---------------------- Project Requirement -----------------------

For a decision matrix project aimed at categorizing tasks based on priority, we need to define a set of detailed requirements. These requirements will cover the functionality of the decision matrix, user interaction, data handling, and any specific features that are needed for prioritizing tasks. Here's an outline of the project requirements:

### 1. Project Overview

- \*\*Objective\*\*: To develop a decision matrix application that helps users categorize tasks based on urgency and importance, allowing them to decide whether a task needs immediate attention, can be scheduled for later, should be delegated, or can be deleted.

- \*\*Target Audience\*\*: Individuals or teams needing to prioritize tasks to enhance productivity and time management.

### 2. Functional Requirements

- \*\*Task Input\*\*:

- The system must allow users to input tasks with descriptions.

- Users should be able to assign deadlines and estimated completion times to tasks.

- \*\*Task Categorization\*\*:

- Implement an algorithm that categorizes tasks into four categories based on user input:

1. Complete Now

2. Schedule for Later

3. Delegate

4. Delete

- The algorithm should consider task urgency, importance, and deadlines in categorization.

- \*\*User Interface\*\*:

- Provide a simple, intuitive interface for entering and viewing tasks.

- Tasks should be visually differentiated based on their categories (e.g., color coding).

- Support for filtering and sorting tasks based on category, deadline, and other criteria.

- \*\*Notifications and Reminders\*\*:

- The system should notify users of upcoming deadlines for tasks categorized under "Complete Now" and "Schedule for Later."

- Optional reminder setups for delegated tasks to follow up.

- \*\*Delegation Features\*\*:

- Allow tasks to be marked as delegated, with the option to input delegatee contact information.

- Option to send automated or manual reminders to the delegatee.

- \*\*Data Storage\*\*:

- Secure storage of tasks and user preferences.

- Cloud synchronization for accessibility across multiple devices (optional requirement based on scope).

### 3. Non-Functional Requirements

- \*\*Performance\*\*: The application should load and respond to user inputs quickly, with task categorization processing within seconds.

- \*\*Usability\*\*: The interface must be user-friendly, allowing users to navigate and perform tasks without extensive training.

- \*\*Scalability\*\*: The system should be scalable, capable of handling an increasing number of tasks without degradation in performance.

- \*\*Security\*\*: User data must be encrypted and protected against unauthorized access.

### 4. Technology Stack (Optional)

- Frontend: [Technologies suited for web/mobile app development, based on platform choice]

- Backend: [Server-side technology for logic processing and data handling]

- Database: [Suitable database technology for storing user data and task information]

### 5. Compliance and Standards

- Ensure compliance with data protection regulations applicable in the user's region (e.g., GDPR, CCPA).

- Follow best practices in software development and UI/UX design.

This outline forms a solid base for developing a decision matrix application. Depending on the project scope, additional features like team collaboration, integration with external calendars, or AI-based task prioritization might also be considered.

----------------- User stories and journeys ------------------------

Based on the project requirements for the decision matrix project aimed at helping users prioritize tasks, let's define some user stories and journeys that outline how different users would interact with the application. These stories and journeys will focus on the main functionalities such as task input, categorization, user interface, notifications, reminders, delegation, and data handling, considering the non-functional requirements for performance, usability, scalability, and security.

### User Stories

1. \*\*As a user, I want to add a new task with a description and a deadline so that I can keep track of my work.\*\*

- \*\*Acceptance Criteria\*\*:

- The system allows the input of task names and descriptions.

- Users can assign deadlines and estimated completion times.

2. \*\*As a user, I want my tasks to be automatically categorized based on urgency and importance so I can focus on what matters most.\*\*

- \*\*Acceptance Criteria\*\*:

- The application implements an algorithm that categorizes tasks into "Complete Now", "Schedule for Later", "Delegate", and "Delete".

- The categorization takes into account the urgency, importance, and deadlines.

3. \*\*As a user, I want to be able to easily enter and view my tasks through a simple and intuitive interface.\*\*

- \*\*Acceptance Criteria\*\*:

- The application provides an easy-to-use interface for entering and viewing tasks.

