**Shapper-A task Managing Software**

[**Main functionalities of shapper** 1](#_Toc162137522)

[**Requirements** 2](#_Toc162137523)

[**Manage Tasks:** 2](#_Toc162137524)

[**Create Task** 2](#_Toc162137525)

[**View Task List** 2](#_Toc162137526)

[**Update Task** 2](#_Toc162137527)

[**Delete Task** 2](#_Toc162137528)

[**Set Task Details:** 3](#_Toc162137529)

[**Task Scheduling and Resource Allocation** 3](#_Toc162137530)

[**Create Project:** 3](#_Toc162137531)

[**Monitor Progress:** 3](#_Toc162137532)

[**Constraints** 3](#_Toc162137533)

[**Resource Availability** 3](#_Toc162137534)

[**Task Dependencies** 3](#_Toc162137535)

[**Time Constraints** 3](#_Toc162137536)

[**Priority Conflicts** 4](#_Toc162137537)

[**Use case Diagram** 4](#_Toc162137538)

[Use Case –Specification 4](#_Toc162137539)

[**Create Task** 4](#_Toc162137540)

[**View Task List** 5](#_Toc162137541)

[**Update Task** 5](#_Toc162137542)

[**Delete Task** 5](#_Toc162137543)

[**Set Task Details** 7](#_Toc162137544)

[**Create Project** 8](#_Toc162137545)

[**Monitor Progress** 8](#_Toc162137546)

[**Update Task Status** 9](#_Toc162137547)

## **Main functionalities of shapper**

**Manage Tasks**

* Create Task
* View Task List
* Update Task
* Delete Task

**Set Task Details**

* Set Priority
* Add Tags
* Attach Files

**Task Scheduling and Resource Allocation**

* Create Project
* Monitor Progress

## **Requirements**

### **Manage Tasks:**

#### **Create Task**

1. Take Task name, description, due date, priority level, and other optional details (e.g., tags, categories).
2. Validate input, create a new task with a unique identifier, and store it in the database.
3. Confirmation message indicating successful task creation.

#### **View Task List**

1. When user request to view tasks, Retrieve the user's tasks from the database.
2. Display a list of tasks with details like task name, due date, and priority level.

#### **Update Task**

1. User will select the task from the displayed tasks, details of that task should be shown on the screen
2. When the user select update option, The edit mode should open
3. User will update the details, like name , due date, priority
4. System will validate the updated data to maintain data integrity and save the task
5. A successful updating message should be shown after the process

#### **Delete Task**

1. User will select the task from the displayed tasks
2. user select delete option
3. Remove that task from the database.
4. Confirmation message indicating the task has been deleted.

#### **Set Task Details:**

1. User can set detail by add priority (can be in form of number, colors etc.), can add tags and attach a file or document (image, word, pdf, excel sheet etc.)
2. The system will save the updates and show a confirmation message

### **Task Scheduling and Resource Allocation**

#### **Create Project:**

1. Project Manager can creates a new project within the app.
2. Project manager can add project name, description, start date, end date, and assigned team members.
3. Project Manager can defines more details of the task such as, description, estimated duration, priority, dependencies and required resources
4. Project Manager can assign start and end dates to tasks
5. Project Manager assigns tasks to team members based on their roles, skills, and workload capacity.
6. Team members receive notifications or task assignments within the app.

#### **Monitor Progress:**

1. Team members update task progress regularly within the app, indicating completed, in progress, or delayed tasks.
2. Project Manager monitors overall project progress tracking.

### **Constraints**

**Resource Availability**

* Provide a dashboard showing each team member's workload, availability, and scheduled time off.
* Set up automatic alerts for over allocated resources or scheduling conflicts.
* Use an algorithm to balance task assignments based on workload and availability.

**Task Dependencies**

* Allow users to define and visualize dependencies between tasks with an interactive dependency graph or Gantt chart.
* When a preceding task is delayed, automatically adjust the schedules of dependent tasks and notify the relevant stakeholders.
* Provide alerts or notifications when a task that has dependencies is at risk of not meeting its deadline.

