**< Online Job >**

**System Requirements Specification**

**Part-2**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **ID** | **Group** |
| 1 | Donia Yasser Saied | 20201230 | S5 |
| 2 | Sarah Tareq Abd El-Alim | 20201087 | S5 |
| 3 | Reham Ashraf Fathee | 20201079 | S5 |
| 4 | Aziz Samy Aziz | 20200327 | S5 |
| 5 | Mohammed Samy Hmam | 20190442 | S5 |
| 6 | Belal Ahmed Gharib | 20200769 | S5 |

Project contact member email : st473614@gmail.com

Project contact member mobile : 01141723072

TA. <Eng: Ahmed Abd El-Aziz>

1. System architecture for your system

* **Motivation for selecting a Layered Architecture:**

**A layered architecture is a common architectural pattern that divides an application into logical layers, each responsible for a specific set of functionalities. Each layer provides services to the layer above it and interacts only with adjacent layers. This architecture aligns well with the specified non-functional requirements in the following ways:**

1**-Performance:**

**A layered architecture can contribute to performance by providing clear separation of concerns. The separation allows for efficient resource allocation and optimization within each layer. For example, the data access layer can be optimized for database interactions, while the presentation layer can focus on rendering and delivering a smooth user experience. This division of responsibilities within layers helps ensure fast response times and optimal performance.**

