A task management engine with the ability for users to view, approve, or reject tasks typically requires a combination of backend and frontend components. Here's a general outline of the specifications for such a system:

1. User Authentication and Authorization:
   * User registration and login functionality to ensure secure access.
   * Role-based access control to define user permissions (e.g., admin, manager, employee).
2. Task Creation and Assignment:
   * Ability for authorized users to create new tasks.
   * Assign tasks to specific users or groups.
   * Include relevant details such as task name, description, due date, priority, and attachments.
3. Task Dashboard:
   * A user-friendly interface where users can view their assigned tasks.
   * Filter and sort options to organize tasks based on various criteria (e.g., due date, priority).
   * Search functionality to quickly find specific tasks.
4. Task Status and Updates:
   * Display the current status of each task (e.g., pending, in progress, completed).
   * Allow users to update the status of their assigned tasks (e.g., mark as completed, progress percentage).
   * Provide notifications or alerts for task updates.
5. Task Comments and Collaboration:
   * Enable users to add comments or notes to tasks for collaboration and communication.
   * Support attachments and file sharing related to tasks.
   * Notify users of new comments or updates.

A Users and Permissions Management System is responsible for managing user accounts, roles, and permissions within an application or system. Here are specifications for building such a system:

1. User Registration and Authentication:
   * Enable user registration with necessary information (e.g., username, email, password).
   * Implement authentication mechanisms such as username/password
   * Ensure password security measures, including hashing and encryption.
2. User Roles and Permissions:
   * Define different user roles based on their responsibilities and access levels (e.g., admin, manager, user).
   * Assign role-based permissions to control user access to specific features, functions, or data within the system.
   * Allow customization of roles and permissions to match the specific needs of the application.
3. User Profile Management:
   * Provide functionality for users to manage their profile information (e.g., name, contact details, profile picture).
   * Allow users to update their personal settings and preferences.
4. User Search and Filtering:
   * Allow administrators to search for users based on criteria such as username, email, or role.
   * Implement filters and sorting options for efficient user management.
5. User Activity Logging and Auditing:
   * Log user activities, such as login attempts, role changes, or permission modifications.
   * Provide audit trails for user actions to track and investigate any suspicious or unauthorized activities.
6. Password Management:
   * Allow users to reset their passwords in case of forgotten or compromised credentials.