CMSC 204

Assignment #2

Fall 2019

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.

**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. **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

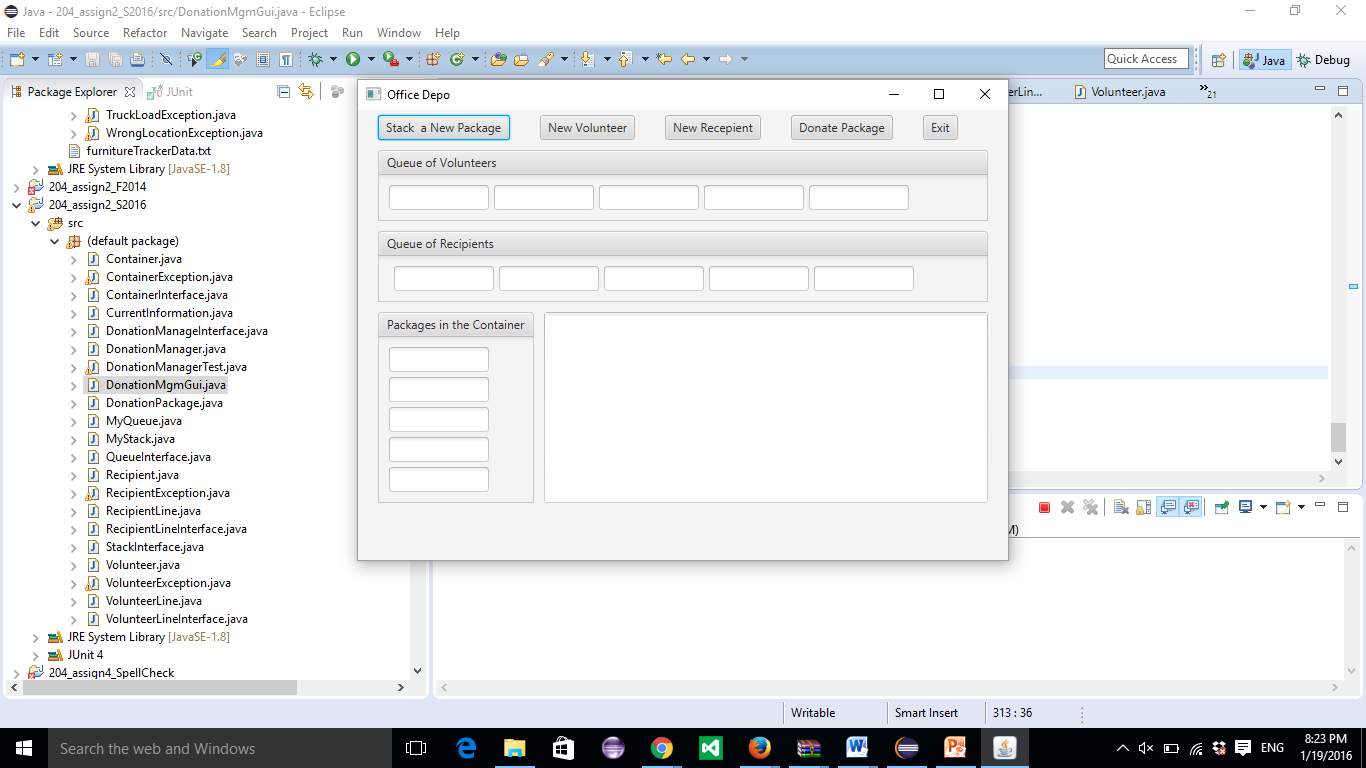
**You may add additional methods and attributes to your classes.**

