

<JobHub>

System Requirements Specification

#	Name	ID	Group
1	Hussein alyamnii zein eldeem hussein	20200157	S3
2	Shady mohamed abdelgawad	20200246	S3
3	Hady saad eldein roshdy mohamed	20190606	S4
4	Laila abdelaaty mohammed abdelaal	20211072	S2
5	Fatima mohammed ahmed amin	20211068	S1
6	Sara salah mohamed	20200216	S1

Project contact member email : ***hussein.yamany11@gmail.com***

Project contact member mobile : ***01125385922***

TA. < Reham >

1. System architecture for your system

- For the Online Job Recruitment application, the most suitable system architecture would be the Model-View-Controller (MVC) pattern. The MVC pattern separates the application into three interconnected components - Model, View, and Controller, providing a modular and scalable structure. Here's how the MVC architecture addresses two non-functional requirements of the system:

1. Scalability:

Explanation: The MVC pattern promotes the separation of concerns, allowing for the independent development and modification of each component. This separation makes it easier to scale the application by adding or modifying specific functionalities without affecting the entire system.

Example: As the user base grows, the system may need to handle a larger number of concurrent users. In an MVC architecture, it is possible to scale the application horizontally by deploying multiple instances of the Controller and View components. This scalability ensures that the system can handle increased user traffic without compromising performance.

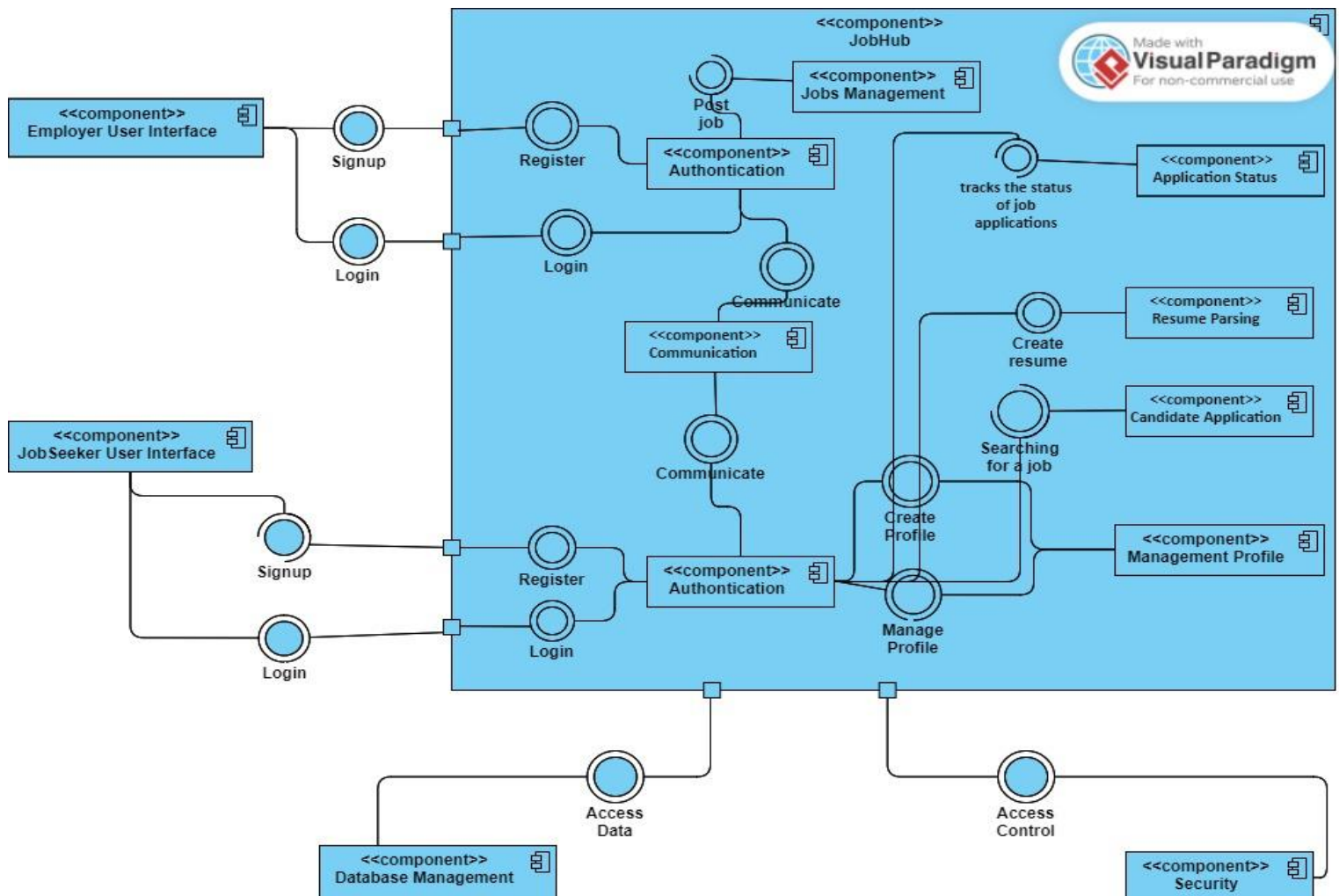
2. Maintainability:

