SAMANTA DEY (MDS202032)

DCBD Final Exam 2020-2022

Ans:1. (1) Here, N= 4, M=2

Given: Atleast one muddy child.
The children don't know M's value.

Let d denote the set of children with mud [#d=2]
c denote " " " " without mud. [#c=2].

ROUND 1 :-

All of d and c say "No" since they don't know if they are muddy. This is because, none of them knows the no. of muddy foreheads.

ROUND 2 :-

- Inference after round 1: # muddy foreheads = >1. [This is because, if there was exactly one muddy forehead, the child with muddy forehead would see no other muddy forehead and could answer "Yes".
- The children in d set answer "Yes" [they know if they are muddy]
- The children in c answer "No" [they don't yet know if they are muddy]

Reason:- The children in d know that there are >1 muddy foreheads and each of the children in d can see only other child in d's muddy forehead.

The children in c can see two muddy foreheads (of those in d) and cannot conclude anything about theire own.

ROUND 3 :-

- Inference from Round 2 :- # muddy foreheads = 2. [since, both the children in d know they are muddy].
- The children in d answer "Yes"
- The children in c answer "Yes"

[∵ all of them know whether each of them have mud on their own forehead]

Thus, after Round 3, all responses converge.

Ans:1 (2) $N=4$, $M>0$, $M=$ even

Notations and symbols: same as 1.(1)

~~▨ M=2~~

~~ROUND 1:-~~

~~All of d and c say "No" since none of H~~

▨ M=2 :-

Same as Ans:1(1).

▨ M=4 :-

Round 1:-

All of d and c say "No" since all see 3 muddy foreheads.

Round 2:-

- Inference from Round 1:- # muddy foreheads > 1 [same as 1.(1)].
- All of d and c say "No" since they are not yet able to figure out if they are muddy because all of them see 3 muddy foreheads, and the total no. of foreheads with mud is 3 or 4.

Round 3:-

- Inference from Round 2:- # muddy foreheads > 2 because o.w children in d would have said "yes".
- Still none of d and c can answer since they are not properly able to figure out if they themselves have mud on their foreheads. All say "No"

Round 4:-

- Inference from Round 3:- # muddy foreheads > 3.
- All of d and c answer "Yes" since they are now sure that all of them have mud on their foreheads.

∴ Responses converge after Round 4.

Ans:2. Our RESTful web service will mainly do the following two tasks :—

- create a new registration for an user who had completed one dose of vaccination or has not taken any dose of vaccination
- return the vaccine information for the users who have administered a single dose of vaccine [information about the type/vial].

