North South University

Department of Electrical and Computer Engineering

CSE327 - Software Engineering

Quiz 1

Total Marks - 30

Name:	Mohammed Rakib
Student ID:	1731176042
Section:	01
Date:	12th August 2020

Sl.	CO Description	Question#
CO1	**Identify** the requirements of a software system, including technical-functional requirements, non-technical requirements, and wider societal impact.	**1,2**
CO2	**Design** an object oriented software architecture and **express** the architecture using UML or other standard tools under a set of requirements and/or constraints,.	**3**

Instructions: Please read the Case Study carefully. There are three questions. You are required to answer all three questions in order.

Case Study

A company wants to automate its task allocation process. Currently, employees are assigned task either verbally or via email, and it is very hard to keep track of who is doing what, and how much is done. The company wants to move to a web-based task allocation system to do the following.

Anyone can create a task. Each task gets a corresponding ticket number. One user is assigned as “responsible” for the task, and others can be assigned as “helpers”. Admins create and delete user accounts (individually, or in bulk via a CSV file).

When a task is created, by default the creator is “responsible” for it unless it is changed. Admins can also change the responsible person. The creator or the current responsible person can also change the assignment. The responsible person can add/remove helpers.

Responsible persons and helpers can post against a task. Once a task is complete, its status has to be changed to complete. Tasks can be in different statuses: Not initiated, initiated, progressing, stalled, waiting for resources, completed, unresolved being some of them.

When creating a task, the creator has to give a task title and a short description of no less than 50 words. He/She can assign the responsible person and/or helpers. He/she can also attach watchers to the task. All stakeholders, the creator, responsible person, helpers, and watchers should be notified with every post on the task.

A task can be dependent on other tasks, meaning they will never achieve the complete status, unless the other tasks are completed. Tasks can also have sub-tasks (not the same as a dependency), which can be created and assigned to other responsible people, but the responsible person on the main task will automatically become a watcher on the sub-tasks, as well as the tasks on which there is a dependency.

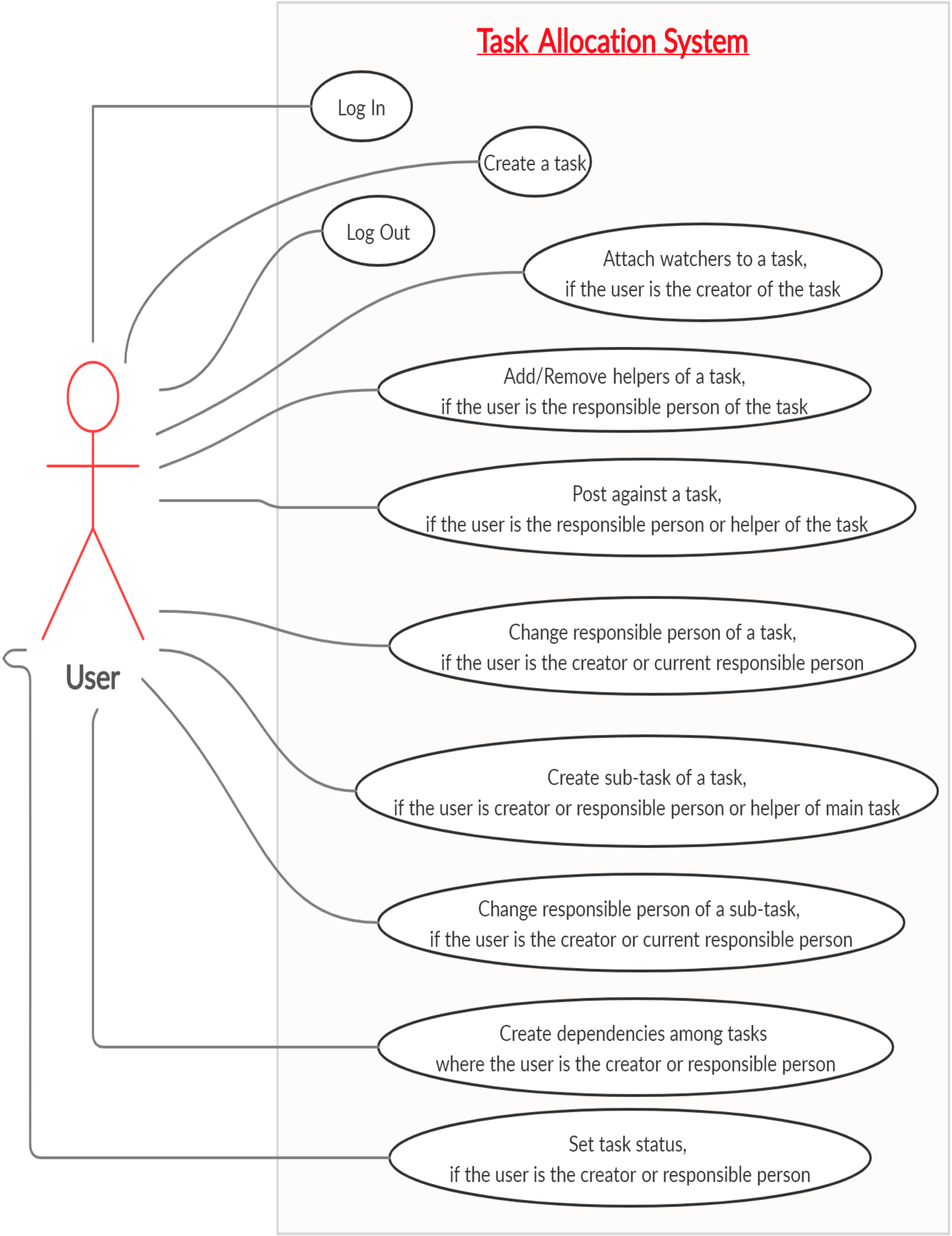
Task posts are mainly text, but can also be attachments. Special processing should be done on certain types of attachments. These are:

