CMSC 204

Assignment #2

Spring 2020

Office Depo, an office supply retailing company, donates its surpluses to colleges at the end of each year. Volunteers will help deliver packages of supplies to representatives of colleges (Recipient of supplies). Write an application to simulate delivering packages from the container of packages by the volunteers to recipients. Upload the initial files and your solution to the repository in GitHub created in Lab 1, in a directory named Assignment\_2, and take a screen shot of it.

**Concepts tested:**

Generic Queue

Generic Stack

Exception handling

**Data Elements**

**Volunteer-** Holds the relevant information for a Volunteer: *name*

**DonationPackage-** Holds the information of the package to be donated: *description, weight*

**Recipient-** Holds the information of the recipients: *name*

**Data Structures**

1. Create a generic queue class called **MyQueue** that implements the **QueueInterface** given you.
2. Create a generic stack class called **MyStack** that implements the **Stack** **Interface** given you.
3. Create a class **VolunteerLine** that implements **VolunteerLineInterface**.

This class contains a queue of Volunteers and simulates queuing and dequeuing volunteers to and from the volunteers’ line.

1. Create a class **RecipientLine** that implements **RecipientLineInterface** .

This class contains a queue of Recipients and simulates queuing and dequeuing Recipients to and from the recipients’ line.

1. Create a class **Container** that implements **ContainerInterface**.

This class contains a stack of **DonationPackage** and simulates adding and removing package from the container’s stack.

**Assumptions**

* When a volunteer enters a queue they will stay in the queue and every time that the volunteer is removed from the line they will be go back to the end of the line.
* When a recipient receives a donation package they will leave their queue.
* **Data Manager**  
  The **DonationManager** class will manage adding a new package to the container, a new volunteer to the volunteer queue line, a new recipient to the recipient queue and donating package by the volunteer to the recipient. The three methods named **managerArrayPackage, managerArrayRecipient,** and **managerArrayVolunteer** place their data elements in an array. They all call the methods in the **container, volunteer line, and recipient line** respectively, named toArrayPackage, toArrayVolunteer, and toArrayRecipient. Each of those methods calls toArray in MyStack or MyQueue.

Notice that because of type erasure in the generic MyStack and MyQueue, if their toArray methods return an array of type T, i.e., T[], they are actually returning Object[]. But since the individual elements of the array are still Packages, Recipients, or Volunteers, we can copy them one by one into a new array of the appropriate type and cast each one to the appropriate type.

**DonationManager** will implement the interface **DonationManager Interface**.

**Exception Classes**

1. RecipientException:

**public** **class** RecipientException **extends** RuntimeException {

**public** RecipientException(){};

RecipientException(String message) {

**super**(message);

}

If the user attempts to add new Recipient and the recipient line is full, throw RecipientException(“Recipient Queue is full”). If user attempts to remove a recipient and the recipient is empty, throw RecipientException(“Recipient Queue is empty”);

1. VolunteerException – similar to RecipientException except throws VolunteerException when the Volunteer Queue is full or empty
2. ContainerException – similar to RecipientException except throws ContainerException when the Volunteer Stack is full or empty

