VERSION 0.4

APRIL 17, 2017

UML DESIGN

MPP CS401 – SHAFQAT ALI SHAD

PRESENTED BY: GERMÁN SEGURA, ERDENEZAYA ERDENEPIL

MUM

1000 N 4ST

PURPOSE

A CRM is a tool to manage customer interactions throughout the customer lifecycle, with the objective of improving business relationships with customers. We want to develop an Insurance Broker CRM to be used for individual brokers, to have a Database of clients and have a different kind of features to improve their sales.

FEATURES
Client Manager: This module permit to create, update and delete the information of the clients, each client has a category (Lead, Customer, Contact and Opportunity)
Client Relations: This is a submodule of Client Manager, permit to create relations between different clients and give a discount for each sale: spouse (15%), child (13%), familiar related (6%):
Actions: This Module Permit to create different actions concerning to contact a client like: Email, call, meetings; each one has to be recorded in the Database for easy consult, these actions has a date of execution, and hour and a duration.
Activities: Permits to retrieve the different actions concerning to a specific client with different filters like: period ranges, type of action.
Campaigns: This module permit to send a message to different clients, (There is no contemplated to send email through this system)
Sales: This module permits to create the different insurance sales for each client, each type of insurance has a different percent of gain: Health Insurance (10%) Car Insurance (7%) Home Insurance (12%) Life Insurance (11%) Disability Insurance (9%). With these percent the module generate the total gain