- Zip files and binary files should be scanned for viruses.
- Images should be limited to a maximum resolution of 1024 x 800. If necessary, the system should be resized on the fly.
- URLs should be stored as it is, but when displaying the URL, the title of the target HTML content should be rendered as well.
- Video files should be displayed in two parts:
    - a download link.
    - and a video player embedded in the view so that the viewer can play the video in the system itself.
- GIT URLs should be linked to the GIT repository as well as a small floating test showing how the repository can be cloned.
- When viewing PDF files should have a download link and an icon. Clicking on the icon should render the file on the system itself instead of downloading it.

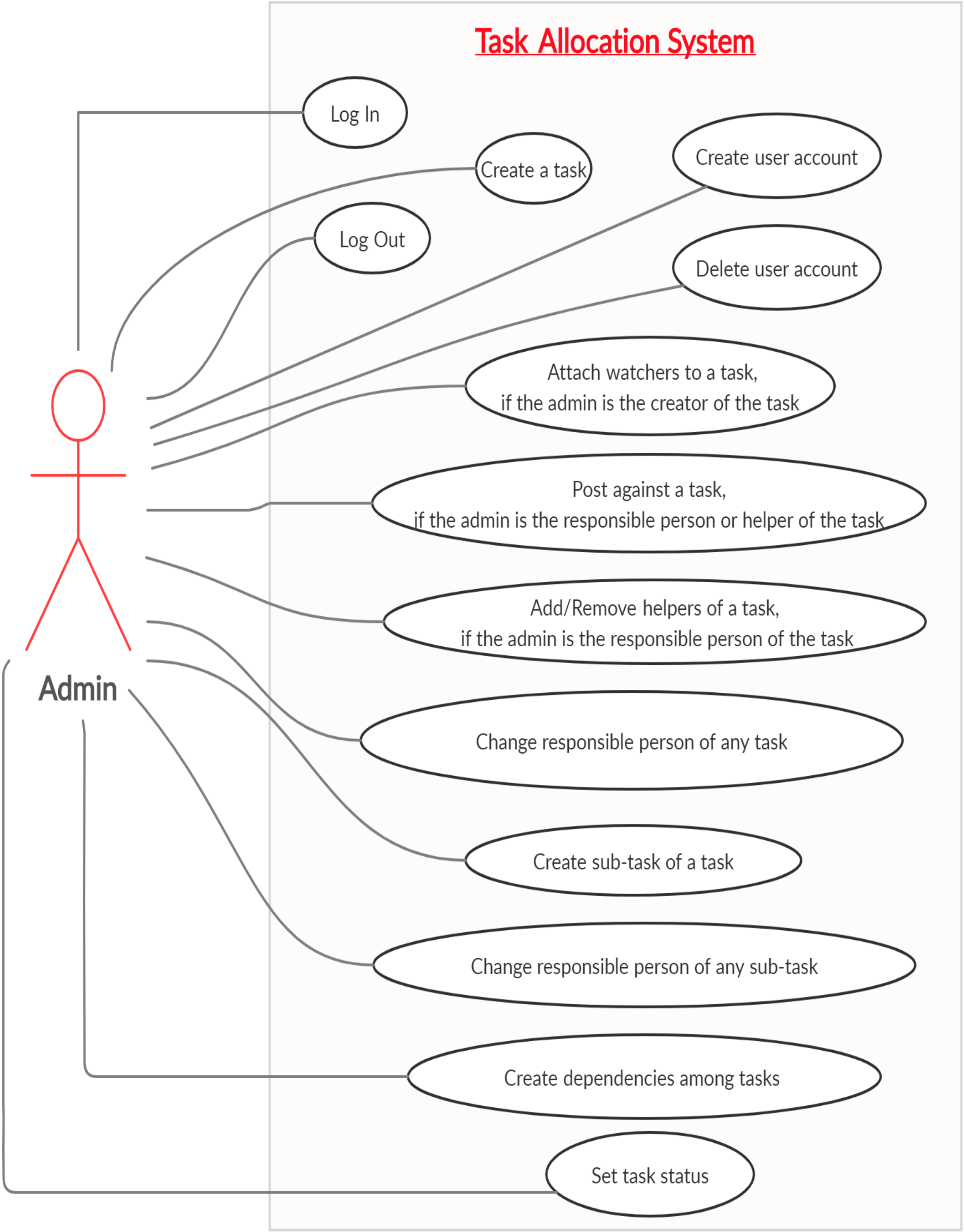
Questions:

Q1. Identify all actors and use cases. Draw a UML Use Case Diagram for the system described in the case study. (10)

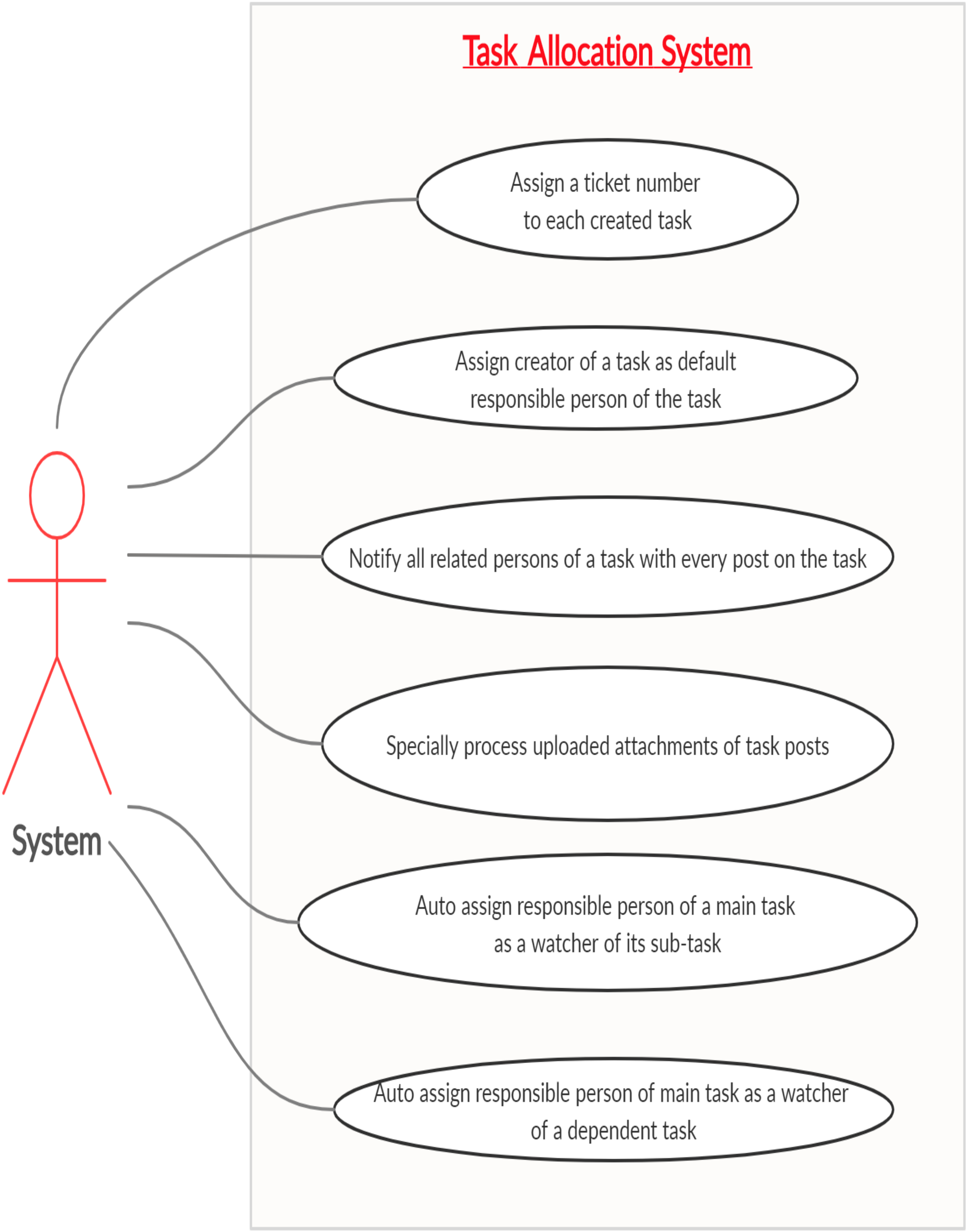
Answer to the question no. 1


Task Allocation System
Log In
Create a task
Log Out
Attach watchers to a task,
if the user is the creator of the task
Add/Remove helpers of a task,
if the user is the responsible person of the task
Post against a task,
if the user is the responsible person or helper of the task
Change responsible person of a task,
if the user is the creator or current responsible person
Create sub-task of a task,
if the user is creator or responsible person or helper of main task
Change responsible person of a sub-task,
if the user is the creator or current responsible person
Create dependencies among tasks
where the user is the creator or responsible person
Set task status,
if the user is the creator or responsible person
User

Answer to the question no. 1(contd.)


Task Allocation System
Log In
Create a task
Create user account
Log Out
Delete user account
Attach watchers to a task,
if the admin is the creator of the task
Post against a task,
if the admin is the responsible person or helper of the task
Add/Remove helpers of a task,
if the admin is the responsible person of the task
Change responsible person of any task
Create sub-task of a task
Change responsible person of any sub-task
Create dependencies among tasks
Set task status
Admin

Answer to the question no. 1(contd.)


Task Allocation System
Assign a ticket number to each created task
Assign creator of a task as default responsible person of the task
Notify all related persons of a task with every post on the task
Specially process uploaded attachments of task posts
Auto assign responsible person of a main task as a watcher of its sub-task