Explanation: The modular structure of the MVC pattern makes the application more maintainable. Changes to one component (Model, View, or Controller) can be made without affecting the others. This separation of concerns simplifies maintenance, updates, and bug fixes, as developers can focus on specific components without disrupting the entire system.

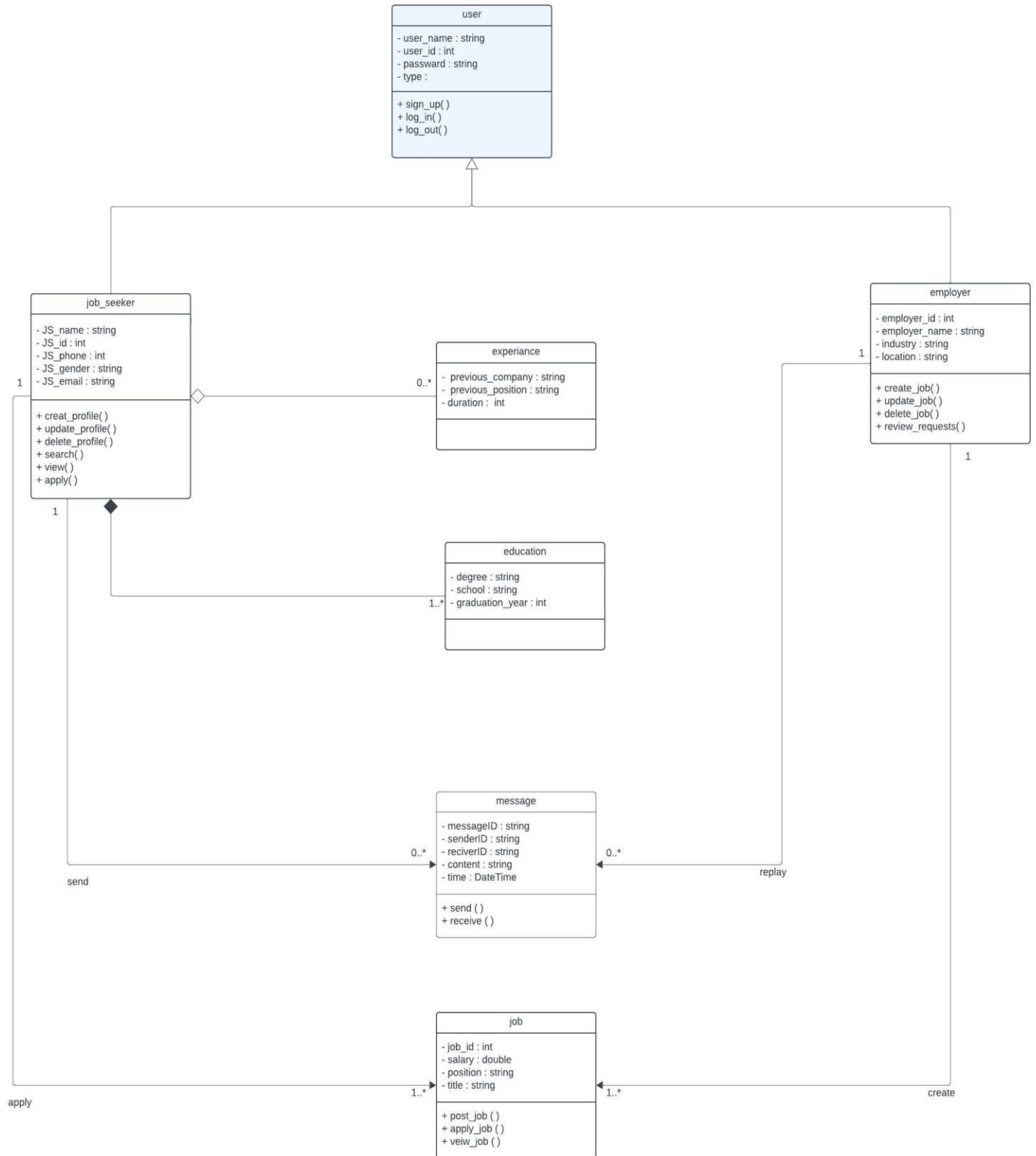
Example: If there is a need to update the user interface (View) to improve the user experience, developers can make changes to the View component without altering the underlying application logic (Model) or user input handling (Controller). This ensures that maintenance tasks are localized and don't introduce unintended consequences.

