North South University

Department of Electrical and Computer Engineering

PROJECT REPORT

CSE323
Operating System Design
Section- 5

Deadlock Solution

Group Members:

Akmam Hasan	**-- 1911192042**
Rafi Majid	**-- 1922111642**
Kazi Rabiul Alam	**-- 1921335042**

Date of Submission: 12.01.2022

Deadlock

Deadlock occurs in operating system when a process or thread enters a waiting state. Deadlock occurs because requested system resource is held by another waiting process, which in turn is waiting for another resource, held by another waiting process.

Banker’s Algorithm

Banker’s Algorithm is resource allocation and deadlock avoidance algorithm which test all the request made by processes for resources, it checks for the safe state, if after granting request system remains in the safe state it allows the request and if there is no safe state it doesn’t allow the request made by the process.

Resource Allocation Graph

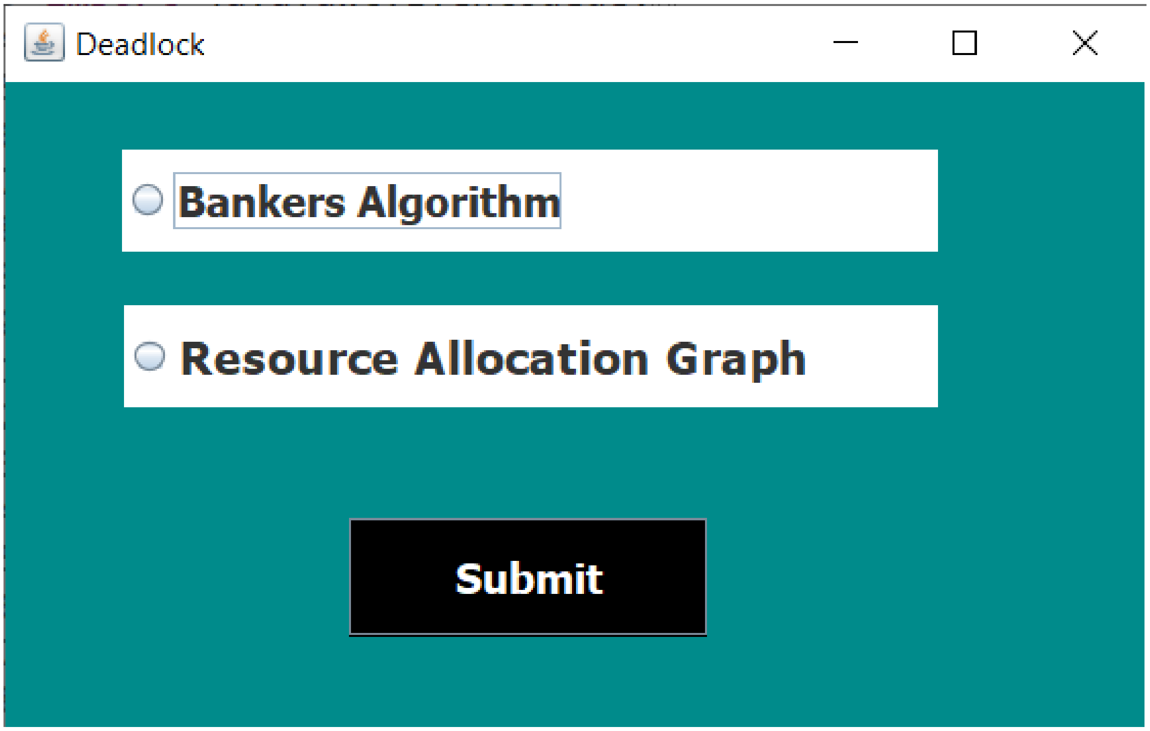
The resource allocation graph is the pictorial representation of the state of a system.

