MIDDLE EAST TECHNICAL UNIVERSITY
NORTHERN CYPRUS CAMPUS

Computer Engineering Program

# CNG 495
# CLOUD COMPUTING

# FALL 2024 – TERM PROJECT PROPOSAL

**Team members:**

Raed H. Manna – 2550911

Omar A. Mourad – 2487080

# Contents

1. Introduction ........................................................................3
2. Project Description ................................................................3
2.1. Features .............................................................................3
2.1.1. Admin Features: .............................................................3
2.1.2 Normal User Features: .....................................................4
3. Backend Framework .............................................................5
3.1 Why we chose FastAPI? .........................................................5
4. Cloud Services: .....................................................................6
5. Diagrams: .............................................................................7
5.1. Activity Diagrams: ..............................................................7
5.1.1 Task Creation Activity Diagram (Admin): ................................7
5.1.2. Task Update Activity Diagram (User): ....................................9
5.2. Data-Flow Diagram: ............................................................10
5.2.1. level – 0 DFD: ..................................................................10
5.2.1. level – 1 DFD: ..................................................................11
6. Expected Contribution: ..........................................................12

# 1. Introduction

Our project, **ClearTasks**, focuses on developing a cloud-based task management web application using FastAPI and AWS services. The aim is to provide users with an efficient platform to manage, track, and organize their tasks and milestones, ensuring enhanced productivity for individuals and teams. Users can create tasks, set deadlines, and assign tasks to themselves or others. The platform will leverage cloud services to store data, send notifications, and provide real-time updates.

# 2. Project Description

Our Task Management System, ClearTasks, offers two main user roles: Admin and Normal User**.** Each role has specific features that allow them to effectively manage tasks and collaborate with others.

## 2.1. Features

### 2.1.1. Admin Features:

Admins have enhanced capabilities to manage the platform and assign tasks to normal users. Admins can:

- **Create Tasks:** Admins can create tasks with a title, description, deadline, and priority.

- **Assign Tasks to Users:** Admins can assign tasks to specific users and monitor their progress.

- **Monitor Task Progress:** Admins can view the status of tasks (e.g., "In Progress," "Completed") and track individual user performance.

- **Edit/Delete Tasks:** Admins have full control over all tasks and milestones and can edit or delete them as needed.

- **User Management:** Admins can create, edit, or delete user accounts, ensuring they have control over who can access the platform.

- **Notifications:** Admins can send notifications to users via email using AWS SNS, reminding them of upcoming deadlines or overdue tasks.

### 2.1.2 Normal User Features:

Normal users are individuals who can manage their own tasks and collaborate with other users. They can:

- **Create Personal Tasks:** Users can create their own tasks to track their personal goals or projects. Each task can have a title, description, deadline, and priority level.

- **Update Task Status:** Users can mark tasks as "In Progress" or "Completed" as they work through their list.

- **View Assigned Tasks:** Users can see tasks assigned to them by admins or other users and track deadlines.

- **Receive Notifications:** Users will receive notifications for tasks that are approaching deadlines or tasks that have been assigned to them.

## 3. Backend Framework

Our backend will be powered by **FastAPI**, a modern and efficient Python framework known for its speed and ease of use. FastAPI will serve as the core backend framework, connecting the application to various cloud services by handling API requests and responses. we will use the **boto3** library to interact with **AWS S3** for storing tasks and files, **AWS Cognito** for secure authentication, and **AWS SNS** to trigger notifications, enabling seamless integration between the backend and the cloud infrastructure.

### 3.1 Why we chose FastAPI?

- **High Performance:** FastAPI is one of the fastest web frameworks available, handling large numbers of concurrent requests efficiently.

- **Asynchronous Programming:** FastAPI's support for async programming makes it well-suited for real-time features like task updates and notifications.

- **Automatic API Documentation:** FastAPI automatically generates interactive API documentation, which will make it easier for our team to interact with and test the API/Backend during development.

## 4. Cloud Services:

### Amazon AWS S3 (Storage):

We will be using Amazon S3 to store and manage all user data and task information securely. Using S3 it will handle storing our task related files, and provide good availability, which will be the key for users to seamlessly update and access data.

