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updation Date | Updated By | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Project Overviews

1.1 Project Summary

1.2 Purpose , Scope and Objectives

1.3 Assumptions and Constraints

1.4 Project Deliverables

1.5 Schedule Summary

1. References
2. Definition
3. Project Context

4.1 Process model

4.2 Methods , Tools and Techniques

4.3 Product acceptance plan

1. Project Planning

Project work plan

Work Activities

Schedule allocation

Resources allocation

1. Supporting Process PlansRisk Management

Division of work according to the instruction

1. Project Overview :
   1. Project Summary   
      The main aim of this project is to develop a system for handling the laboratories of a department. This system will store the hardware specifications and all the software installed on a specific computer in a lab. System will allow the students and teachers to submit complaints about the software or hardware issues. They would be allowed to request for the installation of a specific software on a computer. The system will be accessible via a website, ensuring ease of use and centralized information management.
   2. Purpose , Scope and Objectives

Purpose :

The main goal of this project is to make it easier to manage the resources in the labs, handle any complaints or issues that arise, and process requests for new software efficiently. This will help improve the overall experience for students and teachers using the labs.

Scope :

The system will cover all the laboratories within the department. It will keep detailed information about each computer, including its hardware and software, and will handle any complaints or installation requests from users.

Objectives :

* + - Keep accurate records of computer hardware and software.
    - Make it easy for users to report problems.
    - Handle and track software installation requests.
  1. Assumptions and Constraints

Assumptions:

* + - All the computers in the labs are connected to the department’s network.
    - Students and teachers will have access to the internet and will be able to use the website to report issues or make requests.
    - The laboratory coordinator will have the necessary access to update and manage the system as needed.

Constraints:

* + - The system’s effectiveness depends on the current IT infrastructure available in the department.
    - The accuracy of the records depends on timely updates and inputs from users, such as reporting issues or making requests.
  1. Project Deliverables:
     + A fully functional website that allows users to manage and access laboratory information.
     + A comprehensive database that contains detailed records of all computer hardware and software in the labs.
     + A system for handling and tracking complaints and software installation requests.
     + User manuals to help students, teachers, and the laboratory coordinator use the system effectively.
  2. Schedule Summary:

|  |  |
| --- | --- |
| Phase | Duration |
| Planning | 1 Week |
| Analysis | 3 Week |
| Design | 4 Week |
| Coding | 4 Week |
| Testing | 1. Week |

1. References

Project Management Institute (PMI):

Website: [https://www.pmi.org](https://www.pmi.org/)

Articles on Agile methodology, project planning, and risk management.

Trello Blog (Project Management Tools):

Website: [https://blog.trello.com](https://blog.trello.com/)

Guides on task management and team collaboration.

Atlassian (Agile Project Management):

Website: <https://www.atlassian.com/agile>

Articles on Agile workflows and iterative development.

1. Definitions

**Laboratory Management System:**

A system designed to help manage and organize information about labs, including computers, software, and user requests.

**Hardware Specifications:**

Details about the physical parts of a computer, like the processor, memory, storage, and other components.

**User:**

Anyone who uses the system, such as students, teachers, or lab staff.

**Laboratory Coordinator:**

The person in charge of managing the labs, including maintaining computers, software, and handling user requests.

**Stakeholder:**

Anyone with an interest in the project, such as students, teachers, lab staff, or department heads.

1. Project Context

#### 4.1 Process Model:

**Agile Approach:**

We’ll use a flexible and step-by-step method to build the system. This means we’ll work in small stages, regularly checking in with students, teachers, and the lab coordinator to make sure we’re on the right track.

**Key Phases:**

The project will move through planning, designing, building, testing, launching, and ongoing support. After each phase, we’ll review progress and make improvements based on feedback.

#### 4.2 Methods, Tools, and Techniques:

**Methods:**

**Talking to Users:** We’ll chat with students, teachers, and the lab coordinator to understand what they need from the system.

**Creating Prototypes:** We’ll build early versions of the system to get feedback before making the final product.

**Testing Along the Way:** We’ll test the system regularly to catch any issues early and make sure everything works smoothly.