About Project

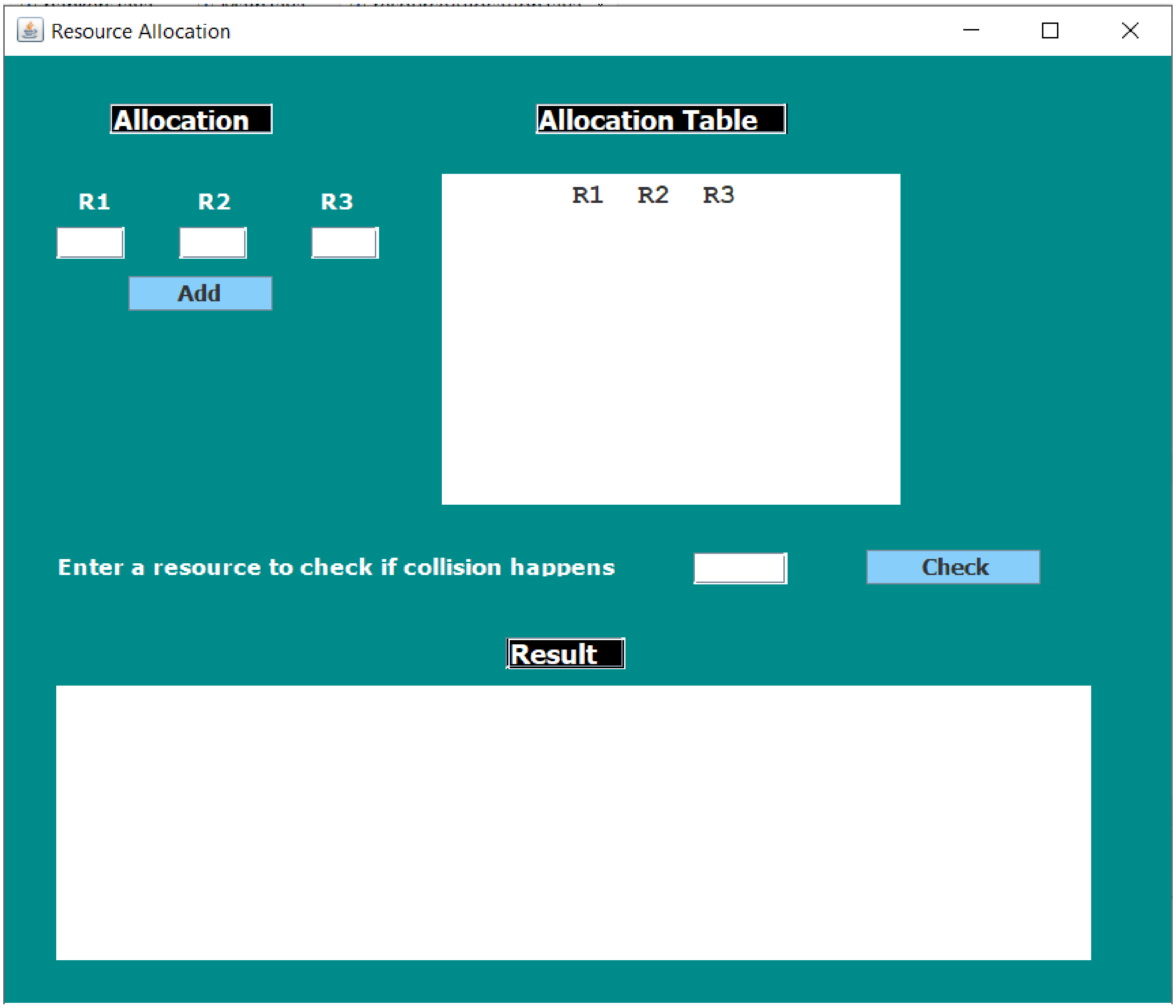
In this project, we used Banker’s Algorithm and Resource Allocation Graph for deadlock avoidance. Banker’s Algorithm is resource allocation and deadlock avoidance algorithm. We have used Java programming language for the development. For GUI implementation we have used Swing which also based on Java. This project contains 3 java classes, for home window, Banker’s algorithm and RAG.