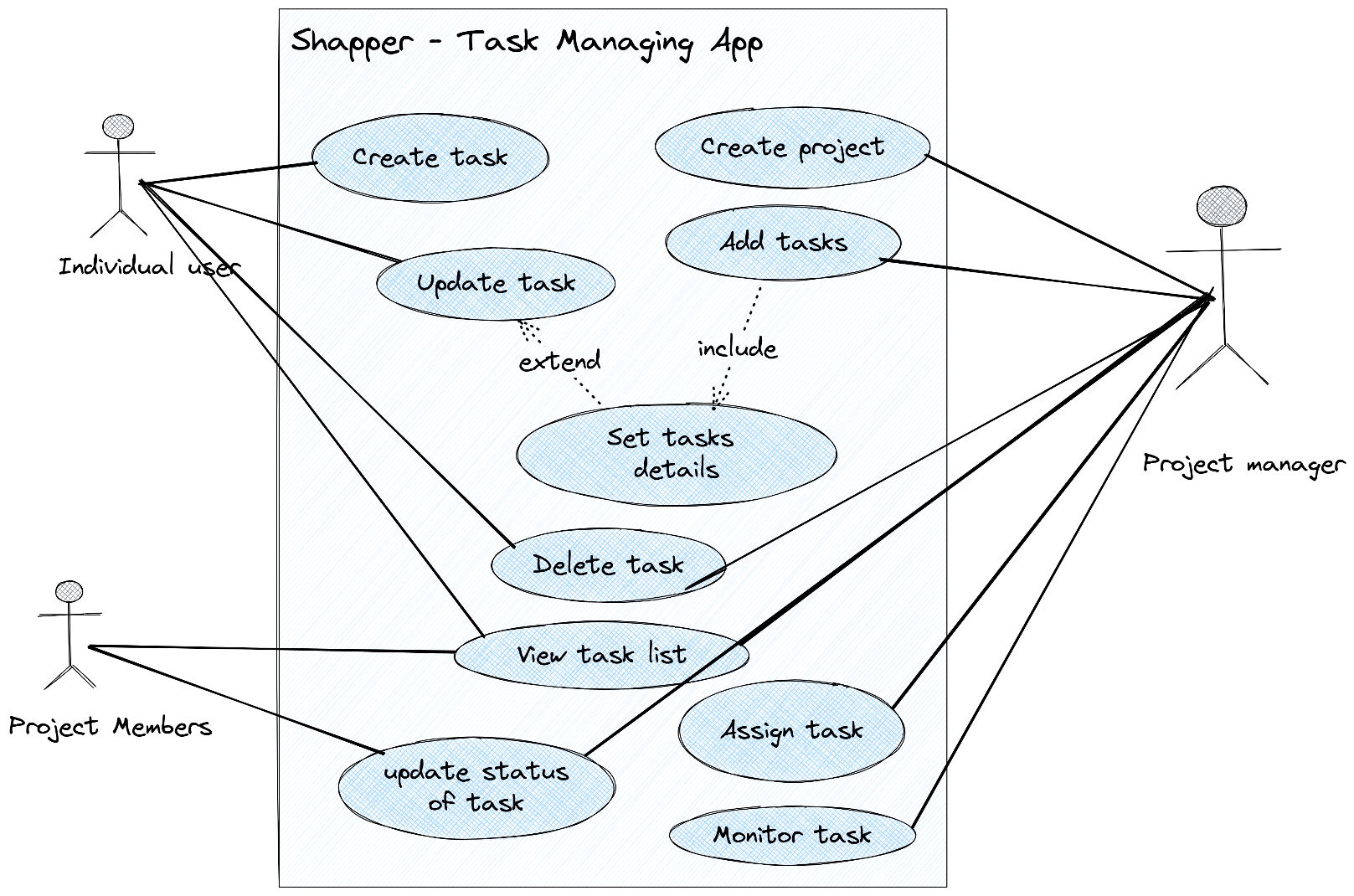
**Time Constraints**

* Send deadline alerts and reminders to keep team members informed.
* Use dynamic scheduling to adjust timelines based on progress updates and changing priorities.
* Track critical milestones and highlight risks to meeting them on time.

**Priority Conflicts**

* Implement clear task prioritization with visual cues like color codes or a matrix.
* Enable dynamic reprioritization with clear communication to stakeholders.
* Provide conflict resolution suggestions and allow quick adjustments to priorities and resource allocation.

## **Use case Diagram**



## **Use Case –Specification**

### **Create Task**

|  |  |
| --- | --- |
| Use case name: | Create Task |
| Scenario: | User Want to add a task |
| Brief Description: | Allows the user to create a new task with specific details. |
| Actor: | User, System |
| Related Use Case: | No use case related |
| Pre-conditions: | User must be logged in with necessary permissions. |
| Post conditions: | A new task is created and stored in the system. |
| Flows Activities: | 1. User selects the option to create a new task. 2. User inputs task details (name, description, due date, priority). 3. System validates and saves the task. 4. User receives a confirmation message. |
| Exception condition: | If input validation fails, the system displays an error and prompts the user to correct the details |
| Alternative Flows | Not applicable. |

### **View Task List**

|  |  |
| --- | --- |
| Use case name: | View Task List |
| Scenario: | User views a list of their tasks. |
| Brief Description: | Users can view all their tasks listed with relevant details. |
| Actor: | User, Project manager, project members, System |
| Related Use Case: | No use case related |
| Pre-conditions: | User is logged in. |
| Post conditions: | User views a list of their tasks. |
| Flows Activities: | 1. User requests to view their task list. 2. System retrieves and displays the tasks. |
| Exception condition: | None |
| Alternative Flows | If no tasks exist, the system informs the user. |

