

## Objectives:

COVID-19 vaccines are on the way to hospitals and clinics across the country. The Centers for Disease Control and Prevention (CDC), other federal agencies, and state public health departments need real-time or near-real-time data on early mass COVID-19 vaccine administration. The objective of our project is to develop a simulated Vaccine Administration Management System (VAMS) system using the data structures we learned in the class, such as LinkedList, Stack/Queue, Binary Search Tree. In this system, we would like to implement four major components: clinician, vaccine recipients, appointment, and administrator.

## Requirements:

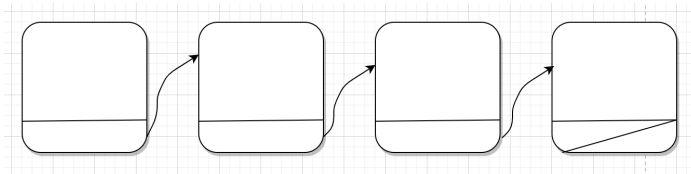
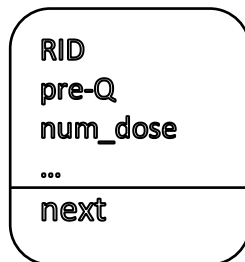
The system includes the following functionalities:

1. **Vaccine recipient component** manages all vaccine recipients records. Vaccine recipient component is responsible for adding/removing a recipients, updating recipient information, and retrieving all recipient information. Each recipient has the following properties: ID, name, prevaccination questionnaire, the number of doses, and receive a vaccination certificate (if received required amount of doses).
2. **Clinician component** manages all clinicians records. Clinician component is responsible for adding/removing a clinician, updating clinician information, and retrieving all clinician information. Each clinician has the following properties: clinician's ID, name, inventory records (if you would like to have a module called "Inventory", then clinician can get inventory from "Inventory" module. For simplicity, you can also make inventory records as a property for clinician).
3. **Appointment component** manages all appointments records. Appointment component is responsible for adding appointments, removing appointments, and updating appointments.
4. **Administrator component** manages the logistics of the above components and it can call methods in the above components. For example, it can add/remove appointments based on vaccine recipients' ID and clinician ID. Appointments can set appointment for first dose, second dose, or walk-ins specific to their clinic (e.g. second dose only clinic). Additional functions such as:
  - a. Input vaccine recipient info:
    - i. vaccine recipient ID/name
    - ii. prevaccination questionnaire
    - iii. the number of doses currently received
  - b. It can print all available clinician' names and allow users to select one of them.
  - c. It can updates appointments of a vaccine recipient based on the appointment order.
  - d. etc.

## Implementation Details:

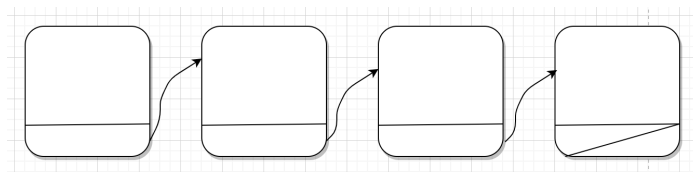
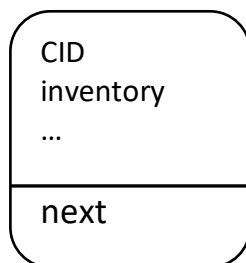
1. All vaccine recipients information can be maintained using a LinkedList, you can call it VaccineRecipientList, which contains a collection of nodes called VaccineRecipientNode (RNode). Each VaccineRecipientNode can have the following fields: RecipientID (RID), RecipientInfo (such as gender, address, etc), pre-vaccination questionnaire, and etc. It may also have a “next” field(which points to the next RNode). The following figure demonstrated the RNode and their connections.

