Admin

username (Primary key)
firstname
lastname
passwd
email
mobile

---

boolean addVendor(Vendor v) ;
   List<Vendor> viewVendors() ;
   List<User> viewUsers() ;
   boolean validateAdmin(Admin a) ;
   boolean updateUserStatus(int uid, String status) ;

1..N

Vendor

vendorid (PrimaryKey)
username (Unique)
passwd
firstname
lastname
street
city
state
country
pincode
email
mobile

---

boolean validateVendor(Vendor v)
boolean addPackage(Packages p)
List<PlanRequest> viewUserRequest()
boolean createQuotation(Quotation q)

user

userid (PrimaryKey)
firstname
lastname
username (Unique)
passwd
doj (Date of Joining)
dob (Date of Birth)
dept (Marketing, Sales, IT)
email
mobile
location
userstatus (Active, Non-active)

User findById(int userId)
boolean registerUser(User u)
boolean updatePassword(int userid, String newPassword)
boolean addPlanRequest(PlanRequest pl) throws SQLException,
InvalidNumberOfPeopleException, PlannedDateException;
List<Quotation> viewQuotation(int uid)
boolean updateQuotationStatus(int quotid, String status)
public void closeConnection()

PlanRequest

private int requestId,
noOfPeopleAttending, userId;
   private String fromdate, todate

PlanRequest findById(long
requestId);
   List<PlanRequest> findAll();
   void save(PlanRequest
planRequest);
   void update(PlanRequest
planRequest);
   void delete(long requestId);
}

Quotation

quotid (PK)
packagetype (Basic, Premium,
Classic)
estimatedamt
vendorid (FK)
userid (FK)
requestid (FK)
quotstatus (Accepted, Pending,
Rejected)

Quotation findById(long
quotationId);
   List<Quotation> findAll();
   void save(Quotation quotation);
   void update(Quotation
quotation);
   void delete(long quotationId);
   void create(Quotation
quotation);

Package

packageid (PK)
packagetype (Basic, Premium,
Classic)
amount
vendorid (FK)

Package findById(long
packageId);
   List<Package> findAll();
   void save(Package aPackage);
   void update(Package
aPackage);
   void delete(long packageId);

1..N