SSJ Package Tracker Testing Plan

TEST PLAN

Sam Anderson
Sanat Bhandari
Jerome Schmidt

Introduction

The objectives of these tests are to ensure that our software system is easy to use, and when it is used as expected, it returns the expected output given in the test cases. If all of our test cases pass, then we will know that our software system fulfills the testing requirements that are derived from the underlying system requirements. If not, then we will modify our software system and proceed with regression testing until all test cases return expected output.

System Overview

SSJ Package Tracker offers clients an intuitive interface to track their couriers dispatched via our partner, Bohn's Drones Inc., in Lincoln and Omaha, Nebraska, area. Our software system consists of a three tier software architecture with a Presentation Tier, an Application Tier, and a Database Tier. The Presentation Tier is the web interface that the user interacts with, the Application Tier is where all necessary business logic takes place, and the Database Tier is where all of the information regarding the drones, packages, customers, and depots will be stored.

The system requirements are as follows:

F1. The system is launched as a web application, with the front-end main page upon system startup being a login screen that allows for users to login as a customer or staff user.

F2. The system database is created and the data therein is accessible and modifiable.

F3. Customer users are able to submit a delivery request, resulting in a drone to be dispatched to pick up the package at their specified location.

F4. When a package is being delivered, both the customer and Bohn's Drones staff can see the progress of the delivery, i.e. whether the drone carrying it is at a depot, between depots, or en route to the delivery address.

F5. Bohn's Drones staff are able to send drones not carrying a package between depots.

F6. Bohn's Drones staff can see which package is aboard a specific SUAS.

F7. When a drone carrying a package arrives at a depot, the package is handed off to another drone to continue delivery.

F8. Errors are thrown when invalid input is entered.

Features to be Tested

Testing Requirement	System Requirement(s)	Short Description
1	1	Start the system
2	2	Data is accessible
3	3	When a delivery is requested, database is updated accordingly
4	4	When checking delivery request and a valid package ID is entered, its location is displayed
5	5	When a staff member orders an empty drone to a different location, the database is updated accordingly
6	6	Staff can check contents of package when corresponding package ID is entered
7	7	When a drone reaches a new depot, the drone that is carrying it in the database is updated accordingly
8	8	When an invalid package ID is entered, an error is thrown

Test Environment

Test Cases

Test Case 1

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2

Initial Conditions

System is not running when test is started

Expected Behavior

Querying the database to return all of the information in the “Packages” table concerning the package information

Input

n/a - (run query "SELECT * FROM Packages" in python file)

Output

(1, '733 N 17th St', '1144 T St', 'Samsung Galaxy S10', None, None, None, None, None, None)

(2, '1600 Vine St', '1 Memorial Stadium Drive', 'Football', None, None, None, None, None, None)

(3, '1400 R St', '440 N 17th St', 'AirPods', None, None, None, None, None, None)

(4, '2200 Dodge St', '4510 Stadium Dr', 'Magnetic Key Holder', None, None, None, None, None, None)

(5, '1600 Court St', '645 N 14th St', 'Handheld Cordless Vacuum Cleaner', None, None, None, None, None, None)

(6, '1430 Vine St', '860 N 17th St', 'Digital Alarm Clock', None, None, None, None, None, None)

(7, '645 N 14th St', '1150 N 14th St', 'Bluetooth Speakers', None, None, None, None, None, None)

(8, '1113 R St', '1601 N 35th St', 'Silverware Set', None, None, None, None, None, None)

(9, '1150 N 14th St', '1215 U St', 'Multivitamin Gummies', None, None, None, None, None, None)

(10, '333 N 48th St', '1101 P St', 'Popcorn Machine', None, None, None, None, None, None)

Test Case 2

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3

Initial Conditions

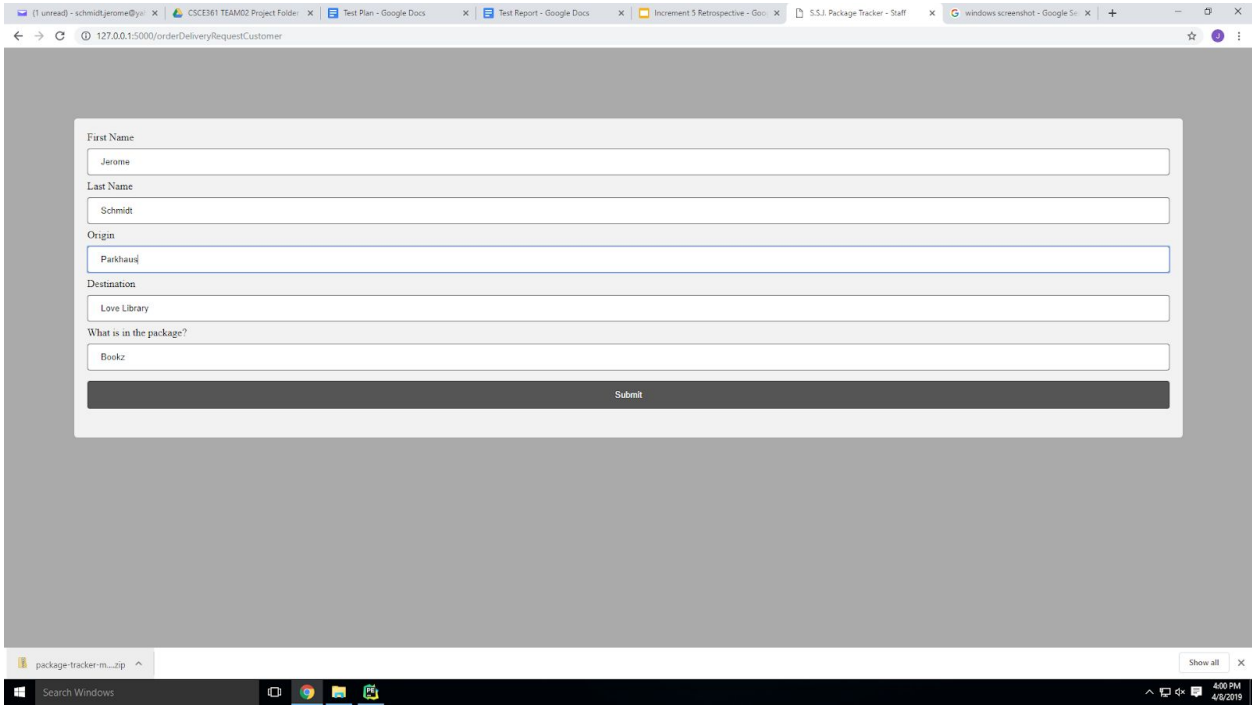
A customer has logged in and wants to request a package delivery

Expected Behavior

Database is updated accordingly with customer's entered package information

Input

//screenshot of page where information is entered



Output

//screenshot of database showed that is was updated with relevant information

PackageID	OriginAddress	DeliveryAddress	PackageName	UserID	DroneID	FirstName	LastName	TimePickedUp	TimeDelivered
1	733 N 17th St	1144 T St	Samsung Galaxy S10	NULL	NULL	SAM	ANDERSON	NULL	NULL
2	1600 Vine St	1 Memorial Stadium Drive	Football	NULL	NULL	SAM	ANDERSON	NULL	NULL
3	1400 R St	440 N 17th St	AirPods	NULL	NULL	SAM	ANDERSON	1:48PM	3:30PM
4	2200 Dodge St	4510 Stadium Dr	Magnetic Key Holder	NULL	NULL	SAM	ANDERSON	NULL	NULL
5	1600 Court St	645 N 14th St	Handheld Cordless Vacuum Cleaner	NULL	NULL	SAM	ANDERSON	2:00PM	NULL
6	1430 Vine St	860 N 17th St	Digital Alarm Clock	NULL	NULL	SAM	ANDERSON	2:53PM	3:15PM
7	645 N 14th St	1150 N 14th St	Bluetooth Speakers	NULL	NULL	SAM	ANDERSON	NULL	NULL
8	1113 R St	1601 N 35th St	Silverware Set	NULL	NULL	SAM	ANDERSON	NULL	NULL
9	1150 N 14th St	1215 U St	Multivitamin Gummies	NULL	NULL	SAM	ANDERSON	NULL	NULL
10	333 N 48th St	1101 P St	Popcorn Machine	NULL	NULL	SAM	ANDERSON	NULL	NULL
16	AVERY HALL	8N LOFTS	BOMB	NULL	NULL	SANAT	BHANDARI	NULL	NULL
▶ 17	PARKHAUS	LOVE LIBRARY	BOOKZ	NULL	NULL	JEROME	SCHMIDT	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Test Case 3

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3, 6

Initial Conditions

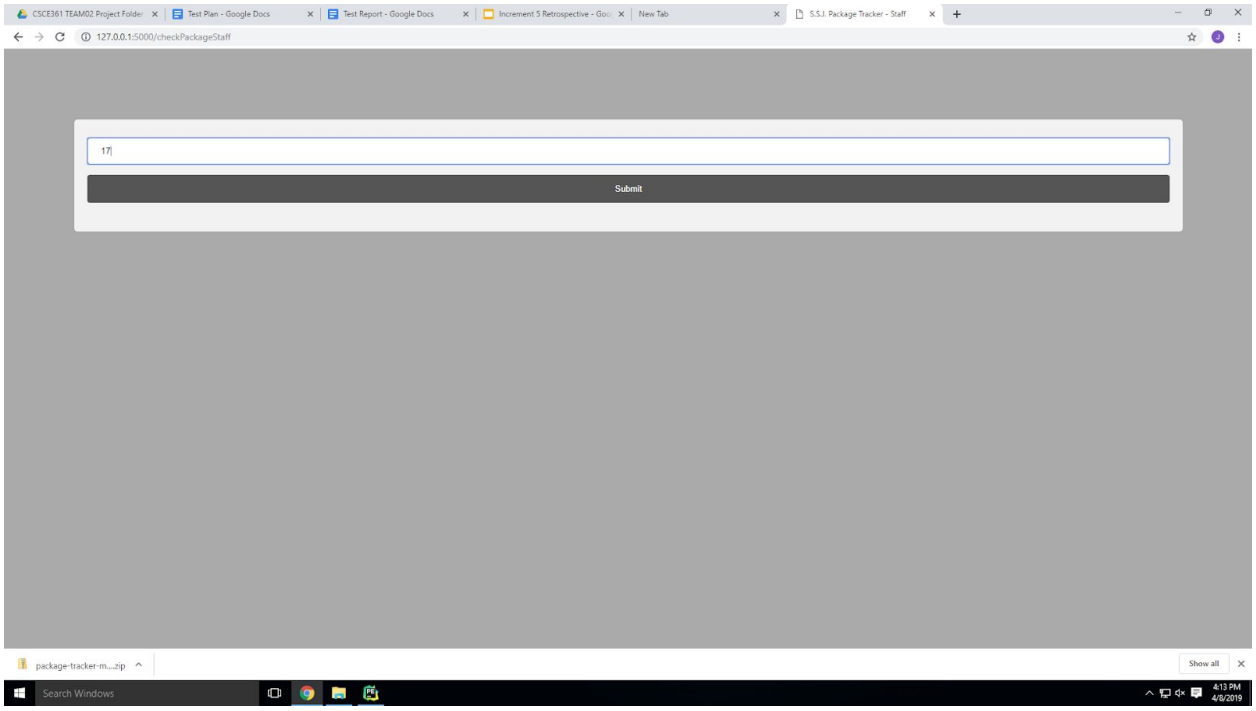
Staff is logged in and wants to know what is in the package specified by its unique identification number

Expected Behavior

Contents of package are returned when the package's unique identification number is entered, requesting said package's information concerning what is in it

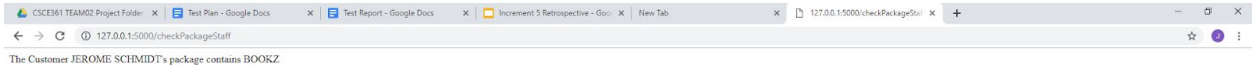
Input

//screenshot of page where staff enters info



Output

//screenshot of page after submit button is hit



Test Case 4

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3, 4

Initial Conditions

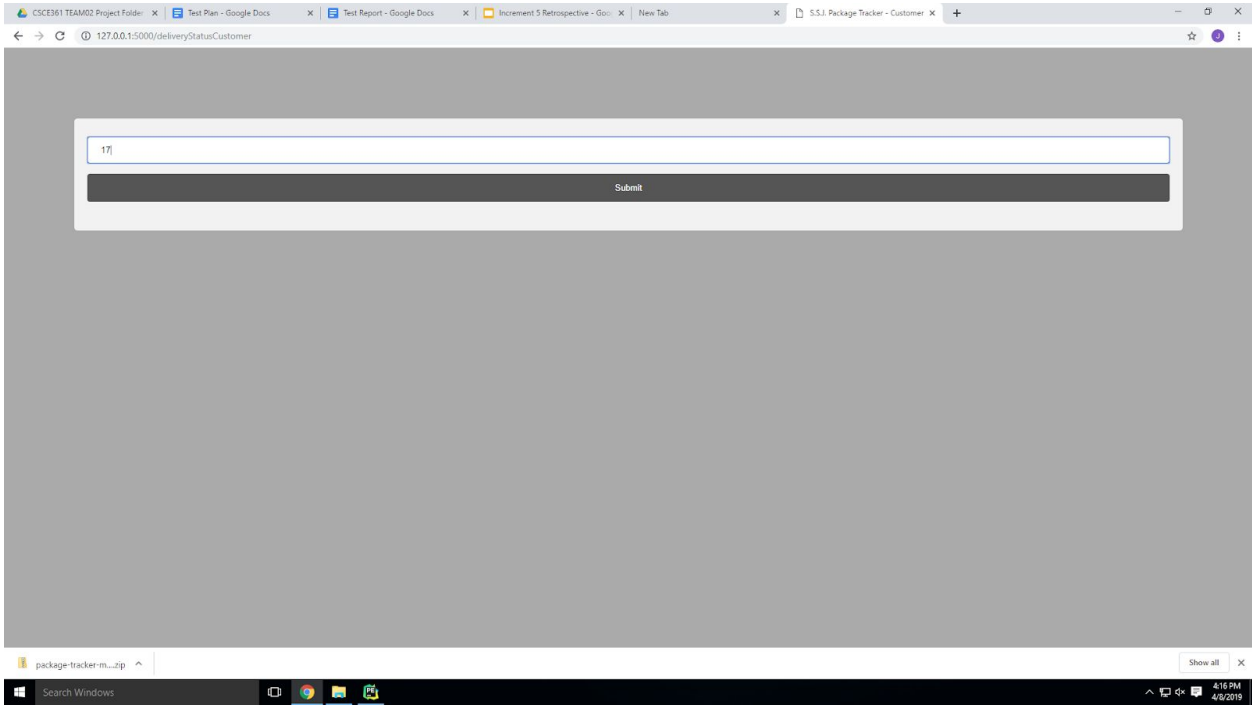
A customer is logged in and wants to check delivery status of a package

Expected Behavior

The requested package's status is returned based on whether it has not yet been picked up by a drone and is currently awaiting pickup at a warehouse, it is currently aboard a drone but not yet delivered, or it has been delivered to its delivery location.

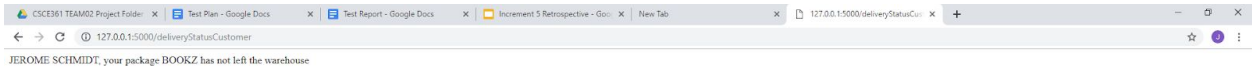
Input

//screenshot of page where package ID is entered



Output

//screenshots showing the page saying where package currently is



Test Case 5

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3, 4

Initial Conditions

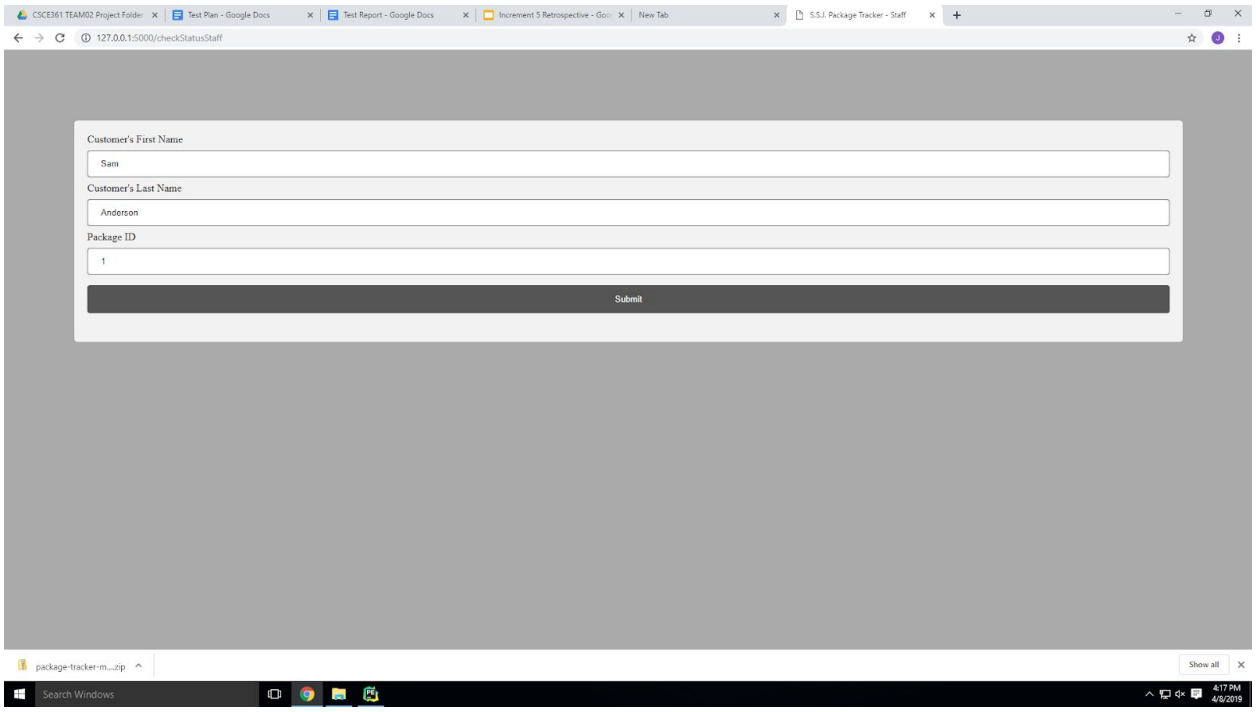
A staff member is logged in and wants to check delivery status of a package

Expected Behavior

The requested package's status is returned based on whether it has not yet been picked up by a drone and is currently awaiting pickup at a warehouse, it is currently aboard a drone but not yet delivered, or it has been delivered to its delivery location. This should function the same as a customer user requesting this information.

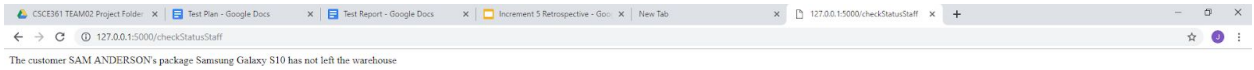
Input

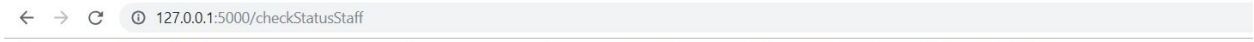
//screenshot of page where package ID is entered



Output

//screenshots showing the page saying where package is





Test Case 6

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3, 5, 7

Initial Conditions

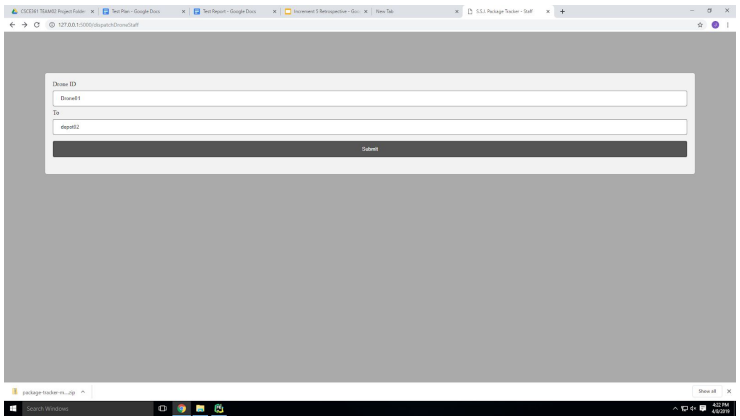
Staff member is logged in and wants to order an empty drone from one location to another

Expected Behavior

The drone is successfully moved from one depot to another, and is thus updated in the database to reflect this

Input

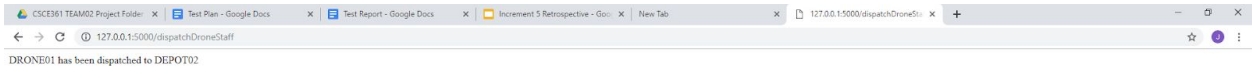
//screenshot of page where staff can enter DroneID and destination, and initial depot that Drone01 is at in the database (Depot01)



DroneID	PackageID	DepotID	DroneStatus
DRONE01	NULL	DEPOT01	Idle
DRONE02	NULL	DEPOT02	Idle
DRONE03	NULL	DEPOT06	Idle
DRONE04	NULL	DEPOT09	Idle
DRONE05	NULL	DEPOT05	Idle
DRONE06	NULL	DEPOT06	Idle
DRONE07	NULL	DEPOT10	Idle
DRONE08	NULL	DEPOT08	Idle
DRONE09	NULL	DEPOT09	Idle
DRONE10	NULL	DEPOT10	Idle
DRONE11	NULL	DEPOT11	Idle
DRONE12	NULL	DEPOT01	Idle

Output

//screenshot of page showing the drone's location has been updated successfully, and the database information that reflects that said drone's depot has been updated (now at Depot02)



DroneID	PackageID	DepotID	DroneStatus
DRONE01	NULL	DEPOT02	Idle
DRONE02	NULL	DEPOT02	Idle
DRONE03	NULL	DEPOT06	Idle
DRONE04	NULL	DEPOT09	Idle
DRONE05	NULL	DEPOT05	Idle
DRONE06	NULL	DEPOT06	Idle
DRONE07	NULL	DEPOT10	Idle
DRONE08	NULL	DEPOT08	Idle
DRONE09	NULL	DEPOT09	Idle
DRONE10	NULL	DEPOT10	Idle
DRONE11	NULL	DEPOT11	Idle
DRONE12	NULL	DEPOT01	Idle
NULL	NULL	NULL	NULL

Test Case 7

Component Under Test

SSJ Package Tracker

Feature(s) to be Tested

1, 2, 3, 8

Initial Conditions

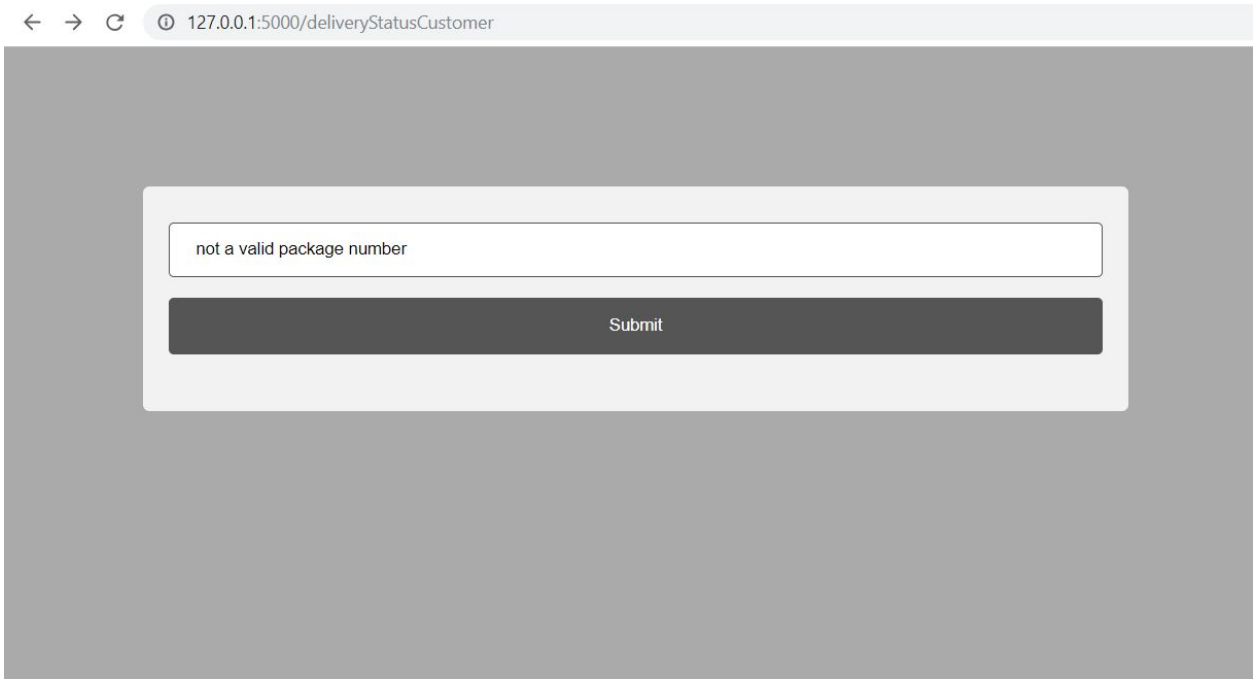
Package ID is entered to check delivery status and package ID is invalid

Expected Behavior

Upon inputting an invalid package ID number, an error is displayed indicating that this package ID number is invalid/not found.

Input

//screenshot of invalid package ID being entered



Output

//error displaying that this package ID number was not found in the database