



ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

ADB ROAD, SURAMPALEM, E.G. DIST.



DEPARTMENT OF



Name:

PIN No.

--	--	--	--	--	--	--	--	--	--

*Certified that this is the bonafide record of
practical work done by*

Mr. / Ms.

a students of with PIN No.....

in the Laboratory during the year

No of Practicals Conducted

--

No. of Practicals Attended

--

Signature - Faculty Incharge

Signature - Head of the Department

Submitted for the Practical examination held on.....

EXAMINER - 1

EXAMINER - 2



CONTENT

Sl.No.	Date	Experiment Title	Page No
1.		Take any real time problem and do the following experiments.	
a)		Write down the problem statement for a suggested system of relevance. Develop Flow-Charts to understand basic problem-solving technique.	
b)		Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.	
c)		Using COCOMO model estimate effort.	
d)		Perform Estimation of effort using FP Estimation for chosen system.	
e)		Analyze the Risk related to the project and prepare RMMM plan.	
f)		Develop Time-line chart and project table using PERT or CPM project scheduling methods.	
g)		Draw E-R diagrams, DFD, CFD and structured charts for the project.	
h)		Design of Test cases based on requirements and design.	
i)		Prepare FTR.	
j)		Prepare Version control and change control for software configuration items.	
2.		Using UML/JAVA, design Abstract Factory design pattern.	
3.		Using UML/JAVA, design Builder design pattern.	
4.		Using UML/JAVA, design Façade design pattern.	
5.		Using UML/JAVA, design Bridge design pattern.	
6.		Using UML/JAVA, design Decorator design pattern.	
7.		User gives a print command from a word document. Design to represent this chain of responsibility design pattern.	

**Experiment – 1 : Take any real time problem and do the following experiments.**

The Task Management App is a web-based application designed to help users organize and manage their tasks effectively. The app provides a user-friendly interface for creating, prioritizing, and tracking tasks, along with features for setting due dates, receiving reminders, and collaborating with team members. Tasks can be categorized, filtered, and sorted based on various criteria to streamline workflow and improve productivity. Additionally, the app offers cross-platform accessibility, allowing users to access their tasks from any device with internet connectivity. With its intuitive design and robust functionality, the Task Management App empowers users to stay organized, focused, and on top of their tasks.



Experiment – 1.a: Write down the problem statement for a suggested system of relevance. Develop Flow-Charts to understand basic problem solving technique.

Problem Statement

In today's fast-paced world, individuals and teams struggle to effectively manage their tasks and deadlines, leading to decreased productivity and increased stress levels. Existing task management tools often lack user-friendly interfaces, robust features, and cross-platform accessibility, hindering users from efficiently organizing and tracking their tasks. There is a need for a comprehensive Task Management App that offers intuitive task management solutions, customizable features, and seamless integration across devices, empowering users to optimize their workflow and achieve their goals with ease.

why i want to work on this problem:

Working on this problem offers an opportunity to combine personal experience, professional insight, and broader significance to drive motivation and passion for finding innovative solutions in the realm of task management.

Personal Experience:

Missed Deadlines and Reduced Productivity

Overwhelmed Workloads and Increased Stress

Professional Insight:

Recognizing the Impact on Organizational Success

Understanding the Importance for Work Culture

Broader Significance:

Contribution to Individual and Team Empowerment

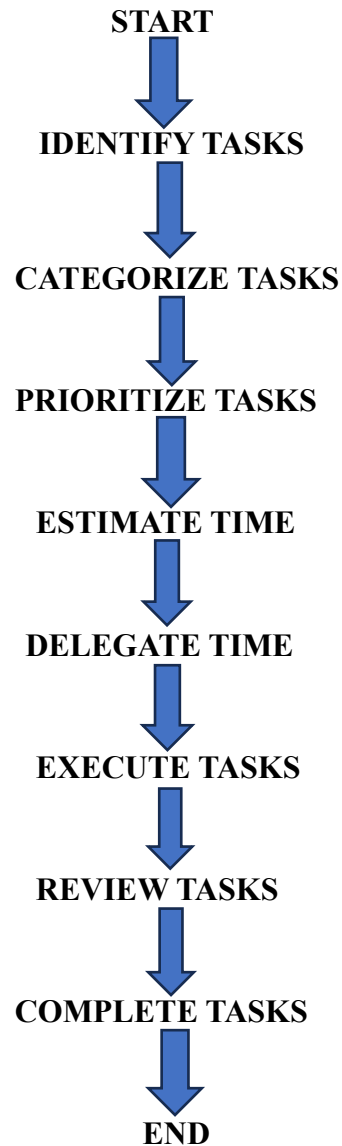
Addressing Universal Relevance and Impact

Motivation:

Passion for Productivity and Goal Achievement

Desire to Make a Meaningful Difference

FLOW-CHART



This flow chart follows intermediate task management model adds additional steps for more structured task management including categorization, time estimation, delegation and periodic review.

Experiment – 1.b: Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.

Introduction:

Effective task management is crucial in today's fast-paced world, yet individuals and teams often struggle to stay organized and productive. The Task Management App offers a solution, providing intuitive interfaces, robust features, and cross-platform accessibility to streamline task organization and enhance productivity. With customizable categories, due date settings, and collaboration tools, users can tailor their task management experience to their needs. This app aims to empower users to conquer their tasks, achieve their goals, and improve overall productivity.

Requirements:

Functional Requirements:

User Registration and Authentication:

Users should be able to register for an account and log in securely.

Authentication mechanisms should be implemented to protect user accounts and data.

Task Management:

Users should be able to create, edit, delete, and prioritize tasks.

Tasks should include attributes such as task name, due date, priority level, category, and description.

Task Organization:

Users should be able to organize tasks into categories or projects.

Tasks should be sortable and filterable based on criteria such as due date, priority, and category.

