Healthcare Pharmaceuticals Ltd.

REQUIREMENT ANALYSIS

1. Project Information	
Project Name:	**Task Management System**
Project Description:	The Task Management Web App is a browser-based application designed to streamline task creation, assignment, tracking, and project-module management for teams. This system will offer a modern, accessible, and user-friendly experience with secure role-based access and real-time updates to improve internal workflow and productivity.

2. Project Team				
Project Manager:	**Name**	**Designation**	**Contact**	**E-mail**
Team Members:	Md. Najim Uddin	Manager (MIS)	01888818557	najim@hpl.com.bd
	Jakia Rahman	Intern (MIS)	01768997799	jakiarahman326@gmail.com
	Kamrun Nahar	Intern (MIS)	01981006695	kamrunnaharurmi2000@gmail.com

3. Project Scope

Project Purpose

Manual or desktop-only systems lack flexibility, collaboration features and remote accessibility. This project aims to solve these challenges by offering a centralized task management platform that enables project managers and team members to monitor project progress, assign responsibilities, and manage modules with transparency and efficiency.

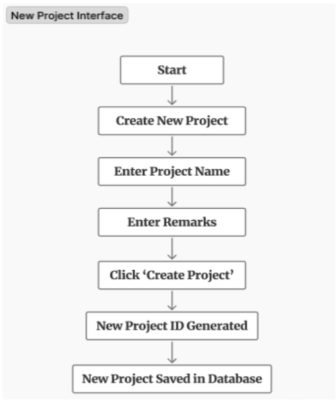
Objectives

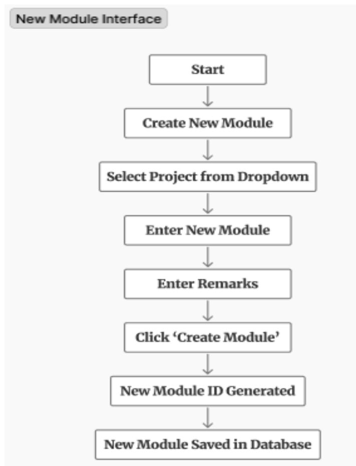
a. Build a responsive web-based task management system accessible to all team members.
b. Design role-based access for Admins, Employees, and Managers.
c. Create a project-module structure that mirrors organizational workflows.
d. Implement real-time task creation, status tracking, and deadline management.
e. Enable filtering, sorting, and reporting of tasks by status, user, or deadline.
f. Automated email notifications to team members when tasks are assigned or updated.
g. Export Task Reports/ Team Member Reports for annual performance tracking.

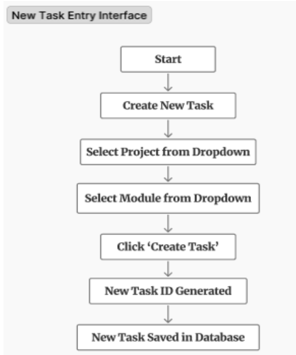
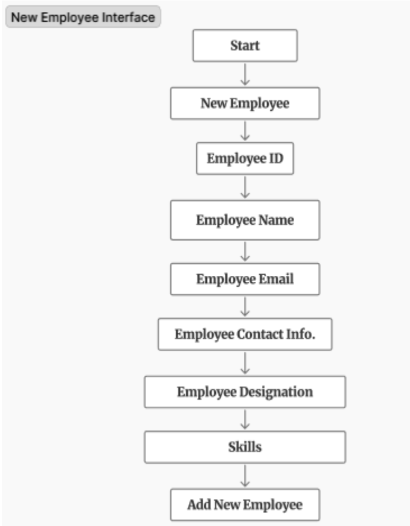
4. Deliverables