Additionally, the MVC pattern aligns well with the requirements of the Online Job Recruitment application:

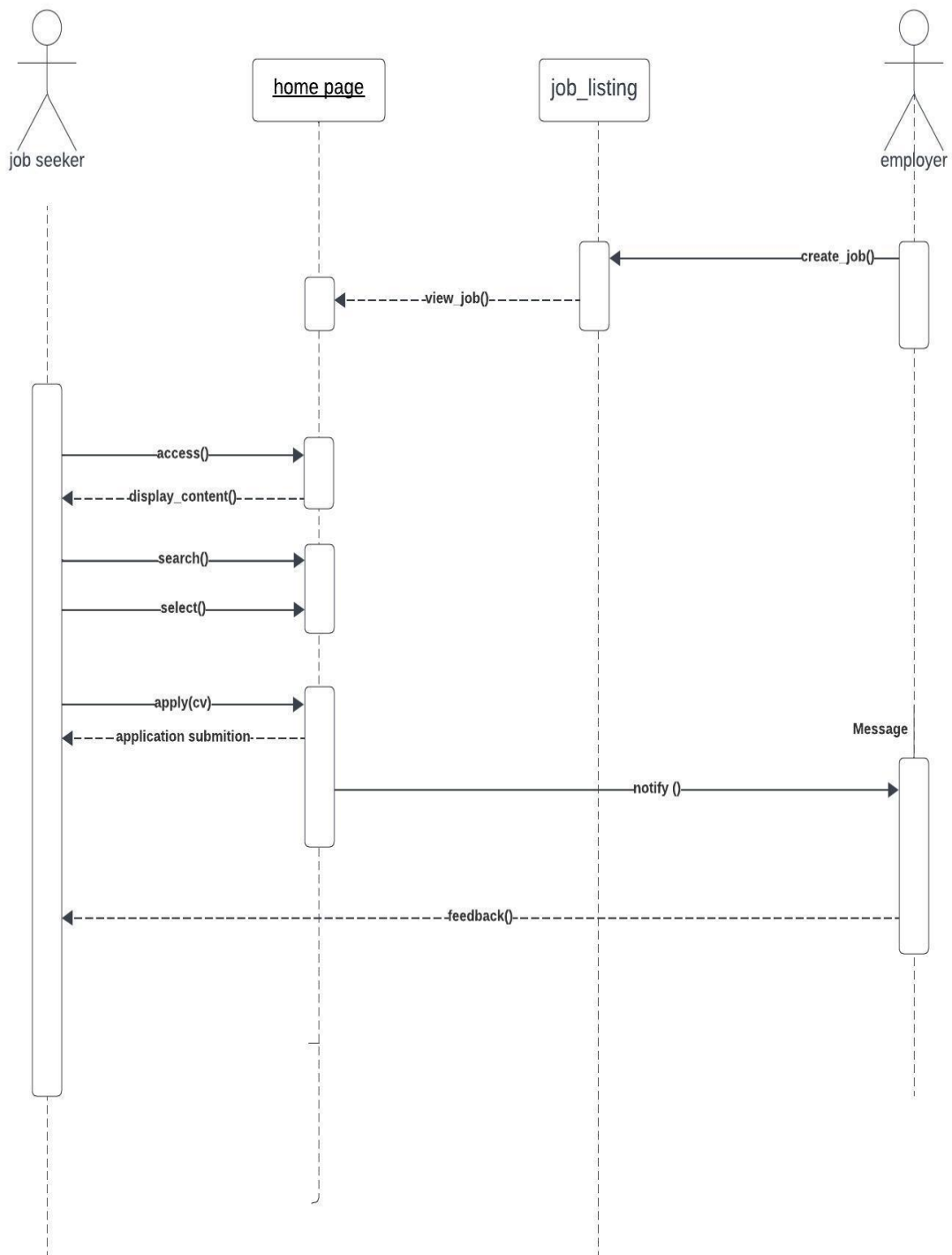
- User Authentication and Profile Management:
 - Model: Manages user data, authentication, and profile information.
 - View: Displays the user interface for authentication and profile management.
 - Controller: Handles user input, processes authentication requests, and updates user profiles.
- Job Listings and Application Process:
 - Model: Manages job listings, company profiles, and application data.
 - View: Presents job listings and application forms to users.
 - Controller: Processes user applications, communicates with the Model to update data, and manages the flow of the application process.