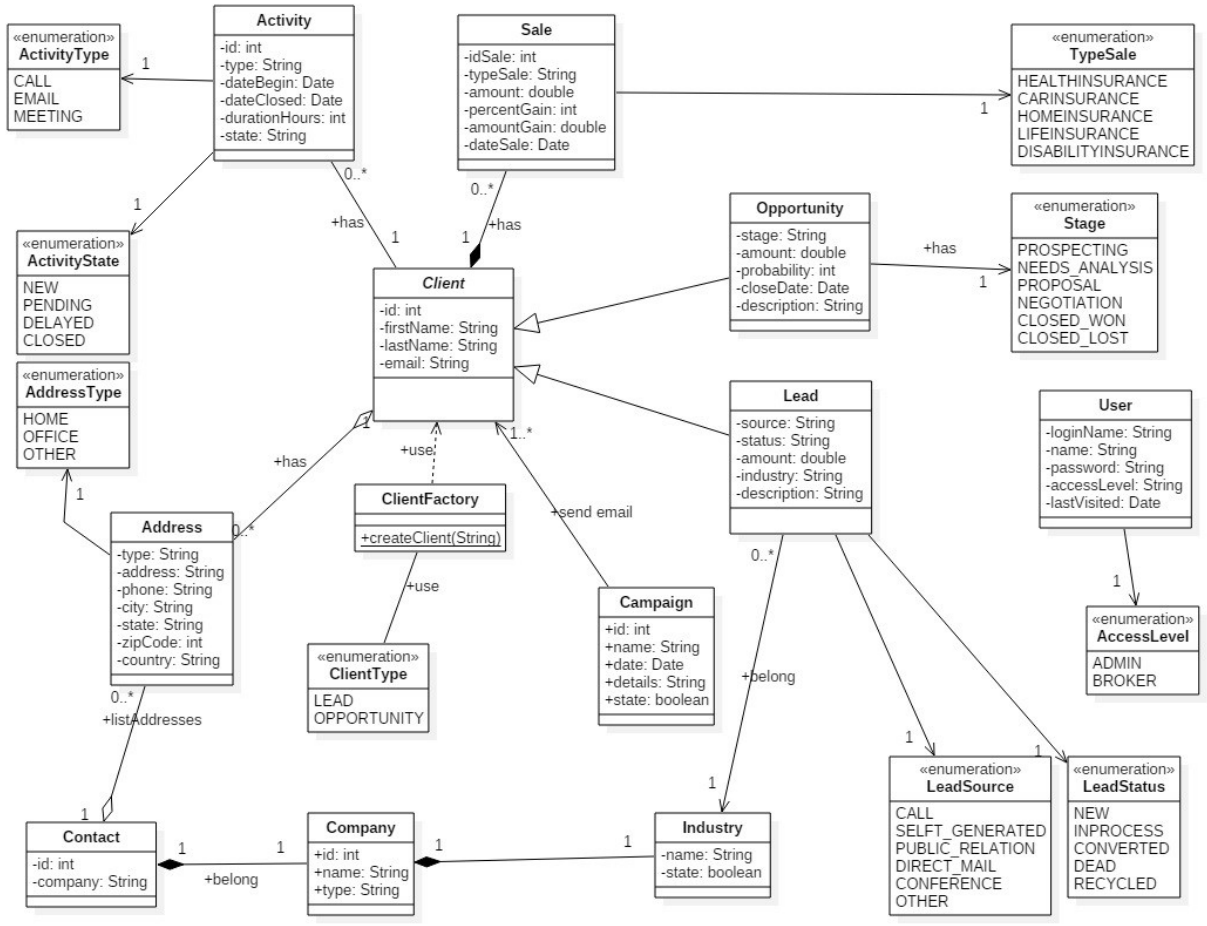
TECHNOLOGYS

This project is a Web Based Project, we used Servlet with Java 1.8, the Frontend layer is built with JSP, HTML5 forms, Datatables for the grids, Ajax to perform the request for the data, Json I used for the data interchange format, the servlet request some services to have a separate layer for the persistence layer. MySql for database. We use GitHub like a repository, and Eclipse EE Neon like a IDE.

ROLES

- Germán Segura: FrontEnd and Servlets
- Erdenezaya Erdenepil: Persistence layer and services

CLASS DIAGRAM



DATA DICTIONARY

CLIENT

The class client is a Domain Class because the objective of the entire Software is to record information of the clients, there are two type of clients: Lead and Opportunity, this class is going to be an abstract class with common attributes for both clients: firstName, lastName, email.

LEAD

This type of client is created when there is only part or all the information of the client, inherits form Client Class. Has a source to know how was created this lead, a status to know is new, etc.; and amount to try to know the possibility of a sell, industry to try to specify form the client come.

OPPORTUNITY

An opportunity is created when there is an opportunity to sell one insurance to a client, so there is an amount of the sell, a probability of the sell and a close date when is going to be the sale completed, a description related to this opportunity, inherits form Client Class.

SALE

This class is one of the most important, this class has a type of the sale, it means in this class is going to be the different types of insurance, total of the sale and also the gain for each sale, also the date when the sales has been done. Also this information is going to be showed in some graphics in the home of the application.

ACTIVITY

When the broker is working in an opportunity he has to make different actions to contact the client, so, these class is going to have all the calls, meeting and emails that the broker has performed to obtain the opportunity positive, so is going to have the type of the action, the date of the beginning and close, the total duration in hours and the state (inactive, active). This activities in a future cab be good to be showed in a calendar.

ADDRESS

The client has different kind of addresses, so this class is going to have the responsibility to have all the related information. The type are Home, Office and other; the address, the number phone, and all the fields related to the location like city, state, zip code and country.

CAMPAIGN

These class is created because the broker is going to create campaigns sending an email to all his clients, so each campaign has a name to identify each campaign, a date to send the emails, the details that is all the information that is sending in the email, and a state for to know is an active campaign or inactive campaign.

