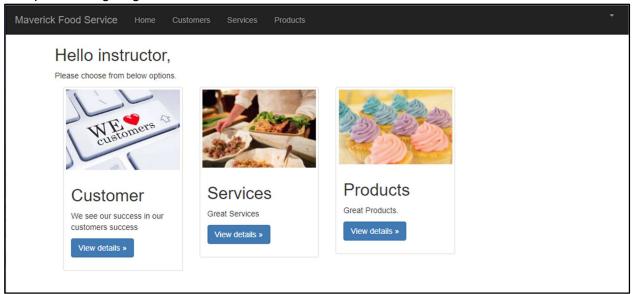
# **Maverick Food Services (MFS):**

MFS is a fictitious organization which provides food services in a campus setting. The catering manager for the food services organization needs to keep track of customers in order to grow their profits by marketing to these customers and their colleagues. Some type of a database is needed to track customers and the products and services they acquire over time. This type of system is often called a customer information database or a customer relationship management database or CRM for short. In this assignment, you will build a CRM for the Maverick Food services organization. The application shell is provided which initiates the application but the remaining functionality must be implemented and deployed to PythonAnywhere.

Sections 1-8 take you through developing part of the application and are included in a separate document. After completing Sections 1-8, complete sections 9 and 10 to add functionality to the application.

# Section 1 - Tour of the application you will be developing for Assignment 2

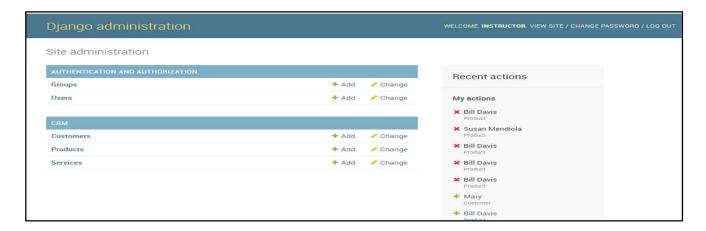
Sample Landing Page for the Maverick Food Service CRM



There are two roles in this sample application:

- The administrators
  - o have access to the information for all customers, services and products.
  - o can add, update and delete customers, services and products
  - o have access to the Admin section of Django to set up new employee accounts.
- Employees can:
  - o Add, update, and delete Customers
  - o Add, update, and delete Products and Services for each customer

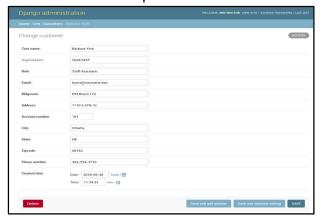
Below you see the Admin panel of the application available to administrators. As you learned in the blog assignment, you add "/admin" to the URL to access it.



The Blog assignment showed the Groups and Users in the admin panel. In this application, a given customer can acquire many services and products. Below is a view of the customers page showing the ability to add, update, and delete customers:

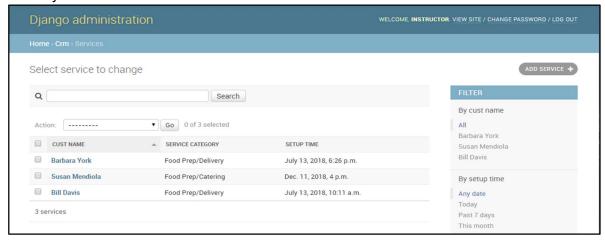


Here is an edit and update of one of the records:

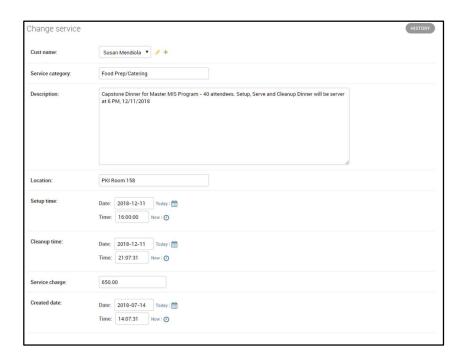


Note the history. This points out the logging capabilities provided by Django.

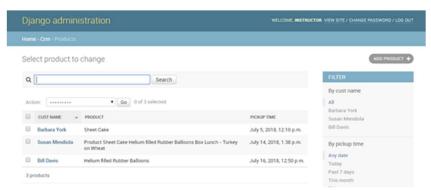
Below you find a list of the service transaction in the Service Table.



As with the customers, you can add, update and delete services for each customer.



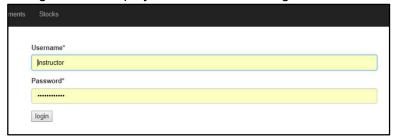
Next we have the page listing of the products purchased by the Maverick Food Service customers.



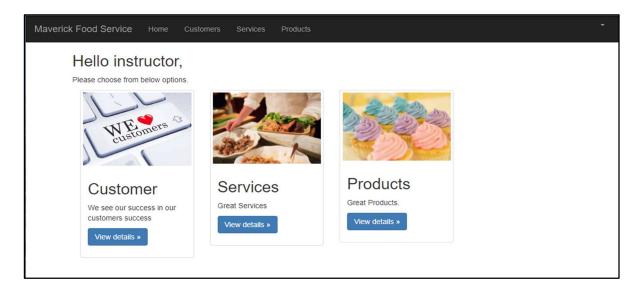
As with Services you can click on a Product and edit or delete the products.

The screens shown below show what may be accessed by any employee of Maverick Food Service. In this scenario, we will assume that employees are provided their ID and PW by the administrator.

To sign in, the employee would click the login button after entering their ID and PW:



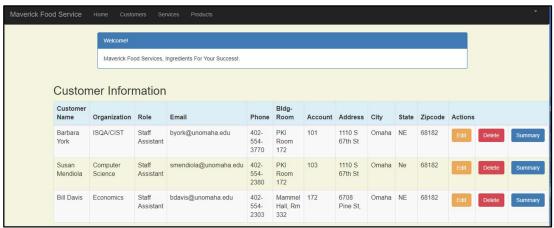
After successfully signing in using the ID and PW provided by the administrator, the employee will see the landing page below:



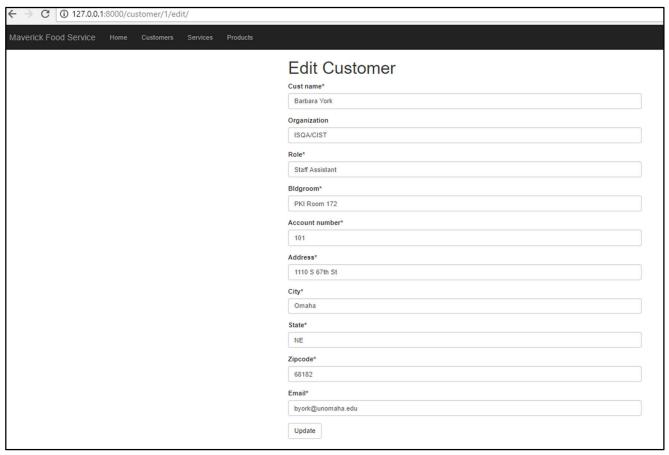
No other features such as "forgot your password" are in the assignment. This feature can be added to earn extra credit.

Employees can view a list of customers and update customer information with a page similar to the one shown below.

### **Customer Screen**

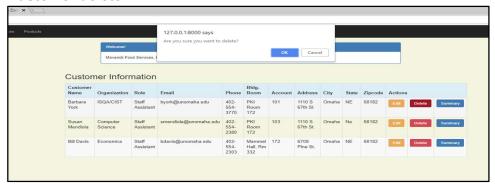


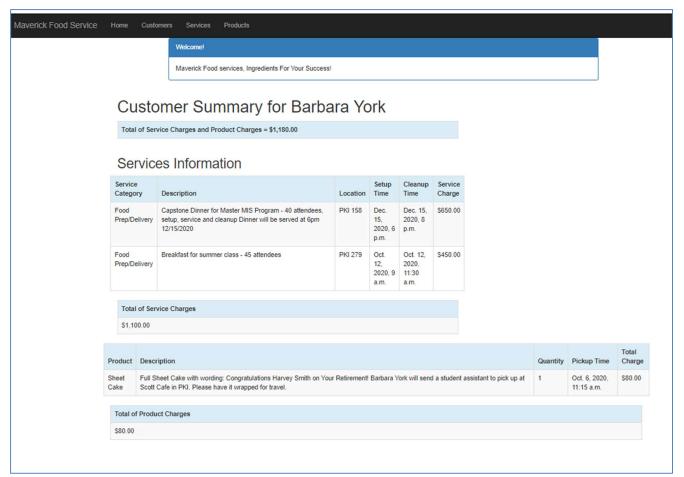
Customer Edit - Employees can edit customer information with the screen shown below:



The Customer delete button brings up a warning. **NOTE**: Since we are using the CASCADE option when setting up the model in Django, deleting a customer will also delete all information in services and products associated with that customer.

#### **Customer delete:**



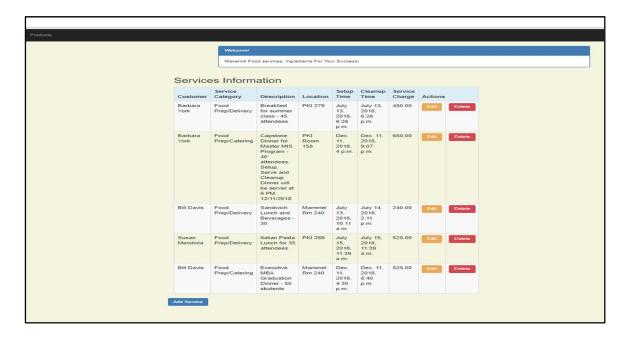


**Report -** The summary function is a report of all services and products acquired by a customer. This is discussed at the end of the tutorial.

According to the report above, Barbara York has ordered \$1,180 services and products from MFS for her department.

## **Service List:**

The service list screen allows employees of MFS (Maverick Food Service) to add services provided to customers, including those working for the university and external parties, requiring food services for events they are hosting in a UNO building. This page allows for adding a new service provided, editing the information, and deleting a service provided. Sample customer, services, and products shown.



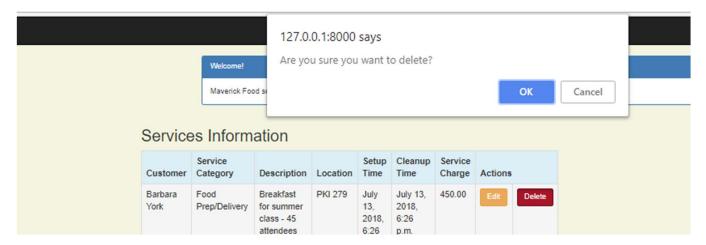
Clicking 'Add Service', brings up the following screen:



Editing an existing record in a similar manner.



Deleting a service presents a warning, but only a single service record is deleted. The customer is not deleted.



# **Products**:

All employees may add, update and delete products sold to customers using screens shown below:

## Product List



#### Add a new product



Edit a Product



Delete product works same as delete service