### **Update Task**

|  |  |
| --- | --- |
| Use case name: | Update Task |
| Scenario: | User updates an existing task. |
| Brief Description: | User selects a task and modifies its details. |
| Actor: | User, Project Manager, System |
| Related Use Case: | Set Task Details |
| Pre-conditions: | User has selected a task they have permission to edit. |
| Post conditions: | The task is updated in the system. |
| Flows Activities: | 1. User selects a task and chooses to update it. 2. User modifies the task's details. 3. System validates and saves the changes. 4. User receives a confirmation message. |
| Exception condition: | None |
| Alternative Flows | If validation fails, prompt user to correct the input. |

### **Delete Task**

|  |  |
| --- | --- |
| Use case name: | Delete Task |
| Scenario: | User deletes an existing task. |
| Brief Description: | User selects a task and deletes it from the system. |
| Actor: | User, Project Manager, System |
| Related Use Case: | None |
| Pre-conditions: | User has selected a task they have permission to delete. |
| Post conditions: | he task is removed from the system. |
| Flows Activities: | 1. User selects a task and opts to delete it. 2. System confirms the action and deletes the task. 3. User receives a confirmation message. |
| Exception condition: | If the task cannot be deleted due to dependencies, inform the user. |
| Alternative Flows | Not applicable. |

### **Set Task Details**

|  |  |
| --- | --- |
| Name: | Set task details |
| Scenario: | User adds or updates details for a task. |
| Brief Description: | User can add or modify priority, tags, and attachments for a task. |
| Actor: | User, Project Manager, System |
| Related Use Case: | Update Task, Add Task |
| Pre-conditions: | User has selected a task to modify. |
| Post conditions: | The task details are updated in the system. |
| Flows Activities: | 1. User selects a task and chooses to add or modify details. 2. User adds/updates priority, tags, and attachments. 3. System saves the changes and confirms to the user. |
| Exception condition: | If saving fails, the user is informed. |
| Alternative Flows | Not applicable. |

### **Create Project**

|  |  |
| --- | --- |
| Use case name: | Create Project |
| Scenario: | Project Manager creates a new project. |
| Brief Description: | Project Manager inputs details to create a new project and assigns tasks. |
| Actor: | Project Manager, System |
| Related Use Case: | No use case related |
| Pre-conditions: | User identified as Project Manager is logged in. |
| Post conditions: | A new project is created and stored in the system. |
| Flows Activities: | 1. Project Manager inputs project details. 2. Project Manager adds tasks and assigns team members. 3. System validates and saves the project. 4. Team members are notified of their assignments. |
| Exception condition: | None. |
| Alternative Flows | If validation fails, prompt to correct input. |

### **Monitor Progress**

|  |  |
| --- | --- |
| Use case name: | Monitor Progress |
| Scenario: | Project Manager monitors the progress of tasks within a project. |
| Brief Description: | This use case allows the Project Manager to check the status and progress of various tasks and overall project advancement. |
| Actor: | Project Manager, System |
| Related Use Case: | No use case related |
| Pre-conditions: | The Project Manager is logged in and has selected a project to monitor. |
| Post conditions: | The Project Manager is updated with the latest progress information on the project and its tasks. |
| Flows Activities: | 1. Project Manager selects a project to monitor. 2. System displays the current progress of tasks and the overall project. 3. Project Manager reviews the progress information. 4. If necessary, the Project Manager takes action based on the progress (e.g., reallocating resources, adjusting timelines). |
| Exception condition: | If the system fails to retrieve or display progress information, it should notify the Project Manager of the error and suggest trying again later. |
| Alternative Flows | If there are no updates or changes in the project progress, the system informs the Project Manager accordingly. |

### **Update Task Status**

|  |  |
| --- | --- |
| Use case name: | Update Task Status |
| Scenario: | A team member updates the status of their assigned task. |
| Brief Description: | This use case allows a team member to change the status of a task they are assigned to, reflecting their progress on the task. |
| Actor: | Project Members, System |
| Related Use Case: | No use case related |
| Pre-conditions: | * The team member is logged in. * The team member has been assigned at least one task. |
| Post conditions: | * The status of the task is updated in the system. * Relevant stakeholders (e.g., Project Manager) are notified of the status change. |
| Flows Activities: | 1. The team member logs into the system and navigates to their list of assigned tasks. 2. The team member selects a task to update. 3. The team member changes the status of the task (e.g., from "In Progress" to "Completed"). 4. The system validates the update and saves the new status. 5. The system confirms the status update to the team member. 6. Notifications are sent to relevant stakeholders about the status change. |
| Exception condition: | If the system encounters an error while saving the task status (e.g., due to a network issue or database error), it should inform the team member of the error and suggest they try the update again later.. |
| Alternative Flows | If the team member selects a task that they are not permitted to update (e.g., a task not assigned to them), the system should display an appropriate message and disallow the status update. |