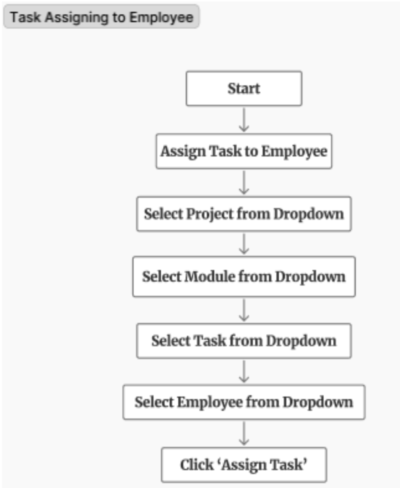
Date	Tasks	Assigned Team Member
	Requirements Analysis	
	Wireframes / UI Design	
	Database Schema & ERD	
	User Authentication Module	
	Project, Module, Task Entry Interface	
	All Feature Management Dashboard	
	Filtering & Search	
	Export / Report Generation	
	QA Test Cases	
	Early Testing Report	
	Bug Fix & Final Revisions	
	Deployment	

<table>
<tr><th colspan="3">5. Functional Features Analysis</th></tr>
<tr><th>Features</th><th colspan="2">Description</th></tr>
<tr><td>1. User Login & Registration Module</td><td colspan="2">◇ Purpose:
To authenticate users (Admin, Manager, Employee) and grant them access to system features based on their role.
◇ Functional Details:
• Login:
○ Users enter email/username and password.
○ System validates credentials.
○ Role-based redirection (e.g., Admin Dashboard, Employee Task board).
○ Invalid credentials show error.
• Registration:
○ Only Admin can register new employees.
○ Input: Name, Email, Role, Department, Designation, Password.
○ Auto-generation of employee ID (optional).
○ Validation of unique email.
○ Confirmation message after successful registration.</td></tr>
<tr><td>2. New Project Entry Interface</td><td>◇ Purpose:
To allow Admin/Project Manager to create new projects and define their scope.
◇ Functional Details:
• Input fields:
○ Project Name
○ Description / Remarks
○ Client Name (optional)
○ Estimated Start and End Dates

◇ Validation:
• Auto-generation of Project ID
• Once created, displays success confirmation box, appears in project list and is available for Module and Task assignment.</td><td>◇ Flowchart:

</td></tr>
<tr><td>3. New Module Entry Interface</td><td>◇ Purpose:
To break down a project into smaller manageable modules or milestones.
◇ Functional Details:
• Selection of Parent Project
• New Module Name
• Description / Remarks

◇ Validation:
• Cannot add module without selecting a valid project.
• Displays success confirmation or error messages.
• Appears in Module List for selected Project.</td><td>◇ Flowchart:

</td></tr>
</table>

4. New Task Entry Interface	◇ **Purpose:** To create specific tasks under a module of a project. ◇ **Functional Details:** • Select Project → Then Module (dependent dropdown) • Task Title and Description / Remarks • On save: Task appears in Dashboard. ◇ **Flowchart:** 
5. New Employee Entry Interface	◇ **Purpose:** To allow the admin or authorized personnel to add new employees into the system. ◇ **Flowchart:**  ◇ **Functional Details:** **Interface Components:** **Employee ID:** ○ Manually entered. **Employee Name:** ○ Full name of the employee. ○ Required field. **Employee Email:** ○ Must follow valid email format. ○ Required for task notifications and login (if applicable). ○ Checked for uniqueness to avoid duplicates. **Contact Info:** ○ Phone number or other relevant contact details. ○ Must follow phone format validation rules. **Designation:** ○ Dropdown or text field to enter roles such as "Developer," "Tester," "Manager," etc. ○ May be used later for filtering or assigning specific roles. **Skills:** ○ Multi-line input or tag-based entry. ○ Lists technical or domain-specific skills (e.g., C#, React, QA Testing). **Add New Employee Button:** ○ Submits the data to the backend for storage. ○ Displays success confirmation or error messages.

