

Task List Management System

Github Repository URL

Group Members:

Jacky Cui (jc10689)

Henry Xu (cx2119)

Warren Wu (yw5928)

05/09/2025

Table of Work

(Please write x in the boxes to mention what each student achieved in this project)

	Jacky Cui	Henry Xu	Warren Wu
Project Description		x	x
Uses Cases Diagram(s)		x	
Sequence Diagrams		x	
Class diagram(s)		x	
Implementation	x		
Conclusion		x	x

Table of Contents

- Terminology Glossary (If needed)
- System Analysis
 - Project Description (One Page)
 - General Description, Goals and Benefits
 - System input(s) and output(s)
 - Special requirements (Performance, Interfaces, Constraints, Reliability, if any)
 - Uses Cases Diagram(s) and use cases description.
- System Design
 - Class diagram(s)
 - Sequence Diagrams
- Conclusion
- Appendix (any related reports, questionnaires, docs.. If any).

Project Description

General Description

The Task-List Management System is a self-contained, command-line Java application that lets individual users create, update and search personal tasks while giving an administrator full oversight (view all users, delete users or tasks, reset data). All information is stored in a single local data file, so the program runs out-of-the-box on any machine.

The URL of Github repository for this project is:

<https://github.com/W0rren12/NYU-CSCI-UA-470-Task-Manager>, which is under one of our group members' Github account.

Goals & Benefits

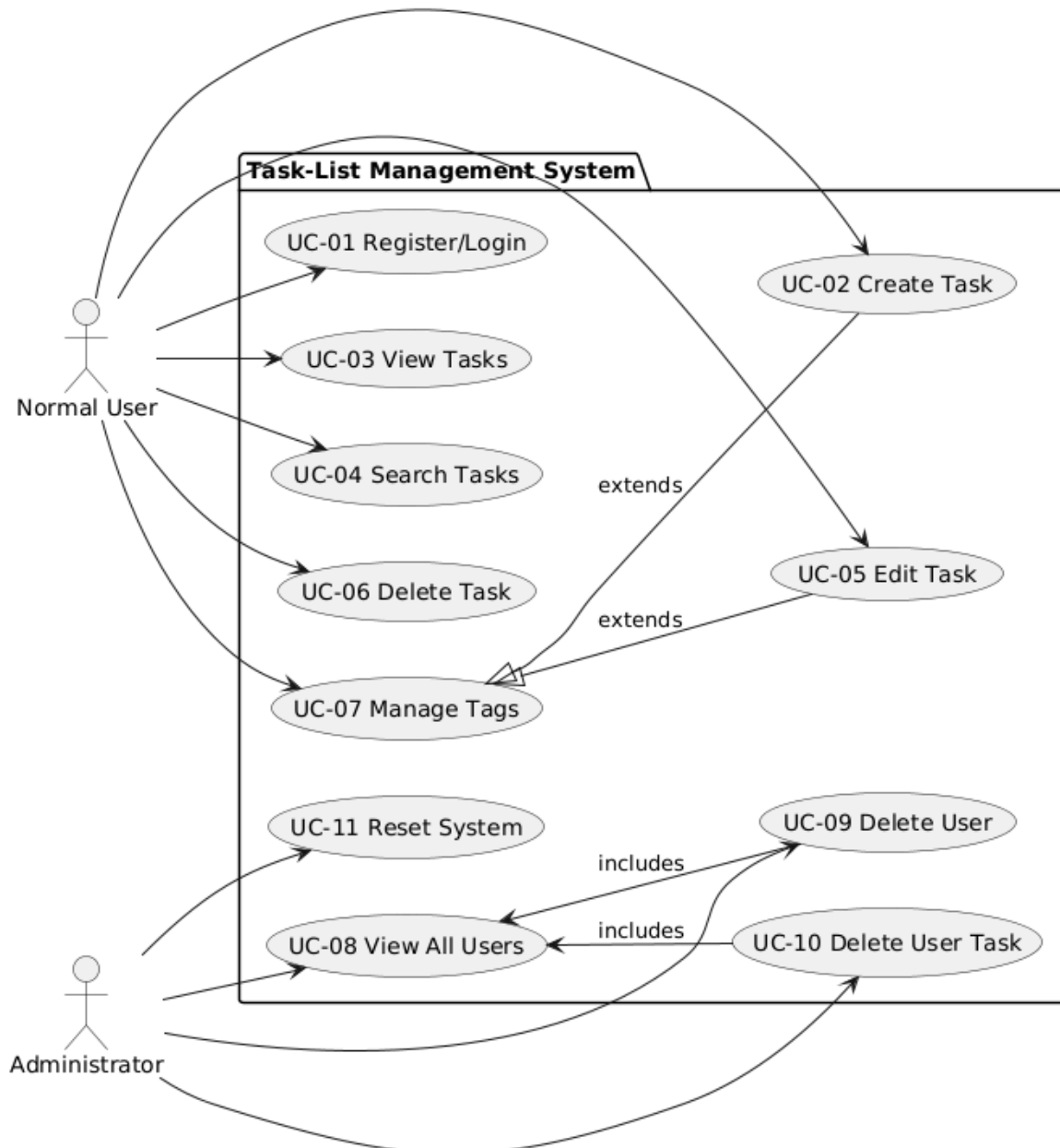
Goal	Benefit
Simplify personal task tracking	Users capture tasks with title, description, priority, due date, status and tags, and therefore reducing missed deadlines.
Provide administrative control	An admin can audit or purge data in one place, making it useful for demos, grading, or multi-user environments.
Ensure data persistence with zero setup	Everything is serialised to data.ser; users can close the program and pick up exactly where they left off.