Task Assignment: Tasks should support assignment to specific users or teams, enabling accountability and collaboration.

Task Prioritization: Users should be able to prioritize tasks based on urgency and importance, allowing for better focus and decision-making.

Reminders and Notifications:

Users should receive reminders and notifications for upcoming task deadlines.

Notifications should be customizable and configurable based on user preferences.

Collaboration:

Users should be able to share tasks with other users or assign tasks to team members.

Collaborative features such as comments, mentions, and file attachments should be available for task discussions.

Cross-Platform Accessibility:

The app should be accessible across different devices and platforms, including web browsers, mobile devices, and desktop computers.

The user interface should be responsive and adaptive to various screen sizes and resolutions.



Non-Functional Requirements:

Performance:

The app should be responsive and performant, even under heavy user load.

Database queries and operations should be optimized for efficiency.

Security:

User data should be securely stored and transmitted using encryption protocols.

Access controls should be implemented to prevent unauthorized access to user accounts and data.

Scalability:

The app should be designed to scale horizontally to accommodate increasing numbers of users and tasks.

Scalability considerations should be made for both frontend and backend components.

Reliability:

The app should be reliable and resilient, with minimal downtime and data loss.

Error handling and recovery mechanisms should be in place to handle unexpected failures gracefully.

Usability:

The user interface should be intuitive and easy to navigate, with clear instructions and feedback.

Accessibility features should be incorporated to ensure usability for users with disabilities.

System Features:

Task Dashboard: Provide users with a central dashboard to view all their tasks, organized by status and priority.

Task Details: Allow users to view and edit detailed information for each task, including description, due date, and assigned user(s). **Task Notifications:** Notify users of upcoming deadlines or changes to assigned tasks via email or in-app notifications.

Task Collaboration: Enable users to collaborate on tasks by sharing comments, attachments, and status updates.

Task Reporting: Provide users with the ability to generate reports on task progress, completion rates, and productivity metrics.

External Interface Requirements:

API Integration: Integrate with external systems or services, such as calendar applications or project management tools, via APIs for data exchange.

Data Import/Export: Support importing and exporting of task data in standard formats for interoperability with other systems.

**Constraints:**

Budgetary Constraints: The system development should adhere to the allocated budget for the project.

Technical Limitations: Consider any technical constraints, such as compatibility with legacy systems or infrastructure limitations.

Assumptions and Dependencies:

User Behavior: Assume that users will actively engage with the system and perform tasks as expected.

System Environment: Depend on the availability of necessary infrastructure and resources for system deployment and operation.

Experiment – 1.c: Using COCOMO model estimate effort.**Basic COCOMO Model:****Assumptions**

Project Size :10000 lines of code (LOC)

COCOMO Model: Organic

Error Adjustment Factor (EAF) :1.0(Neutral Adjustment)

Effort = a x (size) pow(b) x EAF

a=2.4

b=1.05

size = 10000 (Loc)

EAF = 1.0

Effort = 2.4 x (10000) pow (1.05) x 1.0

Effort = 2.4 x (10) pow (1.05)

Effort = 2.4 x 11.047

Effort = 26.5128 persons-months

Time (T) = C (E) pow (d)

= 2.5 (26.5128) pow (0.38)

= 2.5 x 7.6477

T = 19.11925

INTERMEDIATE MODEL:

Project Size: 20000 lines of code(loc)

COCOMO model: Organic

Effort Adjustment Factor (EAF): Detailed Factors Considered

Effort = a x (size) pow (b) x EAF

a = 2.4

b = 1.05

EAF = 1.2

Size = 20000

Where EAF is effort is calculated as;

EAF = (Summation I = 1 to N Rating I)

Where 'n' is the total numbers of cost divers and the rating 'T' is the rating assigned to each cost divers and size is the estimated project size in KLOC.

From, Product Attribute (Nominal)

The Complexity of the product = 1.00

Hardware Attribute (High)

The Required Turn-Around Time = 1.07

Project Attributes (Low)

Use of Software Tools = 1.10

EAF = $1.00 \times 1.07 \times 1.10$

EAF = **1.177**

EFFORT (E) = 1.177×26.5128

E = **31.2055 Persons-Months**

Time (T) = $2.5 \times (31.2055)^{\text{pow}(0.38)}$

= 2.5×3.6966

= 9.2415

T = **9 Months**

People Required (R) = E/T

$R = 31.2055/9.2415$

$R = 3.3744$

R = **3 persons**

**Experiment – 1.d: Perform Estimation of effort using FP Estimation for chosen system.**

Function Point (FP) estimation is a method used to estimate the size, effort, and time required for software development projects. Here are the steps to perform FP estimation for your task management system:

Step 1: Identify the Functional User Requirements

1. **External Inputs (EI):** These are the inputs into the system. For the task management system, these could be:
 - User Registration
 - Task Creation
 - Task Update
 - Task Deletion
 - Project Creation
 - Category Creation
2. **External Outputs (EO):** These are outputs from the system. For the task management system, these could be:
 - Task List Display
 - Project List Display
 - Notification Display
3. **External Inquiries (EQ):** These are inquiries where the user receives data but no significant processing occurs. For the task management system, these could be:
 - Task Search
 - Project Search
4. **Internal Logical Files (ILF):** These are logical groups of data maintained within the system. For the task management system, these could be:
 - User Data
 - Task Data
 - Project Data
 - Category Data
 - Notification Data
5. **External Interface Files (EIF):** These are logical groups of data used by the system but maintained by another application. For the task management system, these might not be applicable if there is no integration with external systems, or they could be:
 - External Calendar Integration

Step 2: Assign Complexity Weights

Assign weights based on the complexity (low, average, high) of each type of function.

