1. **Project Information**

**Project Title: BANKER’S ALGORITHM PROJECT**

**Project Group Number: 20**

**Names / ID’s of members: D Venkata Sai Sri Hari (014533571)**

**Rakshitha Sathyakumar (014511705)**

1. **Abstract:**

Our project is about the banker’s algorithm. It is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for predetermined maximum possible amounts of all resources, then makes an “s-state” check to test for possible activities, before deciding whether allocation should be allowed to continue. It was developed by Edsger Dijkstra. It is named so because this algorithm is used in banking systems to determine whether a loan can be granted or not. The algorithm was developed in the design process for the operating system and originally described (in Dutch) in EWD108. The name is by analogy with the way that bankers account for liquidity constraints. It’s called the Banker’s algorithm because it could be used in the banking system so that banks never run out of resources and always stay in a safe state.

The Banker's algorithm is run by the operating system whenever a process requests resource. The algorithm avoids deadlock by denying or postponing the request if it determines that accepting the request could put the system in an unsafe state (one where deadlock could occur). When a new process enters a system, it must declare the maximum number of instances of each resource type that may not exceed the total number of resources in the system. Also, when a process gets all its requested resources it must return them in a finite amount of time.

For the Banker's algorithm to work, it needs to know three things:

* How much of each resource each process could possibly request?
* How much of each resource each process is currently holding?
* How much of each resource the system currently has available?

Resources may be allocated to a process only if it satisfies the following conditions:

* Request ≤ max, else set error condition as process has crossed maximum claim made by it.
* Request ≤ available, else process waits until resources are available.

Some of the resources that are tracked in real systems are memory, semaphores and interface access. It derives its name from the fact that this algorithm could be used in a banking system to ensure that the bank does not run out of resources, because the bank would never allocate its money in such a way that it can no longer satisfy the needs of all its customers. By using this algorithm, the bank ensures that when customers request money the bank never leaves a safe state. If the customer's request does not cause the bank to leave a safe state, the cash will be allocated; otherwise the customer must wait until some other customer deposits enough.

1. **Weekly Schedule:**(Need to modify based on our midterms and schedule)

**Week 1:** Studying background and objectives

**Week 2:** Approach and Methodology

**Week 3:** Implementation

**Week 4:** Finding and analysis

**Week 5:** Report and preparation