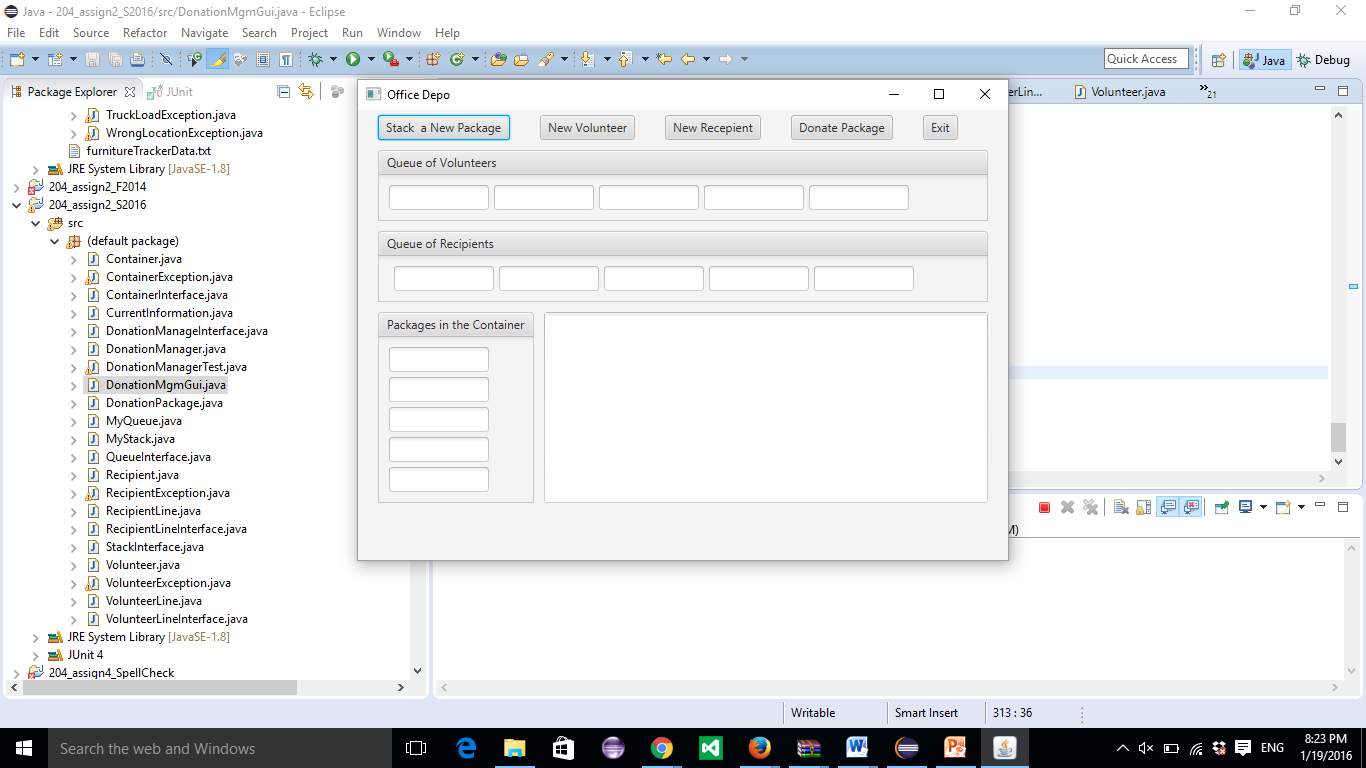
**You may add additional methods and attributes to your classes.**