Two resources :-

- ~~Name~~ Person
- ~~Conta~~ Vaccine

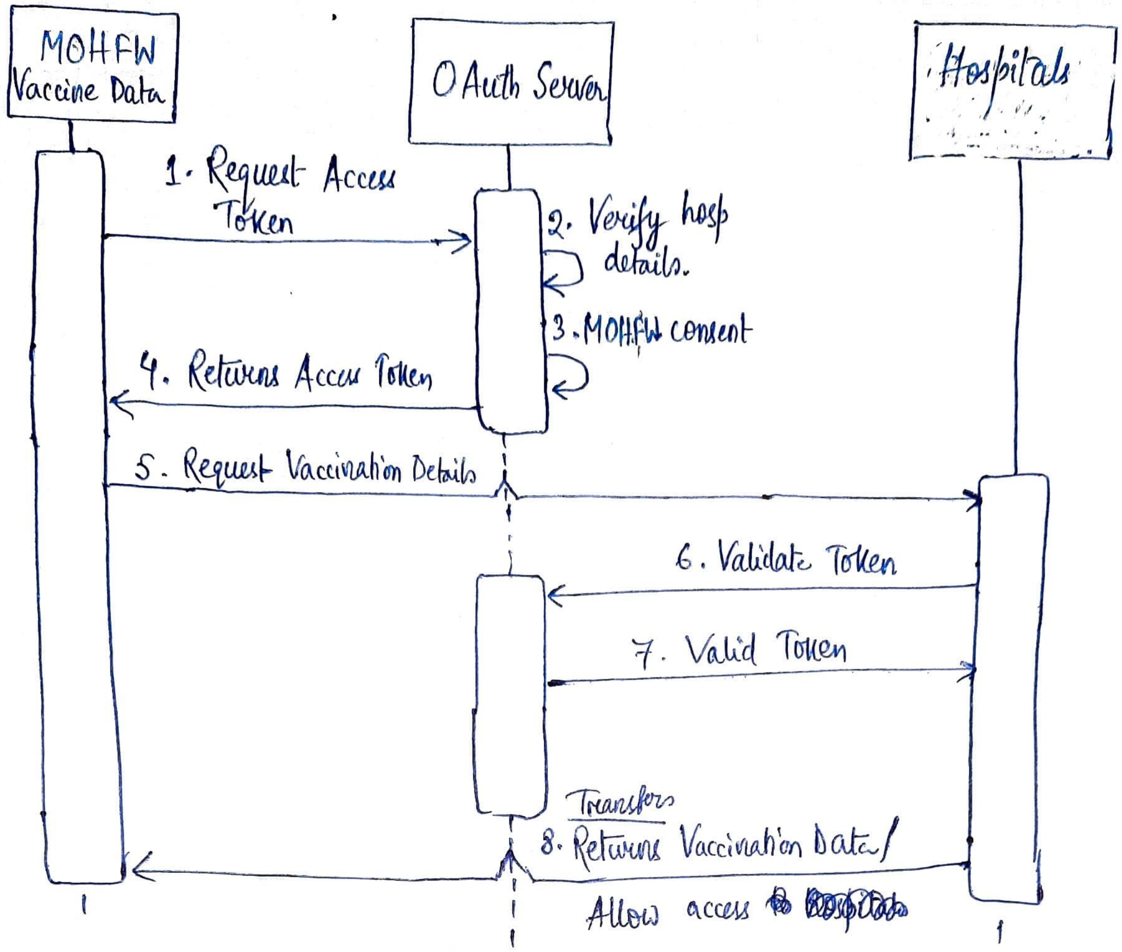
Design of the RESTful API :-

- Object Model :—
  - Person, Vaccine
- Creating Model URIs :—
  - /person/{pname}
  - /person/{pname}/vaccine
  - /person/{p-name}/vaccine/{vaccine-name}
- Determining representations :—
  - Represent all person's information as an XML/JSON
  - Represent all vaccine names as an XML/JSON.
- Assigning HTTP Methods :—
  - create registration for users who are new or completed one dose → HTTP POST
  - return information about type of vaccine taken → HTTP GET

Here, 'POST' is non-idempotent

'GET' is idempotent.

Ans: 3.



When a particular hospital needs to transfer ~~access~~ the resources on vaccination data, through a secure method, the OAuth Server asks the hospitals to login for a secure transaction/transfer of data and after verifying the hospital details, when the ~~hospital~~ MOHFW requests access to the vaccine data, ~~t~~the hospitals validates the token and returns the data and allows access to the data for ~~the hospitals~~ MOHFW.

This has been represented using a sequence diagram as above.

