

**Ahsanullah University of Science & Technolog Department of Computer Science & Engineering**

**Course No: CSE4126**

**Course Title: Distributed Database System Lab**

**Date of Submission: 06/09/22**

**Submitted To: G. M. Shahariar and Sanjana Karim Lora**

**Project name:**

Vaccine distribution system

**Submitted By:**

Name: Md. Atiqur Rahman

ID: 180204024

**Group member:**

Name: Sanjid Islam Chowdhury

ID: 180204005

Name: Ahmed Saad Tanim

ID: 180204015

**Project Description**

Vaccine is the only secured solution to this dire pandemic situation. Proper distribution of such limited stock is absolutely necessary in terms of distributing vaccine in our country. Our project is to establish a system for our country in this solution. We want to make a vaccine management system that tracks the distribution of vaccines across the citizens under all kinds of occupations and age and covid history and set priorities accordingly.

Vaccine distribution System is a project where users can register for vaccine like [Surokkha](https://surokkha.gov.bd/) web app. After successful registration of a user, that user is assigned a random center within his/her city, and a vaccine that is currently available in that center.

After assigning a vaccine to a user, the amount of vaccines is reduced from that center's database and from the server. This process is autonomous. Therefore, global transactions happen. If there is a shortage of vaccines in a particular center, that center can request more vaccines from the vaccine mass storage table which is located on the server. If the vaccine mass storage table has no vaccine to provide, it throws a custom exception with an appropriate message.

There is an option to add vaccines to the mass storage table manually from the server. The site has no permission to do anything to the server.

A global view is also present in this project to show all the information related to vaccines.

**Reference Architecture of DDB**

**Global Schema**

users(nid, cid, name, city, vid)

vaccine\_record(cid, brand)

vaccine\_center(cid, city, cname)

vaccine\_center\_storage(cid, vid, amount)

vaccine\_mass\_storage(vid, amount)

**Fragmentation Schema**

|  |  |
| --- | --- |
| Horizontal Fragmentation | users1 = SLcity = ‘Dhaka’ users  users2 = SLcity = ‘Sylhet’ users |
| vaccine\_center1 = SLcity = ‘Dhaka’ vaccine\_center  vaccine\_center2 = SLcity = ‘Sylhet’ vaccine\_center |
| Derived Fragmentation | vaccine\_center\_storage1 = vaccine\_center\_storage NSJN PJcid vaccine\_center1  vaccine\_center\_storage2 = vaccine\_center\_storage NSJN PJcid vaccine\_center2 |

**Fragmentation Tree**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Allocation Schema**

user1, vaccine\_center1, vaccine\_center\_storage1 at **dhaka\_site**

user2, vaccine\_center2, vaccine\_center\_storage2 at **sylhet\_site**

**Allocation Types of Fragments**

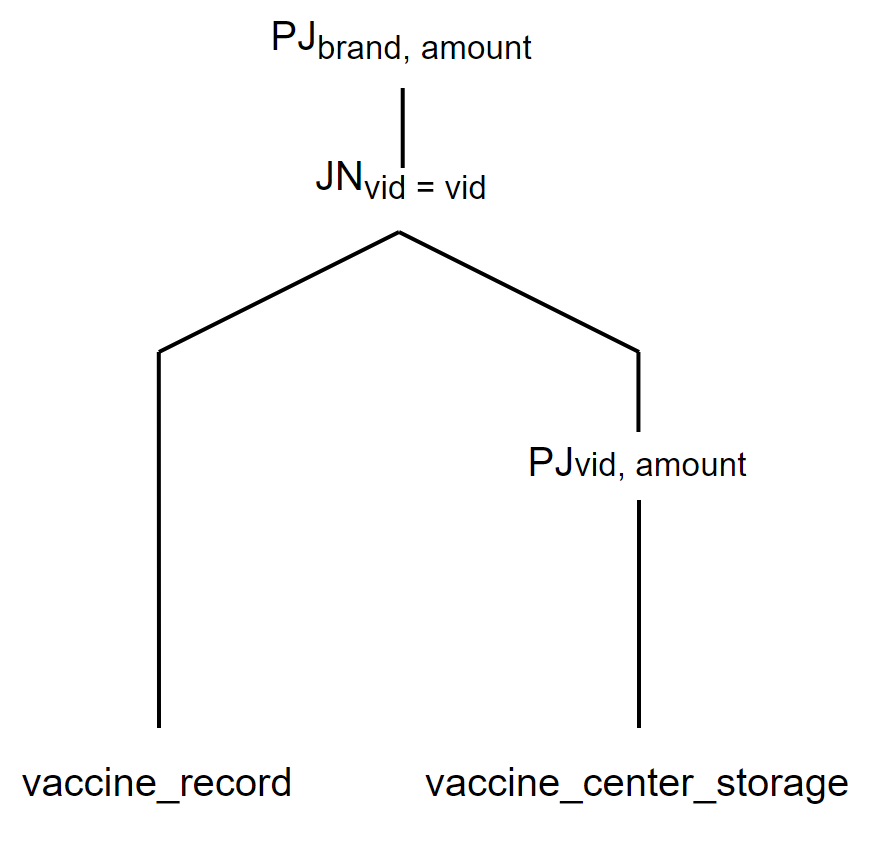
There are two types of allocation, which are:

* Non redundant allocation (one copy one site)
* Redundant allocation (multiple copy in sites)

Our project instilled non-redundant allocation. In other words, one copy of each data is stored in one site only.

**Operator Tree**

PJbrand, amount(vaccine\_center\_storage JNvid = vid vaccine\_record)

****

**Contribution**

I start the project by creating two sites and one server. Those sites are Dhaka and Sylhet. On sites, I added a procedure to register users and created the logic for assigning random vaccine no and random vaccine center. I also created a trigger for inserting user info into the server after successful registration, which is modified later. I also handled exceptions if a registered user registers again by using EXCEPTION. A did the same thing to another site because the functionality is the same in the two sites. So I modified the code accordingly.

I made some tables vaccine\_record, users, and vaccine\_center, and inserted some dummy data for testing purposes.

Initially, when I first created the server, I made a procedure register\_user. By invoking this procedure, I added the locally registered user to the global table to keep a record of all the users. There is a bug inside center\_display in the server which I fixed. I made all the diagrams that have been attached to the report.

**Conclusion**

The vaccine distribution system is a continuation of one of the previous projects that we uploaded into distributes database system from the central database system. We have ensured the usage of topics from both lab and theory, covering distribution transparency, operator trees, views, procedures, conditional statements, cursors, custom exception, fragmentation, and lastly complete reference architecture of DDB.