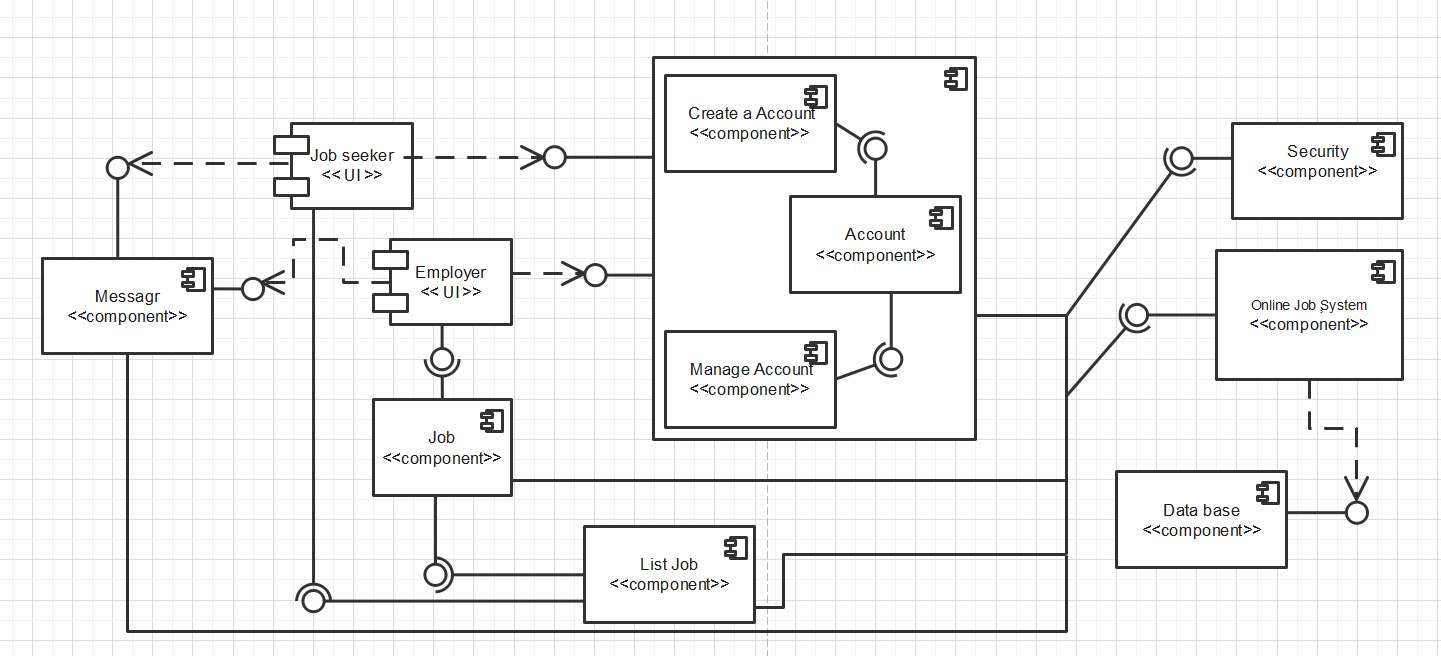
2-**Reliability:**

**Layered architectures promote reliability through encapsulation and modularization. Each layer can be developed, tested, and maintained independently, reducing the risk of introducing errors or causing downtime in other layers. Additionally, the separation of concerns allows for easier fault isolation and troubleshooting. If a specific layer fails or requires maintenance, it can be addressed without affecting the overall system's availability**

