Arab Academy for Science, Technology and Maritime Transport
College of Computing & Information Technology
Database Management Systems

Domestic Tourism System

documentation file

Supervised by	Dr. Mohamed Kholief Eng. Mohamed Galal	
By	Abdallah Mohamed Elsayed Aboudeif	Reg. no. 19235004
	Mohamed Masoud Mohamed Ibrahim	Reg. no. 20135025
	Mohamed Hussein Abdelaal	Reg. no. 20135009

Contents:

1 INTRODUCTION

While countries often tend to focus on international tourism due to the revenue earned through exports; domestic tourism remains the leading form of tourism, representing an important tool for regional economic growth and development.

With over 50% of the global population now categorized as “middle class”, more and more people can afford to travel. Research suggests that domestic tourism demand picks up at an income level lower than international travel.

Absolute size of Travel & Tourism’s domestic spending in 2017, domestic tourism represented 73% of the total global tourism spend (US$3,971 billion). While there are significant variations between countries.

Egypt is one of the most prominent tourist countries in the world, it is distinguished by the abundance of tourist attractions of all kinds, the spread of temples, museums, monuments, historical and artistic buildings and vast gardens on its land, and its possession of strong infrastructure based on serving the tourism sector, including Hotel rooms, villages, tourist resorts, tourism companies and airline offices.

2 Business logic

Domestic tourism system logic stands for offering attractive short to medium travels that aims to spend vacations visiting one place or more inside one or more city, this structure of business requires collecting and storing tourists’ data, offering hostels and transportation for tourists, offering tourist guides for tourists’ groups to help them know about tourist places and coordinate their travels. Marketing is an important part of this business, attracting tourists is make by advertising on a travel at a known media beside to that it’s possible to deal with popular entities by giving their members a discount to encourage their clients or members have travels with our company. It is what we can call a Partner and it’s another important source for customers. Also, it is recommended to store hostel and transport owners and advertising companies’ data as a contact to continuously update their services prices and availability.

3 Roles summary

Business roles is roles can be summary in these points:

- Business goal is to make a profit by making travels inside the country
- Business has an information system consists of Microsoft SQL Server database and Java desktop application supported with JAVA-FX GUI
- Advertising companies make advertisements for our company
- Agents are responsible for working on system
- Agents publish advertisements for a travel on many types of media
- Agents store tourists and their partners on system as they have their information and update it
- Agents stores themselves, tourist guides, tourism places, advertisements and its companies, transports and their owners, hostels and their owners, campaigns
- Agents create travels and register tourism places, tourist guides, transports, campaigns, hostels and tourists in it
- Tourists have balance and it’s decreased by a travel price as they registered in it discounted by their partner discount (if any)

- Guides have a daily rate that calculated as a cost in every travel they registered on according to number of days they worked on
- Every campaign has a publish cost that calculated as a cost on its travel
- Advertisements have a design cost that doesn't calculated as a travel cost while one advertisement consider a business asset and it can be used in many travels for years
- Every hostel has its price and its updateable according to price changes, this price is calculated as a cost in every travel it registered on according to number of nights and rooms
- Transports have its price and its updateable according to price changes, this price is calculated as a cost in every travel they registered on according to number of days it used on
- Places have its ticket price, and it is calculated as a cost in every travel it registered on according to tourists' number
- Tourist Guides are responsible for guides tourists in every travel they registered on
- System should show a travel report showing travel data, registered members numbers, costs, profits and net profit

4 Constraints on business

- Tourist, agent and guide cannot be registered on database more than one time
- It's mandatory to store all person's kind mobile numbers as it's the main way to connect with them
- Advertisements cannot be registered directly in a travel as it have a design cost that consider as on-time-buy and it could be published many times for one travel
- Travel elements cities should be known clearly (not a part of address) to help agents make local compatibility between travel elements
- Partners cannot have a discount more than 100% or less than 0%
- Instead of deleting temporary unused elements, elements have [active/idle] state to ignore idle elements registration on travels while its data still stored
- Travels are created with no elements, and it is eligible for have one or many elements in every type making it flexible to fit all domestic tourist travel types
- Total cost of every registered element is calculated and stored in element's registration table to not affected by future element's price changes