**GUI Driver – provided for you**

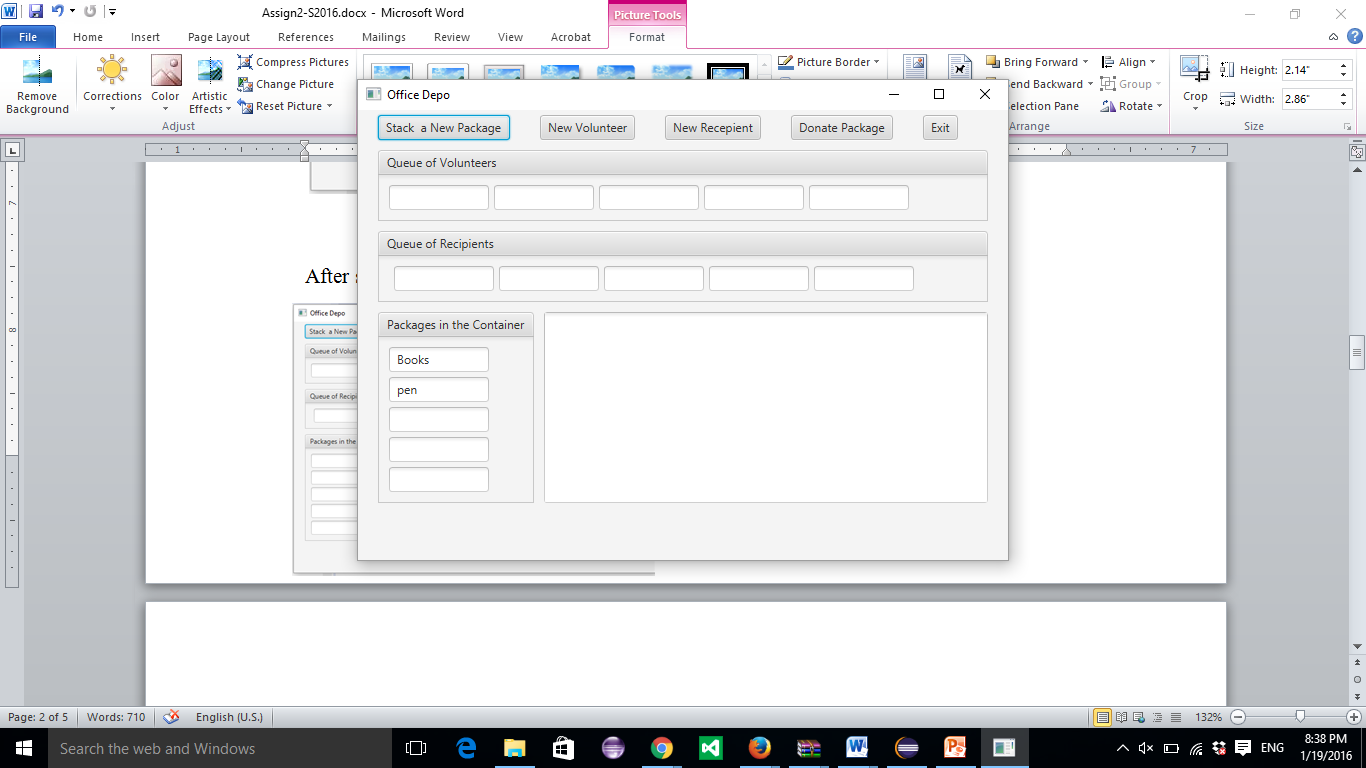
1. Initially there is no package on the container stack, or any volunteers or recipients on the queue.
2. Allows the user to load new packages to the container and add new volunteers to volunteer’s line and new recipient to recipients’ line.
3. Allows the user to simulate the process of donating a package from the container by the volunteer on the front of the Volunteer queue to the recipient on the front of the Recipient queue. When a volunteer finishes donating the package to the recipient, he/she will be placed at the back of the queue of volunteers to continue the process of donation. **Each volunteer takes one package at a time to deliver to the recipient.**

