

→ 2014 ←

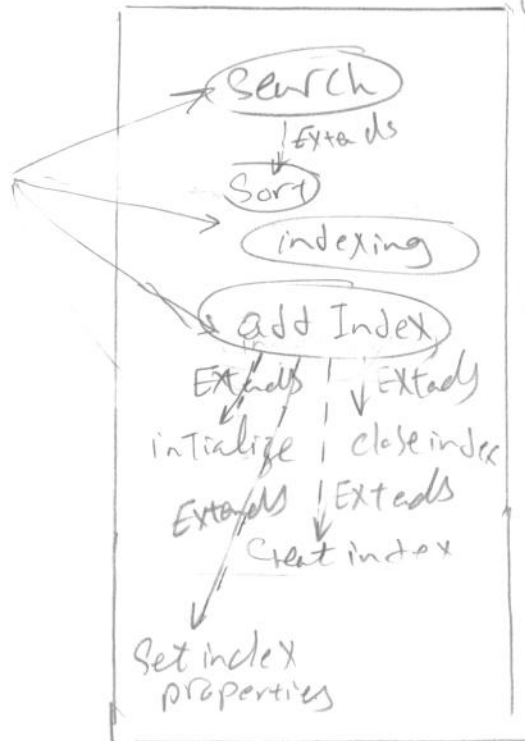
(24) Ziad Ahmed Hendawy

(53) Mohamed Ahmed EL-Rakav

(51) Mohamed Ramadan

(63) Mohamed Youssef EL Gwelly

user



B

There are multiple & different types of users here such as Admin & developer & Client User is the which one of those !!?

Mohamed Saad

Question 2 :-

Structural → iterator :- on result set to reduce time as looping to access data efficiently with uniform way

creational → Factory :- To make it simpler by minimizing classes

→ Singleton :- in Index writer as only one object is created in system

→ pool :- in Index searcher as there is limited number of created instances so we reuse created instance rather than creating new one

1- behavioral design pattern :-

→ State :- as we have two ways of sorting which must be implemented in the system

