**Requirements:**

**User Requirements:**

1. The system should allow users to log in into the system if they already have an account.
2. The system should allow new users to register an account.
3. Any job seeker who logs into the system shall be able to apply to jobs posted in the homepage and also search for jobs.
4. Any company admin who logs into the system shall be able to add job posts or deactivate jobs and also be able to view all the applications on a certain job.

**System Requirements:**

1. Show login form to users and validate their login info with the database.
2. Show registration form to users and authenticate their information before successful registration.

3.1- Show application form to job seekers when the apply button is pressed on any job post.

3.2- The system shall allow a job seeker to search for a specific job through a search bar and return the result of the search.

4.1- The system shall update the job status in the database if the company admin deactivated a job post.

4.2- The system shall show a form to the company admin to fill the details of the new job post when the create post button is pressed.

4.3- The system shall add new job post information in the database with an active status when the company admin creates a new post.

**Functional Requirements:**

1. When a user tries to log into the system, check his info with the database.
2. If the login info is not correct, show error message and allow user to try to login again.
3. Each user is identified with his username and a unique email that is stored in the database.
4. When a user tries to register a new account, update the database with the user’s info after validating the information.
5. If the registration information is not correct/already exists, show error message and allow user to try to register again.
6. Validate a job seeker’s application when the submit button is pressed.
7. Update the database with a user’s application to a job when his application is submitted.
8. Allow company admin to create new job posts if it does not already exist in the database.
9. Add the new job post information to database after validation.
10. Change the status of a job post from active to in-active in the database when a company admin presses the deactivate button in database.
11. Show list of all applications on a job to the company admin.
12. End user session when the logout button is pressed.

**Non-Functional Requirements:**

1. The system shall be responsive and not show any time lag when performing requests.
2. The system shall not fail to update the database when needed.

**Software Process**

**Suggested type of software process:** Agile

**Division of phases:**

**Phase 1)** Model design & validate data with user.

**Phase 2)** Website home page UI & Login and register UI and logic.   
This phase includes implementing:

* Website homepage view
* Login page view
* Job seeker’s register page view
* Login function logic
* Job Seeker ‘s register function logic
* Logout function logic
* Middleware for protecting routes
* Testing

**Phase 3)** Website admin portal page UI design & logic implementation.

This phase includes implementing:

* Admin dashboard view
* Add company page view
* Handle statistics that will be shown to admin
* Add company function logic
* Remove company function logic
* Testing

**Phase 4)** Company portal page UI design & logic implementation.

This phase includes implementing:

* Company dashboard page view
* Add job post page view
* Add job post function logic
* Applications page view
* Applications function logic get data from database
* Delete Job post function logic
* Testing

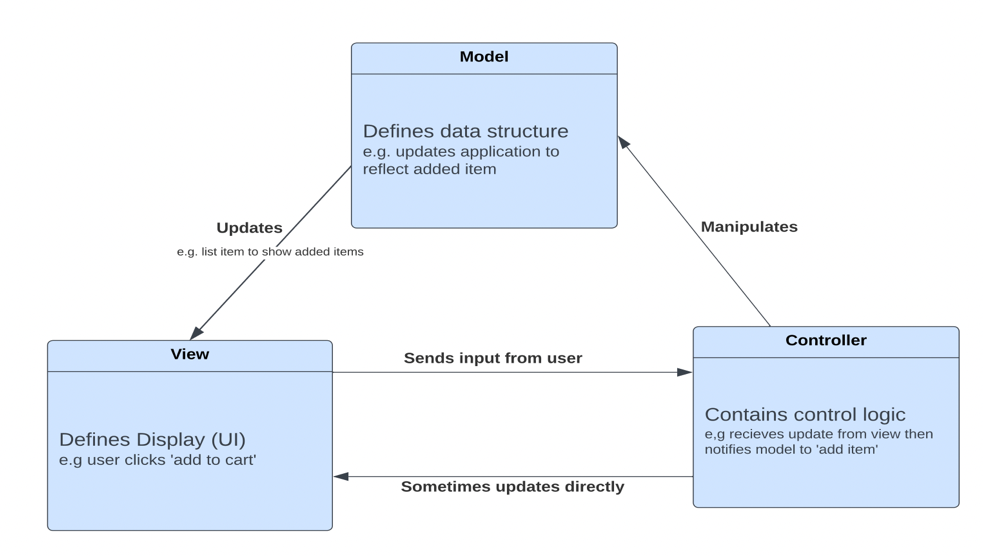
**Phase 5)** User portal page UI design & logic implementation

