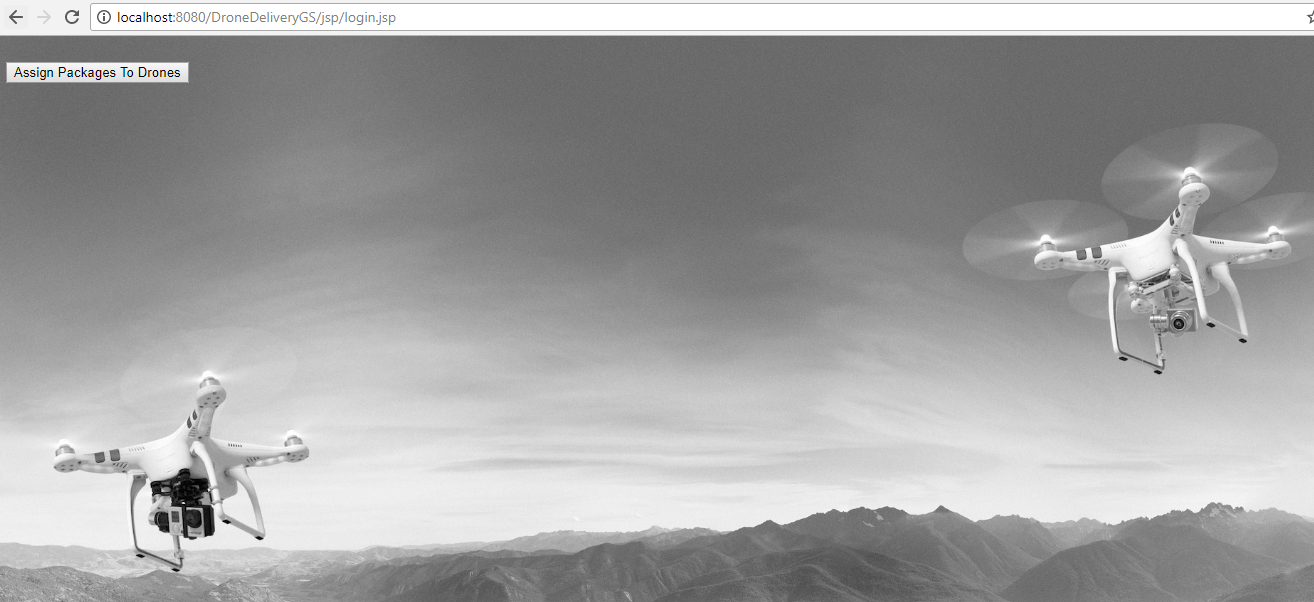
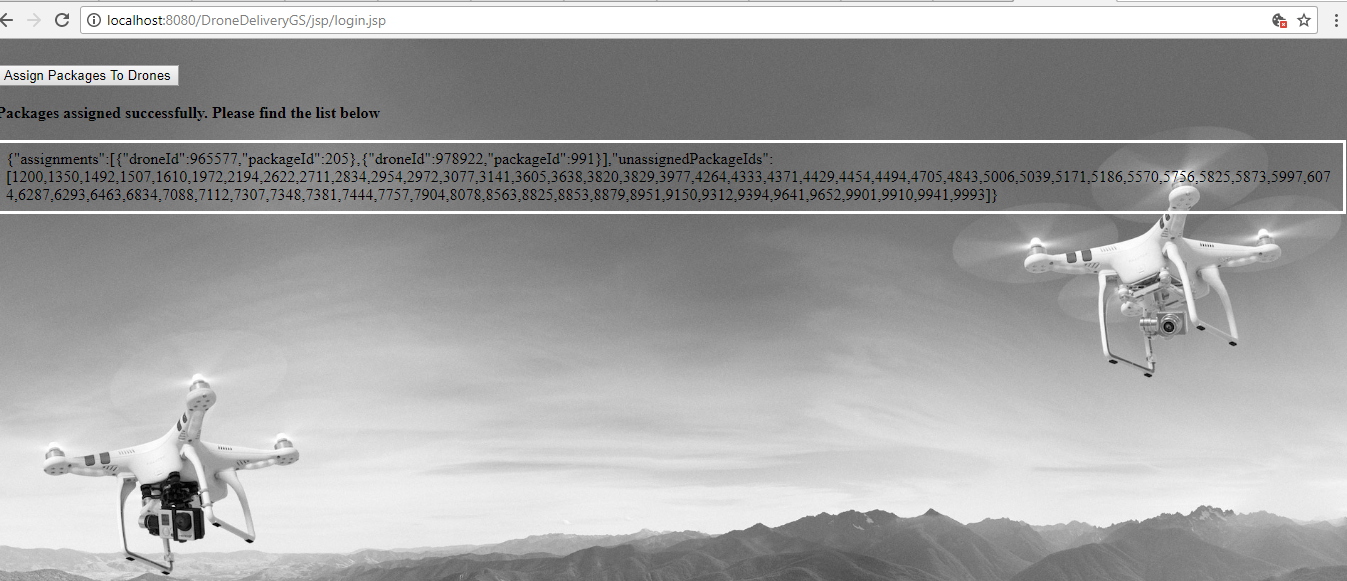
1. The application is written in JAVA and the framework used is struts. The application is accessed using URL <http://localhost:8080/DroneDeliveryGS/jsp/login.jsp> in the local machine. The home page looks like this



1. There is a button **‘Assign Packages to Drones’** which calls the JavaScript function to fetch the list of drones and packages from the API mentioned  [*https://codetest.kube.getswift.co/drones*](https://codetest.kube.getswift.co/drones) and  [*https://codetest.kube.getswift.co/packages*](https://codetest.kube.getswift.co/packages)*.*
2. After pulling the data from the API, a JSON object is created and a restful webservice call is made to service class. The corresponding code can be checked in login.jsp file.
3. The restful webservice call leads us to assignPackagesToDrones() method in AssignPackagesService.java file. In this method, the JSON data from the UI is extracted and beans are populated with this data since working with beans is easy than JSON in the core logic of assigning a drone to a package.
4. There are 2 bean classes used here, DroneBean for storing drone’s details and PackageBean for storing package details
5. Service class then makes a call to assignNewPackages method in AssignPackagesDelegate class. The main logic of assigning drones to packages resides here.
6. In this method, from the list, each package is taken and we try to find the best match drone for this package. The list of drones available is iterated. If the drone is already assigned to a new package, then it is not taken into consideration since a drone can carry only one package at a time.
7. If the drone is on the way to deliver its current package, using haversine equation, the distance left for the drone to deliver the current package, distance from current package destination back to depo, and distance to travel from depo to new destination is calculated.
8. If the drone doesn’t have any package currently, distance to travel from depo to new destination is calculated using haversine equation.
9. Using this distance, since its told that drones travel at 50km/hr, earliest time at which this drone can deliver the package to its destination is calculated
10. When the iteration over available drones is completed, we will be left with one drone which has the minimum delivery time to the package destination. Now we will see if this time is less than the package deadline. If it’s before deadline, then that drone is assigned to that package and then that drone will be added to list of drones to be excluded for the other packages If it is after deadline, that package is added to list of packages that could not be assigned.
11. After iteration over all packages is completed, a JSON object with the list of assigned and un assigned drones/packages and this JOSN is send to jsp, and it is displayed in table. The screen looks like below with the output



Versions used:

* apache-tomcat-7.0.81
* java version "1.8.0\_131"