**GUI Driver – provided for you**

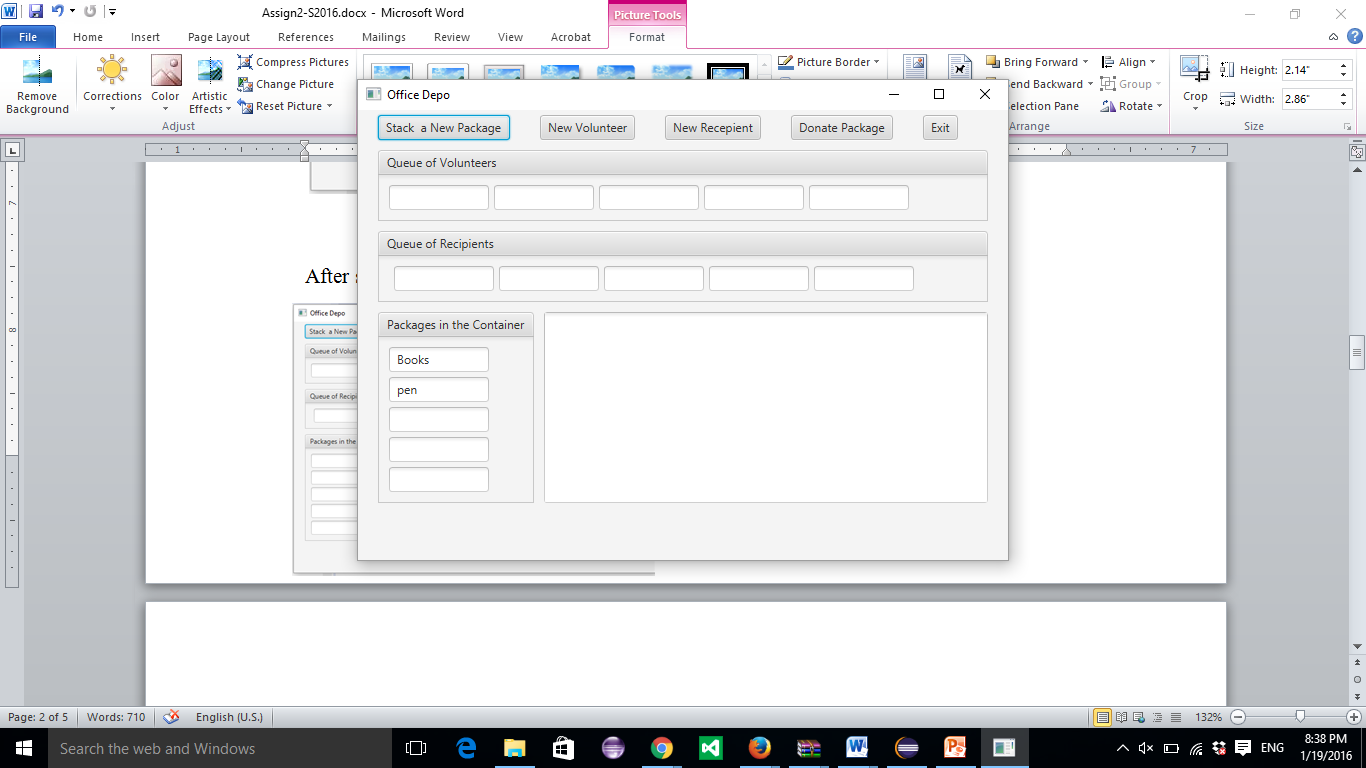
1. Initially there is no package on the container stack, or any volunteers or recipients on the queue.
2. Allows the user to load new packages to the container and add new volunteers to volunteer’s line and new recipient to recipients’ line.
3. Allows the user to simulate the process of donating a package 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