- Tasks are visually differentiated by category, such as through color coding.

4. \*\*As a user, I want to receive notifications about upcoming deadlines so that I don’t miss any important tasks. \*\*

- \*\*Acceptance Criteria\*\*:

- The system notifies users of upcoming deadlines for "Complete Now" and "Schedule for Later" tasks.

- Users can set up optional reminders for delegated tasks.

5. \*\*As a user, I want to mark tasks as delegated and specify delegate information, including the option to send them reminders. \*\*

- \*\*Acceptance Criteria\*\*:

- Users can mark tasks as delegated and enter delegate contact information.

- The system provides options for sending automated or manual reminders to delegatees.

6. \*\*As a user, I want my tasks and preferences to be securely stored and accessible across multiple devices.\*\*

- \*\*Acceptance Criteria\*\*:

- The application securely stores user data and preferences.

- Optional cloud synchronization for multi-device accessibility.

### User Journeys

1. \*\*Jane's First Use Experience:\*\*

- Jane opens the application for the first time and is guided through a simple tutorial on how to add her first task. She enters a task, "Prepare presentation for Monday's meeting", sets the deadline for Sunday, and estimates it will take 3 hours. The system categorizes it as "Complete Now" due to its urgency and importance. Jane appreciates the clear, color-coded categorization and feels more organized already.

2. \*\*Mike's Delegation Process:\*\*

- Mike has a task, "Compile quarterly sales report", that he decides to delegate. Within the app, he marks the task as delegated, inputs his assistant's email, and chooses to send an automatic reminder a week before the deadline. The app updates him when the task is approaching its deadline, and he can check if it's been completed without having to micromanage.

3. \*\*Alex's Daily Review:\*\*

- Every morning, Alex opens the application to review his tasks for the day. He filters tasks by "Complete Now" to focus on urgent and important tasks. Midday, he receives a notification reminding him of a task's deadline in 2 hours, which he had forgotten. Thanks to the reminder, he completes it on time.

Each user story and journey reflects a specific functionality and how it benefits the user, aligned with the project requirements for functionality, usability, and performance. This approach ensures the application is developed with real user needs in mind, focusing on enhancing productivity and efficient task management.

### **Quadrants Definition**

* **Complete Now (Do it):** Tasks that are urgent and important. Immediate action is required.
* **Schedule for Later (Plan it):** Tasks that are important but not urgent. These tasks require planning and scheduling to complete before they become urgent.
* **Delegate (Delegate it):** Tasks that are urgent but not important. These tasks should be done soon but can be completed by someone else.
* **Delete (Delete it):** Tasks that are neither urgent nor important. These are low-priority tasks that can be eliminated.

### **Considering Deadlines in Categorization**

To incorporate deadlines into this matrix, the algorithm can adjust a task's urgency based on its deadline proximity. Here's how tasks can be evaluated and categorized:

* **Evaluate Urgency and Importance:**
  + Urgency is determined by how soon a task needs to be completed. Tasks with impending deadlines are considered more urgent.
  + Importance is determined by the potential impact of completing (or not completing) the task. Tasks with higher stakes or contributions to goals are considered more important.
* **Adjust Urgency Based on Deadlines:**
  + A task with a distant deadline might not be urgent now but could become urgent as the deadline approaches. The algorithm can periodically re-evaluate tasks to adjust their urgency based on changing deadlines.
* **Algorithm Flow:**
  + **Input:** Gather tasks with their urgency, importance, and deadlines.
  + **Process:** For each task, assess its urgency and importance. Adjust the urgency based on the deadline proximity.
  + **Categorize:**
    - If urgent and important, categorize as **Complete Now**.
    - If not urgent but important, check the deadline to decide if it should be **Scheduled for Later**.
    - If urgent but not important, and can be handled by someone else, **Delegate**.
    - If neither urgent nor important, **Delete**.
* **Re-evaluation Mechanism:**
  + Implement a mechanism to re-evaluate the urgency of tasks scheduled for later as their deadlines approach, possibly moving them to the "Complete Now" category if they become urgent.