Function Type	Weight (Low)	Weight (Average)	Weight (High)
External Inputs (EI)	3	4	6
External Outputs (EO)	4	5	7
External Inquiries (EQ)	3	4	6
Internal Logical Files (ILF)	7	10	15
External Interface Files (EIF)	5	7	10



Step 3: Calculate Unadjusted Function Points (UFP)

Identify the count of each function type and assign complexity.

- **External Inputs (EI):**
 - User Registration (Average) = $1 * 4 = 4$
 - Task Creation (Average) = $1 * 4 = 4$
 - Task Update (Average) = $1 * 4 = 4$
 - Task Deletion (Average) = $1 * 4 = 4$
 - Project Creation (Average) = $1 * 4 = 4$
 - Category Creation (Average) = $1 * 4 = 4$
- **External Outputs (EO):**
 - Task List Display (Average) = $1 * 5 = 5$
 - Project List Display (Average) = $1 * 5 = 5$
 - Notification Display (Average) = $1 * 5 = 5$
- **External Inquiries (EQ):**
 - Task Search (Average) = $1 * 4 = 4$
 - Project Search (Average) = $1 * 4 = 4$
- **Internal Logical Files (ILF):**
 - User Data (Average) = $1 * 10 = 10$
 - Task Data (Average) = $1 * 10 = 10$
 - Project Data (Average) = $1 * 10 = 10$
 - Category Data (Average) = $1 * 10 = 10$
 - Notification Data (Average) = $1 * 10 = 10$
- **External Interface Files (EIF):**
 - External Calendar Integration (Low) = $1 * 5 = 5$

Step 4: Compute the Unadjusted Function Points (UFP)

$$\begin{aligned} \text{UFP} &= \sum (\text{count} \times \text{weight}) \\ \text{UFP} &= 4 + 4 + 4 + 4 + 4 + 4 + 5 + 5 + 5 + 4 + 4 + 10 + 10 + 10 + 10 + 10 + 5 = 102 \\ \text{UFP} &= 4 + 4 + 4 + 4 + 4 + 4 + 5 + 5 + 5 + 4 + 4 + 10 + 10 + 10 + 10 + 10 + 5 = 102 \end{aligned}$$

Step 5: Calculate the Value Adjustment Factor (VAF)

Rate the 14 General System Characteristics (GSCs) on a scale of 0 to 5 and compute the VAF.

For simplicity, let's assume an average complexity rating of 3 for all GSCs.

$$\begin{aligned} \text{Total Degree of Influence} &= 14 \times 3 = 42 \\ &= 14 \times 3 \\ &= 42 \text{ Total Degree of Influence} \\ &= 14 \times 3 = 42 \end{aligned}$$

$$\begin{aligned} \text{VAF} &= 0.65 + (0.01 \times \text{Total Degree of Influence}) = 0.65 + (0.01 \times 42) \\ &= 0.65 + 0.42 = 1.07 \\ \text{VAF} &= 0.65 + (0.01 \times \text{Total Degree of Influence}) \\ &= 0.65 + (0.01 \times 42) = 0.65 + 0.42 \\ &= 1.07 \end{aligned}$$

Step 6: Calculate the Adjusted Function Points (AFP)

$$\begin{aligned}
 \text{AFP} &= \text{UFP} \times \text{VAF} \\
 &= 102 \times 1.07 = 109.14 \text{ AFP} \\
 &= \text{UFP} \times \text{VAF} \\
 &= 102 \times 1.07 \\
 &= 109.14 \text{ AFP} \\
 &= \text{UFP} \times \text{VAF} \\
 &= 102 \times 1.07 \\
 &= 109.14
 \end{aligned}$$

Step 7: Estimate Effort

To estimate the effort, you need to determine the productivity rate, which is typically measured in function points per person-month. Let's assume an average productivity rate of 20 function points per person-month.

$$\begin{aligned}
 \text{Effort} &= \frac{\text{AFP}}{\text{Productivity Rate}} \\
 &= \frac{109.14}{20} \approx 5.46 \text{ person-months} \\
 &= \frac{\text{AFP}}{\text{Productivity Rate}} \\
 &= \frac{109.14}{20} \approx 5.46 \text{ person-months}
 \end{aligned}$$

$$\begin{aligned}
 \text{Effort} &= \frac{\text{AFP}}{\text{Productivity Rate}} \\
 &= \frac{109.14}{20} \\
 &\approx 5.46 \text{ person-months}
 \end{aligned}$$


Experiment – 1.e: Analyze the Risk related to the project and prepare RMMM plan.

Risk	Probability	Impact	Severity	Mitigation	Monitoring	Management	Estimated Cost
Technology Obsolescence	Medium	Medium	Medium	Choose technology with strong support, regular updates	Regular review of technology trends	Be prepared to pivot to newer technologies	\$5,000
Integration Challenges	High	High	High	Conduct feasibility study, create robust integration plan	Regular integration testing and validation	Allocate dedicated resources for integration	\$10,000
Performance Issues	Medium	High	High	Implement performance testing, optimization	Regular performance reviews, load testing	Optimize code and infrastructure as needed	\$8,000
Scope Creep	High	Medium	High	Clearly define project scope, use change management	Regular scope reviews, stakeholder meetings	Approve changes only if they add significant value	\$7,000
Time Management	High	High	High	Develop realistic project schedule with buffer times	Weekly progress reviews, milestone checks	Reallocate resources to critical tasks as needed	\$12,000
Cost Overruns	Medium	Medium	Medium	Detailed budget planning, continuous financial monitoring	Regular budget reviews, expense tracking	Adjust project scope or resource allocation	\$6,000
Data Security	Low	High	Medium	Implement robust security measures, regular audits	Continuous monitoring of security systems	Update security protocols, respond to incidents	\$4,000

**Risk Monitoring and Review:**

Regularly review identified risks, report on their status, and develop contingency plans for high-impact risks. Implement a system for continuous risk assessment and communication with stakeholders.