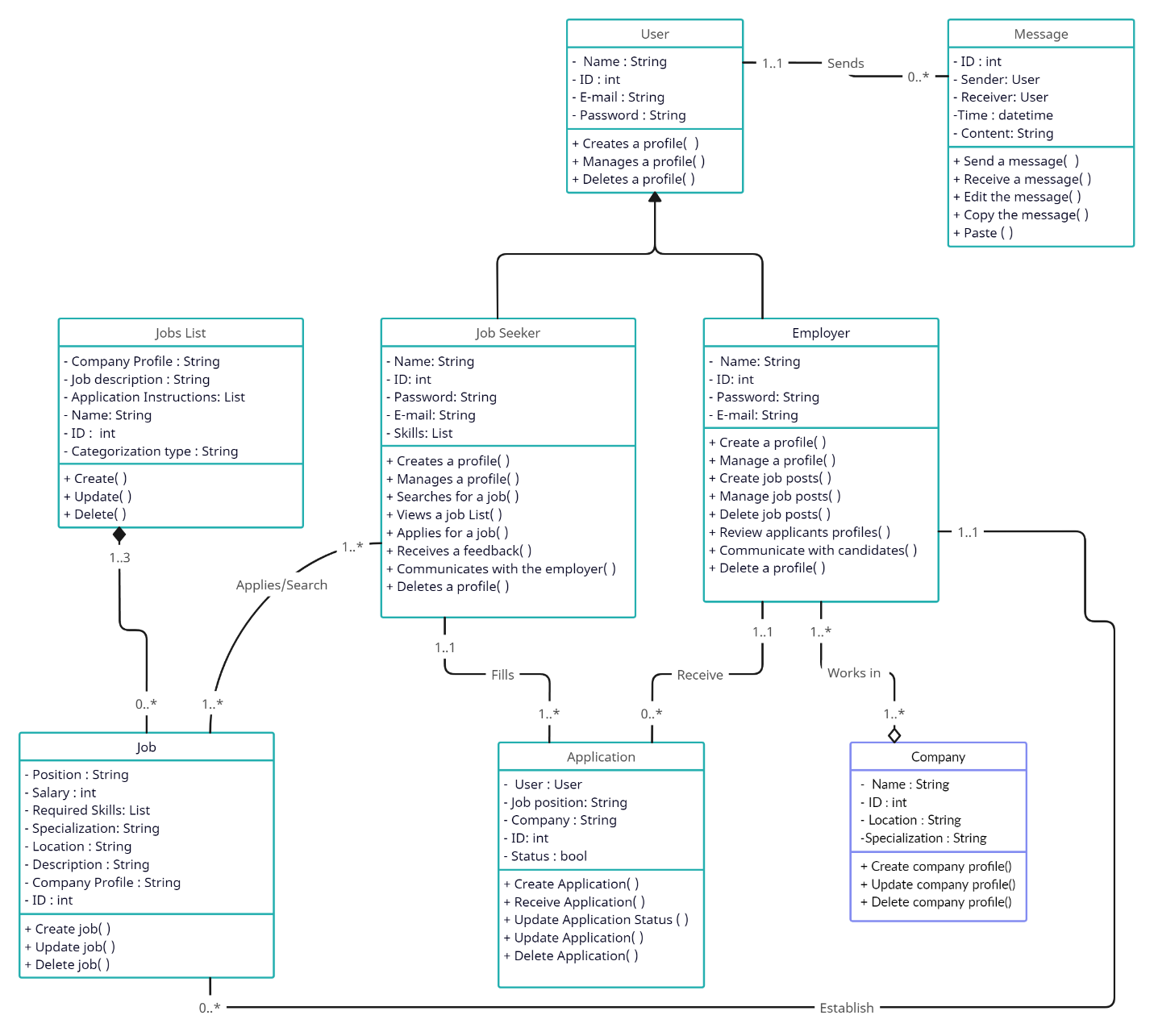
3-**Maintainability:**

**Layered architectures are known for their maintainability. The separation of concerns into distinct layers enables easy maintenance and updates to individual layers without impacting others. Developers can work on specific layers independently, making code changes or enhancements more manageable. This modular approach simplifies testing, debugging, and the introduction of new features or improvements. Mean Time to Repair (MTTR) can be reduced as issues can be localized to specific layers and addressed quickly**