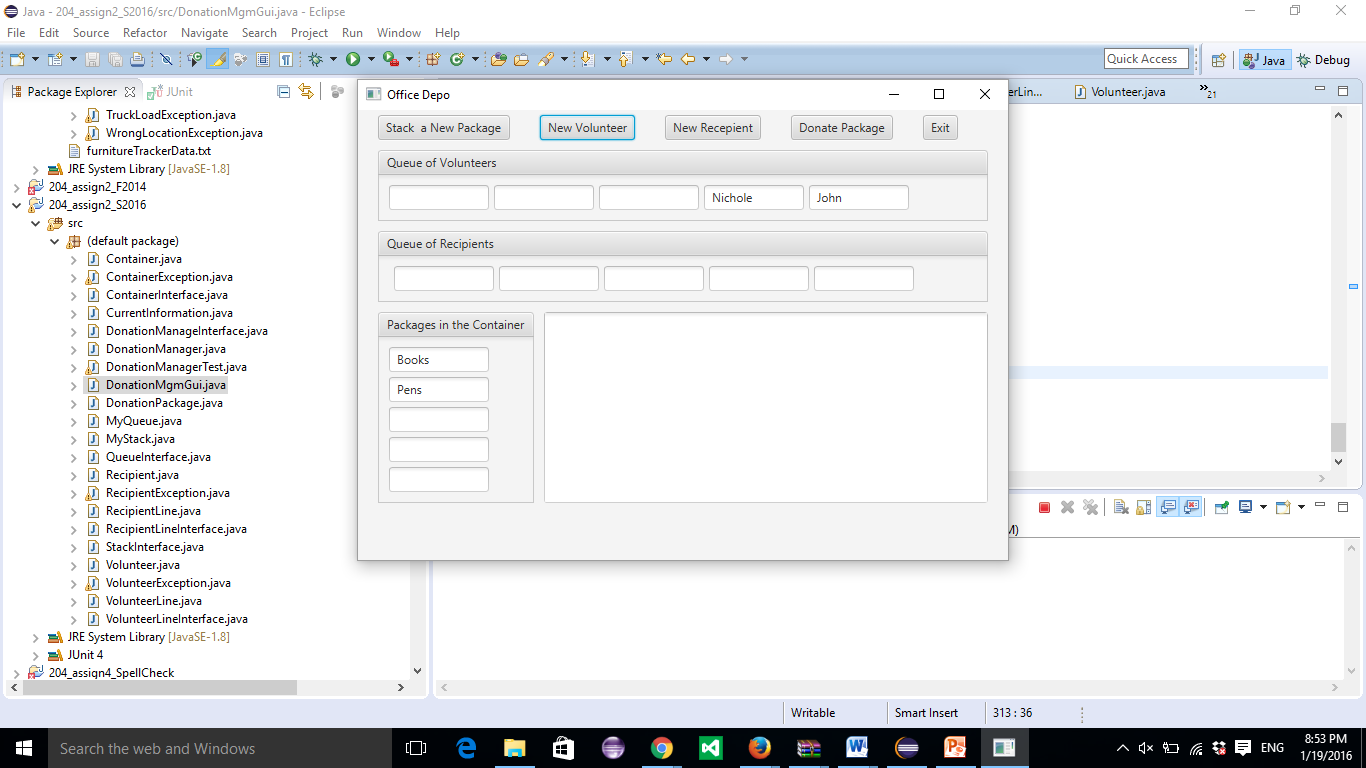
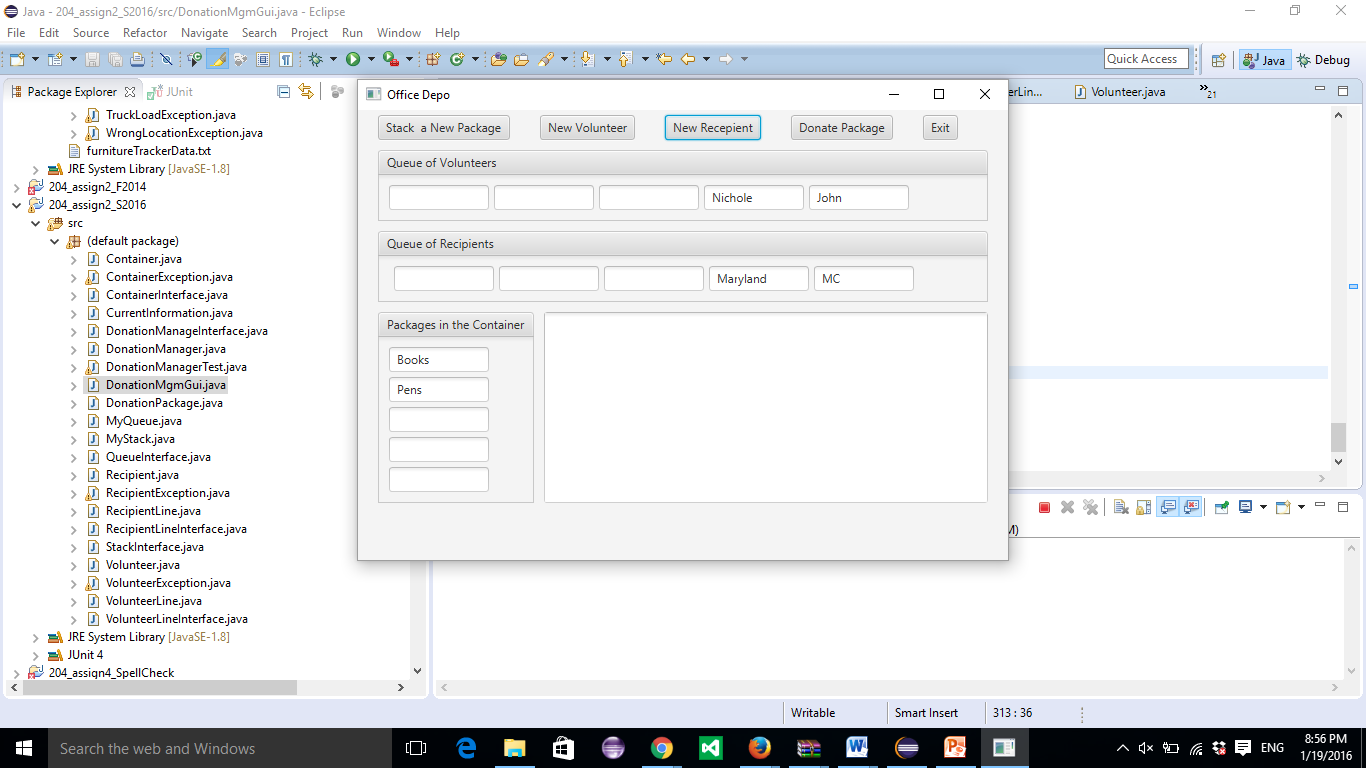