5 Application GUIs

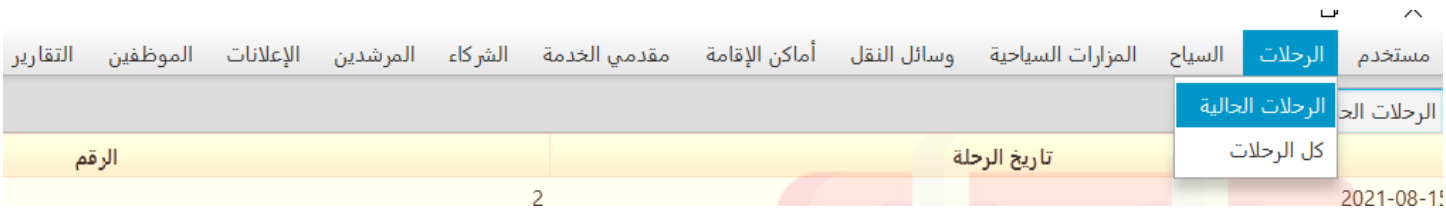
- System supported by many GUI's that help business goals achieved easy way
- Agents are normal users of system so there is login window that allows only registered users work on system using username and password
- The main window GUI contains of many internal GUI's showing and handling all work object's data and supported by search boxes and easy use menu bar to help user find data and work easily
- Many sub-GUIs are created to serve creating and editing objects

6 Examples of GUIs

- Current travels GUI

Application default pane is 'Current travels' which show travels in current week helping agents follow travel time schedule. It shows travels on 2 days before today, today travels and 4 days after.

- Menu bar



Object categorized menu bar helping easy access to all business data and make any needed operation, all menu item is made for handling an objects data

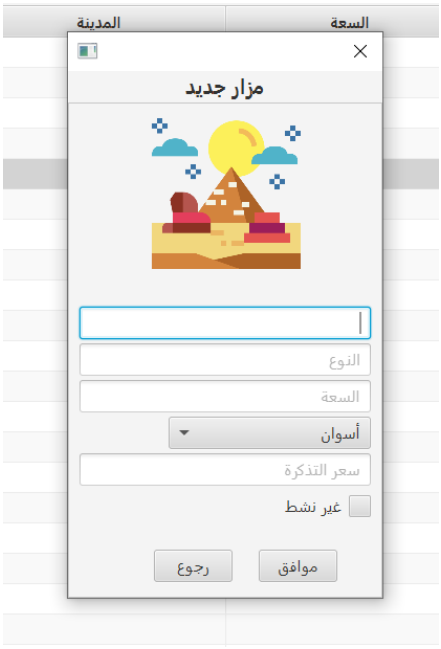
- Object tables

الرقم	الشريك	معلومات إضافية	الرصيد	الاسم	الرقم القومي	النوع	الهاتف
1			5136.0	محمد حسين الشيخ	25636586547521	ذكر	01125632512
2	مؤسسة الشرق		2300.0	علاء الدين خالد	25658541230125	ذكر	01144587541
3	مؤسسة الشرق		3700.0	حمادة أحمد حمادة	20114857965412	ذكر	01253621421
4	مؤسسة الشرق		1000.0	إسماعيل محمد محمد	23365652214875	ذكر	01123665412
5			5000.0	أمين محمد محمود	22012255874698	ذكر	01023201254
6	الشركة المصرية لتو...		1000.0	محفوظ حسنين عبد...	20120165478965	ذكر	01025486594
7	الشركة المصرية لتو...		1000.0	أدم فتحي سعيد	32652141021254	ذكر	01125463254
8	مؤسسة الشرق		2400.0	سامح غريب زاهر	33625145879521	ذكر	01121154254
9	الشركة المصرية لتو...		700.0	كمال الدين عيد كامل	20069987741251	ذكر	01115362112
10			1300.0	راشد محمد شاهين	23311199885541	ذكر	01002255544
11	الشركة المصرية لتو...		900.0	أحمد غزال فاضل	20784585974568	ذكر	01021236543
12	الشركة المصرية لتو...		2000.0	فاضل أحمد قاسم	35410012141259	ذكر	01111566858
13		null	2000.0	السيد فضل عويس	32226598745213	ذكر	01212333666
14			1950.0	مروان عبد القادر ح...	32558745651245	ذكر	01285469855
15	الشركة المصرية لتو...		1700.0	يوسف محمد عباس	26510014121321	ذكر	01263611110
17	مؤسسة الشرق	لا يعيش في الإسما...	1500.0	حسن حسنين	23658754655663	ذكر	01525652223

Objects data showing is done in embedded tables into closable tabs, this makes workspace more customizable and adjustable. Tables are supported by sorting capability and columns scalability too.