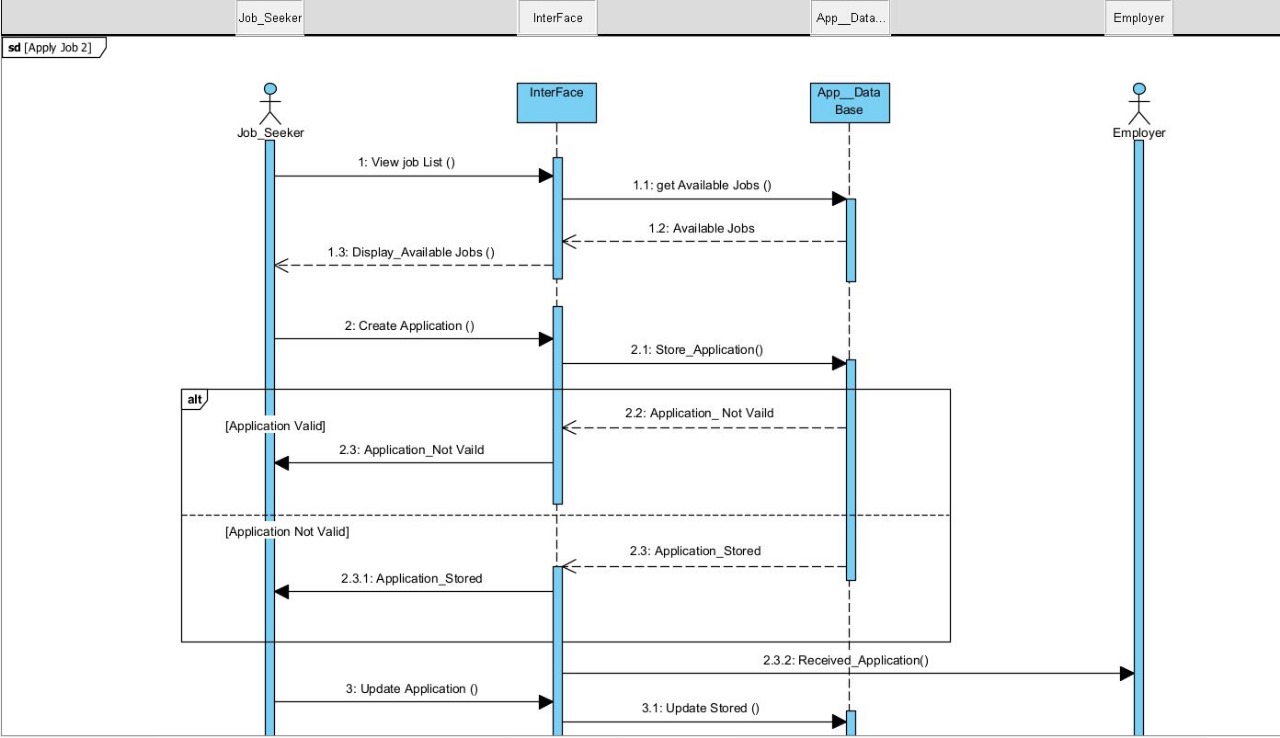
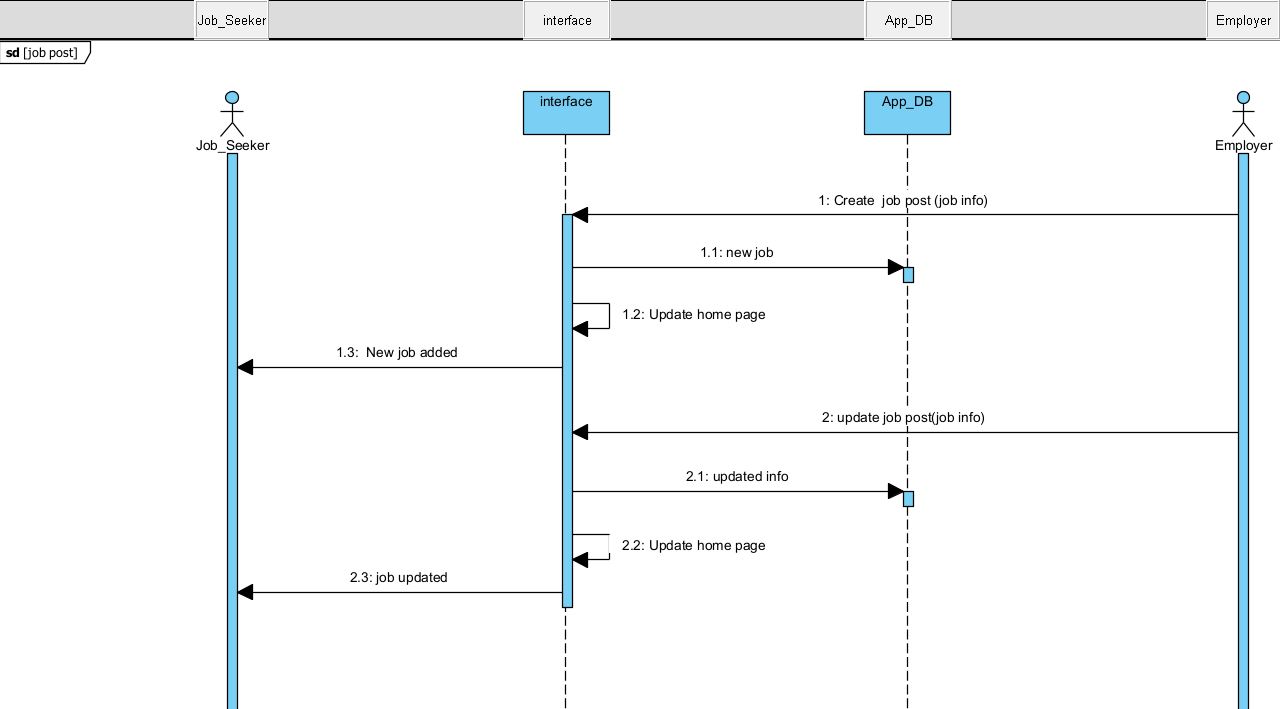
* **component diagram**

