Objectives:

The objective of this project is to design a Hospital Information Management system using the data structures we learned in the class, such as LinkedList, Stack/Queue, Binary Search Tree. In this system, we could like to implement three major components: physicians, patients and visits.

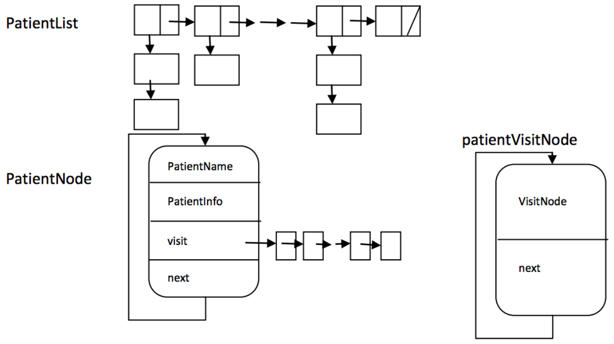
Requirements:

The system includes the following functionalities:

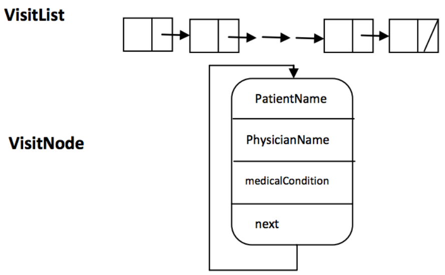
1. Create a new patient. Each patient contains patient name and medical condition (to simplify you can specify only one medical condition per patient)
2. Create a new physician. Each physician contains physician name and specialty (to simplify you can specify only one specialty per physician)
3. Add a new visit for patient:
    a. Input patient info:
        i. patient name
        ii. medical condition
    b. The system should print all available physicians' names and allow users to select one of them.
4. Print all the visits of a patient based on the visit order.
5. Print all the visits of a physician based on the visit order.
6. All patients are stored in a linked list. A binary search tree is build on the names as index.
7. All physicians are stored in a linked list. A binary search tree is build on the names as index.

Implementation Details:

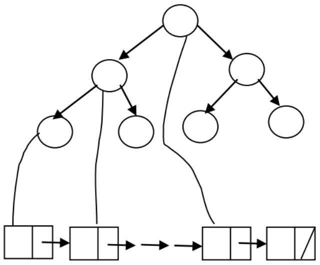
1. All patients information can be maintained using a LinkedList, you can call it PatientList, which contains a collection of nodes called PatientNode. Each PatientNode can have four fields: patientName, patientInfo (such as gender, address, etc), visit, next (which points to the next PatientNode). Each visit can also be considered as a LinkedList, which consists of a collection of PatientVisitNode. Each PatientVisitNode can have two fields: visitNode and next (which points to the next PatientVisitNode)



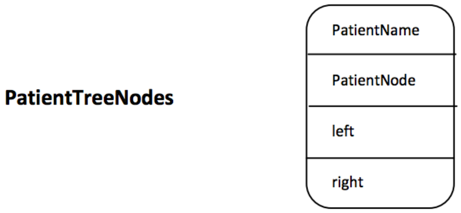
2. All physicians information can be maintained using a LinkedList, you can call it PhysicianList, which contains a collection of nodes called PhysicianNode. Each PhysicianNode can have four fields: physicianName, specialty, visit, next (which points to the next PhysicianNode). Each visit can also be considered as a LinkedList, which consists of a collection of PhysicianVisitNode. Each PhysicianVisitNode can have two fields: visitNode and next (which points to the next PhysicianVisitNode) -------- please refer to the diagram in the previous page
3. All visit information can be maintained using a LinkedList, you can call it VisitList, which contains a collection of nodes called VisitNode. Each VisitNode can have four fields: patientName, physicianName, medicalCondition, next (which points to the next visitNode).



4. A binary search tree is built on the patient names as index. The binary search tree is called PatientIndexTree, which consists of a collection of PatientTreeNodes.



The PatientTreeNodes has four parts: PatientName, PatientNode (which is a reference to a node in the PatientList), left (which points to left PatientTreeNode), and right (which points to right PatientTreeNode).



5. The binary search tree for Physician is similar to PatientIndexTree, so please reference to that.

Project Submission and Evaluation

1. Due date: April 15th, 2020 11:59pm on Blackboard and April 16th, 2018 in class for hard copy
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. The project will be evaluated based on the grading standard posted on the blackboard. Partial credit will be given to incomplete project.
5. 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.