Auto assign responsible person of main task as a watcher of a dependent task
System

Q2. Write an expanded use case for the following (10)

Use Case Name: *Creating a task with the creator being the responsible person.*

Answer to the question no. 2

Use Case Name: Creating a task with the creator being the responsible person

Use Case Type: Primary

Actor(s): User, Admin

Brief Description: An actor can create a task by giving title and short description to it. He/she can also assign helpers and/or watchers to the task while creating.

Typical Course of Events

Actor Actions	System Response
1. Actor goes to the task-allocation system URL.	2. System responds with login page.
3. Actor enters username, password and clicks login button.	4. System verifies credentials and redirects to homepage.
5. Actor clicks on create a task button.	6. System responds with a create task page where there are fields for task title, task description, assign responsible person, assign watchers and assign helpers. The responsible person is selected as creator by default.
7. The actor fills in the task title, task description and optionally adds some helpers & watchers & then clicks the create task button.	8. The system assigns a ticket number to the task and creates it showing a prompt: “Task has been created successfully”.

Alternate Course of Events

4.1 Credentials are wrong. System prompts the actor to re-enter the credentials.

8.1 Actor kept the title field or description field empty. “Cannot create task with empty title field or description field” – message shown with option to go back to previous screen.

8.2 Description less than 50 words. “Description too short. Must be atleast 50 words” – message shown with option to go back to previous screen.