Ans: 4. Class Diagram for MOHFW scenario :—

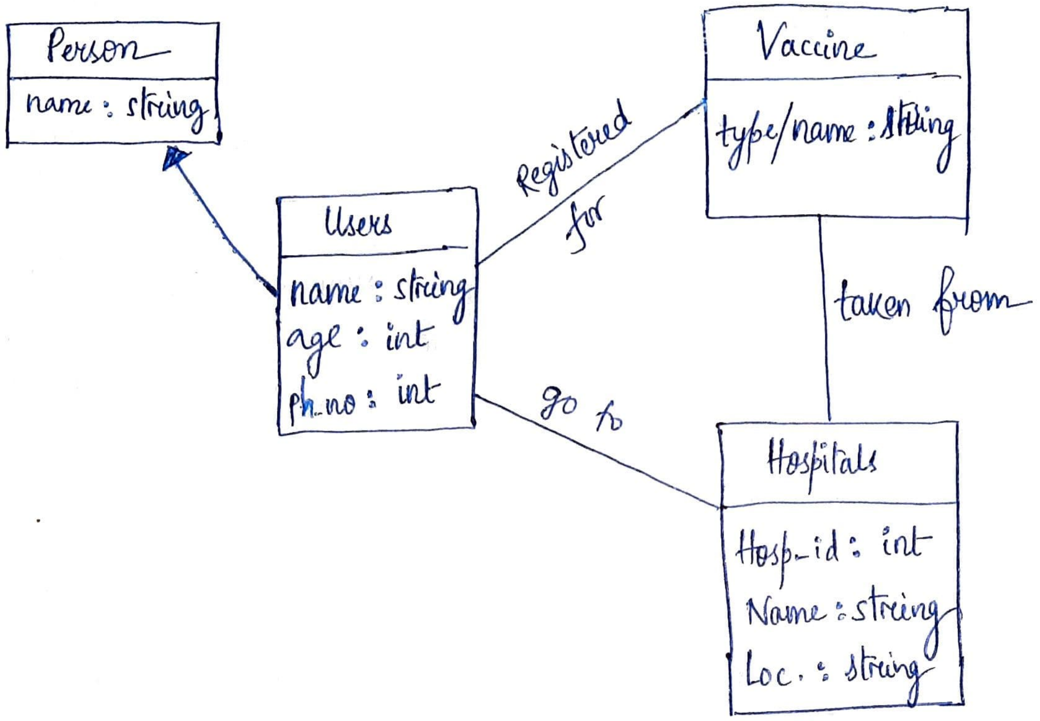


Fig: Class Diagram to demonstrate vaccination situation

Here, the classes are :—

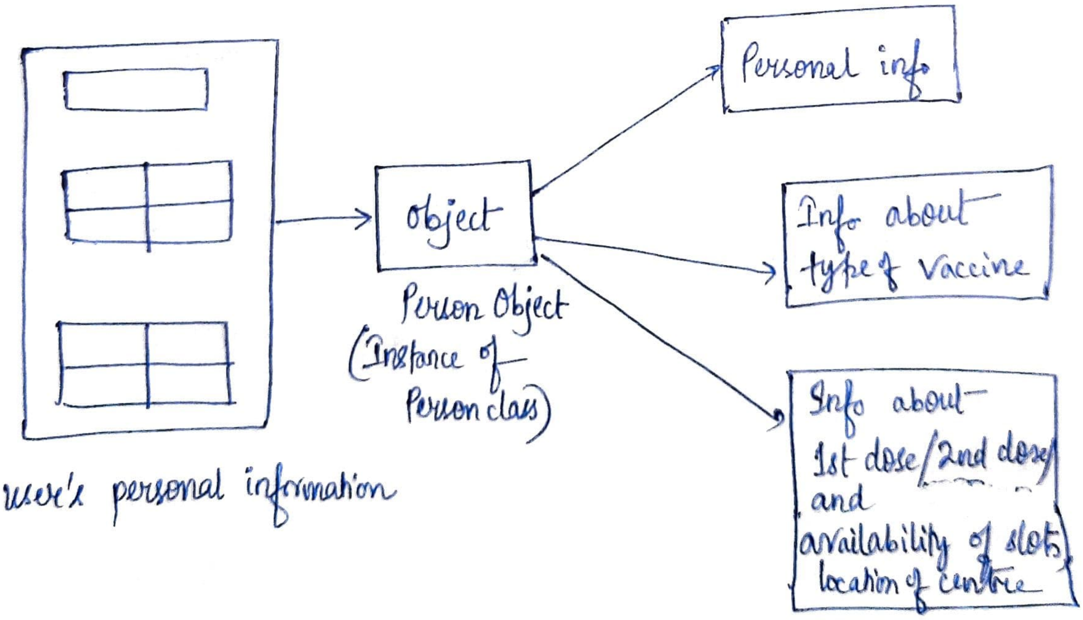
- Users
- Vaccine
- Hospitals

- Users — Person is the "IS-A" Relationship.

ORM can be used in the following ways in this situation :—

- For transfer of vaccine data from hospitals to MOFHW, the hospitals could simply provide its details and the data to MOFHW and the ORM abstracts details like verification through access tokens, generating requests and validating them etc.

- For users' registration at the MOFHW portal for vaccination, the user's just enter credentials or generate request for slot details, vaccine availability and the system provides the details to the user and abstracts all the intermediate steps like separating the user credentials to separate tables, fetching data related to the logged in user from the server etc. from the user.


object
Person Object
(Instance of Person class)
Personal info
Info about type of Vaccine
Info about 1st dose/2nd dose and availability of slots location of centre

A user's personal information

Ans: 7.

(1) Here, we want to choose two processes as co-ordinators from a group of n independent processes.

We assume that every active process in the system has a priority no. associated and the processes can send messages to all other processes in the system.

We want to elect the top two processes with the highest priority nos as the co-ordinators.

If any host thinks that the co-ordinator has failed, it tries to elect itself by sending a message to the highest numbered processors. If any of them answers, the host loses and each of the processors will call election and try to win themselves.

If none of the highest priority no. co-ordinators answer, then the host becomes the co-ordinator.

If a new processor arrives, it again calls for an election.

~~This continues two times~~

After getting the 1st co-ordinator, the above process repeats itself excluding the new co-ordinator to elect the next co-ordinator.

(2) Similarly, for ~~no~~ eleching m processes, continue the procedure m times keeping in mind whether new processes arrive and in that case, recall the election.

Ans:6. Let us consider a vector clock consisting n elements $V = (v_1, \ldots, v_n)$ which can be encoded by distinct prime nos. $p_1, p_2, \ldots, p_n$.

~~A method could be to encode.~~

Here, actually, we have encoded the entire vector V to a unique number using $p_1, p_2, \ldots, p_n$.

The encoded representation of $V = p_1^{v_1} \cdot p_2^{v_2} \cdots p_n^{v_n}$

This works well because the above product is distinct for any permutation of $\{v_1, v_2, \ldots, v_n\}$

This encoding of n-entries to a single no. reduces the space needed to represent vector clocks.