This phase includes implementing:

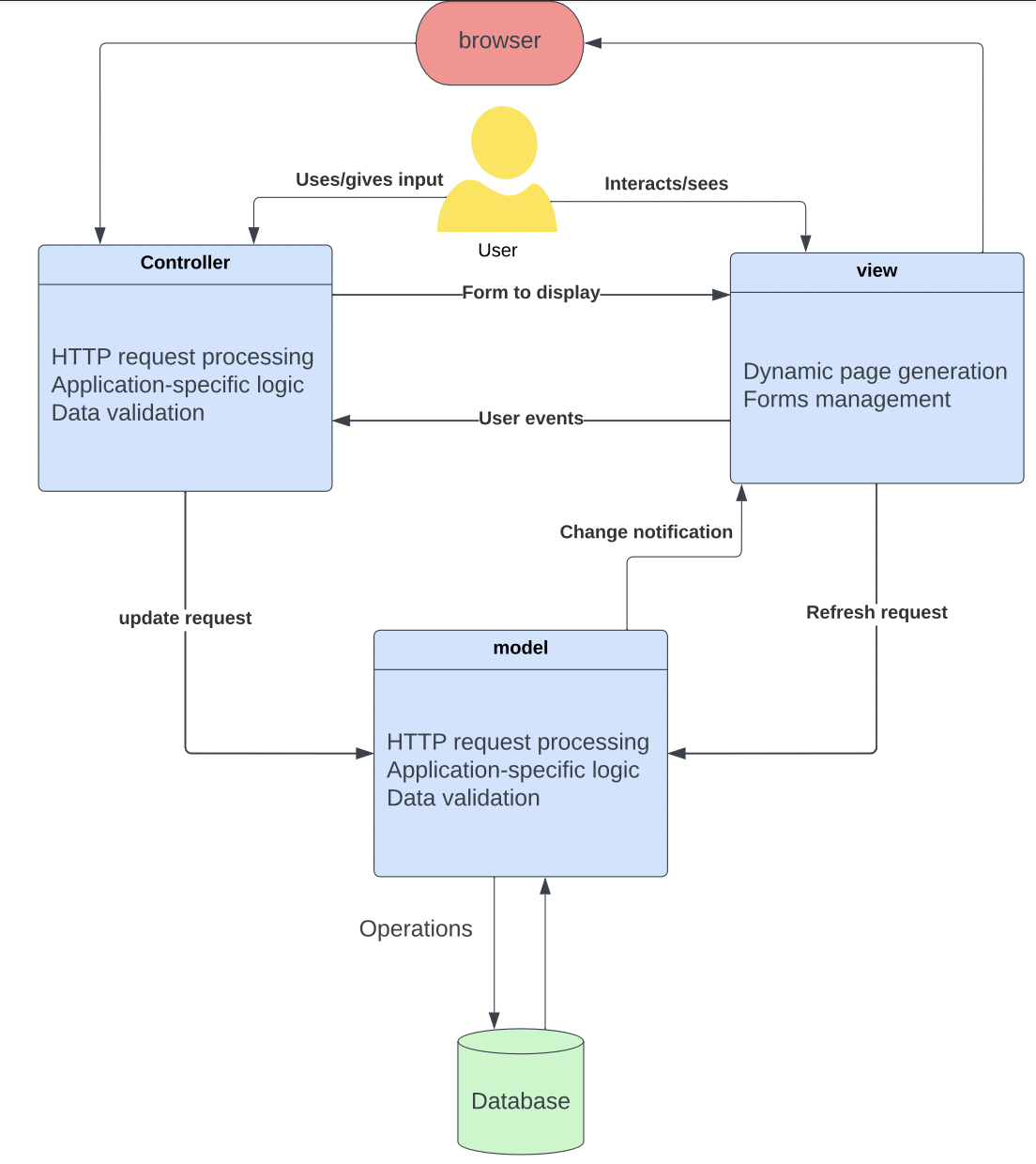
* Jobs’ feed page view
* Job’s feed function logic
* Apply to job page view
* Apply to job function logic
* Testing

