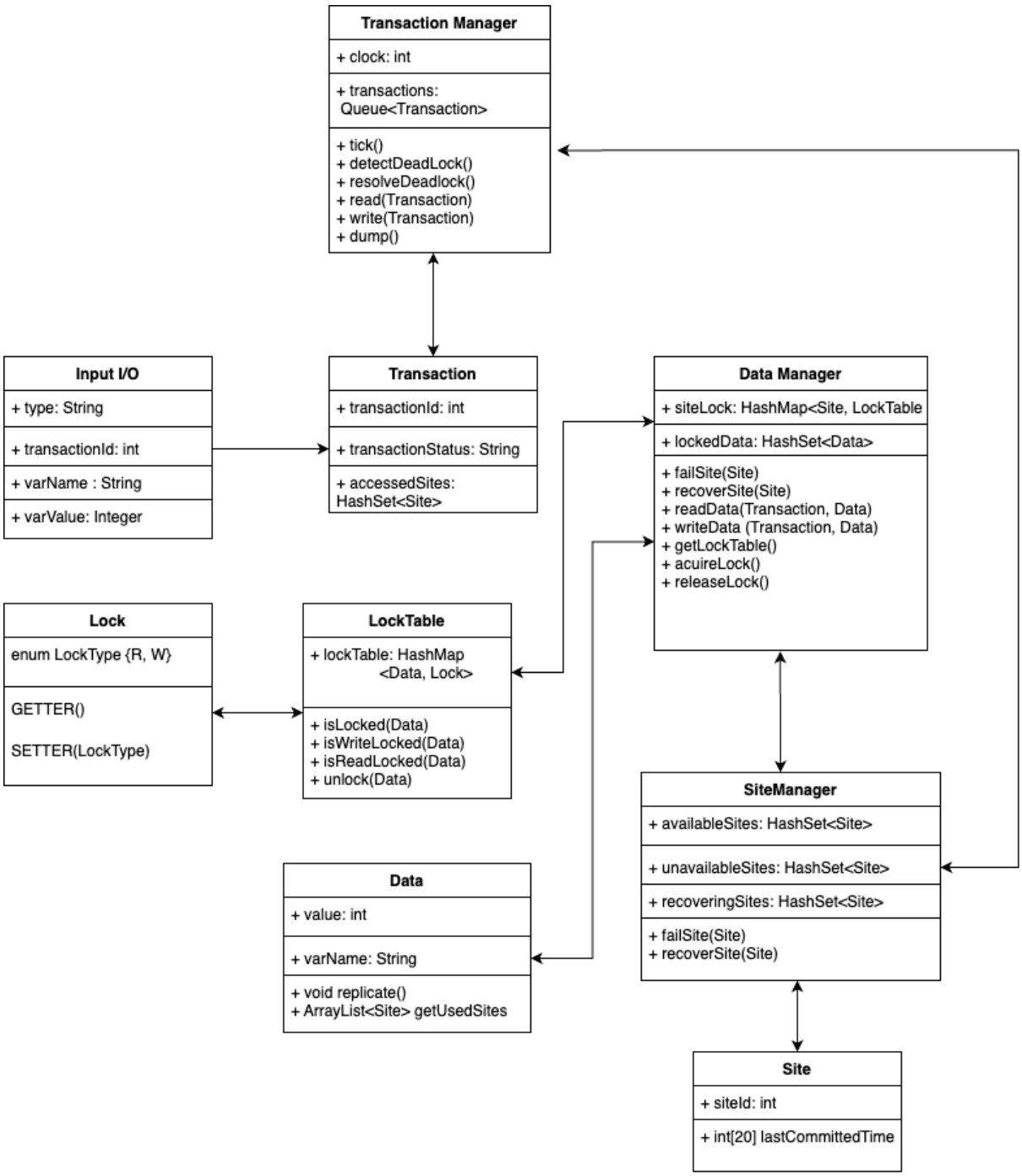
Design Document - Concurrency control and Recovery

Ramyakshi Mallik (rm6030)
Shraddha Iyer (sti5262)



The design for this project involves two main components - the TransactionManager and the DataManager.

The TransactionManager will be responsible for coordinating all transactions, detecting and resolving deadlocks before every tick, and also maintaining the time passed with the help of these ticks. The transactions will be held in a FIFO list(queue). It will have methods to Read, Write, Commit, Start etc which will be used for corresponding operations on the database. The dump function which will be used to print out the committed values of all copies of all variables at all sites, will also be implemented in the TransactionManger. The transactional data is stored in another class called Transaction. This Transaction class will have an id, a string denoting the type of transaction, status of the transaction and a hashed list of sites that the transaction will be accessing, amongst other class members. Another fundamental class is Site, which denotes the sites as explained in the problem statement. As its class members, id, this class will have the time of its last failure. We propose to have different hashed lists of sites for each of the statuses. Sites are managed by a class called SiteManager, which is responsible for failure, recovery and updating status lists. For Input I/O we have a dedicated class to print to the screen the information related to a transaction.

The DataManager is the next main component of this design. This holds the lock table for all sites, as well as coordinates the access and values of all variables. The lock table will have a hashmap with site ids as keys. It will have methods to get the type of lock and which transaction is currently locking it.We will have methods to hold the lock table for a particular site as well as release it. There will also be a supplementary Lock class that will have information on the type of lock. The DataManager will have methods to write, update and read data. Data class will be a separate class to get the list of sites accessed, to check for locks on that data and to replicate the data as required.

All of our implementation will be based in Java and we may revisit our proposed design and update it at the end of the semester..

Shraddha Iyer	Ramyakshi Mallik
TransactionManager, Transaction, Site, SiteManager	Input_I/O, DataManager, Data, LockTable, Lock