### Amazon SNS (Notification):

The notification service from Amazon will be used to push notifications as a way to inform users about task deadlines. An automated alert, will ensure that users will receive either a SMS message or E-mail (to be decided), to help them keep up with deadlines.

**AWS Cognito:**

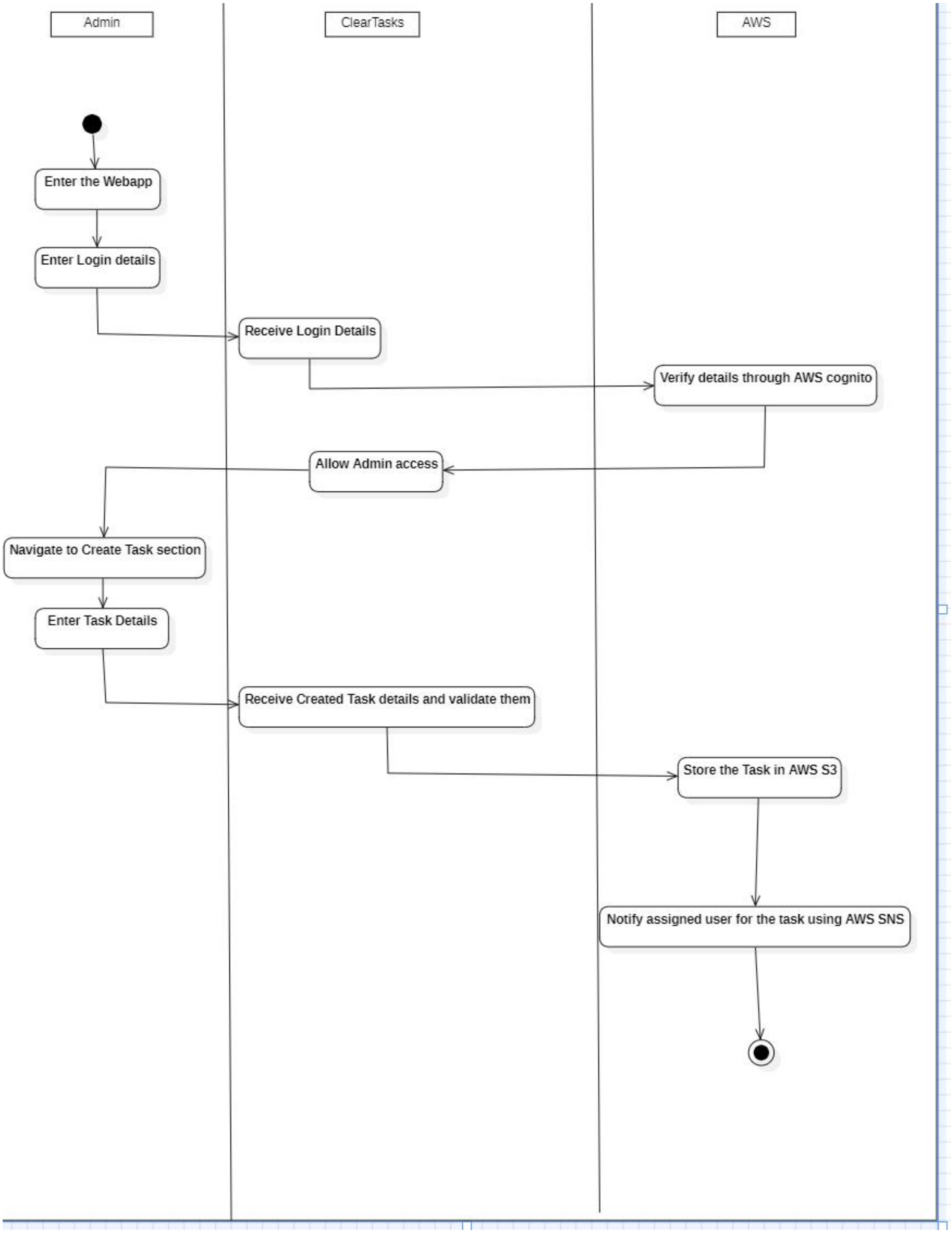
The security service will be used for user authentication and authorization. Cognito will make sure that only the authenticated users are able to access the application.