**Architectural Design:**

General MVC architecture:



Web application MVC architecture:



**UML Diagrams**

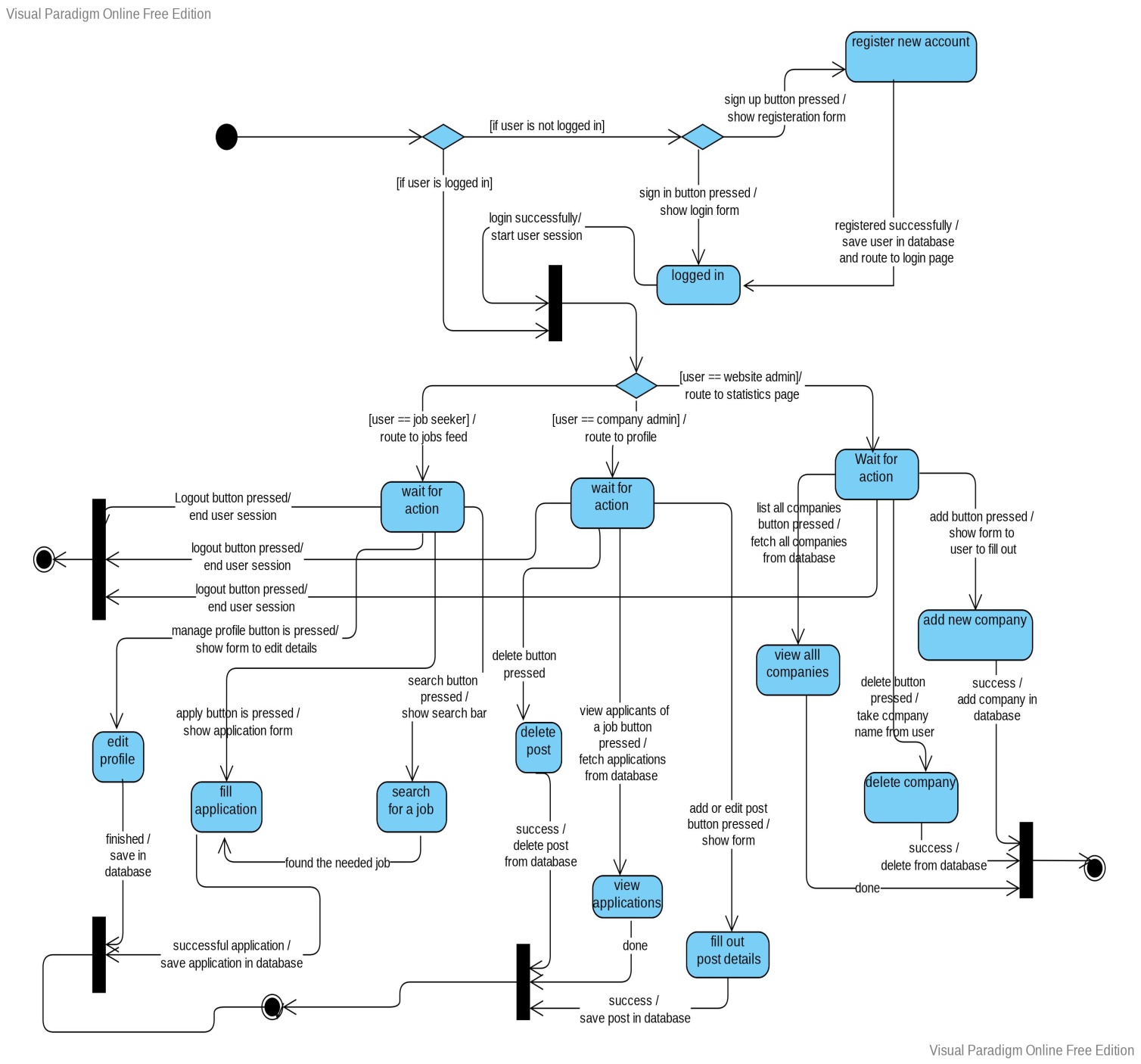
