University of Puerto Rico
Mayagüez Campus

CIIC 4010/ICOM 4015: Advanced Programming Fall 2017

Programming assignment 3: TinyCRM

Due date: January 18, 2017 11:59PM

In this project, you will work in pairs. The objective of the project is to add functions to a very simple customer relationship management (CRM) system. A CRM maintains records primarily of companies (customers), contact information of people (contacts) associated with these companies, and finally, sales opportunities of each company. Some examples of popular commercial CRM's are SalesForce, SugarCRM and SOHO. The latter provides free accounts with some limitations. We recommend that you obtain an account and examine its functionalities.

TinyCRM is our modest CRM developed for the class and it keeps very simple records of clients and contacts and it was developed in Java as a desktop application using Java Swing. Once you register in GitHub Classroom by following the link below, you must proceed to test the application and study its architecture.

TinyCRM was designed following the Model-View-Controller pattern which facilitates the combination of multiple ways to visualize data (views) with multiple ways of representing and storing data (models). You can find lots of information and examples of MVC on the Internet. One of these would be:
https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm.

Your work consists of adding 3 additional functionalities to TinyCRM:

Functionality 1: Validations and fields' formats

Use the Java Pattern and Matcher classes to validate the information that users enter in all fields of customer and contact screens. The validation must prohibit the entry of data that is not specific to the corresponding information field; People's names should only include letters and not digits or other symbols. Email and web addresses must comply with the standard rules. The validation must be implemented following the methods of the controllers provided for these purposes. Once you validate the fields, they should be displayed following a uniform, easy-to-read format. The proper names must begin in capital letters and the telephones must follow the usual format (xxx) xxx-xxxx. Remember that validation and formatting are topics that must be addressed separately to facilitate user input. For example, if a user entering the 11 digits in a telephone field, the validation must not fail, but the format must be fixed.

Functionality 2: Opportunities module

Design a module of opportunities (sales). Each sale must be associated with a customer. In addition, each sale must include a description of the sale, the dollar amount, the estimated closing date of the sale, and the status of the sale (verbal proposal, written proposal, won, lost) among others. You can be inspired by commercial CRMs to determine which fields and what type they should be. The opportunity module must have its own model, controller and display. You must study the design of existing modules to follow the same pattern.

Functionality 3: Contact list panels and opportunities on Customer screen

Add panels to the customer screen that include listings of opportunities and contacts associated with the customer that is being displayed on the screen. The panels must have a maximum height but must allow an arbitrary number of contact records or opportunities using scroll bars.

Organization of two-person teams

You must select a DIFFERENT teammate whom you had in other projects. If you have difficulty finding a teammate, contact the teacher as soon as possible.

Code repository is in GitHub Classroom (we will not use your personal GitHub account)

Once you select your teammate, you must create your repository in GitHub Classroom by entering the following link using your GitHub account:

https://classroom.github.com/g/4YPOHBLu

The first team member to register must name the team. The second one to register must select the team of the partner who registered first so that both have permission to access the repository. The repository will be initialized with the base code automatically. From now on you can copy the repository URI to import it into Eclipse and start collaborating using Git.

Collaboration Guides (SAY NO TO PLAGIO)

During the project it is expected that the members of each team collaborate equally in the work. If you have difficulty contacting or agreeing with your partner and cannot handle it, you should contact the teacher as soon as possible. Each team member is expected to upload their individual contributions (or commits) to the repository through their own GitHub account. It will be assumed that the contribution of each team member will be the one that has uploaded to GitHub with their account.

Dialogue with your teammates from other teams and share / discuss ideas about the assignment is allowed if UNDER NO CIRCUMSTANCES YOU USE ANOTHER STUDENT OR SOURCE CODE WITHOUT DOING DUE RECOGNITION TO THE ORIGINAL AUTHOR OF SAME. All projects must be original of each team. Violation of this rule will constitute a greater lack of academic honesty and will be addressed as specified by the record. A good way to avoid plagiarism is to not show or share your code with other people.

Instructions to submit your project electronically (IMPORTANT)

The staff will correct the project by importing from the repository the latest version that the team has uploaded on or before the deadline. You do not need to send us the URI or two access to your repository.