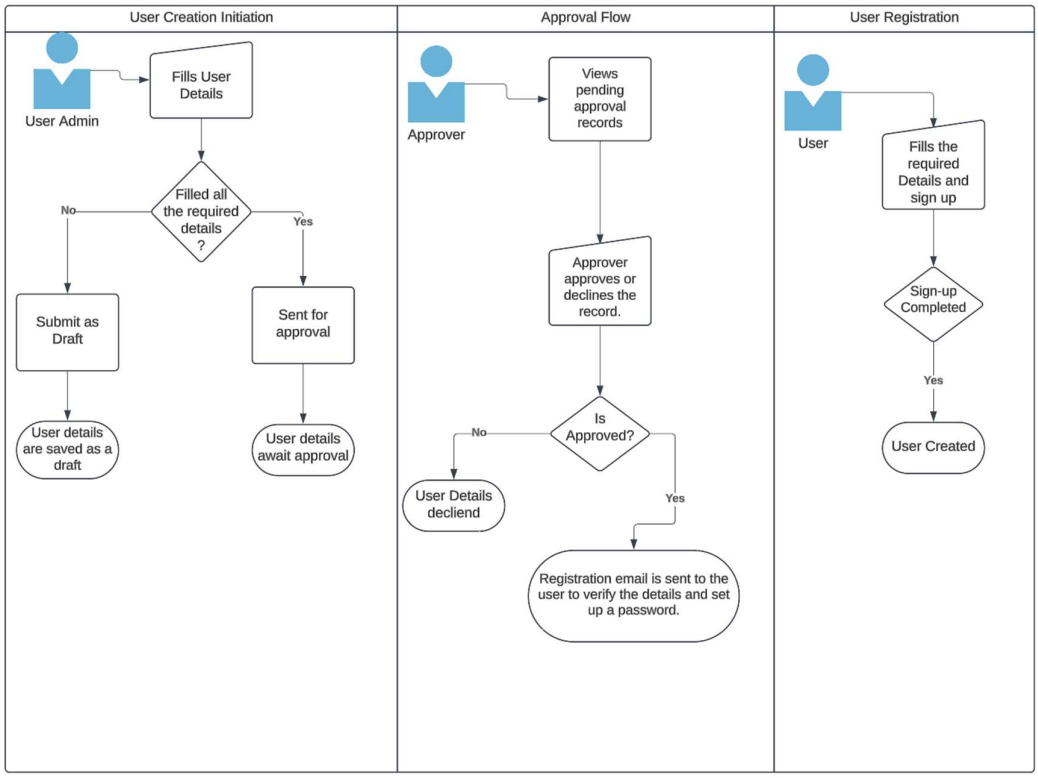
Objective: Design a database schema and develop the application code for a User Creation process adhering to Clean Architecture and Domain-Driven Design (DDD) principles. The implementation should reflect the logic and workflow depicted in the below flowchart.



Flowchart Summary:

The flowchart illustrates the User Creation process through three main phases: User Creation Initiation, Approval Flow, and User Registration.

The steps are as follows:

1. User Creation Initiation:

- **Input:** Admin fills in user details.
    - User Properties to Include:
        - **Email** (Unique, Required)
        - **Name** (Required)
        - **Phone Number** (Optional)

        - **LinkedIn URL** (Optional)
        - **Role** (Required)
        - **Status** (Required)
        - **Alias Name** (Optional)

- **Validation:**
    - If all required details are provided, the request can be sent for approval.
    - If not, the admin can save the request as a draft.
- **Outcomes:**
    - Draft: User details are saved as a draft for later editing.
    - Sent for Approval: The user details await approval.

2. Approval Flow:

- **Input:** Approver reviews pending user creation requests.
- **Actions:**
    - Approver can approve or decline the record.
- **Outcomes:**
    - Declined: User details are declined.
    - Approved: A registration email is sent to the user to verify the details and set up a password.

3. User Registration:

- **Input:** The user receives a registration email and fills in the required details to complete sign-up.
- **Outcome:** If the sign-up process is successfully completed, the user is created.

Requirements:

1. **Database Design:**
    - Design a database schema for this application
2. **Code Implementation:**

- Must Develop the application following **Clean Architecture, Domain-Driven Design (DDD)** and clean code.

3. **Deliverables**:

    - Data Diagram and Db Schema
    - **Code:** Implement the flowchart logic using Clean Architecture, Domain Driven Design in .NET Core

Note:
For any external dependencies, such as sending an email, a complete functional flow is not required. You can use mocks for these cases. We only expect the database design. Stored procedures and database-specific implementations are not expected.

However, it is mandatory to adhere to **Clean Architecture** and **Domain-Driven Design (DDD)** principles. Your implementation should clearly demonstrate how such functionality would be integrated following these principles.

Submission Instructions:

1. Share the database schema design (ER diagram or SQL script).
2. Provide a zip file containing the codebase or a link to the GitHub repository.