Estimated Cost:

The estimated cost associated with addressing these risks depends on various factors such as the resources required for mitigation strategies, potential impact on project timelines, and the complexity of risk management activities. It's essential to allocate a contingency budget to cover unforeseen expenses related to risk mitigation.

Overall, incorporating risk analysis, RMMM planning, and estimated costs into the project budget ensures that potential risks are proactively managed, minimizing their impact on project success and financial outcomes.

**Experiment – 1.f: Develop Time-line chart and project table using PERT or CPM project scheduling methods.**

To estimate the total cost for RMMM, we will sum up the costs for each mitigation strategy

Let's assumed cost for RMMM is \$90,000

RMMM Cost Estimation = Cost Estimation of each risk/Assumed Cost Of RMMM

$$\begin{aligned} & \$50000 + \$10000 + \$8000 + \$7000 + \$12000 + \$6000 + \\ & \$9000 + \$4000 + \$8000 + \$3000 + \$10000 + \$7000 \end{aligned}$$

$$\text{RMMM} = \frac{\text{-----}}{\$90,000}$$

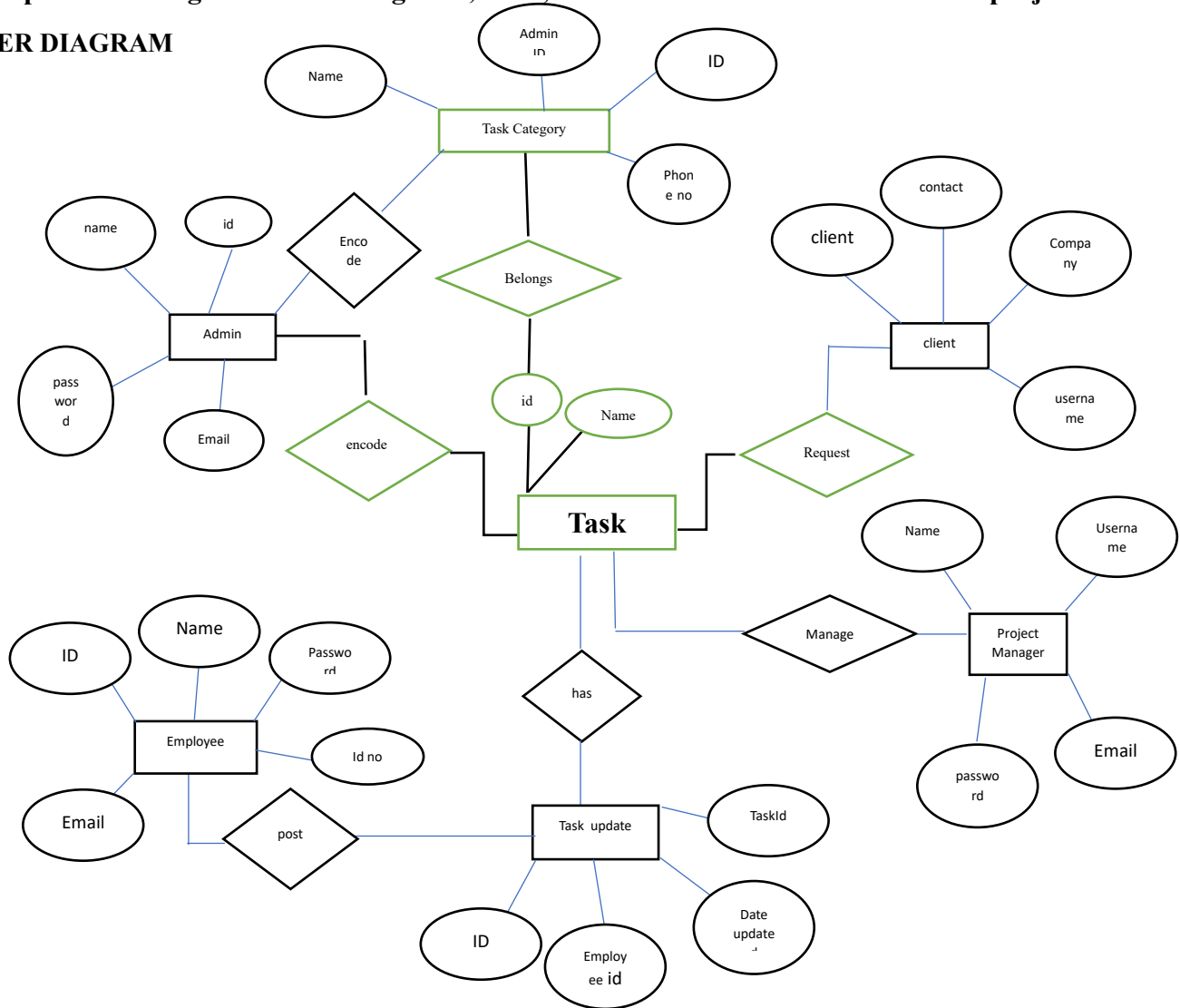
$$\text{RMMM} = \$89,000 / \$90,000$$

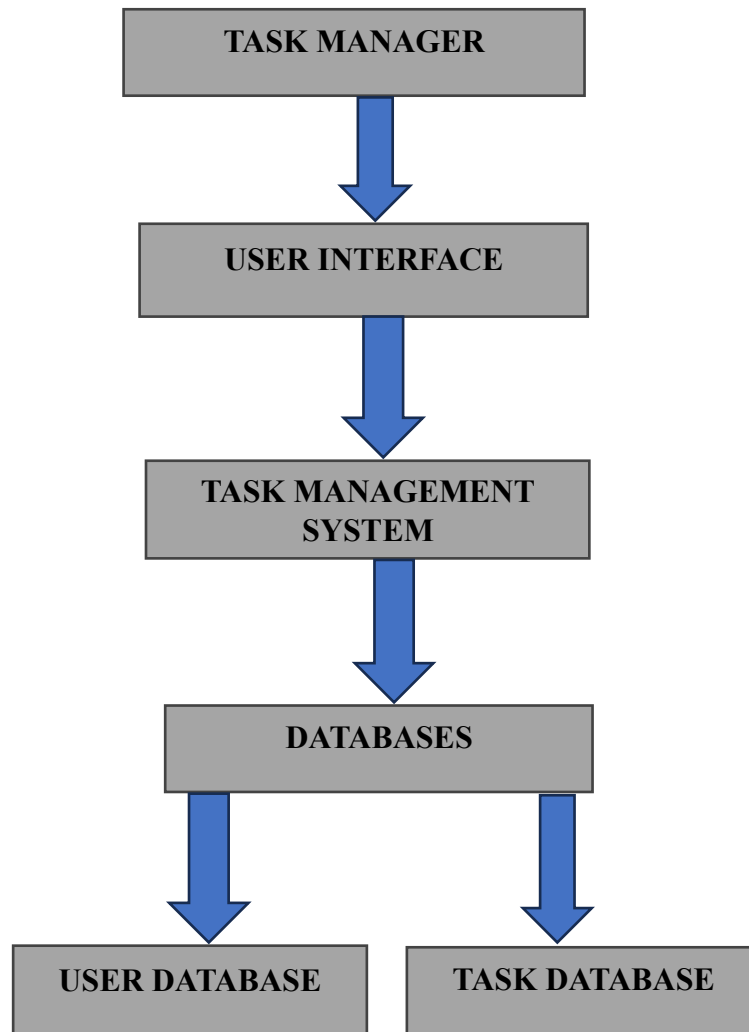
$$\text{RMMM} = 0.9888$$

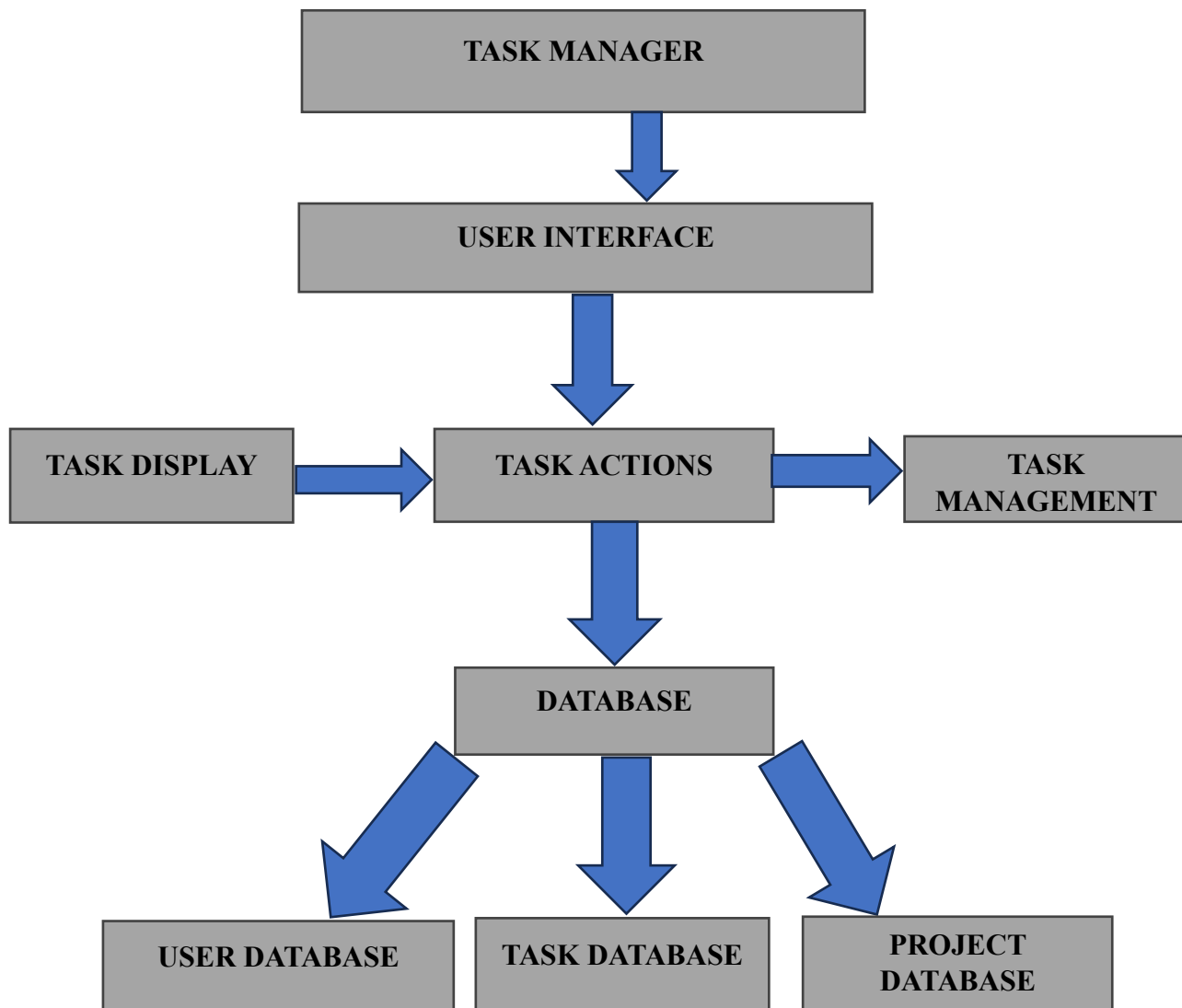


Experiment – 1.g: Draw E-R diagrams, DFD, CFD and structured charts for the project.