After selecting “Stack New Package”

Pen(first) and Books (second).

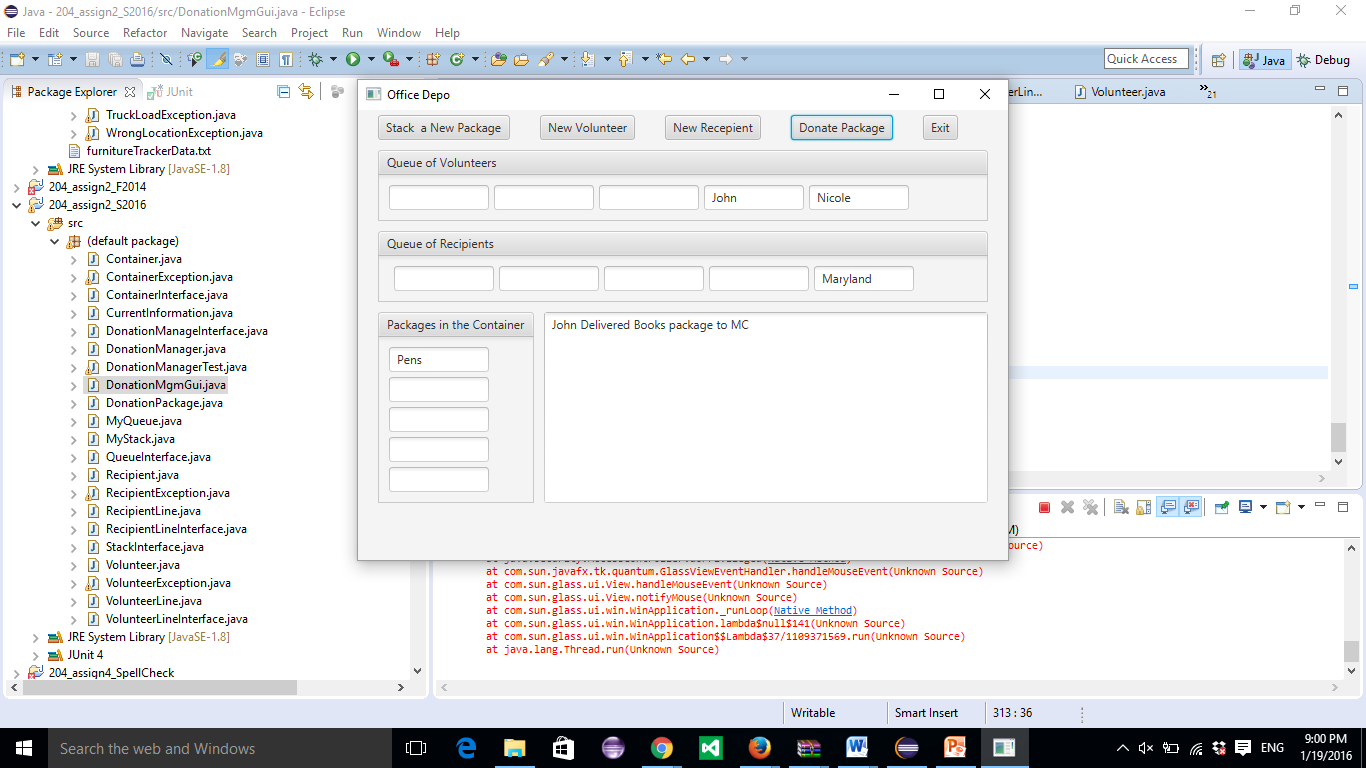