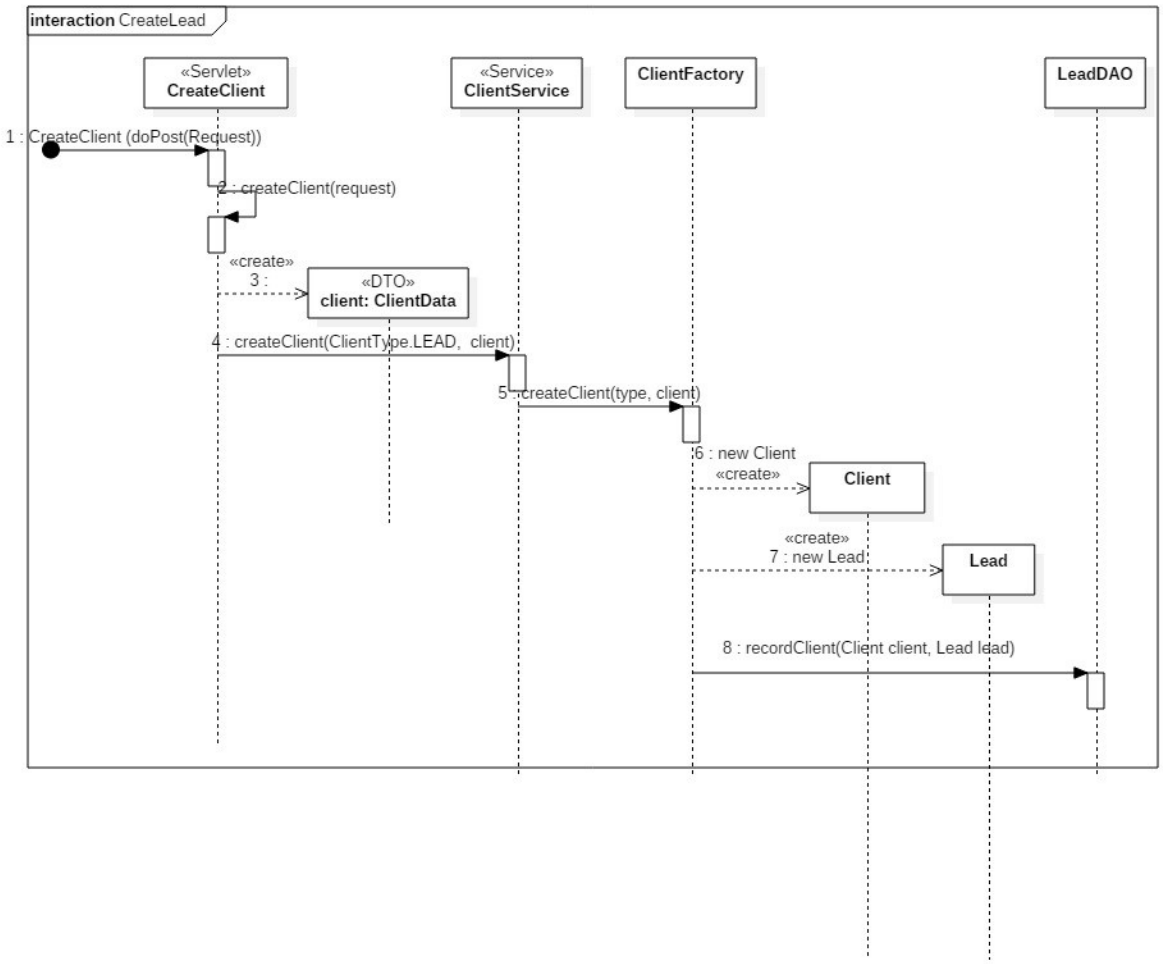
CONTACT

These class is created because is import to have a contact calendar to request different phones from different companies to perform calls and offer all the services that the broker offer. Is has two classes to maintain the company information like Company and Industry.