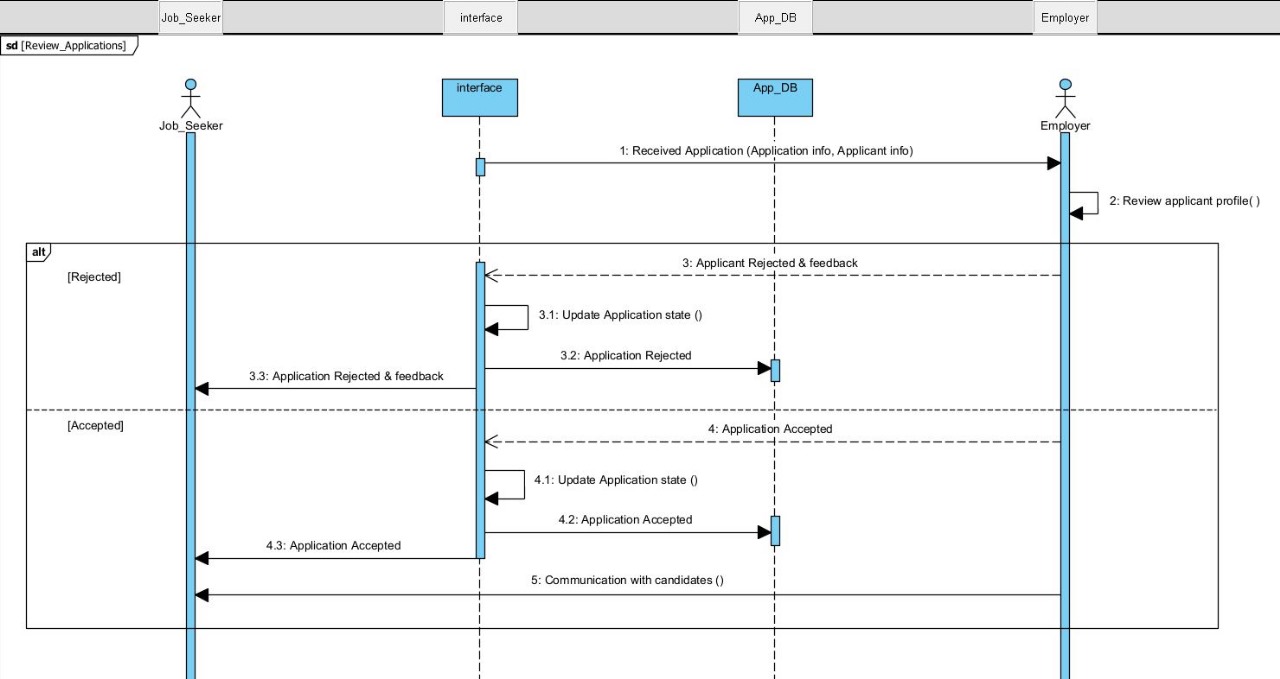
****

1. Class Diagram

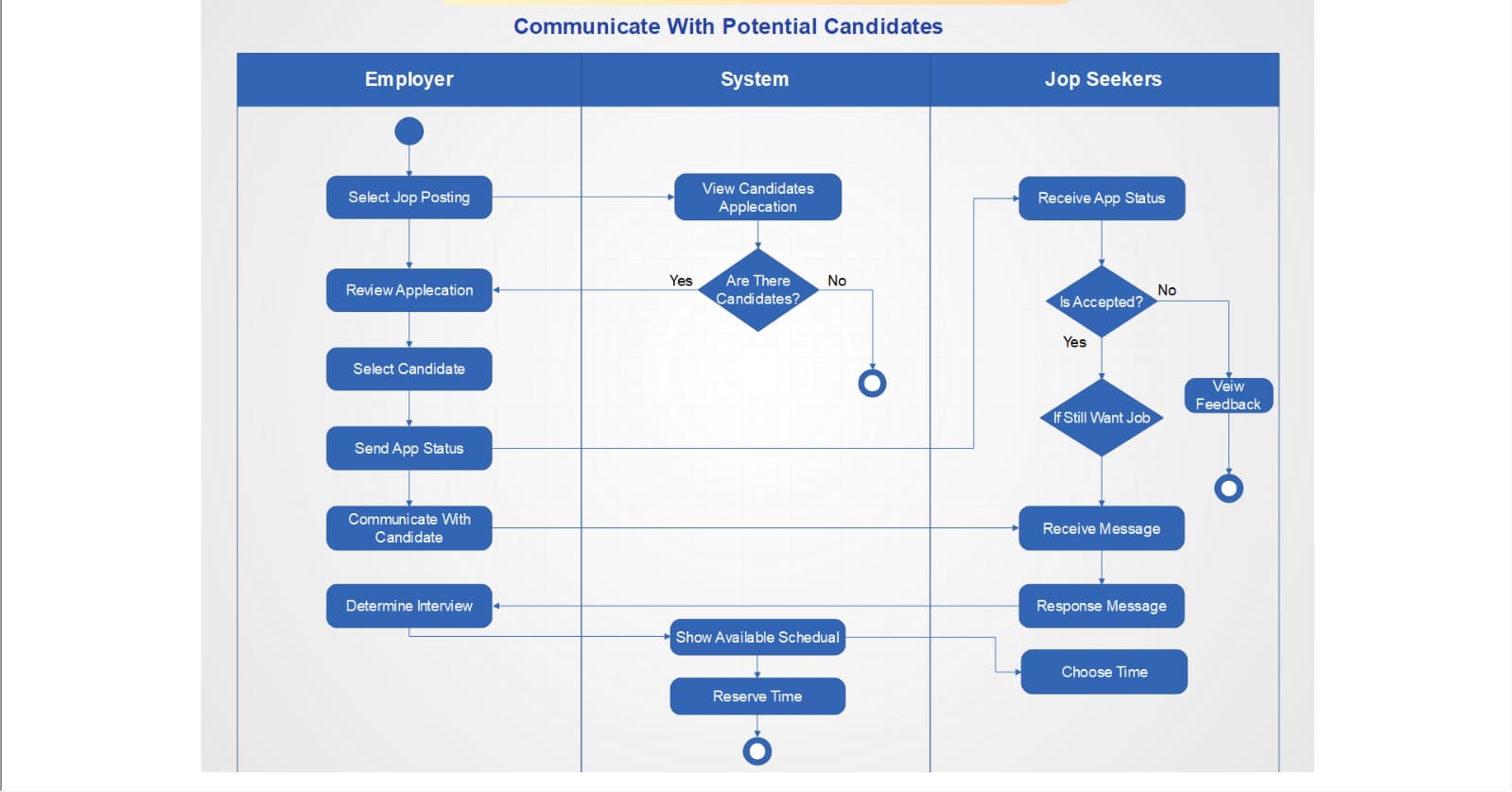