Strategy is better here
Mohamed Saad

Question 3

iterator

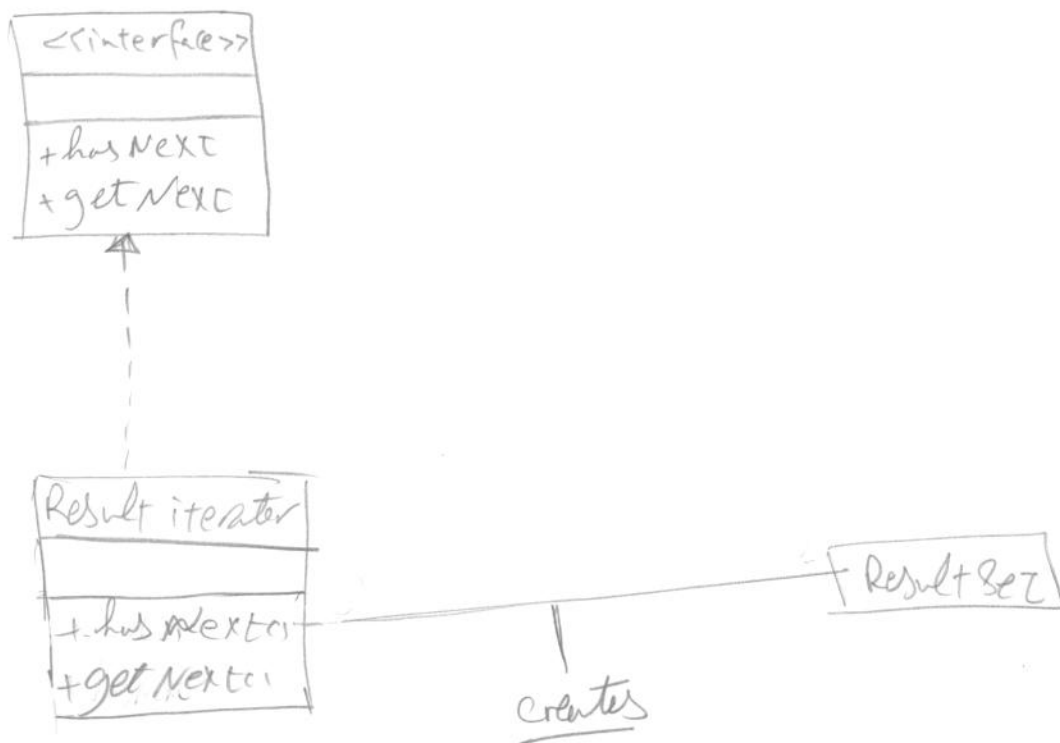