VaccineRecipientNode

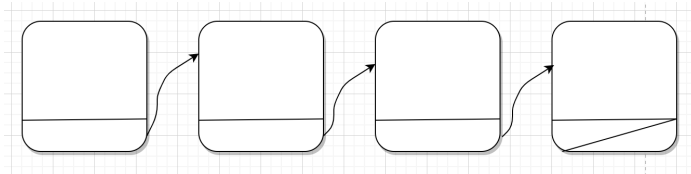
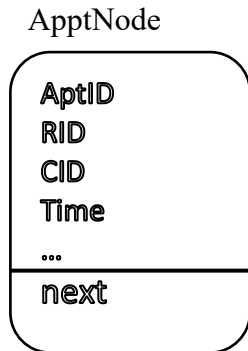


2. All clinicians information can be maintained using a LinkedList, you can call it ClinicianList, which contains a collection of nodes called ClinicianNode (CNode). Each CNode can have the following fields: ClinicianID (CID), ClinicianInfo (name, institution, etc), vaccine inventory, and etc. It may also have a “next” field (which points to the next CNode).

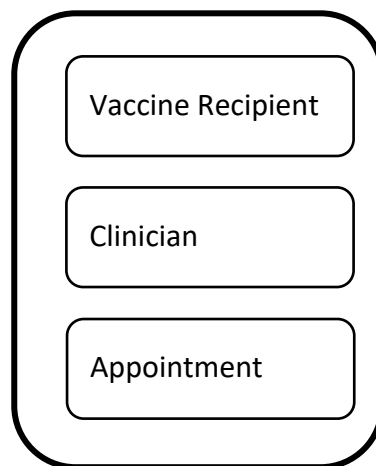
ClinicianNode



3. All Appointment information can be maintained using a LinkedList, you can call it AppointmentList, which contains a collection of nodes called ApptNode. Each ApptNode can have the following fields: AptID, RID, CID, time, and etc. It may also have a “next” field (which points to the next ApptNode).



4. Administrator component can be developed to manage vaccine recipient information as well as clinician’s information. It will maintain the integrity and consistency of the system. For example, if a recipient are removed from the recipient list, then all its appointment information should also be removed.

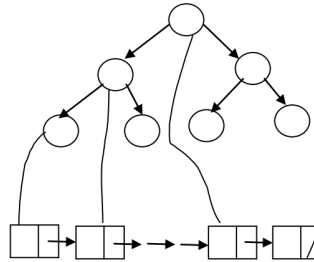


Some additional features for your consideration (optional):

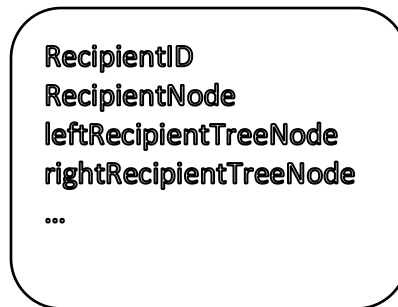
- Add capacity for the clinician (for each clinician there is a maximum number of appointments that can be handled);
- There are multiple types of vaccine for the recipients to choose;

- c. All vaccine recipients/clinician are stored in a data structure (i.e. linkedlist). A binary search tree can be built based on the names as indices;
- d. All clinicians are stored in a data structure (i.e. linkedlist). A binary search tree can be built based on the names as indices.

A binary search tree is built on the vaccine recipient names as index. The binary search tree is called Vaccine recipientIndexTree, which consists of a collection of Vaccine recipientTreeNode.



The RecipientTreeNode has four parts: RecipientID, RecipientNode (which is a reference to a node in the RecipientList), left (which points to leftRecipientTreeNode), and right (which points to rightRecipientTreeNode).



The binary search tree for Clinician is similar to Vaccine recipientIndexTree, so please reference to that.

### Project Submission and Evaluation

1. Due date: April 26th, 2020 11:59pm on Blackboard
2. When you submit your project please submit the following documents: a cover page with group member's signature, a discussion log (available on blackboard), source code, testing code (screenshot), testing results (screenshot).
3. You can either to have a console interface or graphical user interface.
4. Extra credits will be considered based on the complexity and quality of the additional features or algorithms that you incorporate into the project. As many as 10 extra credits will be given toward your final exam grades.
5. The project will be evaluated based on the grading standard posted on the blackboard. Partial credit will be given to incomplete project.
6. Each group also need to prepare for a presentation to demonstrate their system. The grading standard for presentation will also be posted on the Blackboard.