**Examples:**

At startup

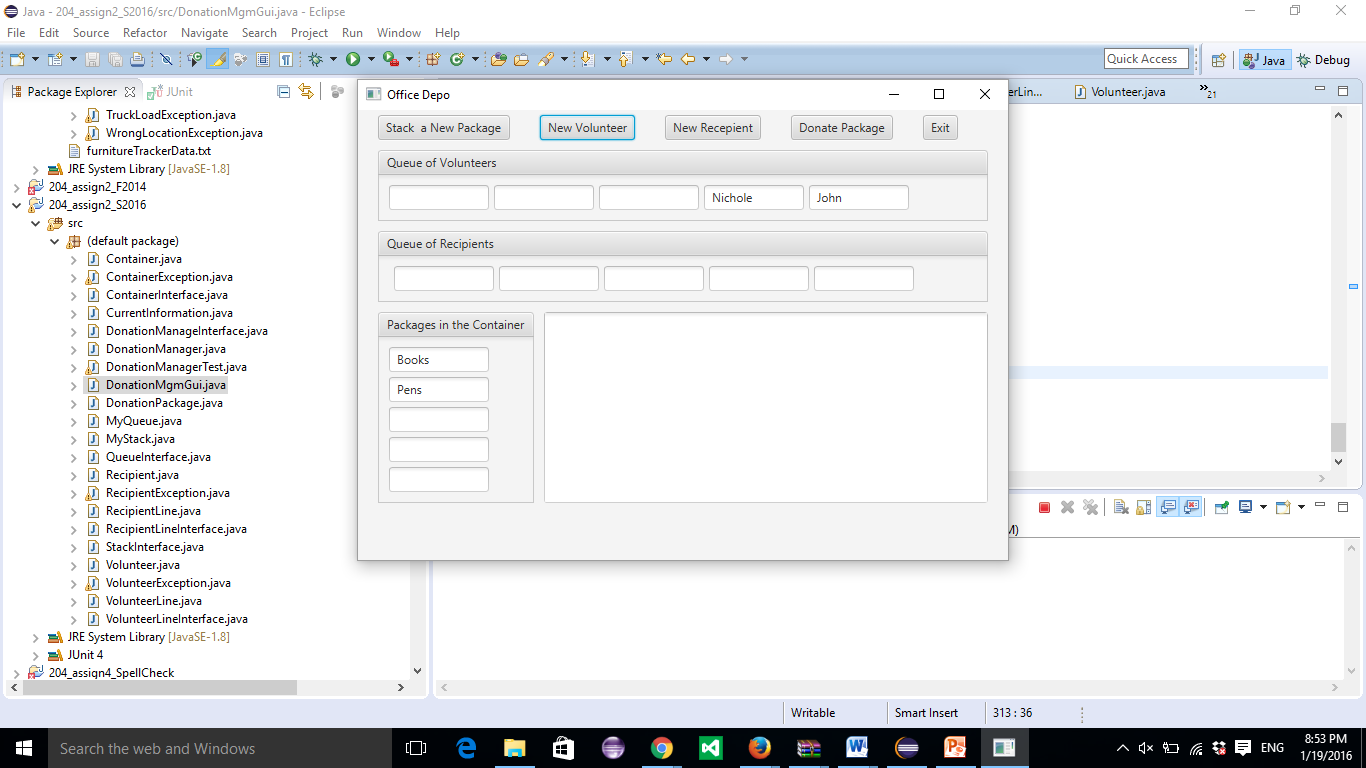
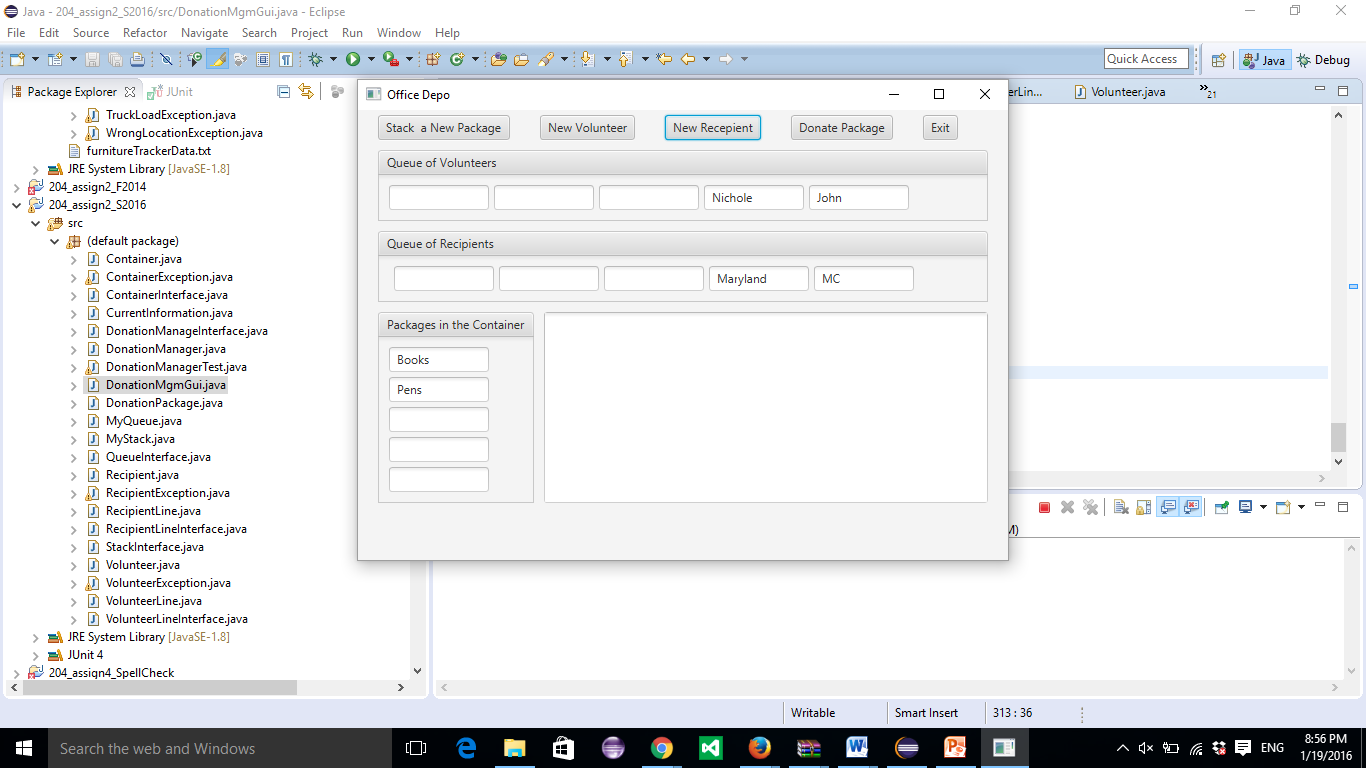