6. Assign Task to Employee	◇ **Purpose**: To provide a structured interface for assigning tasks to specific employees under defined projects and modules. ◇ **Functional Details:** • **Interface Components:** **Dropdown: Project Selection** ○ Lists all active or ongoing projects. **Dropdown: Module Selection** ○ Dynamically loads only the modules under the selected project. **Dropdown: Task Selection** ○ Loads tasks under the selected module. **Dropdown: Employee Selection** ○ Lists all employees. ○ May include optional filters (by role, department, workload—future scope). **Attach files:** Filed to attach briefs, designs, PDFs etc. **Assign Button:** ○ On click 'Assign Task', the selected task is assigned to the chosen employee. ○ Displays a confirmation message upon success. ◇ **Flowchart:** 
7. Filtering & Search	◇ **Purpose:** To quickly locate tasks by filters such as project, module, employee, status, or date range. ◇ **Functional Details:** • Search bar for feature title/keywords • Filter dropdowns: ○ By Project ○ By Module ○ By Employee ○ By Status ○ By Date or Deadline • Instant display of filtered tasks • Export filtered view as CSV or PDF (Future Scope)
8. Employee Information Interface	◇ **Purpose:** To maintain employee profiles and manage internal user data. ◇ **Functional Details:** • List view of all registered employees. • **Detailed info per employee:** ○ Name ○ ID, Role (Admin, Manager, Developer) ○ Email, Phone ○ Designation, Department, Skills ○ Total Tasks Assigned ○ Current Workload • Edit/Update/Delete option for Admin • Status (Active/Inactive)

6. Future Scope

1. Project & Module Management Dashboard	◇ **Purpose:** To monitor the current status and progress of all ongoing projects and their modules. ◇ **Functional Details:** • **Overview cards for each project:** ○ Total Modules ○ Completed Tasks ○ Pending Tasks ○ Overall Progress (%) • Expand each Project to view associated Modules and Task summary • Status color codes (Red for overdue, Yellow for ongoing, Green for completed)

7. Non – Functional Feature Analysis

1. Performance

- The system should load pages within 2 seconds.
- Task assignment, search, and dashboard updates should be real-time or near-instant.

2. Usability

- The interface should be simple and user-friendly, so that even non-technical employees can use it without training.
- Dropdowns, forms, and dashboard must be easy to understand and navigate**.**

3. Reliability

- The system should work consistently without crashes or data loss.
- It should handle multiple users using the system at the same time without issues.

4. Security

- User passwords must be encrypted.
- Only logged-in users can access task/project data.
- Different roles (like Admin, Developer, Manager) should have restricted access to certain features.

6. Maintainability

- The code should be written in a clean and organized way so developers can easily update or fix issues.
- Error messages should be clear for quick debugging.
- The database and UI should be flexible for future feature additions.

7. Portability

- The system should work on modern web browsers like Chrome, Firefox, and Edge.
- It should be deployable on any standard hosting environment.

8. Data Integrity

- The system must ensure that task assignments are accurate and no duplicates are created.
- No two employees can be accidentally assigned the same unique task without tracking.

8. Quality Control

Quality Assurance (QA)

Purpose:
To ensure the development process follows best practices and standards so that quality is built into the product from the beginning.
Details to Maintain:

- QA will be maintained from the planning phase.
- Proper documentation, wireframes, and requirement reviews will be part of QA.
- Code reviews and peer collaboration will be ensured before moving to testing.
- Clear communication will be maintained between developer and tester to avoid misunderstanding of features.

Quality Control (QC)

Purpose:
To check the final product for defects or issues before delivery.
Details to Maintain:

- The final system will be verified against the documented requirements.
- All forms, dashboards, and dropdowns will be tested to ensure they work correctly.
- Task assignment logic will be cross-checked to make sure tasks are not misassigned.
- Validation messages, error handling, and edge cases will be tested.

9. Testing Phase

Purpose:
To identify and fix bugs, confirm that all functional requirements are met, and the system behaves as expected.
Details to Maintain:
a. Test Planning

- A test plan will be created with defined test cases for each feature.
- Testing will cover both functional and non-functional areas.

b. Types of Testing

- **Unit Testing:** Each function (like form submission, dropdown selection) will be tested individually.
- **Integration Testing:** Check how the modules interact (e.g., assigning a task based on selected project and module).
- **System Testing:** Test the system as a whole (full workflow of creating a task and assigning it).
- **User Acceptance Testing (UAT):** Final testing from a user's point of view to confirm it's ready for deployment.

c. Bug Tracking & Fixing

- Bugs found during testing will be logged with details like feature name, steps to reproduce, expected vs. actual behavior.
- Bugs will be fixed and retested before marking the feature as "complete."
- Retesting after bug fixes will be tracked separately to ensure no regression occurs.

10. Notes

Revision History

Version	Date	Name	Description
1.0	20.07.2025	Jakia Rahman (Intern - MIS)	Initial Draft of the Requirement Analysis document.