**Tools:**

* + - **Project Management:** We’ll use tools like Trello or Jira to keep track of tasks and deadlines.
    - **Building the System:** We’ll use programming languages like Python and JavaScript, along with frameworks like Django or React, to create the website.
    - **Database:** We’ll use a system like MySQL or PostgreSQL to store all the information about computers, software, and user requests.
    - **Design:** Tools like Figma or Adobe XD will help us design a user-friendly interface for the website.
  + **Techniques:**
    - **Focus on Users:** The system will be designed to be simple and easy for students, teachers, and the lab coordinator to use.
    - **Regular Updates:** We’ll keep improving the system as we go, based on feedback and testing.

#### 4.3 Product Acceptance Plan:

**Testing the System:**Before the system goes live, we’ll test it thoroughly to make sure it works well. This includes:

* + **Checking Features:** Making sure all parts of the system (like complaint registration and installation requests) work as they should.
  + **User Testing:** Students, teachers, and the lab coordinator will try out the system to make sure it’s easy to use and meets their needs.
  + **Performance Testing:** Ensuring the system can handle lots of users without slowing down.

**Approval Criteria:** The system will be ready to launch when:

* + All features are working correctly.
  + Users are happy with how the system works and find it easy to use.
  + Any issues or bugs found during testing are fixed.

**Final Sign-Off:** Once everything is tested and approved, the lab coordinator and other stakeholders will give the green light to launch the system.

1. **Project Plaining**
   1. **Project Work Plan**

**Plan & Gather Requirements:**

Talk to users, list system needs.

Deliverable: Requirements Document.

**Design System:**

Create website design and database plan.

Deliverable: Design Document & Prototype.

**Build System:**

Develop front-end, back-end, and database.

Deliverable: Beta Version.

**Test & Fix:**

Test features, fix bugs, get user feedback.

Deliverable: Test Reports & Bug-Free System.

**Launch & Train:**

Deploy system, create guides, train users.

Deliverable: Live System & Training Materials.

**Maintain:**

Monitor, fix issues, update system.

Deliverable: Regular Updates & Support.

* 1. **Work Activities**

Plan: Talk to users, list needs.

Design: Draw website, plan database.

Build: Create front-end, back-end, database.