SEQUENCE DIAGRAMS

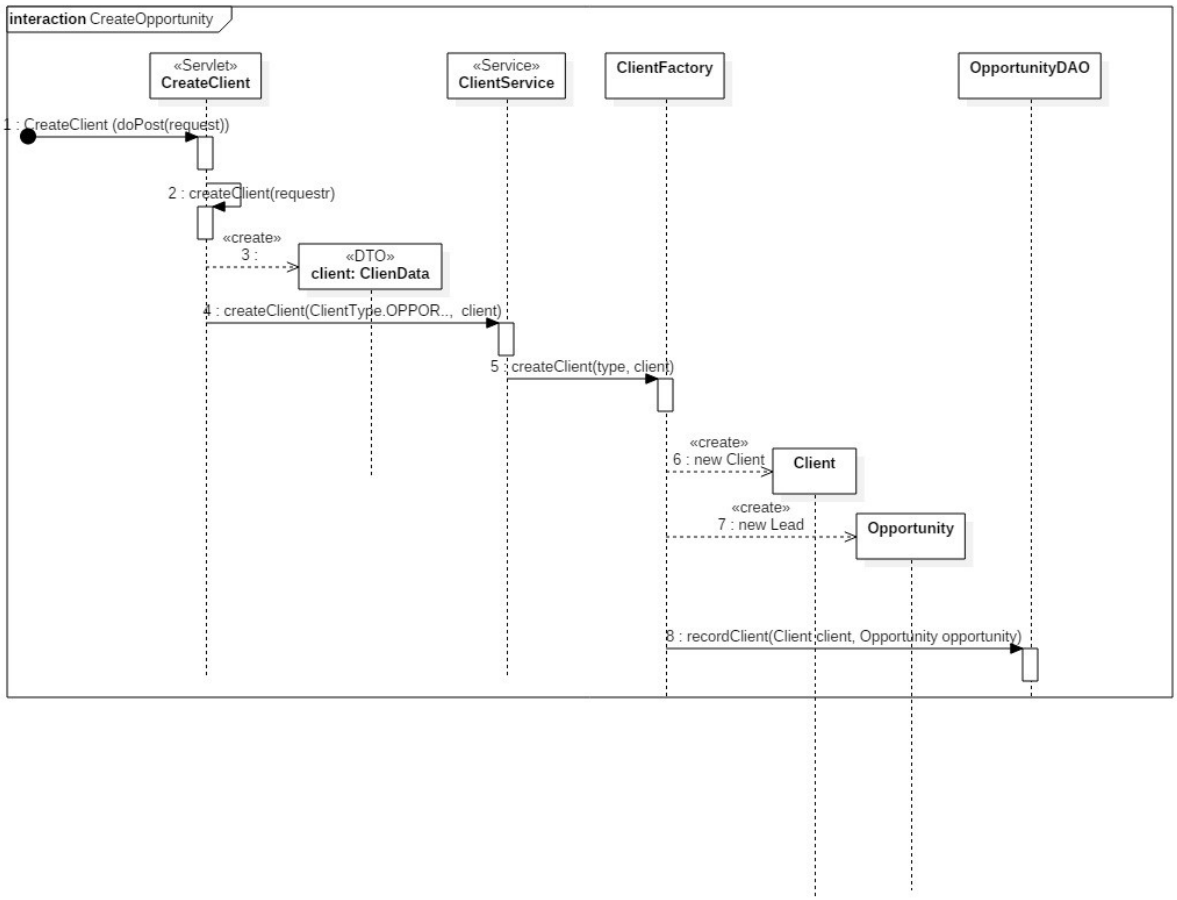
CREATE LEAD

This sequence describes how the client perform a petition to create a new client lead, is going to request to the Servlet CreateClient in its doPost method, the parameter is an HTTPRequest that has all the parameters of the form, is going to build a DTO to put all the information of the client and send it to the service who is in charge of call the factory to create and opportunity or a lead, then the factory call the service to persist the data