ER DIAGRAM



DFD-Level 0

DFR-Level 1



Experiment – 1.h: Design of Test cases based on requirements and design.

Module 1: User Authentication	
Project Name:	Task Management App
Module Name:	User Authentication
Reference Document:	
Created by:	Surendra Babu Dasari
Date of creation:	
Date of review:	

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC-UA-001	Verify user can register successfully	Enter valid details to register	None	1. Navigate to the registration page 2. Fill in details 3. Click "Register"	Valid User Details	Successful registration, user is redirected to the dashboard	User is logged in	Successful registration, user is redirected to the dashboard	PASS
TC-UA-002	Verify user cannot register with duplicate email	Enter existing email to register	None	1. Navigate to the registration page 2. Enter an existing email 3. Click "Register"	Existing Email	Error message indicating email already in use	Registration page is shown again	Error message indicating email already in use	PASS
TC-UA-003	Verify user can log in successfully	Need a registered user account	1. Navigate to the login page 2. Enter valid credentials 3. Click "Log In"	Valid User Credentials	Successful login, user is redirected to the dashboard	User is logged in	Successful login, user is redirected to the dashboard	PASS	
TC-UA-004	Verify user cannot log in with invalid credentials	Enter invalid credentials to login	Need a registered user account	1. Navigate to the login page 2. Enter invalid credentials 3. Click "Log In"	Invalid Credentials	Error message indicating invalid credentials	Login page is shown again	Error message indicating invalid credentials	PASS

Module 2:	Task Management
Project Name:	Task Management App
Module Name:	Task Management
Reference Document:	
Created by:	Surendra Babu Dasari
Date of creation:	
Date of review:	



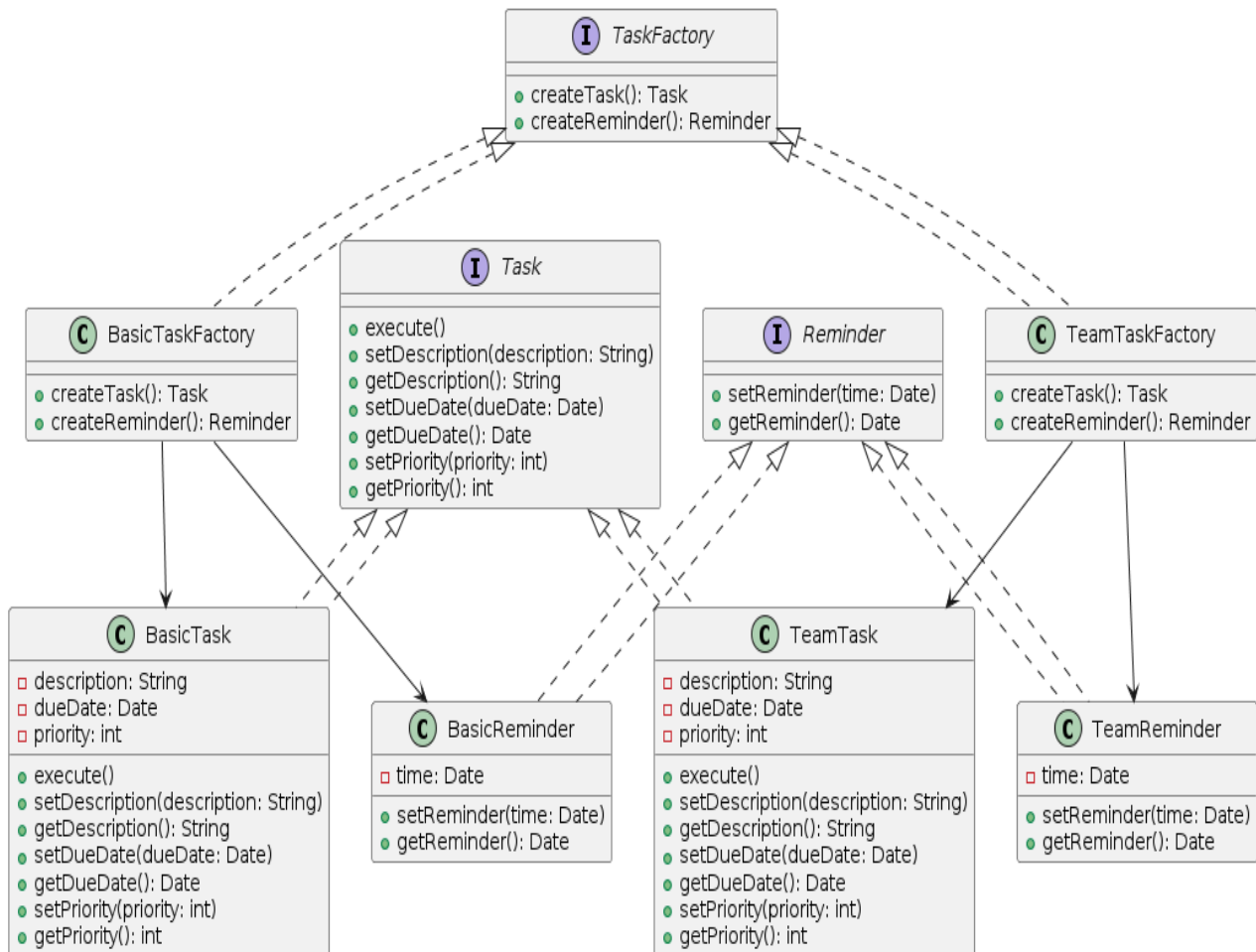
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
TC-TM-001	Create a new task	Enter task details to create a task	User must be logged in	1. Navigate to the "Create Task" section 2. Enter task details 3. Click "Save"	Task Details	Task appears in the task list	Task is created	Task appears in the task list	PASS
TC-TM-002	Edit task details	Modify task details	User must be logged in	1. Navigate to a task 2. Edit task details 3. Click "Save"	Updated Task Details	Task displays updated details	Task is updated	Task displays updated details	PASS
TC-TM-003	Delete a task	Remove a task from the list	User must be logged in	1. Navigate to a task 2. Click "Delete" 3. Confirm deletion	Task ID	Task is removed from the task list	Task is deleted	Task is removed from the task list	PASS
TC-TM-004	Update task status	Change task status	User must be logged in	1. Navigate to a task 2. Change task status 3. Click "Save"	Updated Status	Task's status updates accordingly	Task status is updated	Task's status updates accordingly	PASS

