* Use DB interface
* Use database

Order of Big jobs

* Database
* Server
* Client
* networking
* GUI

Application Overview

* Create client program with GUI that connects to a Database
* A data access system that provides record locking and a flexible search mechanism
* Network server functionality for the Database system
* **Maintainability:** Clear design and limited complexity(nesting etc)
* **Project must adhere to spec!**
* **Modular approach!**

1. **Documentation**

* Create an outline plan, written description of solution design
  + Design decisions and alternatives
  + Classes, Interfaces
  + What does the class or method do? It you have to use the word “and” you might be asking it to do too much.
  + How are exceptions handled? Types of exceptions?
  + Logging?
  + Design Patterns used (Transfer Object Pattern, MVC, facade)
  + Classes that will implement the patterns
  + Package structure?
  + UML class/interface diagrams where applicable – include variables and methods. Include inheritance
  + GUI design – screen shots from Java, Visio, hand drawn etc
* In code comments –
  + Javadoc comments must be used for each element of the public interface of each class
  + Javadoc tool must be used to create full documentation in HTML format.
  + Provide Javadoc documentation for each class that you write <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>
  + don’t comment unnecessarily
* Provide user documentation
  + Text doc format

**2) Architecture**

* 3 Main parts: server side data management, client side GUI, network connection between the two.
* Non-networked Mode
  + Must be able to work in a non-networked mode
  + Database and GUI must run in the same VM and must :
    - perform no networking
    - not use loop back networking
    - no t involve serialization of objects when communication between GUI and database elements
  + Mode must use the same database and GUI from networked mode but none of the networked server code.
  + Operating mode is selected using a single command line argument!(“server” or “alone”
* Network Communication Approach
  + Must use either:
    - Serialized objects over a simple socket connection, or
    - RMI – see guide for restrictions.

**3) GUI**

* Must be composed exclusively from swing components( where available, if not ok to use AWT).
* Must allow user to search for all records
* Must allow user to search for records where name and/or location fields exactly match values specified by user
* Search results must be presented in a Jtable
* Must allow user to “Book” a selected record, updating the database file accordingly.
* Should be designed to include addition of future functionality i.e. swapped out or have new features plugged in – MVC!
* All configuration must be done by GUI and must be persistend between runs of the program
  + Configuration information must be stored in a file called suncertify/properties located in the current working directory.

4) **Server**

* Data access class must
  + be named Data.java
  + be in a package called suncertify.db
  + must implement the “DB” interface
  + any unimplemented exceptions (custom exceptions?) must be created as member classes of suncertify.DB package
    - each must have a no arg constructor and a second constructor that accepts a String serving as the exceptions description.
  + Methods that throw a RecordNotFound Exception should do so if a record does not exist or is marked as deleted in the database file!
* Update DB.Java file to include Javadoc comments
* Network approaches must
  + Program must allow users to specify the location of the database
  + Must accept indication of whether database is local or server ( command line arg?) and if local must bypass all networking.
* Locking!
  + Server must be capable of handling multiple concurrent requests, and thus provide a locking functionality as per interface i.e
    - Locks a record so it can only be updated/deleted by current client
    - Return a value cookie that must be used when a record is unlocked, updated or deleted
    - If record is locked, current thread waits until record is unlocked.
  + Unlocking as per interface
    - Release the lock on a record (notify) when
    - Cookie must be the cookie returned when the record was locked, otherwise throws a security exception
    - Unlock methods are overloaded
  + Locking only needs to lock the server – it is assumed that only 1 program at a time can access the database.

**4) Deliverables**

* Both parts (server and client) must e executable from command line using java –jar <path\_and\_fileName> [<mode]
* Programs must not require use of any other command line args.
* Flag mode is either “server” or “alone”. If none network client and gui must run.

**Packaging submissions**