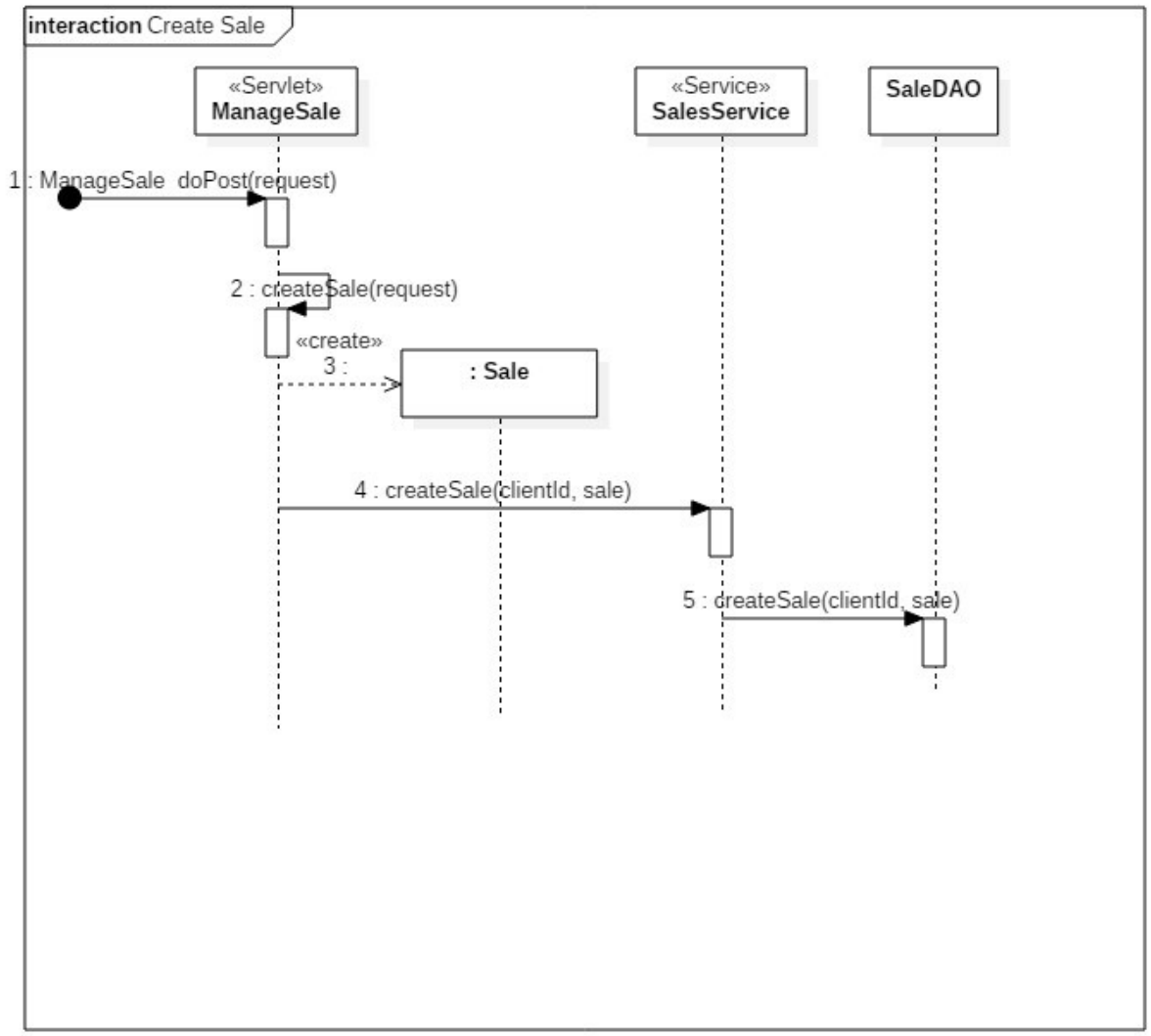
CREATE OPPORTUNITY

This sequence describes how the client perform a petition to create a new client lead, is going to request to the Servlet CreateClient in its doPost method, the parameter is an HTTPRequest that has all the parameters of the form, is going to build a DTO to put all the information of the client and send it to the service who is in charge of call the factory to create and opportunity or a lead, then the factory call the service to persist the data