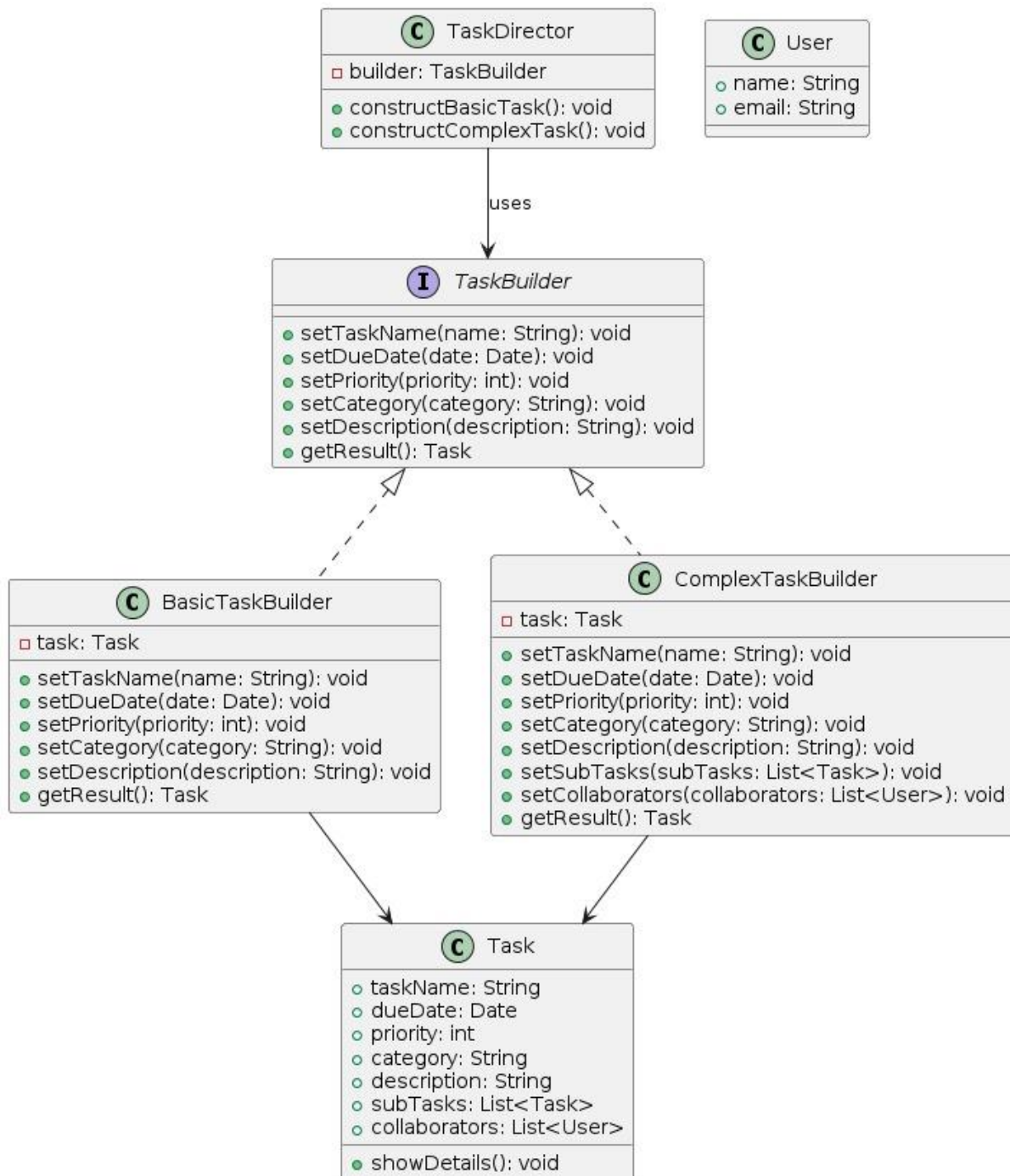
Module 3: Notifications and Reminders

Project Name:	Task Management App
Module Name:	Notifications and Reminders
Reference Document:	
Created by:	Surendra Babu Dasari
Date of creation:	
Date of review:	

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
TC-NR-001	Set due date for a task	Create task with a due date	User must be logged in	1. Create a task 2. Set a due date 3. Click "Save"	Task Details with Due Date	Task displays with the assigned due date	Due date is set	Task displays with the assigned due date	PASS
TC-NR-002	Receive reminder notification	Wait for reminder time	User must be logged in	1. Create a task with a due date 2. Wait for the reminder time	Task with Reminder	Reminder notification is sent at the specified time	Reminder is shown	Reminder notification is sent at the specified time	PASS
TC-NR-003	Receive overdue task notification	Let due date pass without completion	User must be logged in	1. Create a task with a due date 2. Let the due date pass without completing the task	Task with Overdue Date	Overdue task notification is sent	Overdue notification shown	Overdue task notification is sent	PASS
TC-NR-004	Snooze a reminder notification	Postpone a reminder	User must be logged in	1. Receive a reminder 2. Click "Snooze"	Reminder Notification	Reminder is postponed for a set period	Reminder is snoozed	Reminder is postponed for a set period	PASS