2. Class Diagram

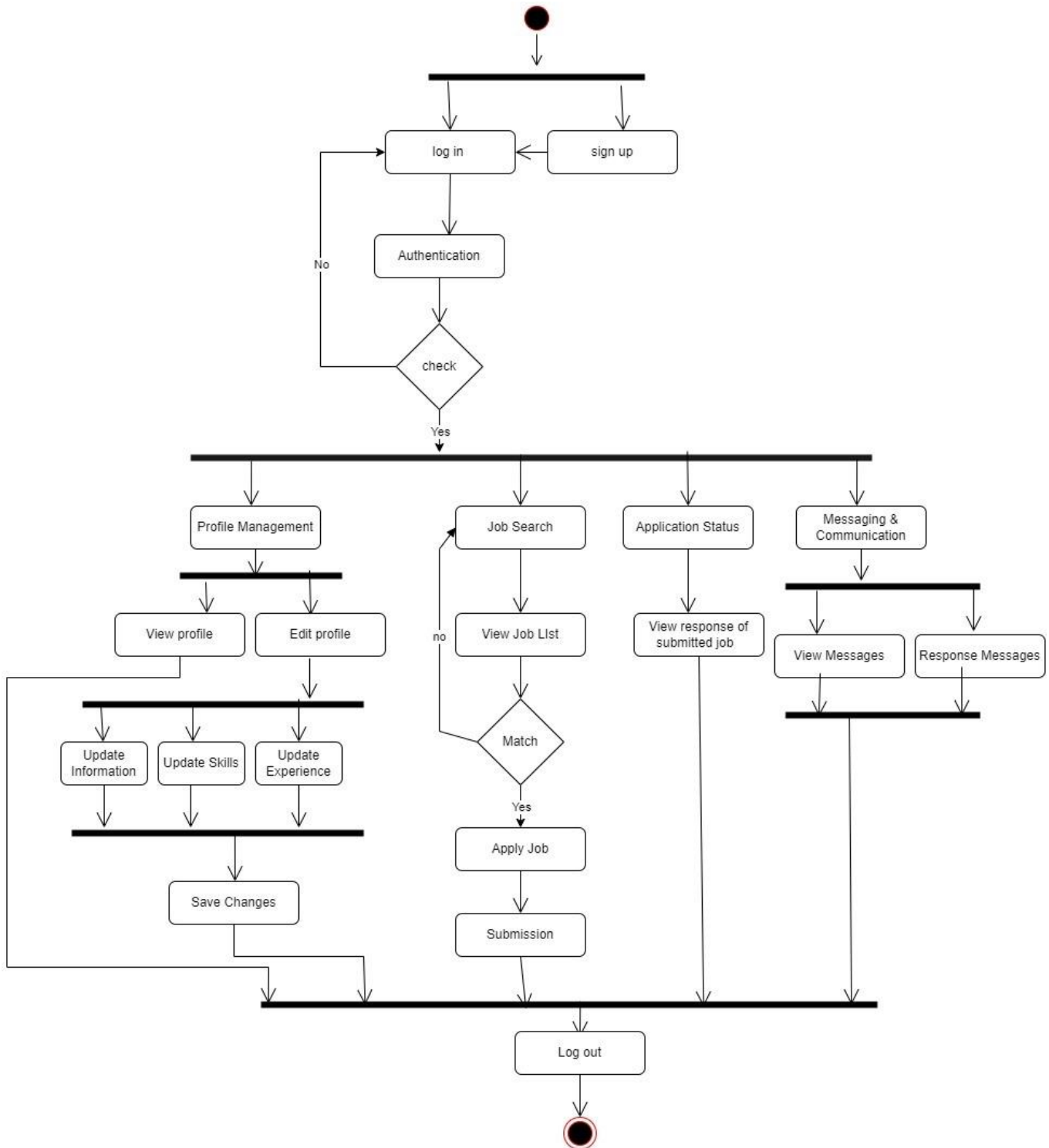


3. Revised Sequence Diagram

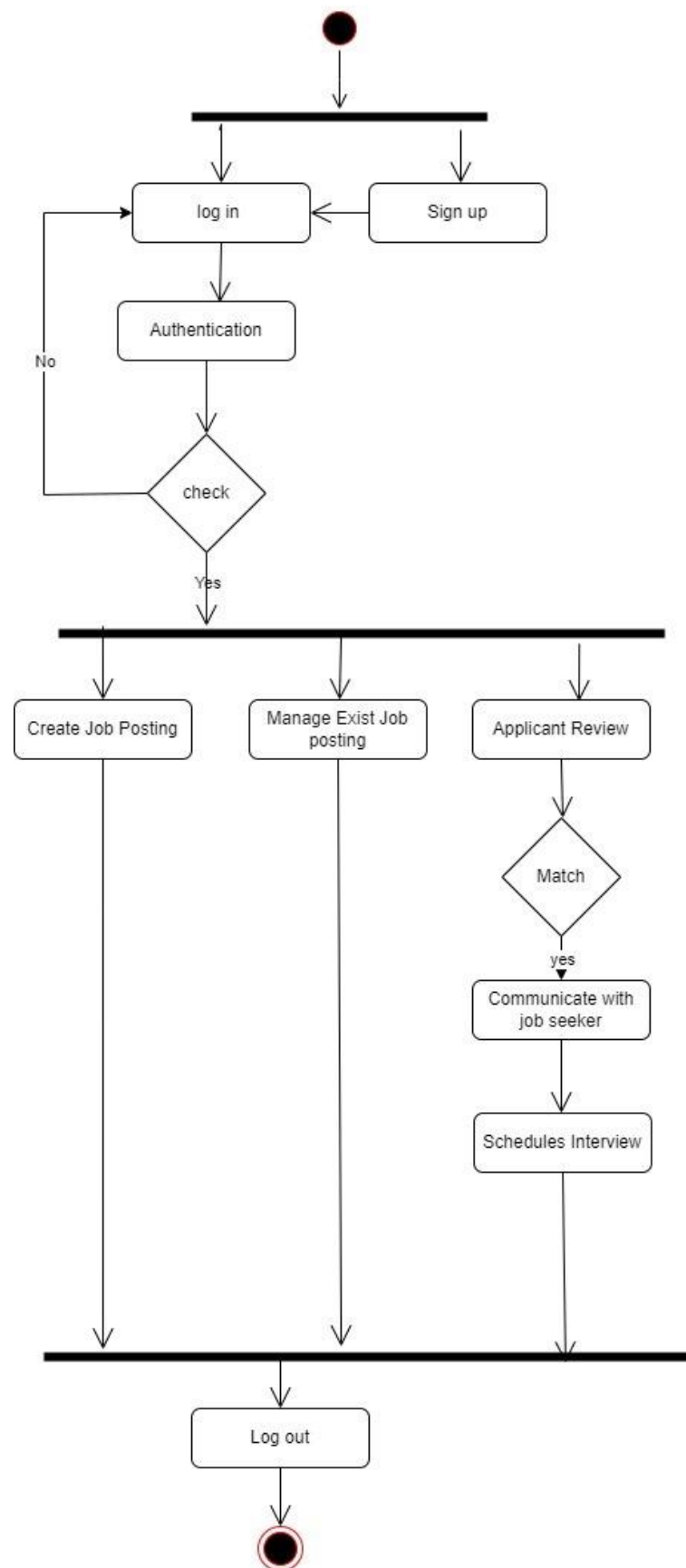


4. Activity Diagram

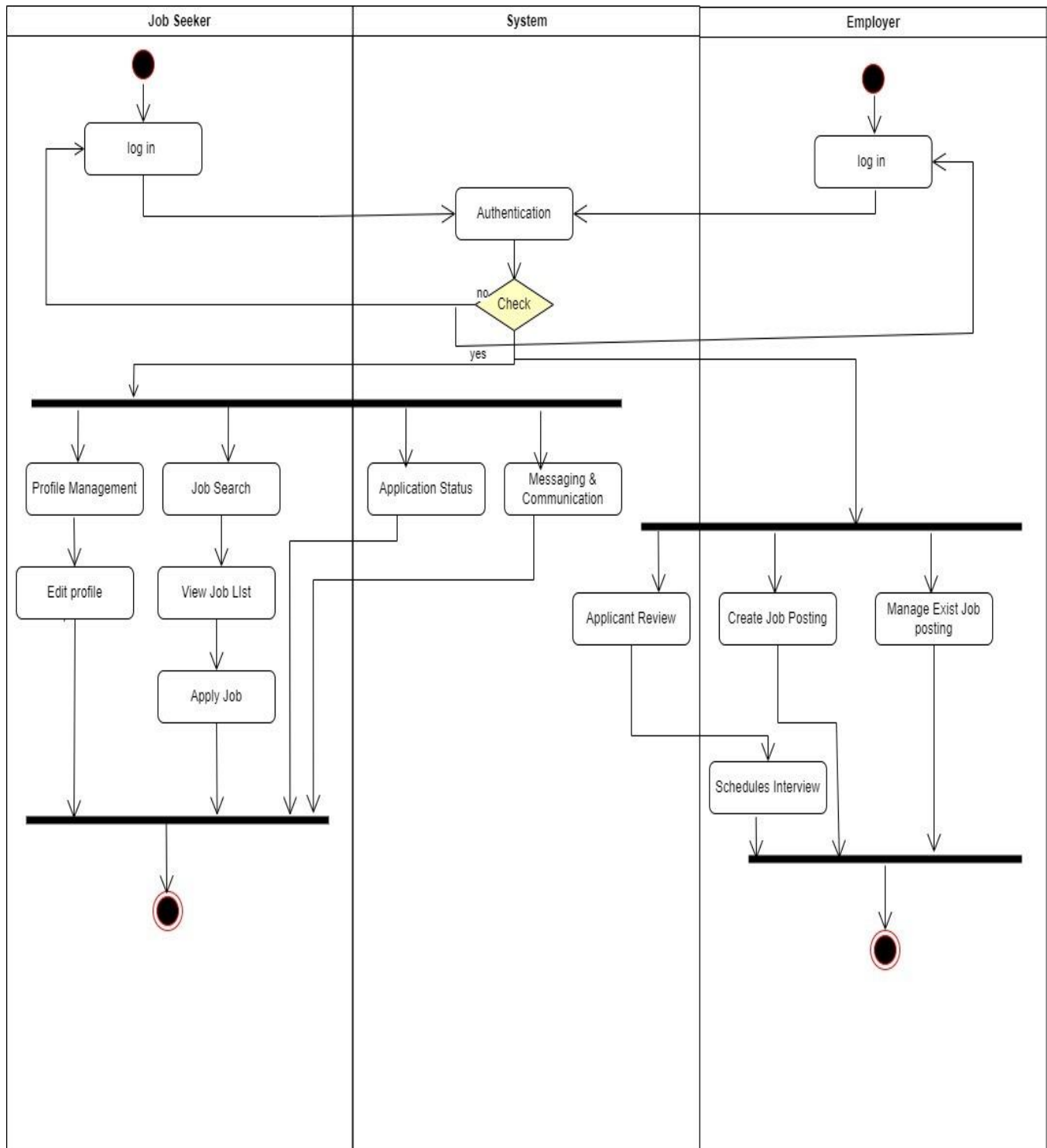
1- job seeker



2- employer



3- job seeker and employer



5. State-transition Diagram

The State Transition Diagram illustrates the different states a system or a component goes through in response to events in the system. As the description doesn't specify a particular feature for state transition, a general representation might include user login, job application submission, and communication states.

the state transition diagram provided is a simplified representation of the interactions between the actors and the system. The diagram illustrates the flow of actions and transitions between different states based on the use case narrative.