We also have created two separate interfaces, one for Banker’s algorithm and another for RAG. For interface, we have used JAVA Swing.

When we execute the program, we would be able to see a interface. In the interface there are two option to select the algorithm. By selecting each we would be able to see the functionality of each program.

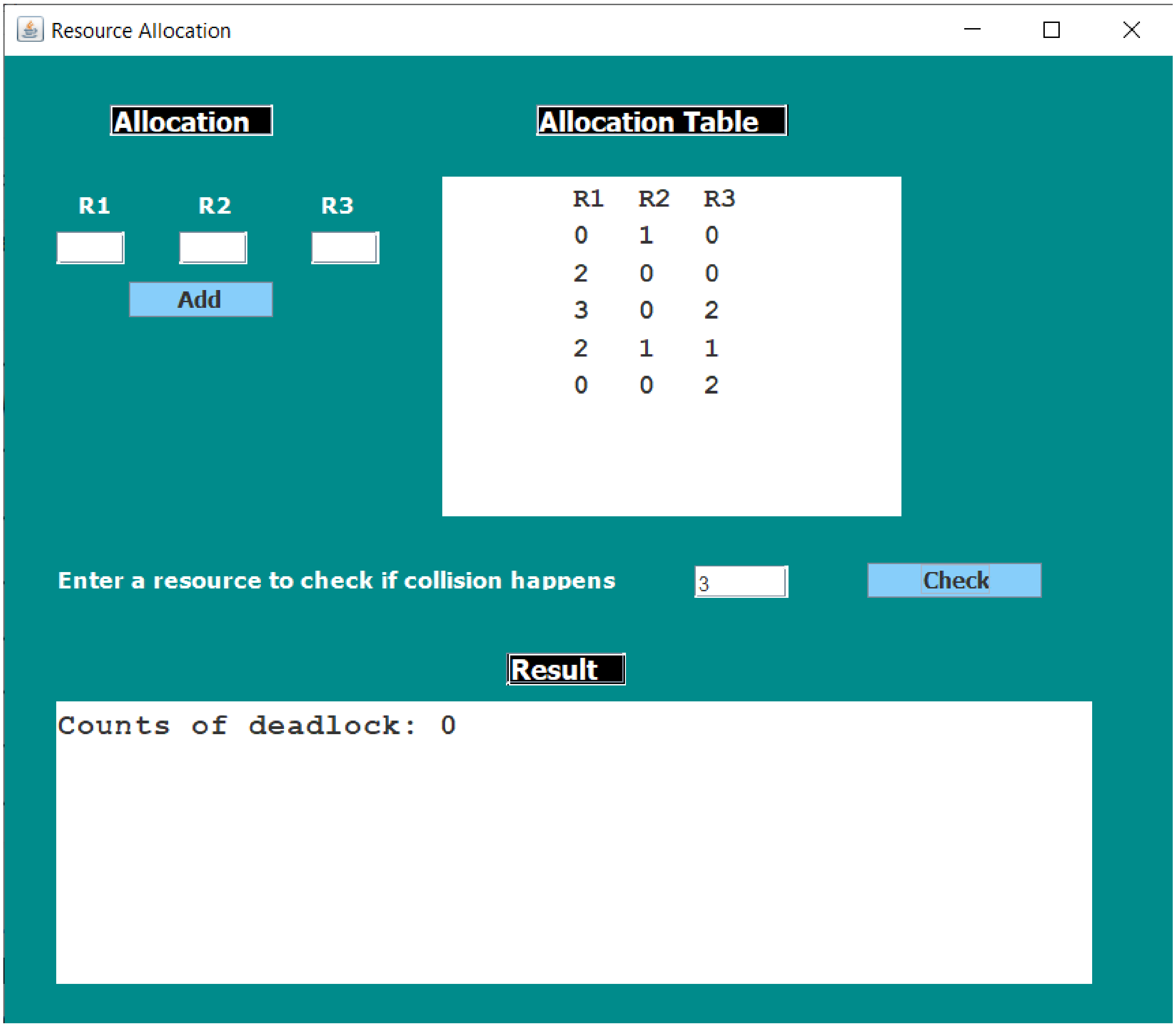


If we select Resource Allocation Graph, then this window will open.