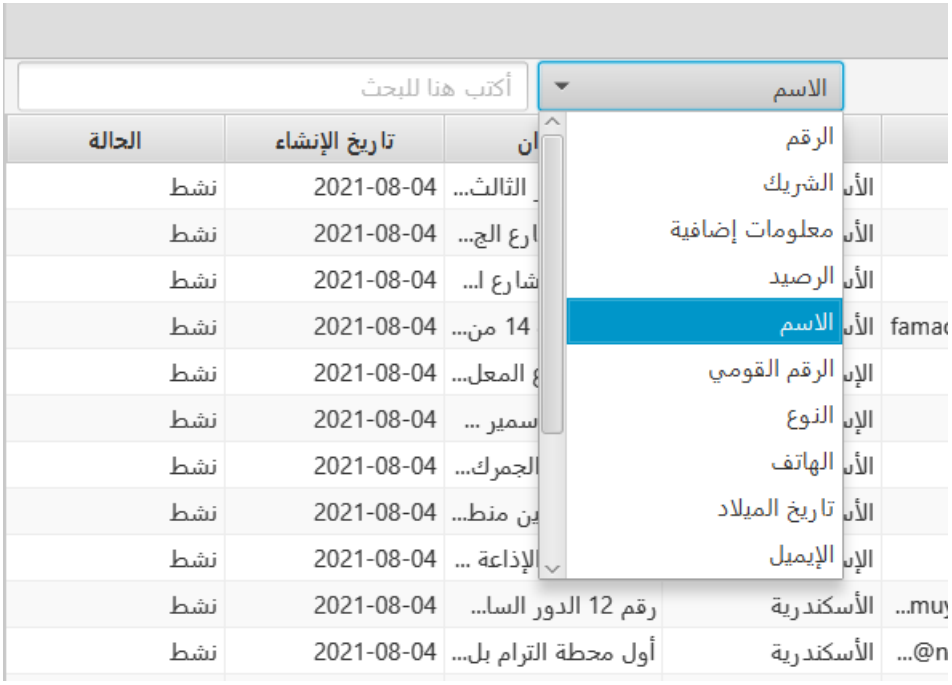
- Sub GUIs

Many Customized sub-GUIs are made for data insertion and updating data of objects



- Search box

Main table supported by immediately search boxes



Code Examples:

Travel class code:

Included encapsulation and

```
import java.util.List;


public class Travel {
    private int id;
    private String title;
    private Date creatDate;
    private String idle;
    private Date startDate;
    private Date endDate;
    private double price;
    private static Travel selectedTravel;
    private List<Campaign> campaign;
    private List<RegTourist> regTourist;
    private List<RegGuide> regGuide;
    private List<RegHostel> regHostel;
    private List<RegTransport> regTransport;
    private List<RegPlace> regPLace;

    public Travel() {
    }

    public int getId() {
        return id;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public Date getEndDate() {
        return endDate;
    }
    public void setEndDate(Date endDate) {
        this.endDate = endDate;
    }
    public Date getStartDate() {
        return startDate;
    }
    public void setStartDate(Date startDate) {
        this.startDate = startDate;
    }
    public Date getCreatDate() {
        return creatDate;
```

```
60
61      public static List<String> getSPList(String cls) throws Exception{
62          List<String> list = new ArrayList<>();
63
64          String sqlString = "SELECT DISTINCT [name] FROM [ServiceProvider] WHERE [class] = '"+cls+"';";
65          ResultSet res = DBConnector.getConnection().createStatement().executeQuery(sqlString);
66          while(res.next())
67              list.add(res.getString("name"));
68
69          return list;
70      }
71      public static ResultSet getSelectiveData(String tableName, String columnName, int value) throws Exception{
72           String sqlString = "SELECT * FROM ["+tableName+"] WHERE ["+columnName+"] = "+value+";";
73          ResultSet res = DBConnector.getConnection().createStatement().executeQuery(sqlString);
74          return res;
75      }
76       public static ResultSet getSelectiveData(String tableName, String columnName, String value) throws Exception{
77           String sqlString = "SELECT * FROM ["+tableName+"] WHERE ["+columnName+"] = '"+value+"';";
78          ResultSet res = DBConnector.getConnection().createStatement().executeQuery(sqlString);
79          return res;
80      }
81
```

