Design Document

Restaurant Menu extraction based on the type and name of the restaurant selected

The main purpose of this application is to provide the users, the accessibility to extract and view the food menu of a particular type and name of the restaurant. The user has the option to select the type of restaurant that he/she wants the information about. Additionally the users can get all the restaurants available under a particular type of restaurant.

The following distributed system consists of three servers and a client. The below is the sequence diagram to depict the functionality of the distributed system.

User Restaurant Type Restaurant Name Restaurant menu

The following provides the communication between the distributed objects.

***For Restaurant Type***

The user/Client gets the restaurant type from the Restaurant Type server.

***For Restaurant Name***

The User/Client gets the restaurant names from restaurant Name server through Restaurant Type server. The user contacts the restaurant Type Server for the Restaurant names. So the restaurant type server now becomes the client for the restaurant name server. It obtains the results through the remote method and in turn passes the information to the client.

***For Restaurant Menu***

The User/Client directly interacts with the Restaurant menu server for the information about the restaurant menu list.

***For Validating Referential integrity***

Referential Integrity: If two references to an object are passed from one JVM to another JVM in parameters (or in the return value) in a single remote method call and those references refer to the same object in the sending JVM, those references will refer to a single copy of the object in the receiving JVM. More generally stated: within a single remote method call, the RMI system maintains referential integrity among the objects passed as parameters or as a return value in the call.

To validate the referential integrity, the client object sends two references to the client object through parameters to Restaurant Type server, which in turn sends these two references as parameters to the Restaurant menu Server. In the restaurant menu server we retrieve the object ids of two references and pass them back to client to get printed as a string.

The objects Ids are found to be the same, which verifies referential integrity.

***Technical Specifications:***

*List of remote Interfaces used:*

1. Restaurant Type Server: RestaurantType
2. Restaurant Name Server: RestaurantName
3. Restaurant Menu Server: RestaurantMenu

*List of remote Methods used:*

1. RestaurantType: getRestaurantType(), getRestaurantNames(int), refIntCheck(Client,Client)
2. RestaurantName: getResName(int)
3. RestaurantMenu: getMenu(String), checkRefInt(Client,Client)

*List of remote objects used:*

1. Restaurant Type Server: RestaurantTypeImpl
2. Restaurant Name Server: RestaurantNameImpl
3. Restaurant Menu Server: RestaurantMenuImpl