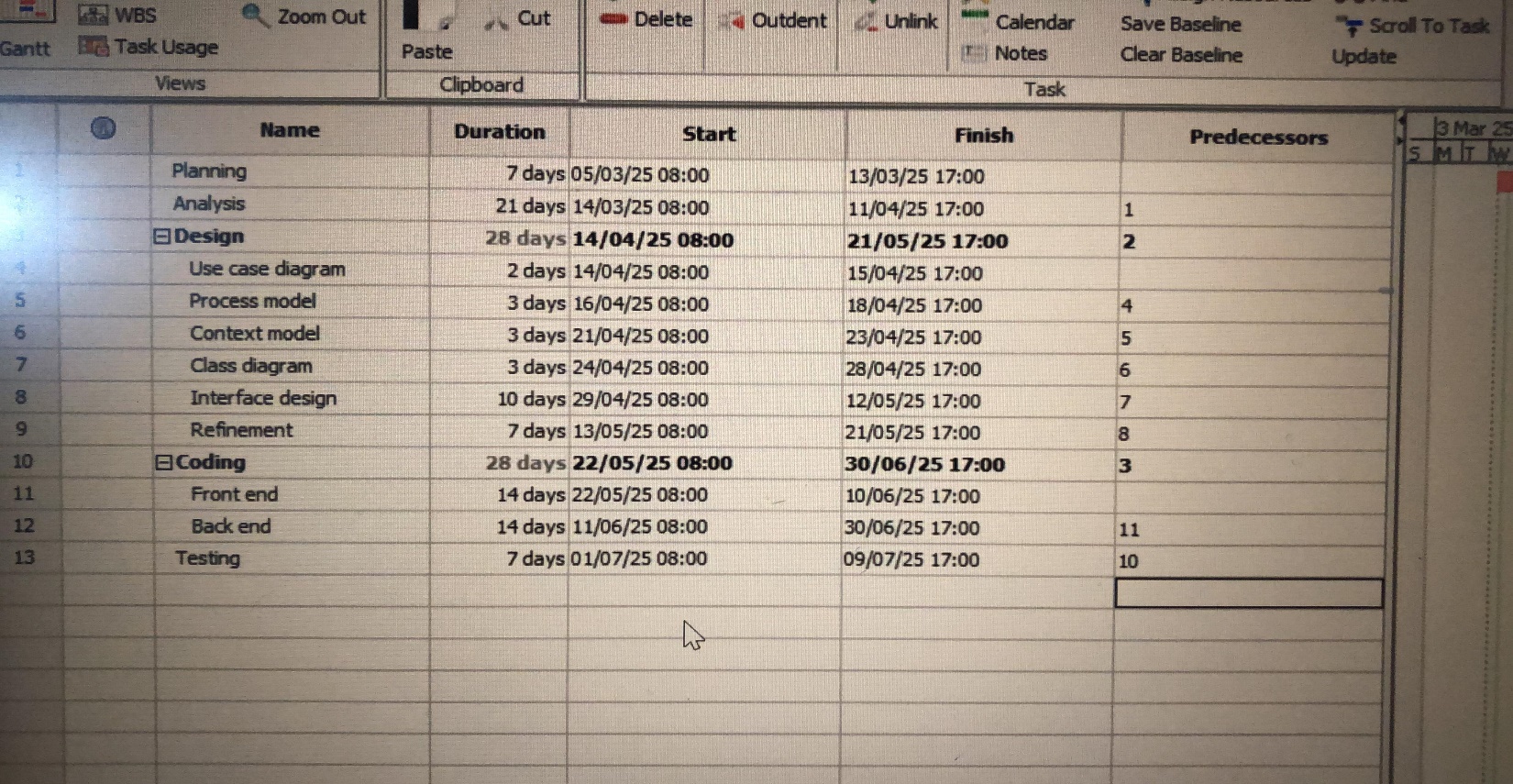
Test: Test features, fix bugs, get feedback.

Launch: Deploy system, train users.

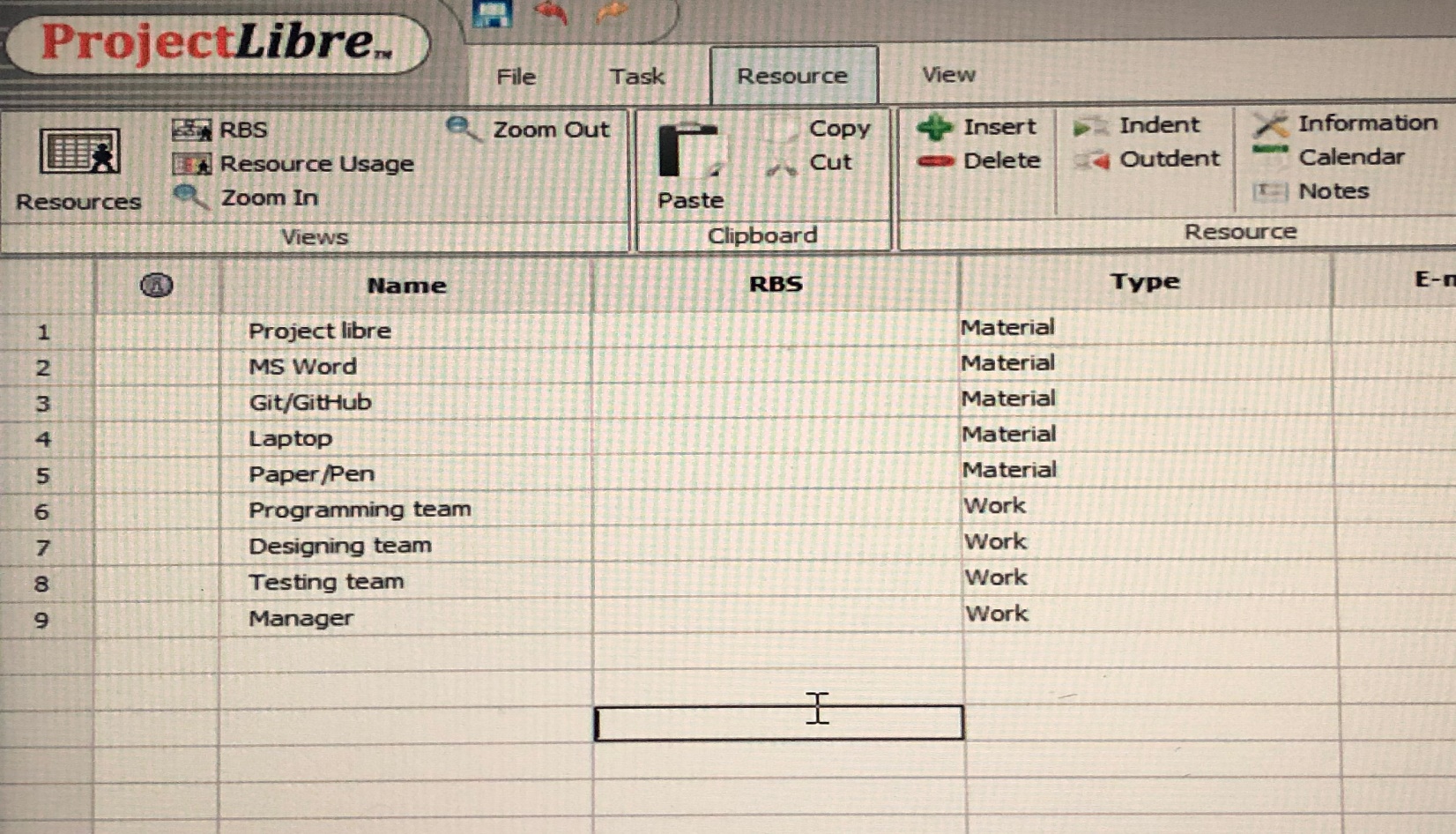
Maintain: Monitor, fix, update.

