GyF

(Grow your firm)

1. Introduction

GyF is an application designed to help every company on earth to have access to a variety of resources based on their preferences, needs and services they provide. This way, anyone can achieve their goals easier and faster than ever before.

1. Description

GyF is a desktop application, with an user friendly interface which offers the following features:

* Login / Logout / Register
* Add / Remove / Modify / View a company
* Add / Remove / Modify / View a resource
* Add / Remove / Modify / View a service
* Create / Remove / Modify / View a request
* Accept request / Make a deal
* Add / Edit / Remove a certificate

Use cases:

* Firstly, any user must login into the application. If the user does not own an account, he must create one in order to use the application.
* Any user can create / edit or remove (requires permissions) a company he belongs to. Also, he can add / edit / remove a resource, a service, a certificate or request or accept a request in the name of the company. He can also search for available resources and to send a deal.
* Before any transaction a certificate of a company must be validated by an admin user known as the “God” (user with id 0).

1. Tables

*Users*: (a table which retains the registered users of the application)

Rows:

* userId ( number, primary key, not null)
* firstName (varchar(50), not null)
* lastName (varchar(50), not null)
* email (varchar(100), not null)
* cellphone (varchar(25))
* password (varchar(32), not null) – this row will retain a md5 of the user password.

*Companies:* (a table which stores details about each company)

Rows:

* companyId(number, primary key, not null)
* name (varchar(100), not null)
* description (varchar (5000))
* location (varchar (100), not null)
* website (varchar (100)
* certificateId (number, not null)

*Owners:* (a table which stores the relation between an user and a company – an user can belong to multiple companies and a company can have multiple users)

Rows:

* userId (number, not null)
* companyId(number, not null)

primaryKey (userId, companyId)

*Services:* (a table which stores the services of a company):

Rows:

* serviceId (number, primary key, not null)
* companyId (number, not null)
* name (varchar(100), not null)
* description (varchar (5000), not null)

*Resources:* (a table which stores the available resources):

Rows:

* resourceId (number, primary key, not null)
* companyId (number, not null)
* name (varchar(100), not null)
* description ( varchar(5000), not null)
* cost (number, not null)

*Requests:* (a table which stores the required resources for a service):

Rows:

* requestId(number, not null, primary key)
* serviceId (number, not null)
* resourceId(number, not null)
* companyId (number, not null)
* status (varchar(10), not null)

*Certificates:* (a table which stores all the certificates)

Rows:

* certificateId(number, not null, primary key)
* userId (number, not null)
* companyId (number, not null)
* authority (varchar(100), not null)
* verified (Boolean, not null)

Relations between the tables:

* an user can register multiple companies and a company can be owned by a single user (owned table)
* an user can register multiple certificates and only one certificate is assigned to a company (certificates table)
* an user can register multiple services and resources and perform various requests for a company (request table)