System Inputs & Outputs

Actor	Typical Inputs (keyboard)	Key Outputs (console)
Normal User	Menu choices; task fields (title, desc, priority, due date, status, tags); search keywords or filters.	Confirmation messages ("Task created"), formatted task lists, filtered search results, error prompts for invalid input.
Administrator	Menu choices to list users, delete a user, delete a user's task, or reset system.	User lists, task lists, success / failure messages ("User deleted"), confirmation prompt before reset, error prompt for invalid input.
Application (startup / exit)	Reads/writes the data.ser file automatically since the	Silent load on launch; silent save after each data-changing operation.

	execution of the program—no user action required.	
--	---	--

Use Case Diagram

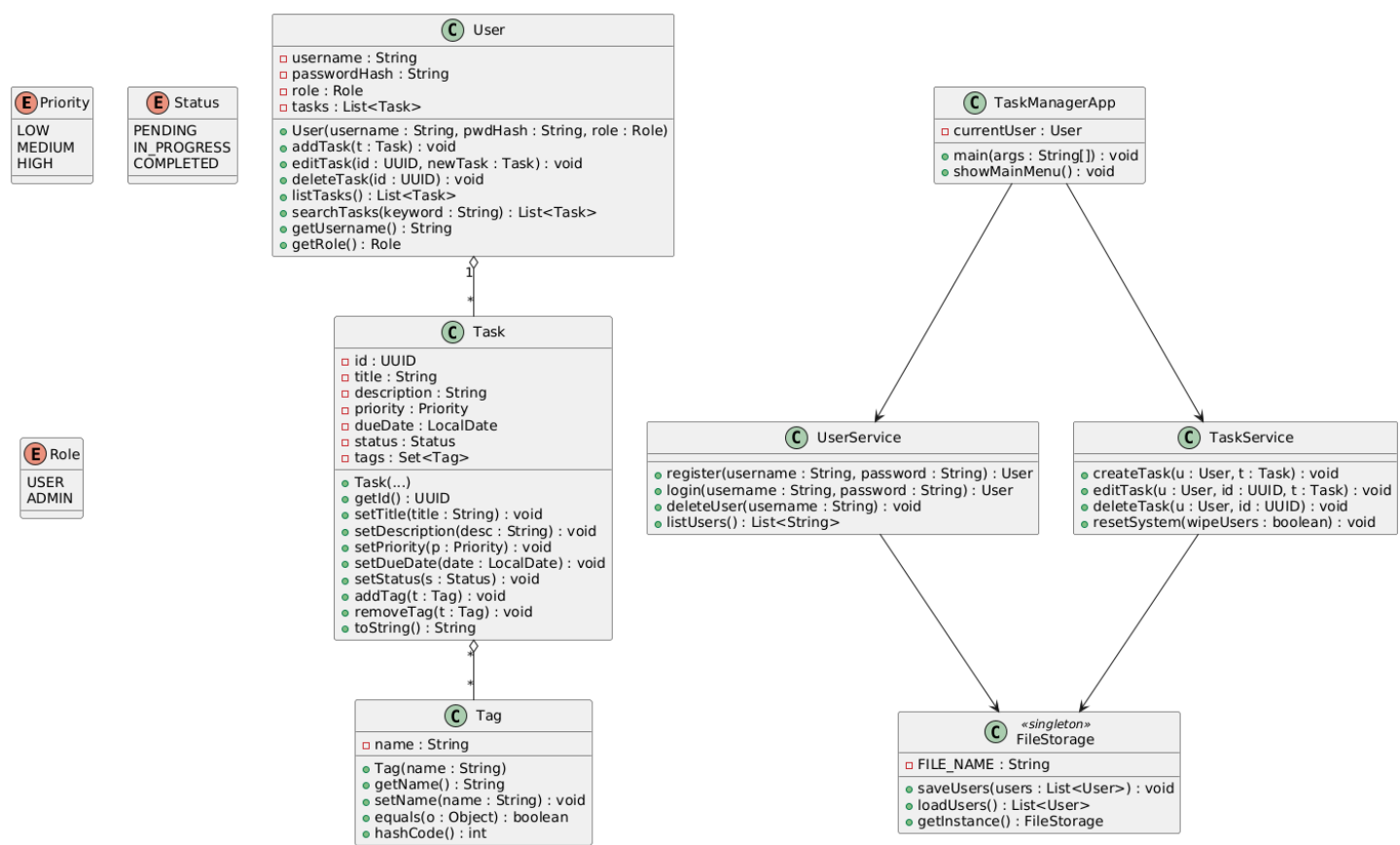


Use Case Reference Sheet

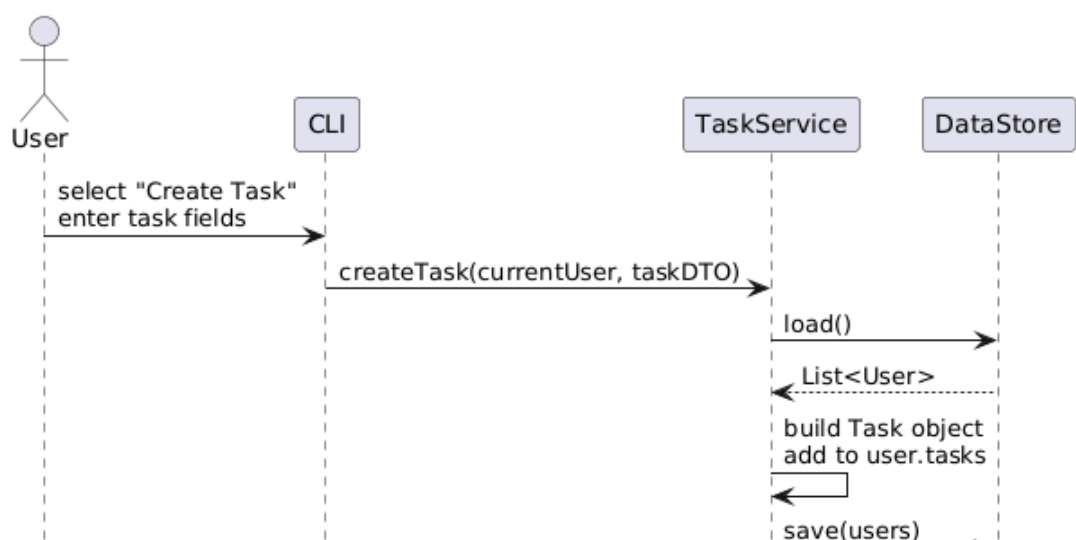
UC Ref	Use-Case Name	Overview	Related UCs	Actors
UC-01	Register / Login	User authenticates or creates account	–	Normal User, Administrator
UC-02	Create Task	Add a task (title, desc, priority, due, status, tags)	UC-07 (extends)	Normal User
UC-03	View Tasks	List all tasks owned by the user	–	Normal User
UC-04	Search Tasks	Filter tasks by keyword, tag, priority, status	–	Normal User
UC-05	Edit Task	Modify any task field	UC-07 (extends)	Normal User
UC-06	Delete Task	Remove an existing task	–	Normal User
UC-07	Manage Tags	Create / rename / delete tags, assign to tasks	–	Normal User
UC-08	View All Users	Show every username in the system	–	Administrator
UC-09	Delete User	Remove a user and (optionally) tasks	UC-08 (includes)	Administrator

UC-10	Delete User Task	Remove a specific task owned by a user	UC-08 (includes)	Administrator
UC-11	Reset System	Wipe all tasks (optionally users)	–	Administrator

Full Class Diagram



Sequence Diagram



Conclusion

The system delivers its two promises: users efficiently manage personal tasks, and admins maintain a good system by managing all the users' information. All the data is extracted, edited and stored in a single local file during execution of the program. Clear OOP structure and upfront UML kept the code tidy, logical and future-proof. The project is fully functional and ready to be upgraded to a GUI or be deployed to a Digital Ocean in the future.