After selecting “Stack New Package”

Pen(first) and Books (second).

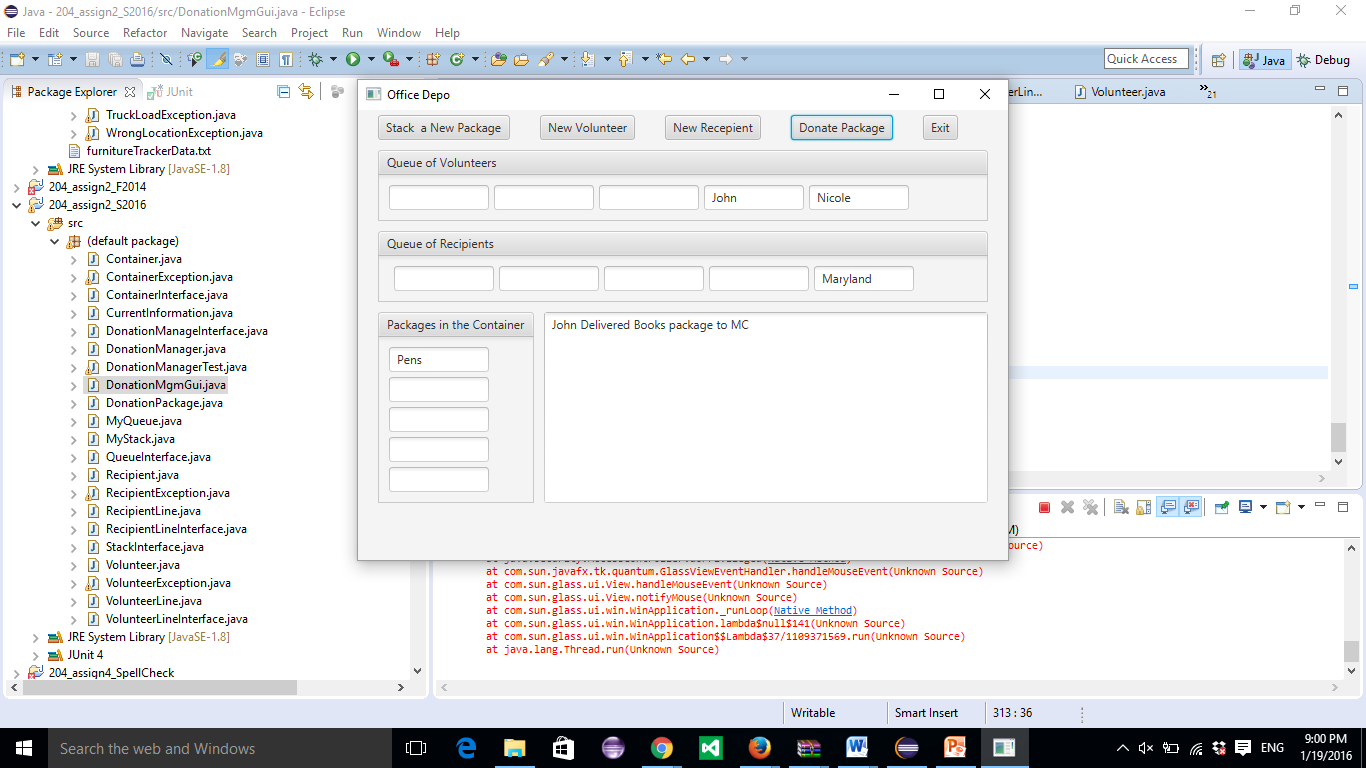