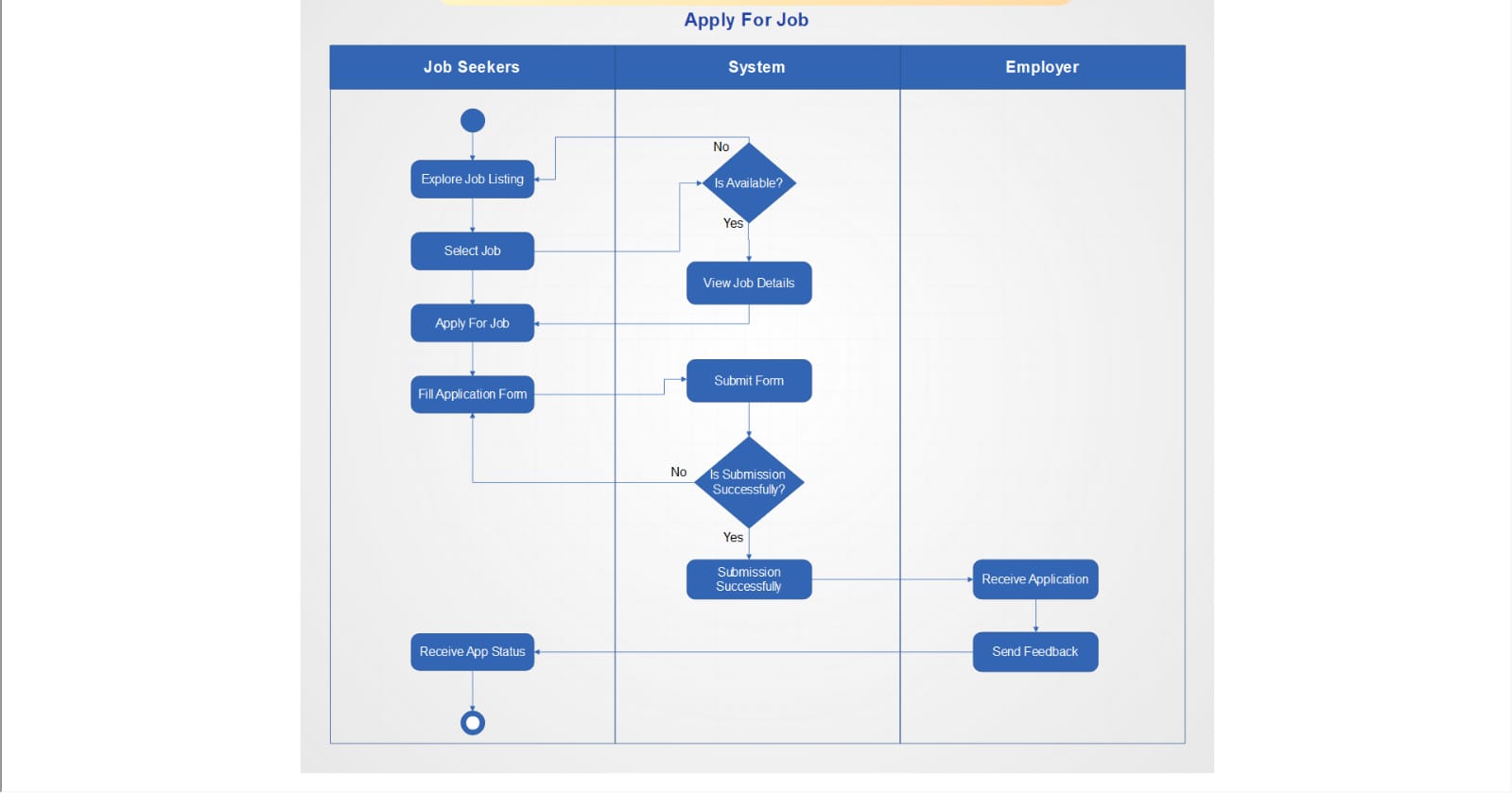
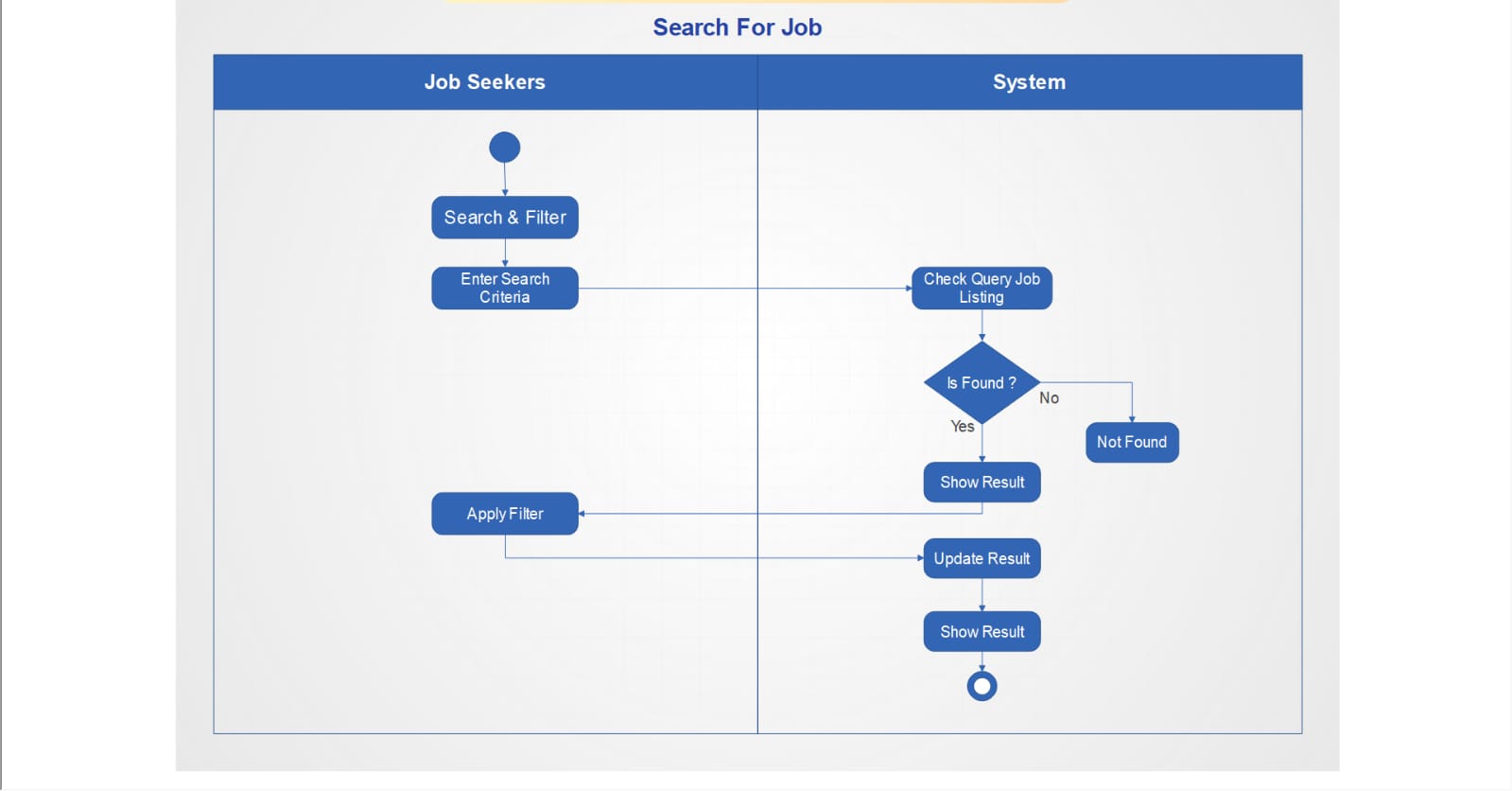
****

1. Revised Sequence Diagram



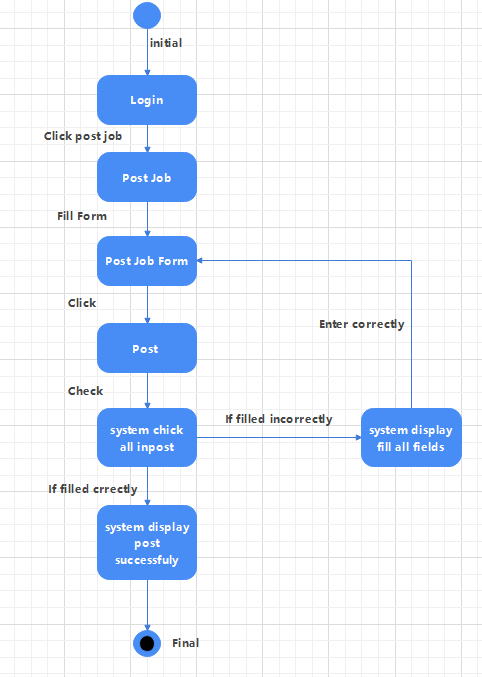


1. Activity Diagram



1. State-transition Diagram

5.1 Post job

****

5.2 Search job