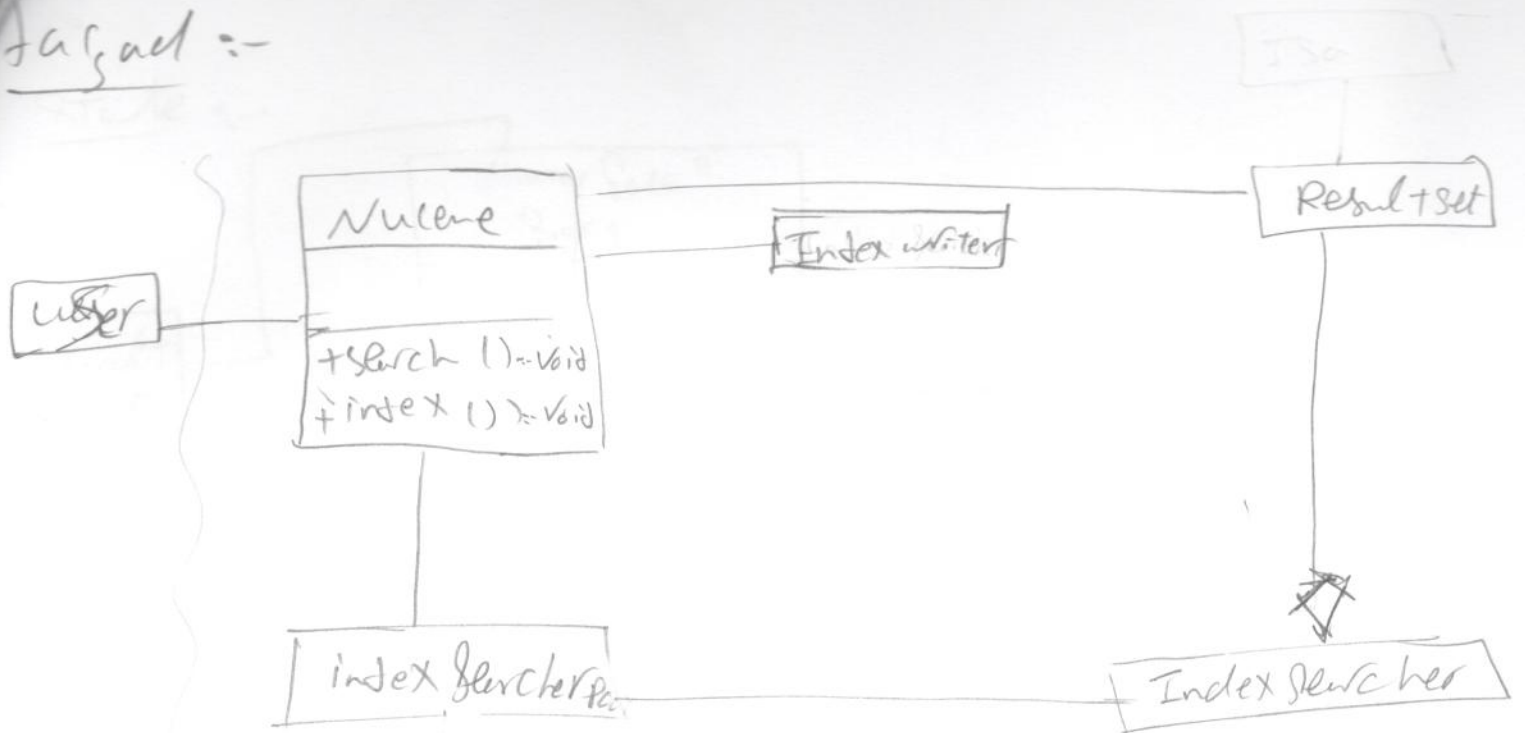
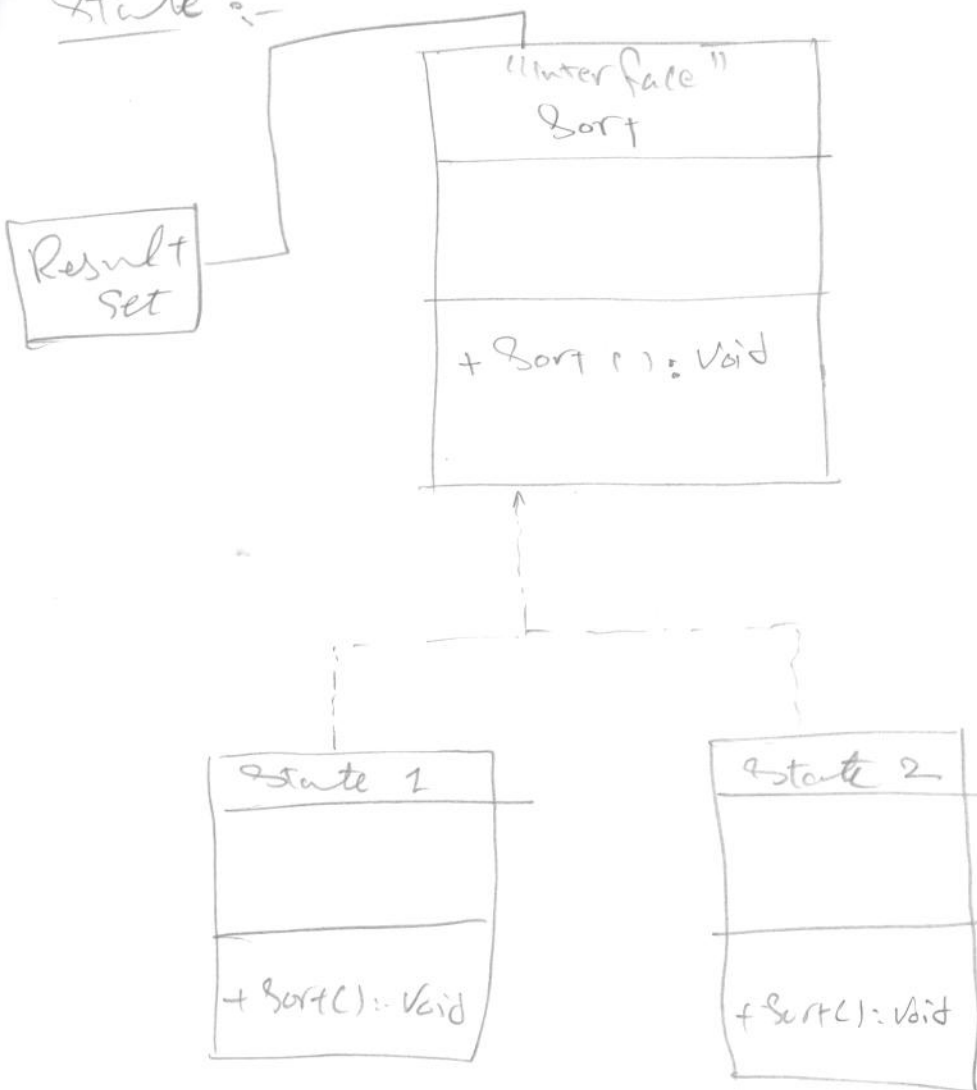