Polymorphism application in getSelectiveData method

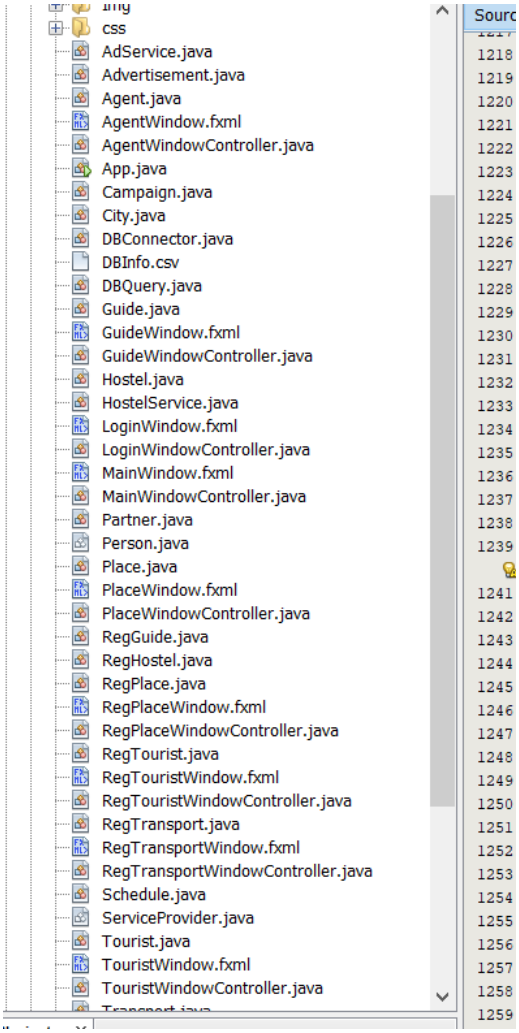
```
public static List<Travel> getResTravel(ResultSet res) throws Exception{
    List<Travel> list = new ArrayList<>();

        while (res.next()) {
        Travel t = new Travel();
        t.setId(res.getInt("id"));
        t.setTitle(res.getString("title"));
        t.setCreatDate(res.getDate("creatDate"));
        t.setStartDate(res.getDate("startDate"));
        t.setEndDate(res.getDate("endDate"));
        t.setPrice(res.getDouble("price"));
        t.setIdle(res.getBoolean("idle"));
        t.setCampaign(getResCampaign(getRegistration("Campaign",res.getInt("id"))));
        t.setRegTourist(getResRegTourist(getRegistration("RegTourist",res.getInt("id"))));
        t.setRegGuide(getResRegGuide(getRegistration("RegGuide",res.getInt("id"))));
        t.setRegHostel(getResRegHostel(getRegistration("RegHostel",res.getInt("id"))));
        t.setRegPLace(getResRegPlace(getRegistration("RegPlace",res.getInt("id"))));
        t.setRegTransport(getResRegTransport(getRegistration("RegTransport",res.getInt("id"))));


        list.add(t);
    }

    return list;
}
```