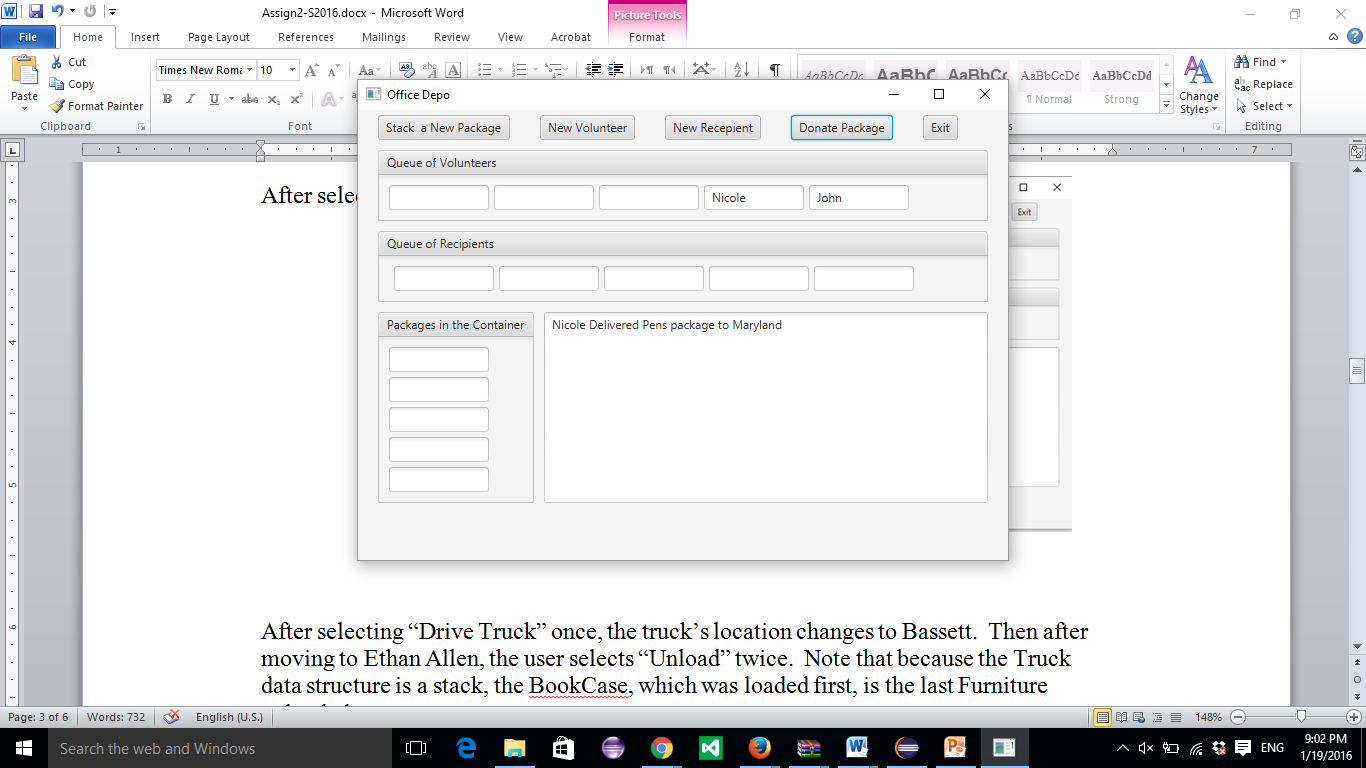
After adding new Volunteers John (first), After adding new Recipients MC (first),