# 5. Diagrams:

## 5.1. Activity Diagrams:

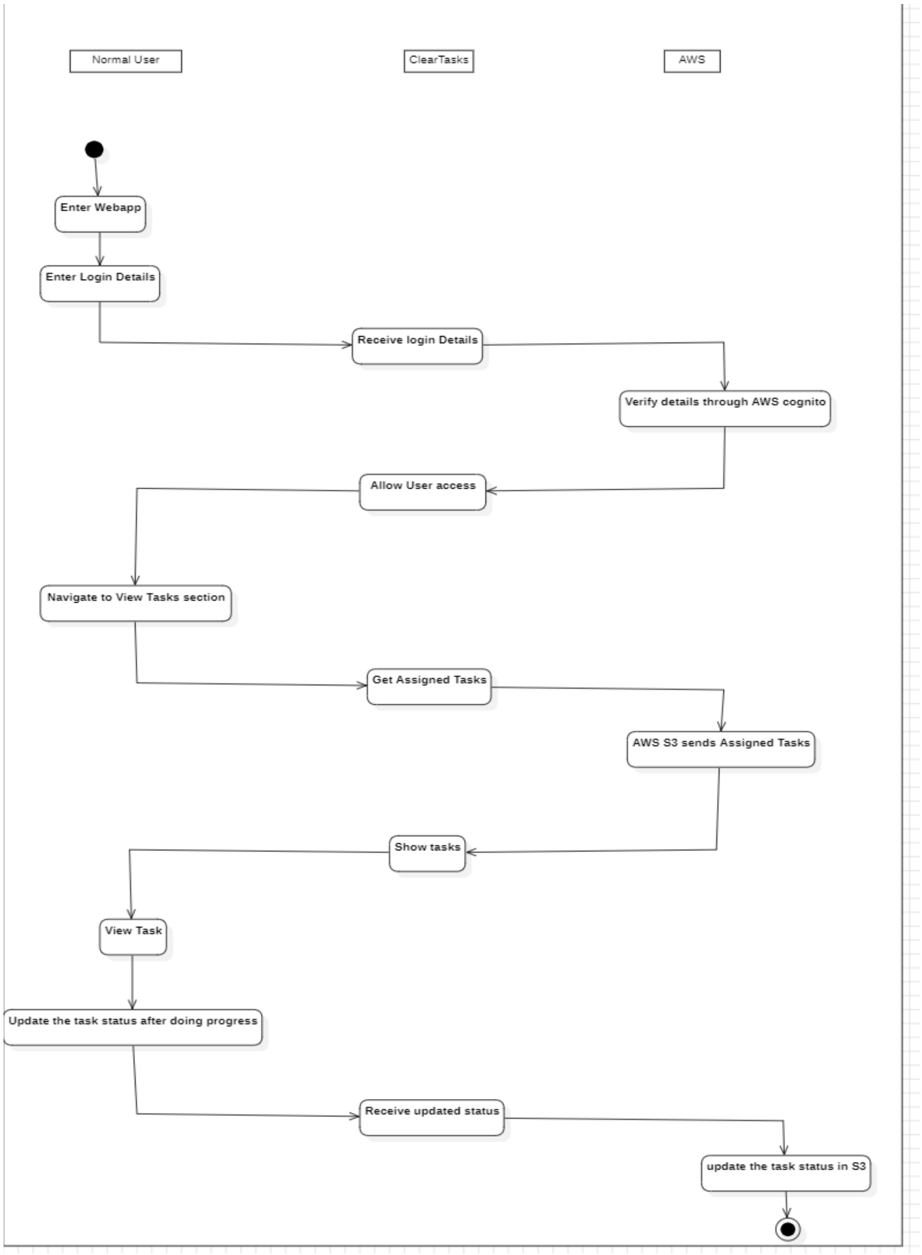
### 5.1.1 Task Creation Activity Diagram (Admin):

In the following activity diagram, we show how an admin creates and assigns tasks:


