

Use Case Diagram

The use case diagram illustrates the interactions between the actors (User and Admin) and the system functionalities of the Task Manager application.

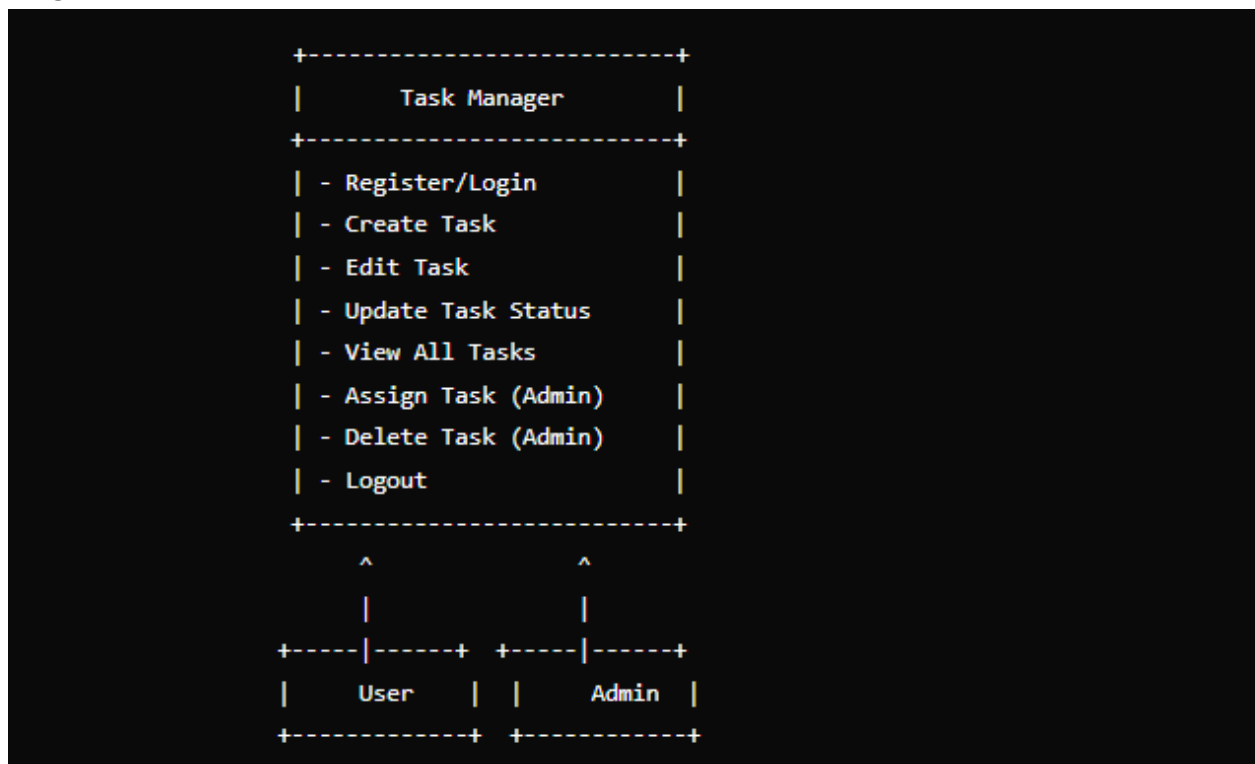
****Actors:****

- User
- Admin

****Use Cases:****

- Register/Login
- Create Task
- Edit Task
- Update Task Status
- View All Tasks
- Assign Task (Admin)
- Delete Task (Admin)
- Logout

Diagram:



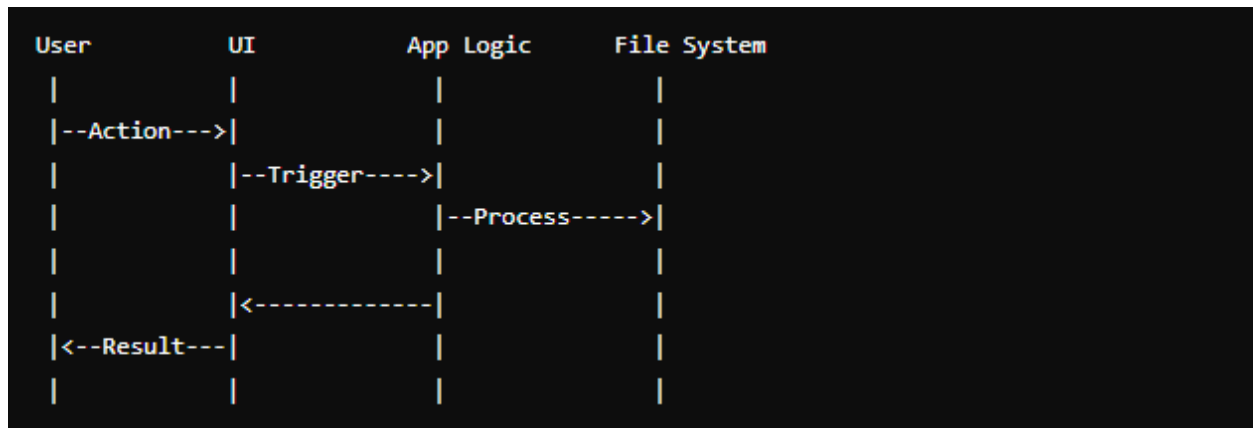
Sequence Diagram

The sequence diagram shows the interaction between the user, UI, application logic, and file system.

****Actors:****

- User
- Admin

Diagram:



Component Concerns Outline

Application Components

- TaskManagerApplication (application.py)**
 - Manages user sessions and main menu display.
 - Handles navigation between different functionalities based on user input.
- Account Management (create_account.py, login.py)**
 - create_account.py:
 - Allows users to create employee or admin accounts.
 - Checks for existing email registrations.
 - Writes new user data to appropriate files.
 - login.py:
 - Manages user login for both employees and admins.
 - Validates credentials against stored data.
 - Loads user information from files.
- Task Management (tasks_logic.py)**
 - Manages creation, editing, deletion, and status updates of tasks.
 - Handles task assignment to employees (admin only).

- Interacts with file system for reading and writing task data.
4. **Objects (objects.py)**
- Defines **Task**, **Employee**, and **AdminUser** classes.
 - Manages task and user attributes and behaviors.

Class Diagram

The class diagram represents the structure of the Task Manager application with visibility modifiers for methods and relationships between classes.

Classes:

- User
- Task
- TaskManager

Diagram:

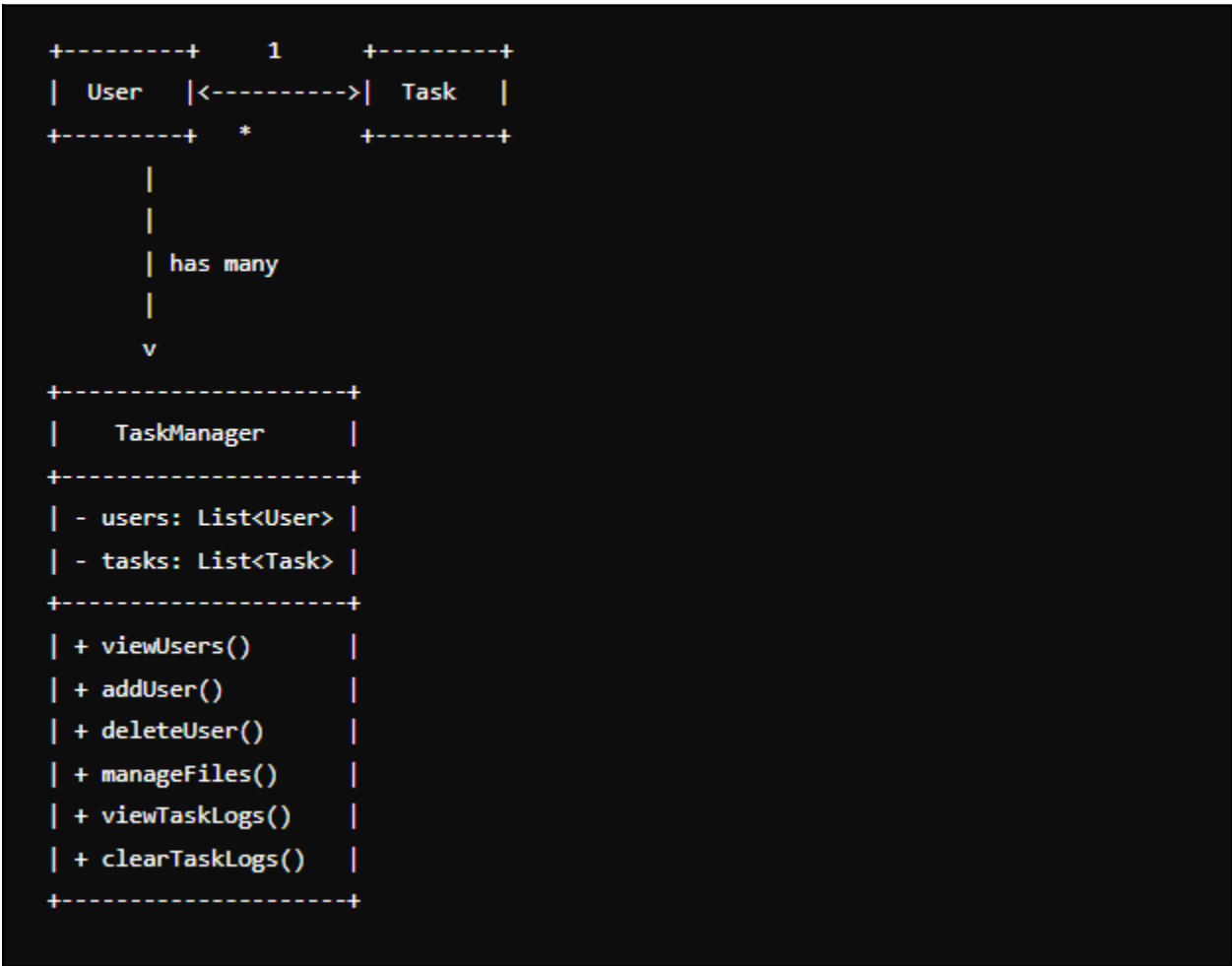
```
+-----+
|      User      |
+-----+
| - first_name: str |
| - last_name: str  |
| - email: str      |
| - password: str   |
| - tasks: list[Task] |
+-----+
| + __init__()      |
| + __str__()       |
| + register()      |
| + login()         |
| + viewTasks()     |
| + addTask()       |
| + editTask()      |
| + deleteTask()    |
| + setPriority()    |
| + setDueDate()    |
| + searchTasks()   |
| + filterTasks()   |
+-----+
```

```
+-----+
|      Task      |
+-----+
| - task_name: str |
| - task_description: str |
| - task_is_complete: bool |
+-----+
| + __init__()     |
| + __str__()      |
| + create()       |
| + read()         |
| + update()       |
| + delete()       |
+-----+
```

```
+-----+
|   TaskManager   |
+-----+
| - users: list[User] |
| - tasks: list[Task] |
| - current_user: User|
| - all_tasks_file: str|
+-----+
| + __init__()      |
| + display_menu()  |
| + viewUsers()     |
| + addUser()       |
| + deleteUser()    |
| + manageFiles()   |
| + viewTaskLogs()  |
| + clearTaskLogs() |
+-----+
```

Class Relationships

Diagram:



CRUD Matrix:

Feature	Create	Read	Update	Delete
Tasks	Yes	Yes	Yes	Yes
Employees (Admin only)	Yes	Yes	No	No
Admin Users	Yes	Yes	No	No
Task Assignment	Yes	Yes	Yes	No

This table summarizes the Create, Read, Update, and Delete functionalities available for tasks, employees, and admin users within the Task Manager application.