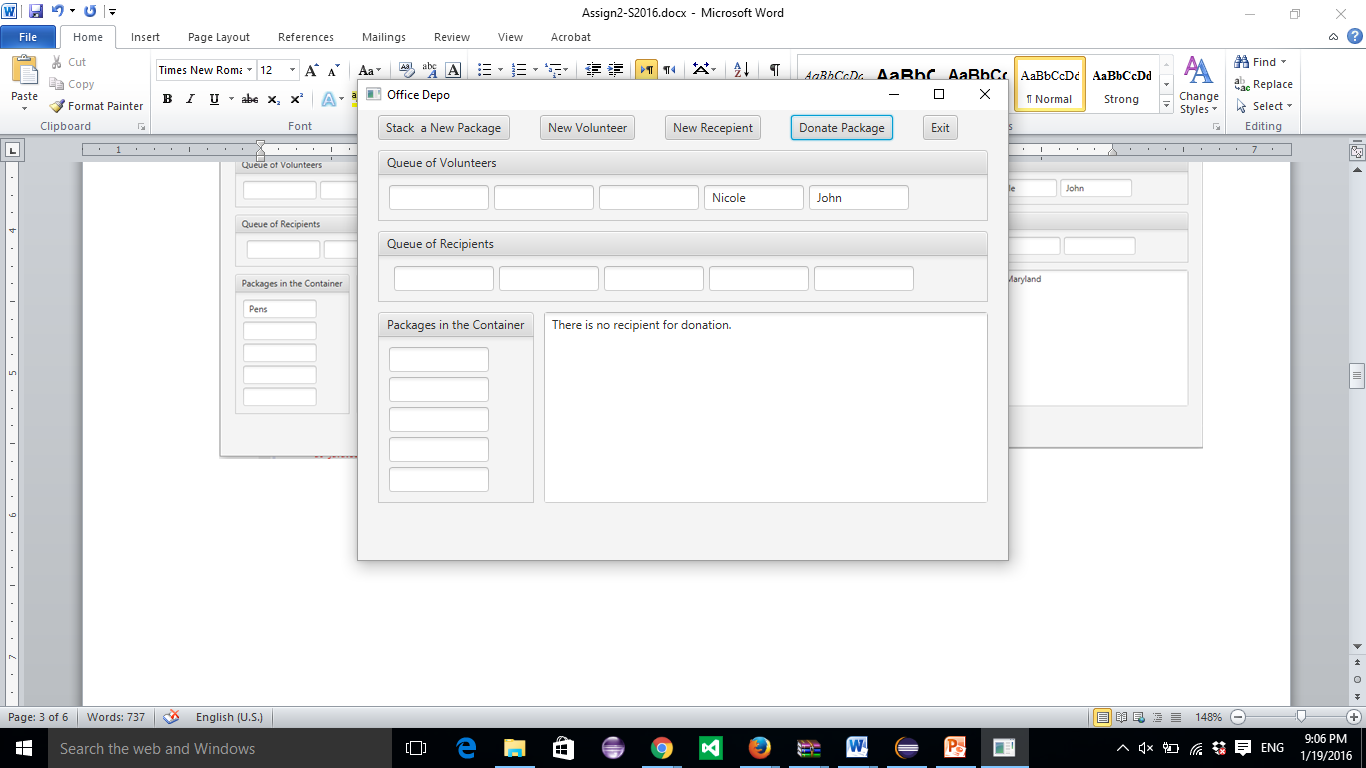
Nicole (second) Maryland (second)