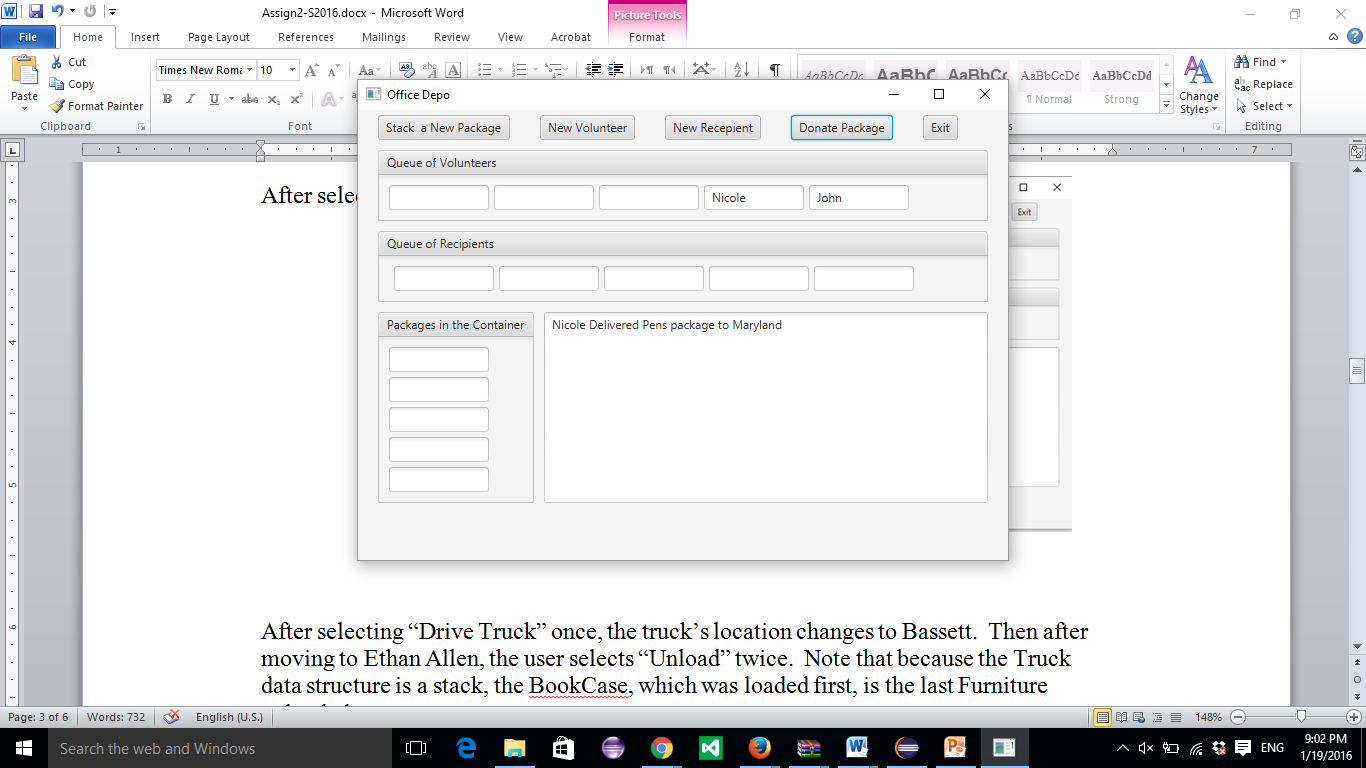
After adding new Volunteers John (first), After adding new Recipients MC (first),

