Department of Computer Engineering 

University of Puerto Rico

Mayaguez Campus



**Team Members:**

Kevin Santiago Ortíz

José A. Rodríguez Rivera

Jariel O. Laureano Cruz

**Emails:**

[kevin.santiago1@upr.edu](mailto:kevin.santiago1@upr.edu)

[jose.rodriguez37@upr.edu](mailto:jose.rodriguez37@upr.edu)

[jariel.laureano@upr.edu](mailto:jariel.laureano@upr.edu)

**Introduction**

We live on a society that depends on science and technology. As technology advance our goal is making things easier. Things like cars, houses, and clothes advertisement are now performed online. The reason is not a mystery, is a cheaper and more efficient way to do it.

Moving on this line of reasoning, DealIt is a web application that offers a user friendly interface were can be placed sales/trades and special offers. The sales/trade transactions could be anything such as: vehicles, houses, electronics, furniture, and miscellaneous. As mentioned before sales/trades could be anything, so this web application targets everyone willing to sale/buy/trade something that fits in one of the above categories.

The technologies employed to develop DealIt are as follows:

*Front End*

* Angular JS
* Javascript
* Html
* Css
* Bootstrap

*Middleware*

* Java Play

*Back End*

* PostgreSql

**Client App Description**

DealIt web application will have the following capabilities:

1. *User registration:*

Users that want to sign up would need to supply some information such as:

full name, city, cell phone number, email and a password. After submitting their

credentials a notification for email validation would be send for completing the

process. Another option

b) *Selling:*

For posting something for sell, a form will be supplied. To those that are

already registered, the contact information will be automatically fill up. In

change, those that want to continue as guest will need to supply all the required

data, and a post code will be send via text to the user for future post editing.

A description field would be supplied and the option for uploading a picture of the

item.

c) Buy/Trade:

A few methods will be supported in terms of buying or trading items. From

the most traditional ones such as: contacting the owner through the information

supplied on the advertisement. Also is possible to pay through credit card in

some cases. All the above mentioned would depend on the product owner and

obviously what classification we are talking about.

d) Product Filtering:

The interface will count with a filter that would be capable of search trough

the classifications making easier find things within the web app.

**Server App Description**

The server side of the web application will make possible the communication between the client and the database. In this way the information from the item and the people who own them will be stored at the database. The client side of the application would be worked using Angularjs which facilitates a lot things like the ajax calls for getting or posting data and other things such as the interface response.

Our database system is PostgreSql based and will probably have some tables like the ones described below:

*Guest Users --* Will provide an ID for those users that want to post things without

registering on the web application.

*Registered Users --* Will provide an ID and store all the information from the user. Will have

have an active column to distinguish between the active users, and the

inactive ones.

*Deleted User --* The deleted registered users will be stored here (Their ID) after setting their

active column to false on the Registered Users table.

*Items --* The items posted will be stored here. Each item need a classification field that has to

be consistent with the available classifications on the Items Classification table. As

the Users table this table will also have a active column.

*Items Classification --* This table will contain all available classifications assigning them an

ID that can be used for the classification field on the Items table.

*Items Deleted --* The deleted items post would be stored here (Item ID) after setting their

active column to false on the Items table.

*User Payment Info --* This table will bind each user with the credit card that each one supplied.

*User Watch List --* Bind the user with the items that he adds to the watch list.

**Division of Labor**

For the project we will have a work distribution as follows:

* Jariel - Implementation of schema. Establish connection between the RESTful API and the database. UI implementation.
* José - Design RESTful web services, Front End / Back End integration
* Kevin - Design Schema, Back-End Development and UI Design