Creating object and its owned objects

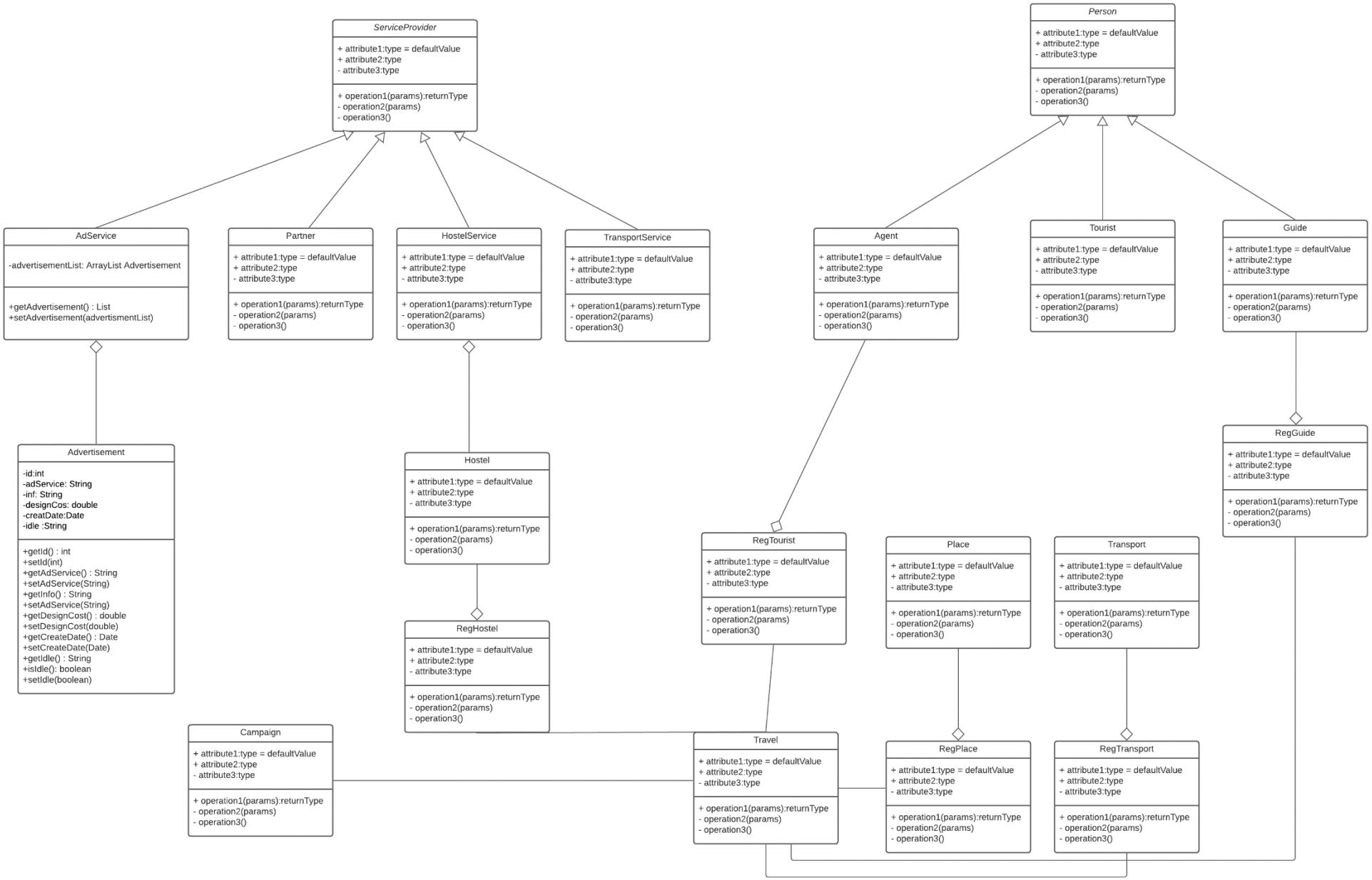
composition

8 OOP hierarchy

- Classes Are:
  - Tourist
  - Guide
  - Agent
  - Travel
  - Hostel
  - Transport
  - Advertisement
  - Place
  - AdService
  - Campaign
  - City
  - HostelService
  - Partner
  - RegGuide
  - RegHostel
  - RegPlace
  - RegTourist
  - RegTransport
  - Schedule
  - ServiceProvider
  - TrvReport
  - Person



- There are tow abstract classes ‘Person’ that is the parent of ‘Agent’, ‘Tourist’ and ‘Guide’ classes and ‘ Service Provider’ class the parent of all company type classes ‘Hostel Service’ , ‘Transport Service’ , ‘Ad Service’ and ‘Partner’ classes
- Travel information represented in object of ‘Travel’ class that owns its history with other classes in ‘Reg Tourist’, ‘Reg Transport’, ‘Reg Guide’, ‘Reg Hostel’, ‘Campaign’ and ‘Reg Place’
- Service provider objects owns its services objects such as Hostel and Transport and Advertisement
- Registration objects owned by travel aggregates with classes registered with travel


ServiceProvider
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Person
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
AdService
-advertisementList: ArrayList Advertisement
+getAdvertisement() : List
+setAdvertisement(advertismentList)
Partner
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
HostelService
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
TransportService
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Agent
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Tourist
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Guide
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Advertisement
-id:int
-adService: String
-inf: String
-designCos: double
-creatDate:Date
-idle :String
+getId() : int
+setId(int)
+getAdService() : String
+setAdService(String)
+getInfo() : String
+setAdService(String)
+getDesignCost() : double
+setDesignCost(double)
+getCreateDate() : Date
+setCreateDate(Date)
+getIdle() : String
+isIdle(): boolean
+setIdle(boolean)
Hostel
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
RegHostel
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
RegGuide
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
RegTourist
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Place
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Transport
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Campaign
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Travel
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
RegPlace
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
RegTransport
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()