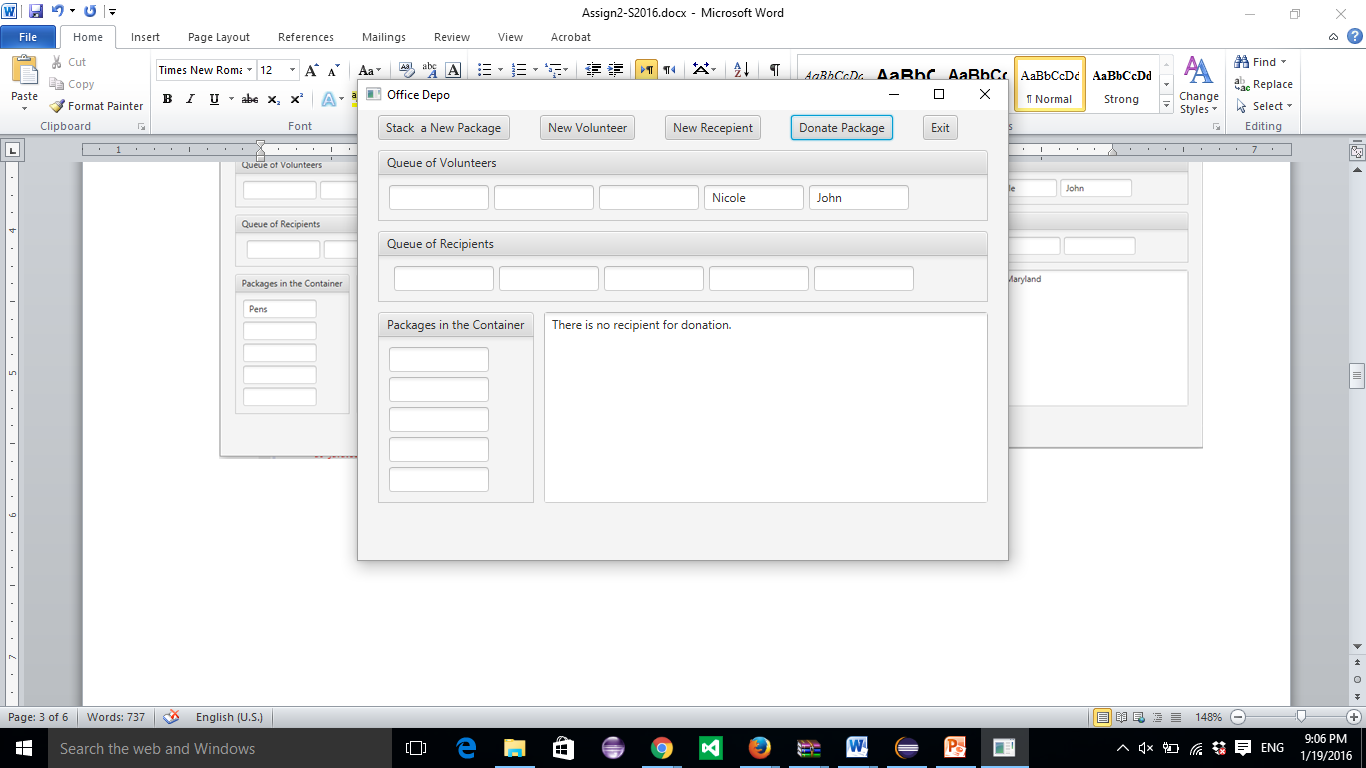
Nicole (second) Maryland (second)