**Diagram, schematic

Description automatically generatedUse Case:**

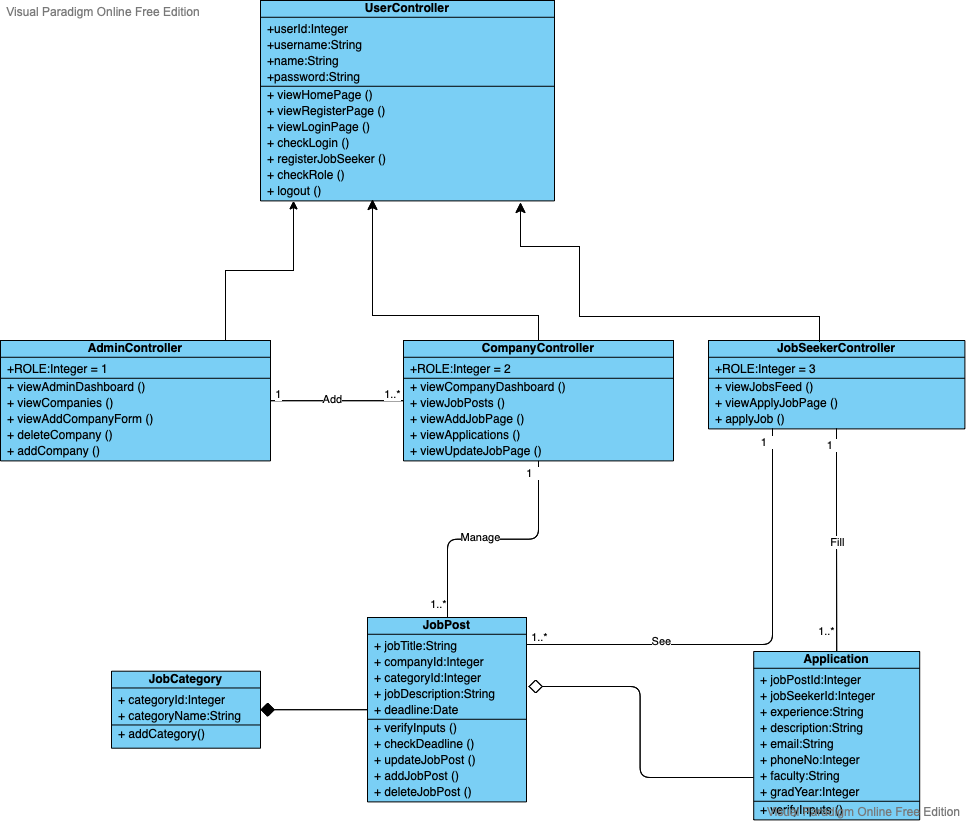
Diagram

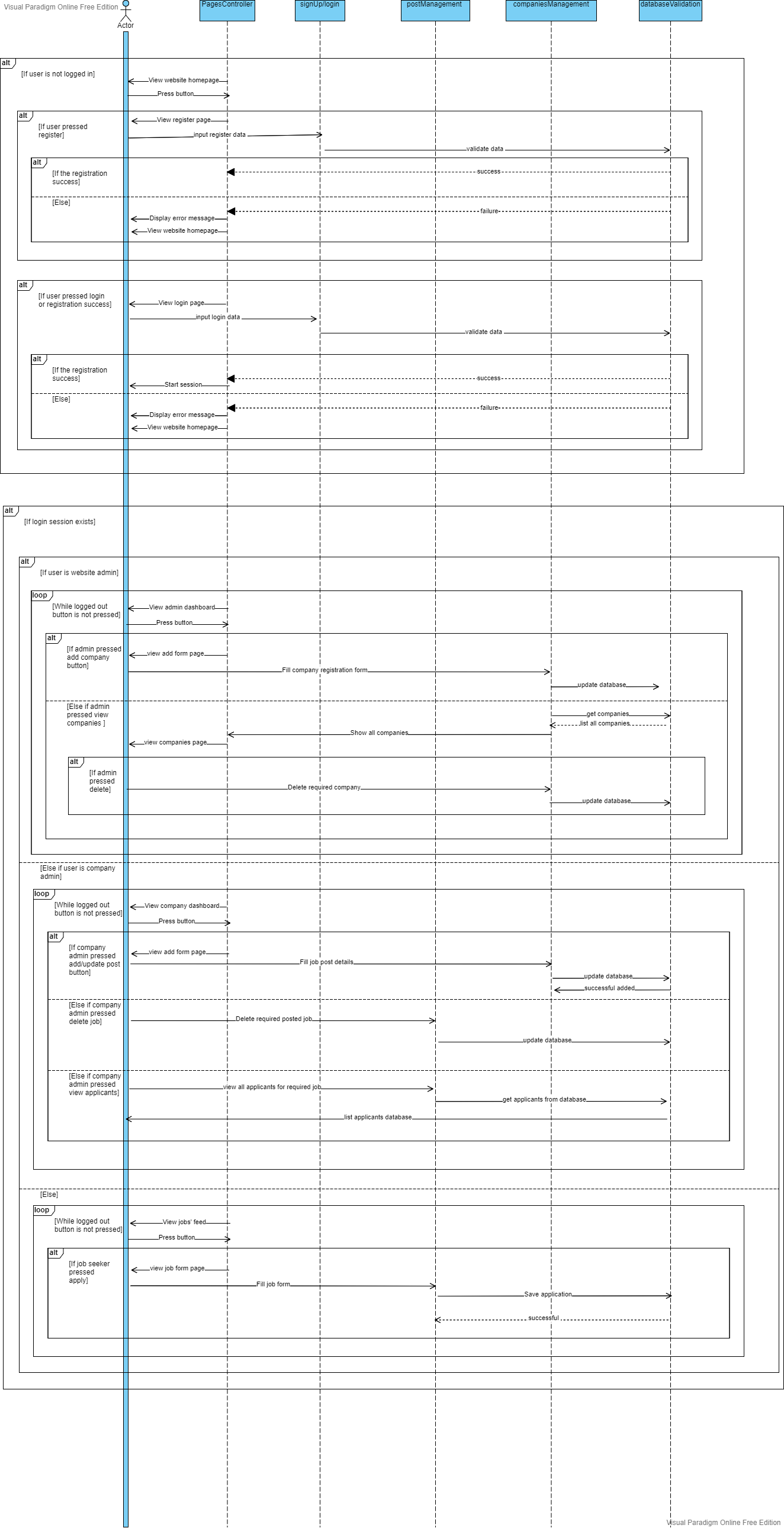
Description automatically generated**Activity Diagram:**

**State Diagram:**

****

**Class diagram:**

****

**Sequence Diagram:**