package com.seva.user.dao.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.geo.GeoResults;

import org.springframework.data.geo.Metrics;

import org.springframework.data.geo.Point;

import org.springframework.data.mongodb.core.MongoOperations;

import org.springframework.data.mongodb.core.MongoTemplate;

import org.springframework.data.mongodb.core.aggregation.Aggregation;

import org.springframework.data.mongodb.core.index.GeospatialIndex;

import org.springframework.data.mongodb.core.query.Criteria;

import org.springframework.data.mongodb.core.query.NearQuery;

import org.springframework.data.mongodb.core.query.Query;

import org.springframework.stereotype.Repository;

import org.springframework.data.mongodb.core.geo.\*;

import com.seva.user.dao.SpUserDetailsDao;

import com.seva.user.model.Address;

import com.seva.user.model.SpUser;

import com.seva.user.model.SpUserD;

@Repository

public class SpUserDetailsDaoImpl implements SpUserDetailsDao{

@Autowired

MongoTemplate mongoTemplate;

public SpUserDetailsDaoImpl(){

}

@Override

public void createUser(SpUser user) {

MongoOperations mongoOperation = (MongoOperations) this.mongoTemplate;

try

{

mongoOperation.insert(user,"spusers");

}

catch(Exception ex )

{

System.out.println("error::" + ex.getMessage());

}

}

@Override

public void updateUser(SpUser user) {

// TODO Auto-generated method stub

}

@Override

public void deleteUser(SpUser user) {

// TODO Auto-generated method stub

}

@Override

public SpUser getUser(SpUser user) {

MongoOperations mongoOperation = (MongoOperations) this.mongoTemplate;

Point point = new Point(-91.493619, 44.240309);

user.getAddress().setLoc(point);

//Query searchUserQuery = new Query(Criteria.where("mobileNum").is(user.getMobileNum()));

Query searchUserQuery = new Query(Criteria.where("address.loc").nearSphere(point).maxDistance(20D));

SpUser dbuser = mongoOperation.findOne(searchUserQuery, SpUser.class,"spusers");

//mongoOperation.find({ location: { $nearSphere: { $geometry: { type: "Point", user.getAddress().getLoc(), $maxDistance: 5 \* METERS\_PER\_MILE } } })

return dbuser;

}

@Override

public List<SpUser> fetchNearByUsers(String zipcode) {

MongoOperations mongoOperation = (MongoOperations) this.mongoTemplate;

Point point = new Point(-91.493619, 44.240309);

// Query searchUserQuery = new Query(Criteria.where("address.loc").nearSphere(point).maxDistance(1D));

// List<SpUser> dbusers = mongoOperation.find(searchUserQuery, SpUser.class,"spusers");

NearQuery query = NearQuery.near(point.getX(),point.getY()).spherical(true).maxDistance(10,Metrics.MILES);//maxDistance(new Distance(radius,Metrics.MILES));

GeoResults<SpUserD> dbusers = ((MongoOperations)mongoTemplate).geoNear(query, SpUserD.class);

//final Aggregation a = newAggregation(geoNear(query, "distance"));

List<SpUserD> ls = db

return ls;

}

}