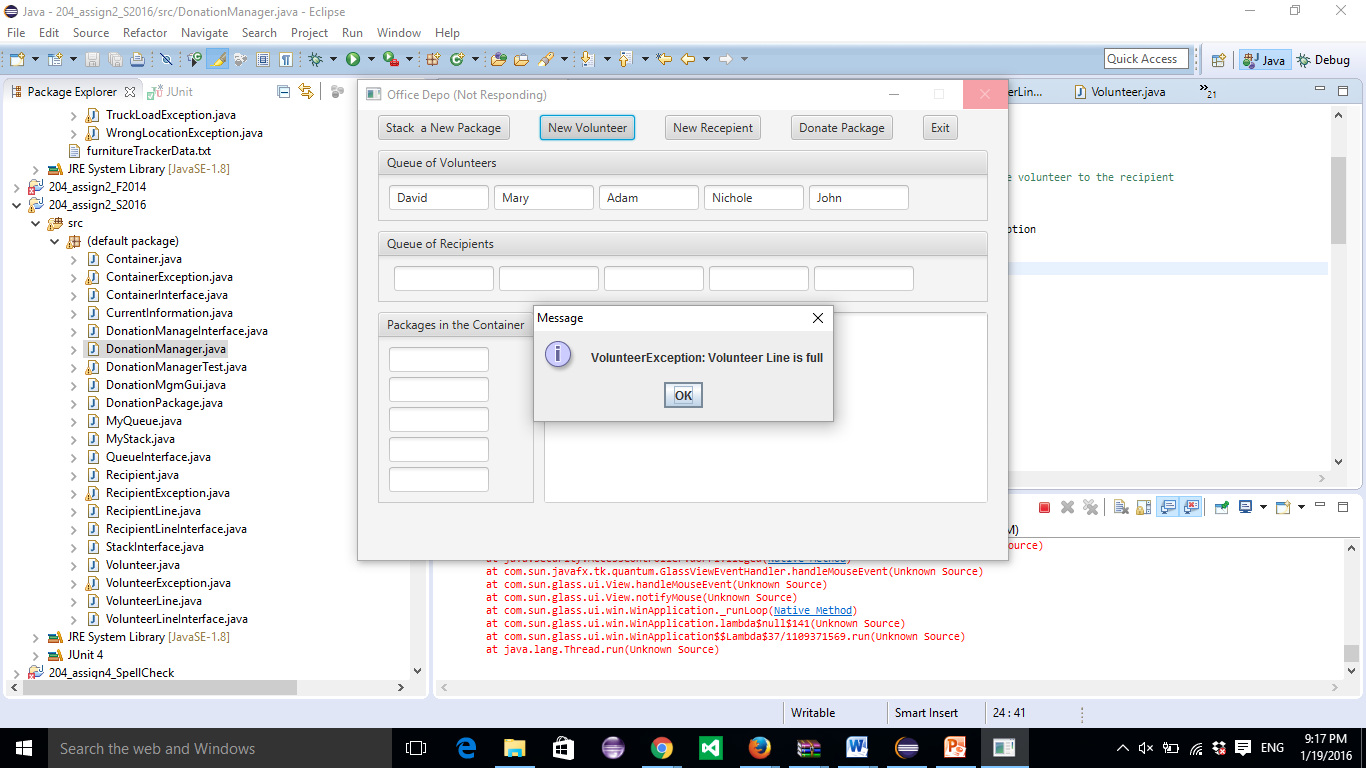


After selecting Donate Package After selecting Donate Package Second time

First Time



After selecting Donate Package Third time

 If the user tries to add a new volunteer and

The line of volunteers is full:

**Grading Rubric - CMSC 204 Project #2**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

**PROGRAMMING** (100 pts)

Compiles 40 pts \_\_\_\_\_

Accuracy

Passes public JUnit tests (provided) 10 pts \_\_\_\_\_

Passes STUDENT tests 5 pts \_\_\_\_\_

Execution: runs without errors (either run-time or logic errors) 45 pts \_\_\_\_\_

Possible Sub-total 100 pts \_\_\_\_\_

**REQUIREMENTS** (Subtracts from Programming total)

**Documentation:**

Javadoc was not provided - 5 pts \_\_\_\_\_

Documentation within source code was missing or incorrect - 5 pts \_\_\_\_\_

Description of what class does was missing

Author’s Name, @author, was missing

Methods not commented properly using Javadoc @param, @return

UML diagram was not provided - 5 pts \_\_\_\_\_

JUnit Test Class: STUDENT methods were not implemented - 5 pts \_\_\_\_\_

In 3+ paragraphs, highlight your lessons learned and learning experience - 5 pts \_\_\_\_\_\_

from working on this project. How did you do? What have you learned? What

did you struggle with? How will you approach your next project differently?

**Programming Style:**

Incorrect use of indentation, statements, structures - 5 pts \_\_\_\_\_

User interface - 6 pts \_\_\_\_\_

Not clear to user how data is to be entered

UI does not follow requirements

Output is easy to understand

**Design:** Classes do not have the functionality specified, i.e.,

1. Data Structures classes - 6 pts \_\_\_\_\_

* Generic Stack class for Container
* Generic Queue for volunteer line
* Generic Queue for Recipient line

1. Data Elements – DonationPackage, Volunteer, Recipient - 6 pts \_\_\_\_\_

* Contains required attributes
* Contains toString and any other necessary methods

1. Data Manager - DonationManager - 6 pts \_\_\_\_\_

* Implements DonationManager Interface
* Creates Volunteer, Recipient, and DonationPackage objects
* Donate Package implemented
* Correctly handles exceptions

1. Exceptions - does not handle exceptions correctly - 6 pts \_\_\_\_\_

Possible decrements: -60 pts \_\_\_\_\_

Possible total grade: 100 pts \_\_\_\_\_