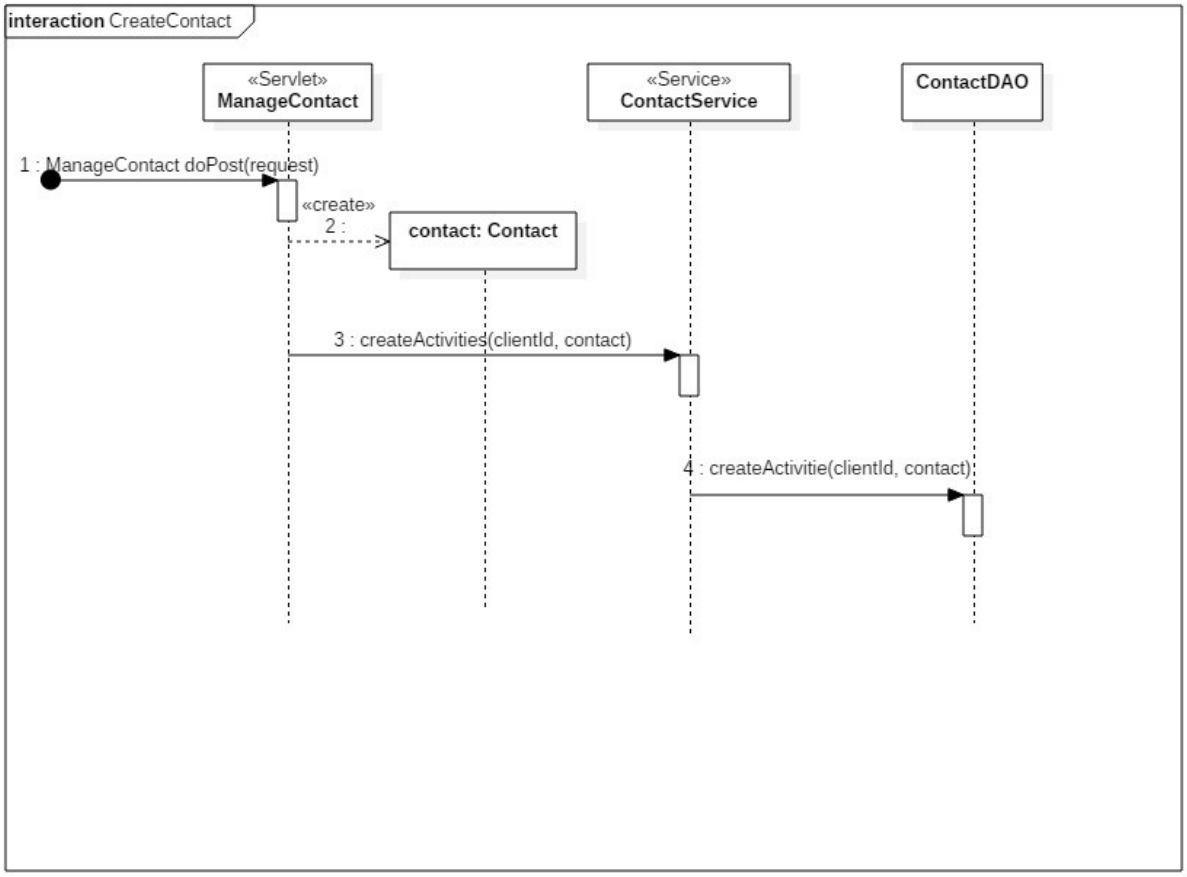
CREATE SALE

The sale is going to be processed by the client selecting first an opportunity and filling out a form with all the related data, then a Servlet call ManageSale is going to create one instance of the class Sale and request the service to persist the data.



CREATE CONTACT

The contact is going to be processed by the client filling out a form with all the related data, then a Servlet call ManageContact is going to create one instance of the class Contact and request the service to persist the data.



CREATE ACTIVITY

The activity is going to be processed by the client filling out a form with all the related data, then a Servlet call ManageActivity is going to create one instance of the class Activity and request the service to persist the data.