Department of Electrical and Computer Engineering
University of Puerto Rico
Mayagüez Campus

ICOM 5016 – Introduction to Database Systems
Fall 2017

Term Project – Backend System for Disaster Site Resources Locator
Phase I – Conceptual Design

E-R Model Report

December 5th, 2017

Christian Cancel 802-12-1070
Michael J. Mercado 844-10-4968
Neysha M. Matos 802-11-4385

As part of the deliverables for phase 1 of the project, we have designed a E-R Diagram illustrating the data to be stored in our site. The aforementioned has been attached along with this report. The diagram includes the following entities: Administrator, Person in need, Supplier, Requests, Stock, Transactions, Resources and Category. A brief explanation of each entity follows:

Administrator: This table contains the information of the Administrator. This entity stores the record of each administrator of the site. The administrator will validate requests and transactions and well as overall duties of the site. Its attributes are:

- **A_id:** Serial Primary Key. The entity's primary key.
- **A_Name:** This attribute will store the Administrator's name, up to 50 characters.
- **A_Location:** Stores the town the administrator is located in.
- **A_GPSC:** A float attribute, this contains the GPS coordinates of the person in need.

Person In Need: Abbreviated as PIN in many of our code, Person In Need entity follows a similar schema to the administrator entity. This entity stores the records of each person in need participating in the site, registered as person in need. It's attributes are:

- **PIN_id:** Serial Primary Key. The entity's primary key.
- **PIN_Name:** This attribute will store the Person In Need's name, up to 50 characters.
- **PIN_Location:** Stores the town the person in need is located in.
- **PIN_GPSC:** A float attribute, this contains the GPS coordinates of the person in need.
- **PIN_payment:** This attribute contains the payment method the person in need will use.

Supplier: Supplier entity follows a similar schema to the administrator and Person In Need entity. This entity stores the records of each supplier participating in the site, registered as supplier. It's attributes are:

- **S_id:** Serial Primary Key. The entity's primary key.

- **S_Name:** This attribute will store the Supplier's name, up to 50 characters.
- **S_Location:** Stores the town the Supplier is located in.
- **S_GPSC:** A float attribute, this contains the GPS coordinates of the supplier.

Requests: Request entity is the bridge between a Person in Need and the Resources it wants. This table will serve as a record for all the requests made. Also, it will hold the id of the person making a request. Lastly the quantity of resource requested by a Person in Need.

- **Req_id:** Serial Primary Key. The entity's primary key.
- **R_id:** Foreign key for the Resource entity.
- **PIN_id:** Foreign Key to Person in Need that requests a resource ID.
- **Req_Qty:** An integer value containing the quantity of a resource request.

Stock: The stock entity will serve as the bridge between supplier, transaction and resource. This entity will be in charge of showing available resources. Also, it checks which supplier provides each resource.

- **ST_id:** The stock's primary key, in serial.
- **S_id:** A foreign key to the Supplier entity for supplier.
- **R_id:** A foreign key to the Resources entity for the desired resource.
- **ST_Res:** Boolean value for reserved Resources.
- **ST_Qty:** Stock quantity of requested resource.
- **ST_PPU:** A float type containing the total price per unit.

Transactions: Transactions entity will serve as a record for the transactions that will occur on the website. To keep track of the transaction, the table stores the stock id, and the person in need id. In future phases, it is within our plans to add additional attributes. Its attributes are currently:

- **T_id:** The transaction's primary key, in serial.
- **ST_id:** A foreign key to the Stock entity for the desired resource.

- **PIN_id:** A foreign key to Person in Need that is requesting the resource in the transaction.
- **T_total:** A float type containing the total price of a purchase.

Resources: The Resources table will keep track of all resources in the database, regardless of whether or not it is currently in stock. It will have stored the resource's name, id, and its category. Its attributes are:

- **R_id:** The resource's serial primary key.
- **C_id:** A foreign key for the resource's category.
- **R_name:** The resource's name.

Category: The Category is an entity that will classify the resources. Its attributes are:

- **C_id:** The category's serial primary key.
- **C_Name:** the name of a given category.

The most important part of the diagram, is the relationship between the entities explained above. A brief explanation of each relationship is as follows:

The **Person In Need** entity connects to the **Requests** entity. Their relationship is one to many, as in one person in need may have many requests however the requests belong solely to the person who requested it. A person in need may solicit a request. Each time a request is submitted, a new and unique record in the Request entity is created.

The **Request** entity verifies that the resource or resources being requested in the **Stock** are available. Also, it has the information of each request made, the person who made it and the resources they are requesting.

The **Supplier** entity connects to the **Stock** entity. Their relationship is many to many, as in many suppliers may have different goods in stock, and the stock may contain goods that other suppliers may offer as well. A supplier provides goods that he has in stock. A supplier may add what supplies he has available in the stock entity.

The **Stock** entity connects to the **Resource**, **Transaction**, and **Supplier** entities. The relationship between resources is many to many because many suppliers can have more than one resource and many resources can be in a stock. The relationship with **Transaction** is one to one.

The **Transaction** entity connects to the **Stock** entity in a many to many relationship, as a transaction may have multiple resources from stock as well as any given resource, as long as there's still left in stock, can be in multiple transactions. Only once the amount of resources from the request has been verified with the stock, can the transaction go through for validation with an administrator.

The **Resources** entity connects to the **Category** entity in a many to one relationship, because many resources can belong to a category, but a resource should only belong to a single category. Each resource has a category such as fuel, food, medicine, etc.

The **Administrator** entity will validate the **Requests** being made by the Person In Need. This relationship is many to many, as in many administrators will be able to validate different requests. For security reasons, more than one administrator will be able to verify the same Request.