Admin
ClearTasks
AWS
Enter the Webapp
Enter Login details
Receive Login Details
Verify details through AWS cognito
Allow Admin access
Navigate to Create Task section
Enter Task Details
Receive Created Task details and validate them
Store the Task in AWS S3
Notify assigned user for the task using AWS SNS

## 5.1.2. Task Update Activity Diagram (User):

The following activity diagram shows how a user updates task status. Same steps for admin updating other aspect of the task:


Normal User
ClearTasks
AWS
Enter Webapp
Enter Login Details
Receive login Details
Verify details through AWS cognito
Allow User access
Navigate to View Tasks section
Get Assigned Tasks
AWS S3 sends Assigned Tasks
Show tasks
View Task
Update the task status after doing progress
Receive updated status
update the task status in S3

## 5.2. Data-Flow Diagram:

### 5.2.1. level – 0 DFD:

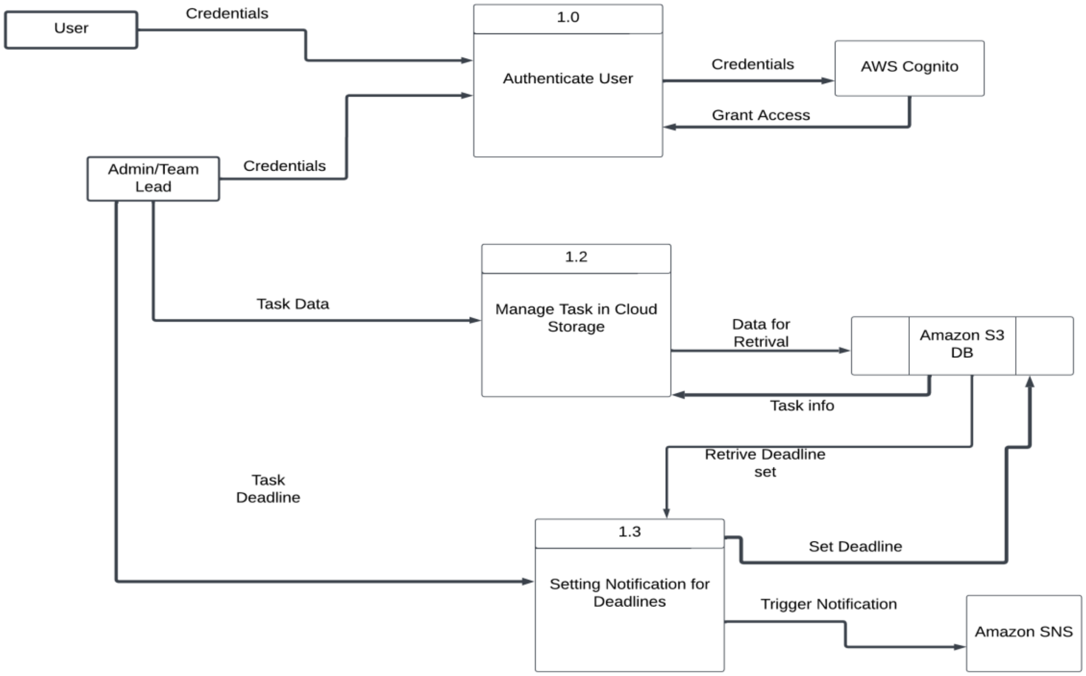
We mainly have Our services, the basic outline of the processes that will be used and our two main entities that will be using the connected services.

Both the User and Admin will have their login checked by AWS Cognito.

The admin will be providing the data needed to store in Amazon S3 DB so that we can retrieve the data later for Display, edit, delete, or view.

Lastly the we will me managing the notifications to be sent through amazon SNS by checking the deadline and after a trigger it will contact SNS to send a message through email.