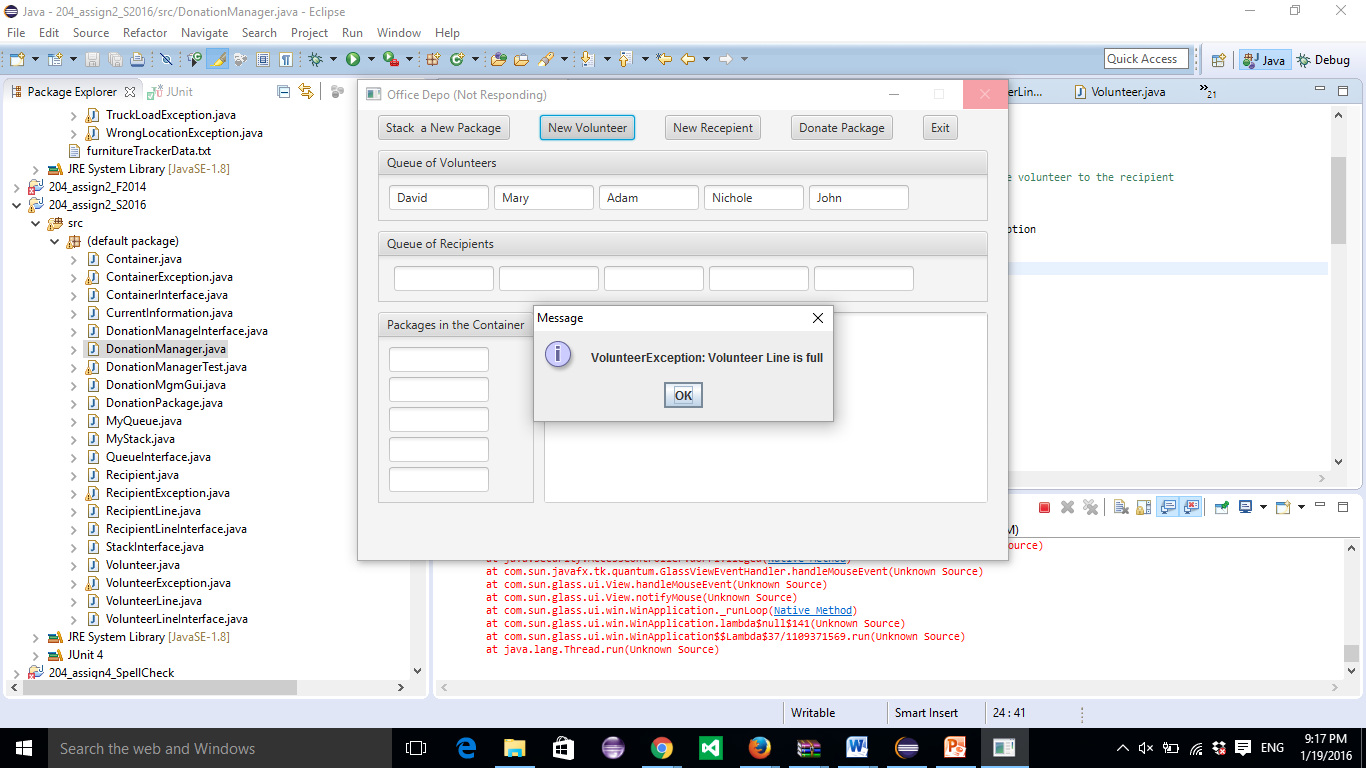
After selecting Donate Package After selecting Donate Package Second time

First Time



After selecting Donate Package Third time

If the user tries to add a new volunteer and

 The line of volunteers is full: