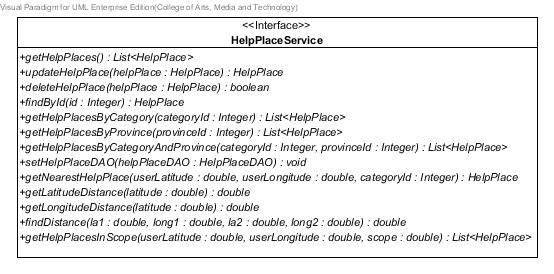


1. HelpPlaceService



**Figure 9** HelpPlaceService **class**

**Method details**

1. getHelpPlaces

public List<HelpPlace> getHelpPlaces ()

**Description:** Get list of all help places from the database.

**Return:** List of all help places in database.

1. findById

public HelpPlace findById (Integer id)

**Description:** Get help place by id from the database.

**Parameter:** id-Integer uses for getting the help place.

**Return:** The help place in database.

1. updateHelpPlace

public HelpPlace updateHelpPlace(HelpPlace helpPlace)

**Description:** Update information of the help place to database.

**Parameter:** helpPlace-HelpPlace uses for update information of the help place to database.

**Return:** Help place object that needs to update.

1. deleteHelpPlace

public boolean deleteHelpPlace(HelpPlace helpPlace)

**Description:** Delete the help place in the database.

**Parameter:** helpPlace-HelpPlace uses for delete the help place from database.

**Return:** Boolean, if the help place has been deleted, will return true.

1. getHelpPlacesByCategory

public List<HelpPlace> getHelpPlacesByCategory(Integer categoryId)

**Description:** Get help places from the database by the selected category id.

**Parameter:** categoryId-Integer uses for getting the help place from database.

**Return:** List of help places that contain the selected category, in the database.

1. getHelpPlacesByProvince

public List<HelpPlace> getHelpPlacesByProvince(Integer provinceId)

**Description:** Get help places from the database by the selected province id.

**Parameter:** provinceId-Integer uses for getting the help place from database.

**Return:** List of help places that contain the selected province, in the database.

1. getHelpPlacesByCategoryAndProvince

public List<HelpPlace> getHelpPlacesByCategoryAndProvince (Integer categoryId, Integer provinceId)

**Description:** Get help places from the database by the selected category id and province id.

**Parameter:** provinceId–Integer and categoryId-Integer use for getting the help place from database.

**Return:** List of help places that contain the selected category and province, in database.

1. setHelpPlaceDAO

public void setHelpPlaceDAO(HelpPlaceDAO helpPlaceDAO)

**Description:** Set HelpPlaceDAO value into help place’s attribute.

**Parameter:** helpPlaceDAO–HelpPlaceDAO use for setting help place DAO.

XXX: getNearestHelpPlace

public HelpPlace getNearestHelpPlace(double userLatitude, double userLongitude, Integer categoryId)

**Description:** Get nearest help place from the database by the selected category id.

**Parameter:** userLatitude-double use for finding nearest help place.

userLongitude-double use for finding nearest help place.

categoryId-Integer use for getting the help place from database.

**Return:** Nearest help place by the selected category.

XXX: getLatitudeDistance

public double getLatitudeDistance(double latitude)

**Description:** Get surface distance per 1 degree change in latitude (meter).

**Parameter:** latitude-double use for finding surface distance per 1 degree change in latitude.

**Return:** Surface distance per 1 degree change in latitude in unit meter.

XXX: getLongitudeDistance

public double getLongitudeDistance(double latitude)

**Description:** Get surface distance per 1 degree change in longitude (meter).

**Parameter:** latitude-double use for finding surface distance per 1 degree change in longitude.

**Return:** Surface distance per 1 degree change in longitude in unit meter.

XXX: findDistance

public double findDistance(double la1, double long1,double la2, double long2)

**Description:** Get distance between two difference locations measures in unit meter.

**Parameter:** la1-double use for indicate latitude of location one.

long1-double use for indicate longitude of location one.

la2-double use for indicate latitude of location two.

long2-double use for indicate longitude of location two.

**Return:** Distance between two locations in unit meter.

XXX: getHelpPlacesInScope

public List<HelpPlace> getHelpPlacesInScope(double userLatitude, double userLongitude, double scope)

**Description:** Get list of help places in the setting area scope.

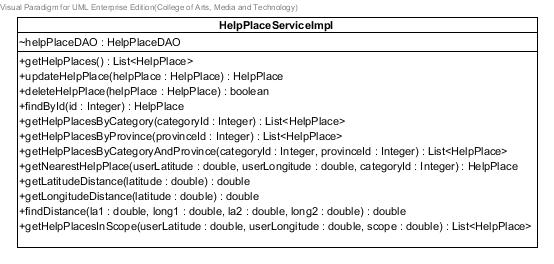
**Parameter:** userLatitude-double use for indicate user current location.

userLongitude-double use for indicate user current location.

scope-double use to define the scope for getting help places.

**Return:** List of help places locate in the selected scope.

1. HelpPlaceServiceImpl



**Figure 10** HelpPlaceServiceImpl **class**

**Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Type** |
| 1 | helpPlaceDAO | Contained HelpPlaceDAO object | HelpPlaceDAO |

**Method details**

1. getHelpPlaces

public List<HelpPlace> getHelpPlaces()

**Description:** Get help places from the database through getHelpPlaces method of HelpPlaceDAO class.

**Flow:**

1. Use method getHelpPlaces()of HelpPlaceDAO class to get all help places.
2. Return list of all help places.

**Return:** List all help places in the database.

1. updateHelpPlace

public HelpPlace updateHelpPlace(HelpPlace helpPlace)

**Description:** Update information of the help place to the database through updateHelpPlace method of HelpPlaceDAO class.

**Parameter:** helpPlace-HelpPlace uses for update information of the help place to the database.

**Flow:**

1. Use method updateHelpPlace(helpPlace)of HelpPlaceDAO class to update help place.

**Return:** Help place object that needs to update.

1. deleteHelpPlace

public boolean deleteHelpPlace(HelpPlace helpPlace)

**Description:** Delete the help place from database through deleteHelpPlace method of HelpPlaceDAO class.

**Parameter:** helpPlace-HelpPlace uses for delete the help place from database.

**Flow:**

1. Use method deleteHelpPlace(helpPlace)of HelpPlaceDAO class to delete help place.

**Return:** Boolean, if the help place has been deleted, will return true.

1. findById

public HelpPlace findById(Integer id)

**Description:** Get help place from the database through findById method of HelpPlaceDAO class.

**Parameter:** id-Integer uses for getting the help place.

**Flow:**

1. Use method findById(id)of HelpPlaceDAO class to find the help place by using its id.

**Return:** The help place in the database.

1. getHelpPlacesByCategory

public List<HelpPlace> getHelpPlacesByCategory(Integer categoryId)

**Description:** Get help places from the database by selected category id through getHelpPlacesByCategory method of HelpPlaceDAO class.

**Parameter:** categoryId-Integer uses for getting the help place from database.

**Flow:**

1. Use method getHelpPlacesByCategory(categoryId)of HelpPlaceDAO class to get list of help places.
2. Return list of help places.

**Return:** List of help places in the database.

1. getHelpPlacesByProvince

public List<HelpPlace> getHelpPlacesByProvince(Integer provinceId)

**Description:** Get help places from the database by selected province id through getHelpPlacesByProvince method of HelpPlaceDAO class.

**Parameter:** provinceId-Integer uses for getting the help place from database.

**Flow:**

1. Use method getHelpPlacesByProvince(provinceId) of HelpPlaceDAO class to get list of help places.
2. Return the list of help places.

**Return:** List of help places in the database.

1. getHelpPlacesByCategoryAndProvince

public List<HelpPlace> getHelpPlacesByCategoryAndProvince (Integer categoryId,Integer provinceId)

**Description:** Get help places from the database by selected category id and province id through getHelpPlacesByCategoryAndProvince method of HelpPlaceDAO class.

**Parameter:** provinceId–Integer and categoryId-Integer uses for getting the help place from database.

**Flow:**

1. Use method getHelpPlacesByCategoryAndProvince(categoryId,provinceId)of HelpPlaceDAO class to get list of help places.
2. Return the list of help places.

**Return:** List of help places in the database.

1. setHelpPlaceDAO

public void setHelpPlaceDAO(HelpPlaceDAO helpPlaceDAO)

**Description:** Set HelpPlaceDAO value into help place’s attribute.

**Flow:**

1. Get parameter to set value of helpPlaceDAO attribute.

**Parameter:** helpPlaceDAO–HelpPlaceDAO use for setting help place DAO.

XXX: getNearestHelpPlace

public HelpPlace getNearestHelpPlace(double userLatitude, double userLongitude, Integer categoryId)

**Description:** Get nearest help place from the database by the selected category id.

**Parameter:** userLatitude-double use for finding nearest help place.

userLongitude-double use for finding nearest help place.

categoryId-Integer use for getting the help place from database.

**Flow:**

1. Use method getHelpPlacesByCategory(categoryId) to get list of help places .
2. Check if the list is empty, then return null.
3. If not, get distance of all help places in list by use method findDistance(double la1, double long1, double la2, double long2), by compare latitude and longitude in parameter with latitude and longitude of each help place in list.
4. Keep all distances in distance list.
5. Find the minimum distance from each value in distance list.
6. Return help place which have a minimum distance value.

**Return:** Nearest help place by the selected category.

XXX: getLatitudeDistance

public double getLatitudeDistance(double latitude)

**Description:** Get surface distance per 1 degree change in latitude (meter) which is a tolerance value.

**Parameter:** latitude-double use for finding surface distance per 1 degree change in latitude.

**Flow:**

1. Use latitude from parameter divided by 15 and convert result into integer, to find number of index in an array.
2. Get value “Surface distance per 1 degree change in latitude” in the array, multiply with 1000 to change unit into meter.
3. Return Surface distance per 1 degree change in latitude in unit meter.

**Return:** Surface distance per 1 degree change in latitude in unit meter.

XXX: getLongitudeDistance

public double getLongitudeDistance(double latitude)

**Description:** Get surface distance per 1 degree change in longitude (meter) which is a tolerance value.

**Parameter:** latitude-double use for finding surface distance per 1 degree change in longitude.

**Flow:**

1. Use latitude from parameter divided by 15 and convert result into integer, to find number of index in an array.
2. Get value “Surface distance per 1 degree change in longitude” in the array, multiply with 1000 to change unit into meter.
3. Return “Surface distance per 1 degree change in longitude” in unit meter.

**Return:** Surface distance per 1 degree change in longitude in unit meter.

XXX: findDistance

public double findDistance(double la1, double long1,double la2, double long2)

**Description:** Get distance between two difference locations measures in unit meter.

**Parameter:** la1-double use for indicate latitude of location one.

long1-double use for indicate longitude of location one.

la2-double use for indicate latitude of location two.

long2-double use for indicate longitude of location two.

**Flow:**

1. Find “Surface distance per 1 degree change in latitude”, and “Surface distance per 1 degree change in longitude” to use calculate distance.
2. Get parameter and calculate to get distance between two locations.
3. Return distance.

**Return:** Distance between two locations in unit meter.

XXX: getHelpPlacesInScope

public List<HelpPlace> getHelpPlacesInScope(double userLatitude, double userLongitude, double scope)

**Description:** Get list of help places in the setting area scope.

**Parameter:** userLatitude-double use for indicate user current location.

userLongitude-double use for indicate user current location.

scope-double use to define the scope for getting help places.

**Flow:**

1. Get list of all help places by using method getHelpPlaces().
2. Find a distance between each help place in the list, and userLatitude, userLongitude.
3. Check each distance is in the scope or not.
4. Keep the help place which its distance is in the setting scope.
5. Return list of help places where locate in the setting scope.

**Return:** List of help places locate in the selected scope.