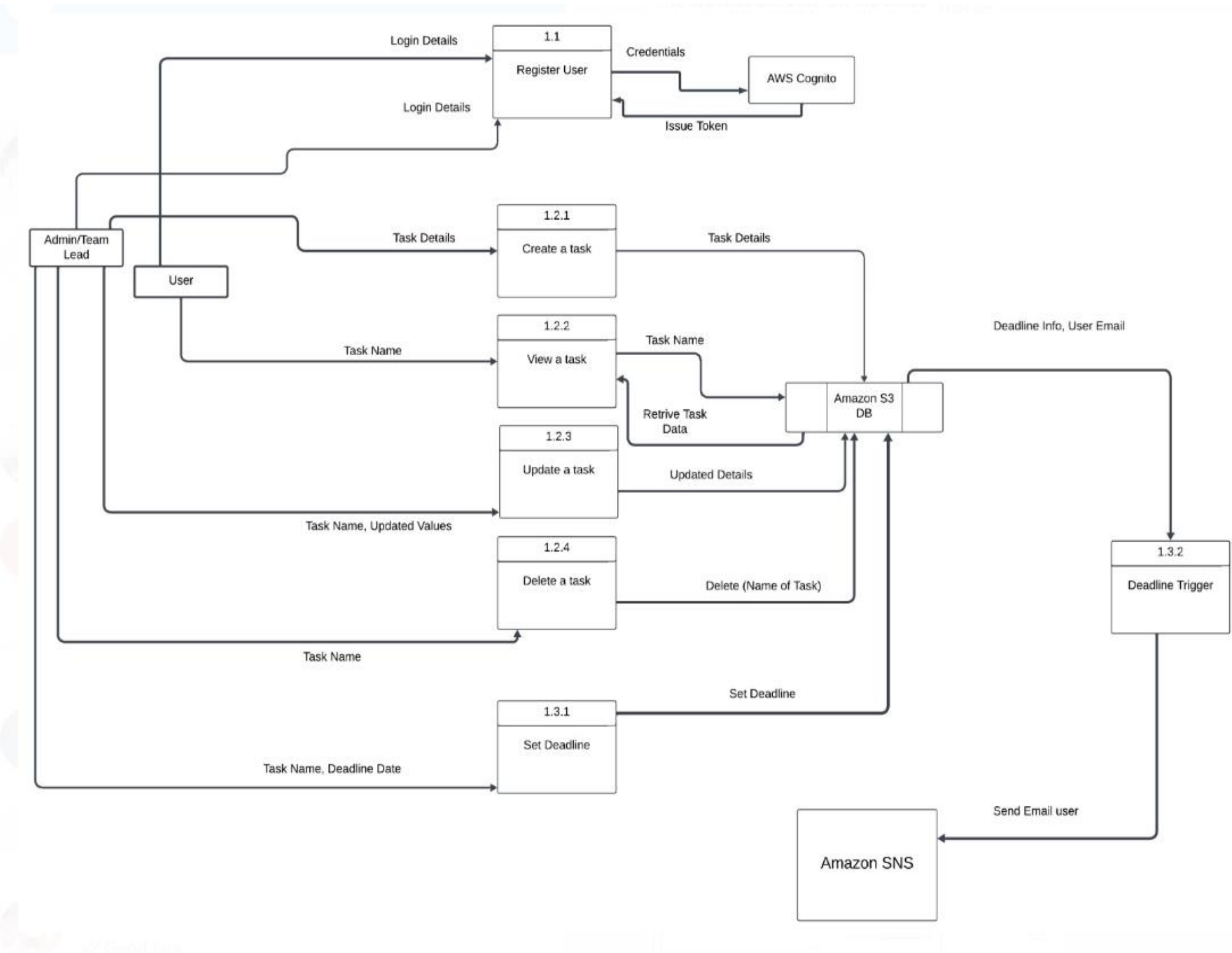
The following figure represents level – 0 DFD:



### 5.2.1. level – 1 DFD:

Same structure as our level 0 diagram with more details into each step and what happens in it.

The following figure represents level – 1 DFD:



## 6. Expected Contribution:

Given the scope and scale of the **ClearTasks** project, we believe that co-working on all major areas will maximize productivity and ensure a cohesive outcome. However, one of us may focus more on some areas of the project than the other (be the decision maker).

Expected primary focus of each team member:

**Front-End → React:** Raed

**Back-End → Python – FastAPI:** Omar & Raed

**Cloud Services → AWS S3, Cognito, SNS:** Omar

**Our next step:** preparing the GitHub repository and getting more familiar with the cloud services before starting to use them