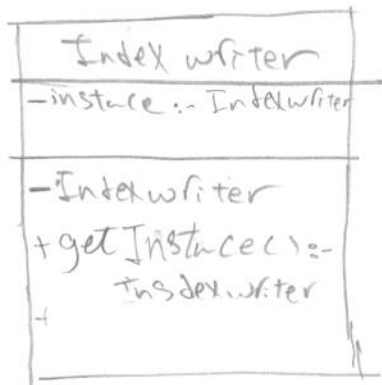
Diagram :-



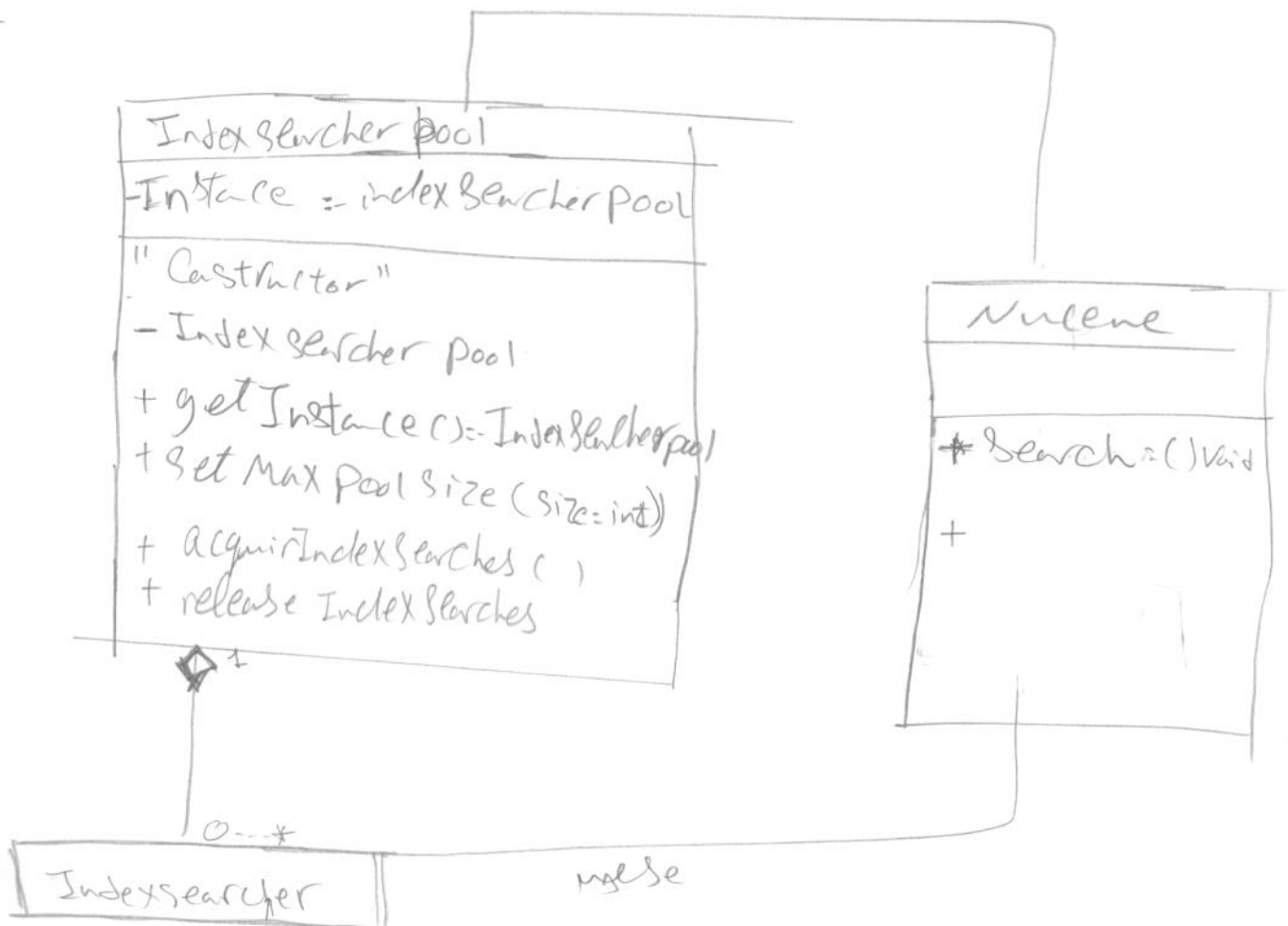
State :-



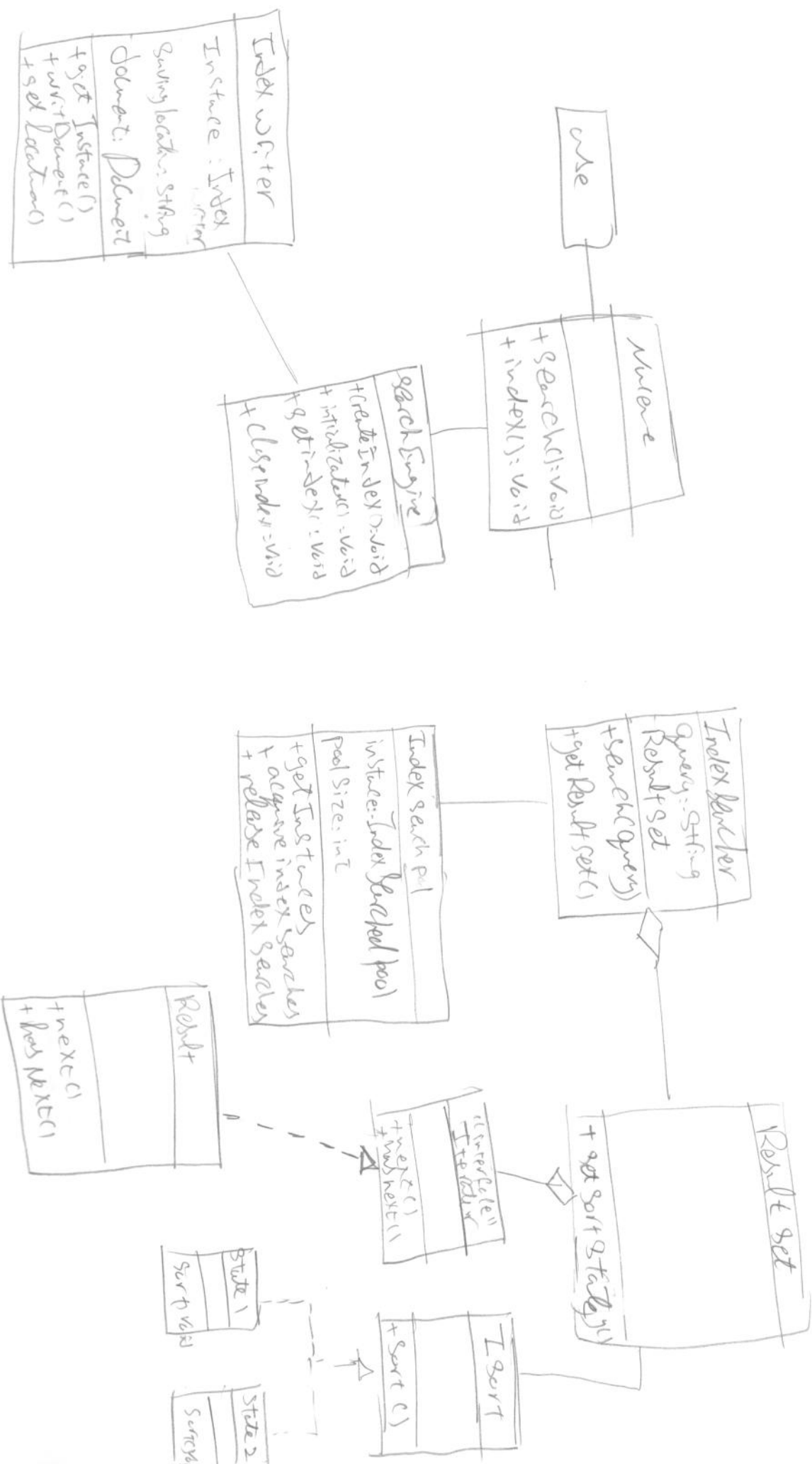
Singleton



Pool :-



Question Four



Q5, Q6, Q7

We need Sequence diagram in Q5
Mohand Saad

```
public class UI implements Observer {
```

```
    public void update (Observable obs, Object o)
    {
```

```
        if (obs instanceof IndexSearcherPool) {
```

```
            if (obs.poolSize() >= 99) {
```

```
                button.setColor (Color.RED);
```

```
                return;
```

```
            }
```

```
        } else {
```

```
            if (obs.getDocumentNumber() >= 1000000) {
```

```
                button.setColor (Color.RED);
```

```
                return;
```

```
            }
```

```
        }
```

```
        button.setColor (Color.GREEN);
```

```
    }
```

```
}
```

```
public class IndexSearcherPool extends Observable {
```

```
    private int poolSize = 25;
```

```
    private static instance = null;
```

```
    private IndexSearcherPool() { }
```

```
    public static IndexSearcherPool getInstance() {
```

```
        if (instance == null)
```

```
            return new IndexSearcherPool();
```

```
        else {
```

```
            instance = new IndexSearcherPool();
```

```
            return instance;
```

```
        }
```



```

public IndexSearcher acquireIndexSearcher() {
    notify();
    if (haveReusableObjects()) {
        return getReusableObject();
    } else {
        poolSize++;
        return new IndexSearcher();
    }
}

```