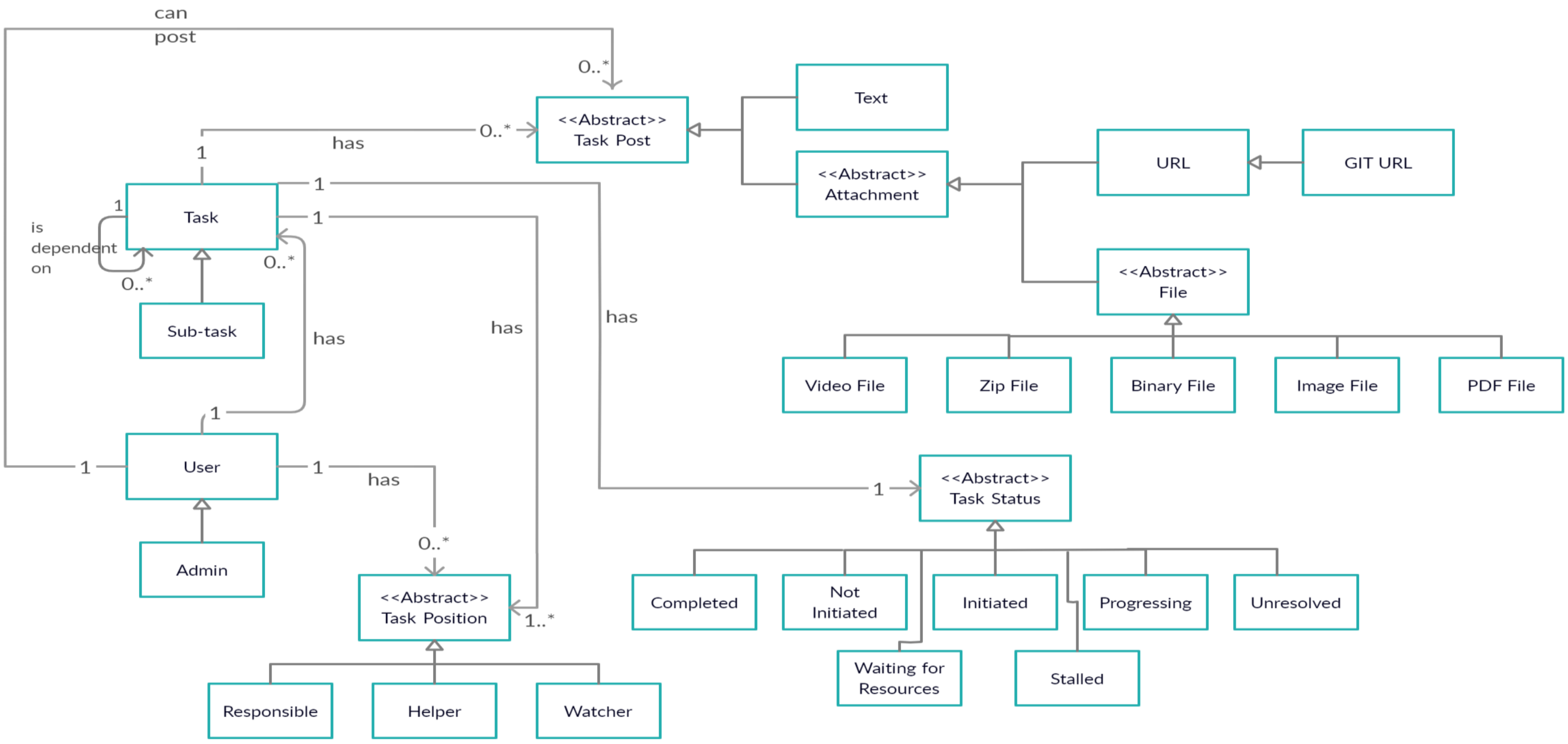
8.3 Non-existent username added to the helper field or watcher field. “Username not found. Please add an existing user as helper or watcher” – message shown with option to go back to previous screen.

PTO

Q3. Assuming that an MVC architecture is used. Draw a high-level UML Class Diagram of the *Model* classes of the system. You do not need to show attributes and methods. (10)

Answer to the question no. 3

UML Class Diagram of Model Classes of Task Allocation System



THE - END