We can allocate resources and also enter a resource to check for collusion. The test area allocation table will show the allocated resources. After clicking on check button for collusion, the text area of Result will show the count of deadlocks.

One example is shown below:

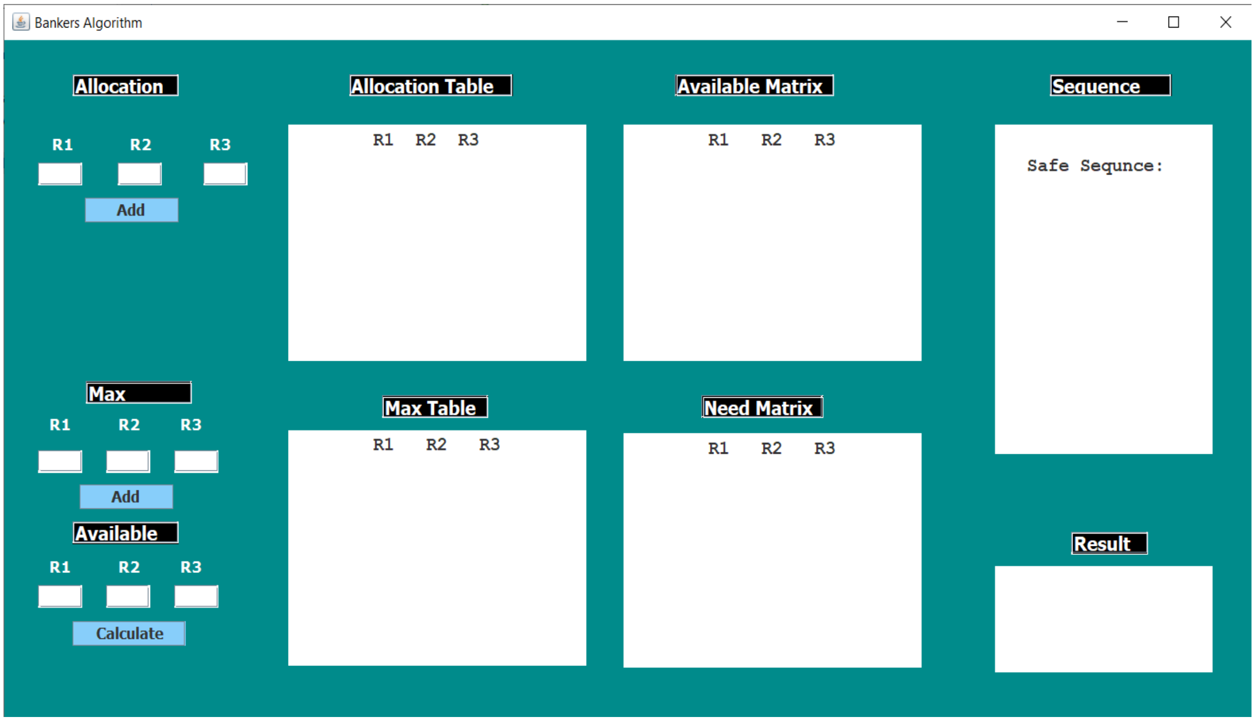


R1	R2	R3
0	1	0
2	0	0
3	0	2
2	1	1
0	0	2

Enter a resource to check if collision happens 3

```
Counts of deadlock: 0
```

After selecting Banker’s Algorithm, this window will open.

