**CS 475**

Dr. Sanjeev Setia

**Name**: Shohinjon Khamidjanov & Avelino Inguane

**Login IDs**: ainguane & skhamidj

**RMI: Creating Distributed Applications**

**General Steps**

1. Designing and implementing the components of distributed application.

2. Compiling sources.

3. Making classes network accessible.

4. Starting the application

1. Designing and implementing the components of distributed application.
   1. Defining the remote interfaces. A remote interface specifies the methods that can be invoked remotely by a client. Clients program to remote interfaces, not to the implementation classes of those interfaces. The design of such interfaces includes the determination of the types of objects that will be used as the parameters and return values for these methods. If any of these interfaces or classes do not yet exist, you need to define them as well.
   2. Implementing the remote objects. Remote objects must implement one or more remote interfaces. The remote object class may include implementations of other interfaces and methods that are available only locally. If any local classes are to be used for parameters or return values of any of these methods, they must be implemented as well.
   3. Implementing the clients. Clients that use remote objects can be implemented at any time after the remote interfaces are defined, including after the remote objects have been deployed.

|  |
| --- |
| **DESIGN** |
|  |

|  |  |
| --- | --- |
|  | **Calendar Overview** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 | **Notification Overview** |
| 5 |  |
|  |  |

|  |
| --- |
|  |
| Notification Functionality  **Client:** An Open Event    **Server:** broadcast to other clients (Alert other clients of other client availability)    **Other Clients:** |
| Compiling sources  &  Starting the application   1. Start the Server   javac CalendarManager.java  rmiregistry &  java -Djava.security.policy=policy.txt CalendarManager & |
|  |
| 1. Start the Client   javac Client.java  java Client Alice |
| Making classes network accessible  SERVER        CLIENT |
| 1. For each client |
| 1. Create three clients for Alice, Bob and Tom respectively        1. Try to create another **client** for Alice – should report error        1. Insert public and private events into Alice, Bob and Tom’s calendars including an event that results in an alarm        1. If necessary, insert open events too.      1. Enter events with mistakes (test robustness of interface)   ????   1. From Alice’s client, look at Bob and Tom’s calendars.      1. Insert group event into all three calendars.      1. Insert group event into Alice and Bob’s calendars.      1. Check ability to delete events {group or individual} |
|  |
|  |