#### 

#### ****Schedule Allocation****



#### ****Resource Allocation****



1. Risk Management for the Laboratory Management System

Managing risks is critical to ensure the system works smoothly, stays secure, and meets user needs. Below are key risks and strategies to address them:

* 1. **Technical Risks**
  + **System Downtime:**

Risk: The website or database crashes, disrupting lab operations.

Mitigation:

Use reliable cloud hosting or backup servers.

Regularly test backups and implement auto-recovery tools.

* + **Data Security Breaches:**

Risk: Sensitive data (e.g., user complaints, installation requests) is hacked.

Mitigation:

Encrypt user data and use secure authentication (e.g., two-factor login).

Conduct regular security audits.

* + **Software/Hardware Compatibility Issues:**

Risk: New software installations conflict with existing lab hardware or software.

Mitigation:

Test all software updates in a controlled environment before deployment.

Maintain a compatibility checklist for lab computers.

* 1. **Operational Risks**
  + **User Resistance:**

Risk: Students or teachers avoid using the system due to complexity.

Mitigation:

Design a simple, intuitive interface.

Provide training sessions and user guides.

* + **Inaccurate Records:**

Risk: Outdated hardware/software data due to untimely updates.

Mitigation:

Assign the lab coordinator to review and update records weekly.

Automate reminders for updates.

* 1. **Project Risks**
  + **Delays in Development:**

Risk: Missing deadlines due to technical challenges or team bottlenecks.

Mitigation:

Break tasks into smaller milestones with buffer time.

Use project management tools (e.g., Trello) to track progress.

* + **Budget Overruns:**

Risk: Costs exceed estimates for hosting, tools, or training.

Mitigation:

Plan a contingency budget (e.g., 10-15% extra).

Prioritize free/open-source tools where possible.

* 1. **External Risks**
  + **Dependency on Existing IT Infrastructure:**

Risk: Poor network quality or outdated lab computers slow down the system.

Mitigation:

Work with the IT department to upgrade infrastructure.

Design the system to work on low bandwidth as a fallback.

* + **Regulatory Changes:**

Risk: New policies (e.g., data privacy laws) require system updates.

Mitigation:

Monitor regulations and build a flexible system to adapt to changes.