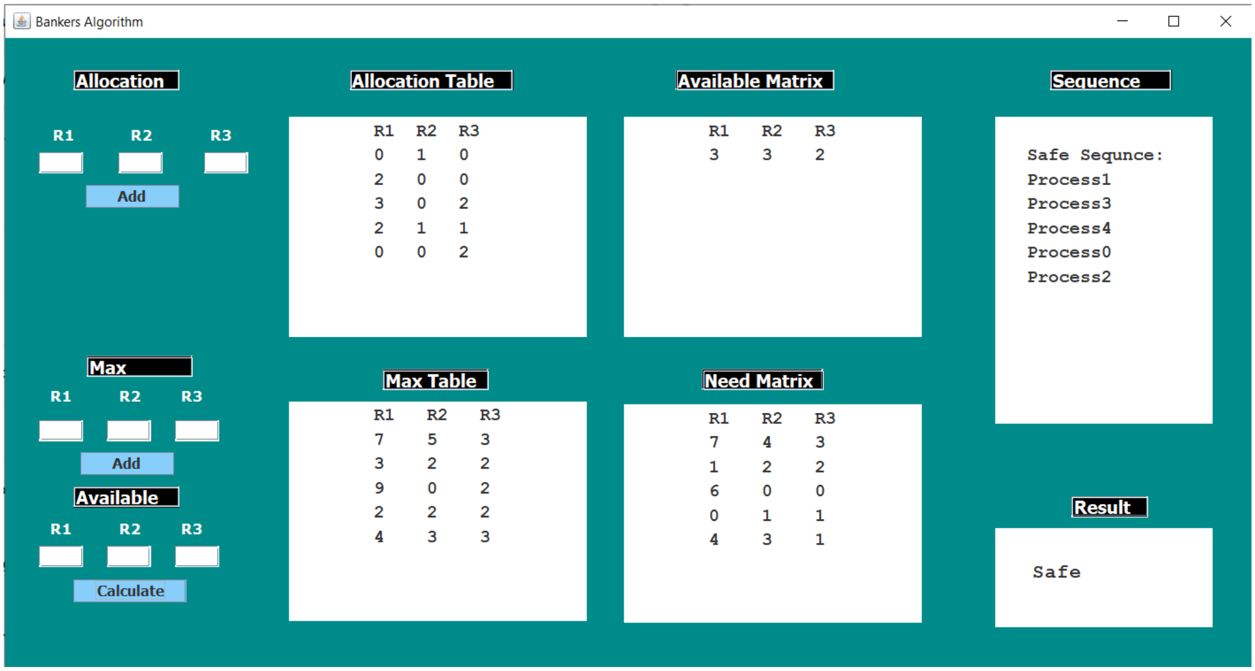


```
Safe Sequnce:
```

We have to add values of resources of each process separately which will be displayed on text area titled “Allocation table”. Then we need to provide the max values of resources which will be display on “Max table”. After giving the values of available resource we need to click on “Calculate” to see the safe sequence and result. After clicking on “Calculate” we will be seeing the value of available resources on the text area titled” Available Matrix” and the need values on text area titled “Need Matrix”. The calculated safe sequence would be displayed on text area titled “Sequence” and the result would be showed on text are titled “Result”.

It will show the safe sequence and detect whether there is any safe sequence or not.

An example is shown below:



Allocation Table

```
R1  R2  R3
0   1   0
2   0   0
3   0   2
2   1   1
0   0   2
```

Available Matrix

```
R1   R2   R3
3    3    2
```

Sequence

```
Safe Sequnce:
Process1
Process3
Process4
Process0
Process2
```

Max Table

```
R1   R2   R3
7    5    3
3    2    2
9    0    2
2    2    2
4    3    3
```

Need Matrix

```
R1   R2   R3
7    4    3
1    2    2
6    0    0
0    1    1
4    3    1
```

Result

```
Safe
```