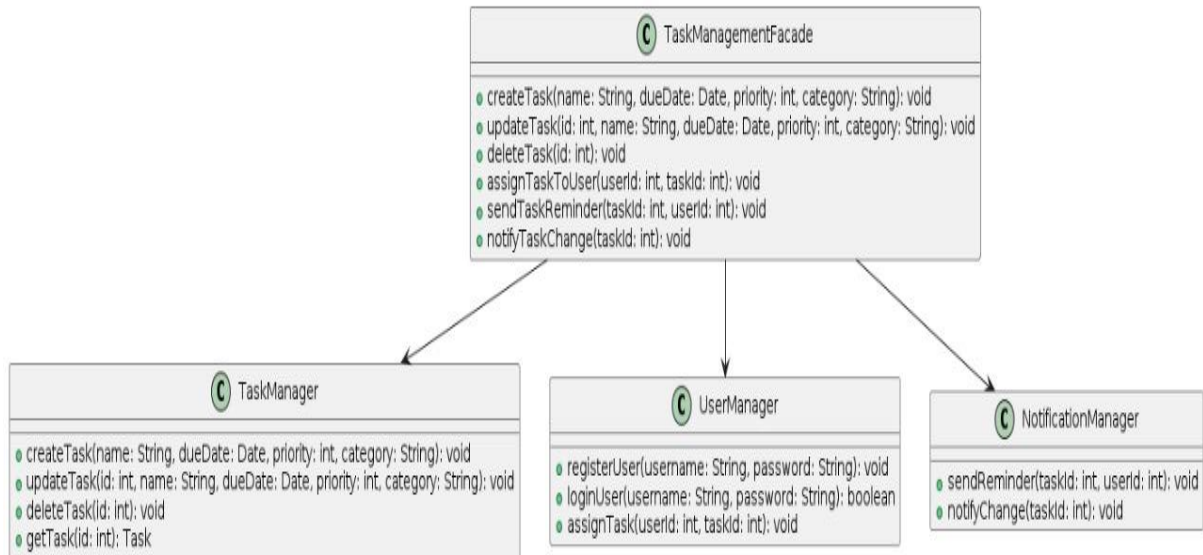
Experiment – 2: Using UML/JAVA, design Abstract Factory design pattern.



Experiment – 3: Using UML/JAVA, design Builder design pattern.**Task Management App Builder Design Pattern**

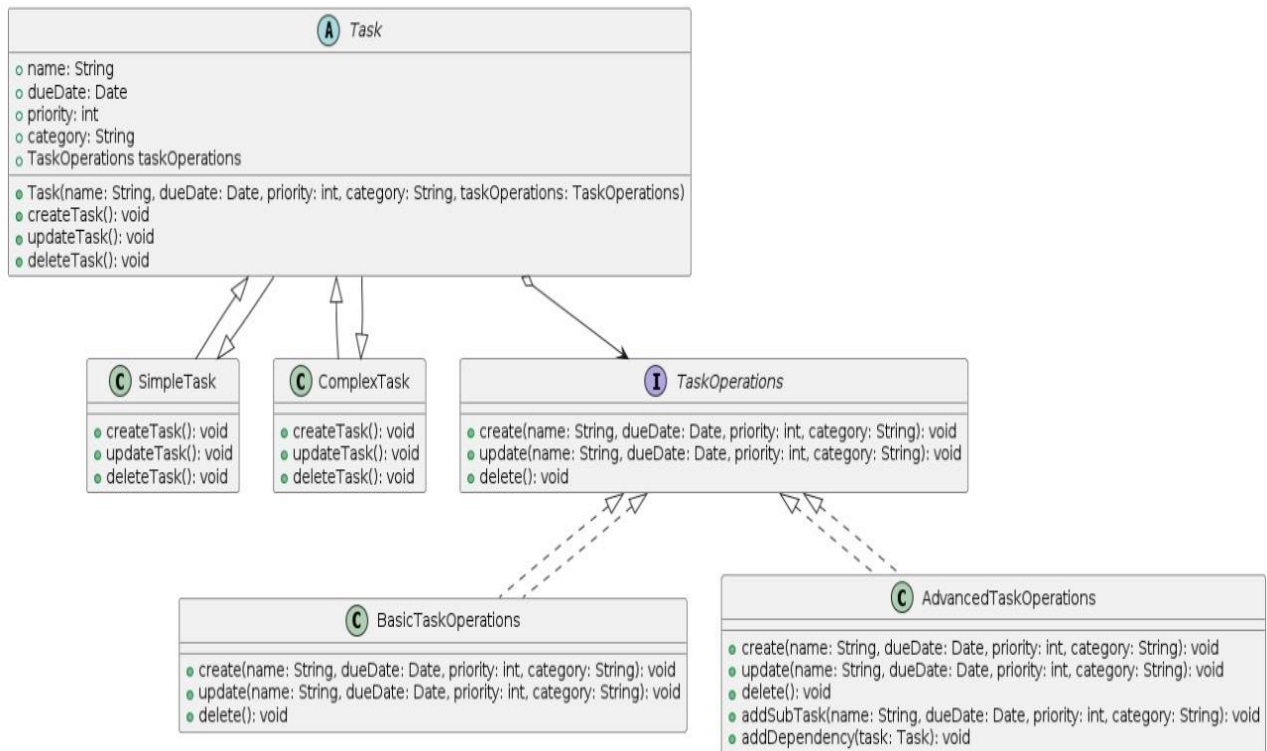
Experiment – 4: Using UML/JAVA, design facade design pattern.

Task Management System - Facade Design Pattern



Experiment – 5: Using UML/JAVA, design Bridge design pattern.

Task Management System - Bridge Design Pattern



Experiment – 6: Using UML/JAVA, design Decorator design pattern.**Task Management System - Decorator Design Pattern**