CS 458
Software Verification and Validation
2018-2019 Summer
Homework #3
INTRODUCTION TO TEST DRIVEN DEVELOPMENT (TDD)

Grup Şurup

Ayça Begüm Taşçioğlu - 21600907
Muhammet Said Demir - 21602021
Zeynep Nur Öztürk - 21501472

1. With Test Driven Development (TDD) develop a responsive page (that can run on laptops, mobile devices etc., no mobile native code required) for the following tasks:

a. You enter your coordinates of your location and it shows your city (via Google Map API etc.).

1. True Case: Enter valid location

- Test Code:

```
package homework3;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
		Thread.sleep(1000);
		WebElement searchBox;
		searchBox = driver.findElement(By.linkText("Enter Your
Coordinates and Find Your City!"));
		searchBox.click();
		Thread.sleep(1000);
		WebElement searchBox1;
		searchBox1 = driver.findElement(By.id("latlng"));
		searchBox1.clear();
		searchBox1.sendKeys("30,25");
		Thread.sleep(1000);
		driver.findElement(By.id("submit")).click();
		Thread.sleep(5000);
		driver.quit();
	}
}
```

- Print Out of Code:

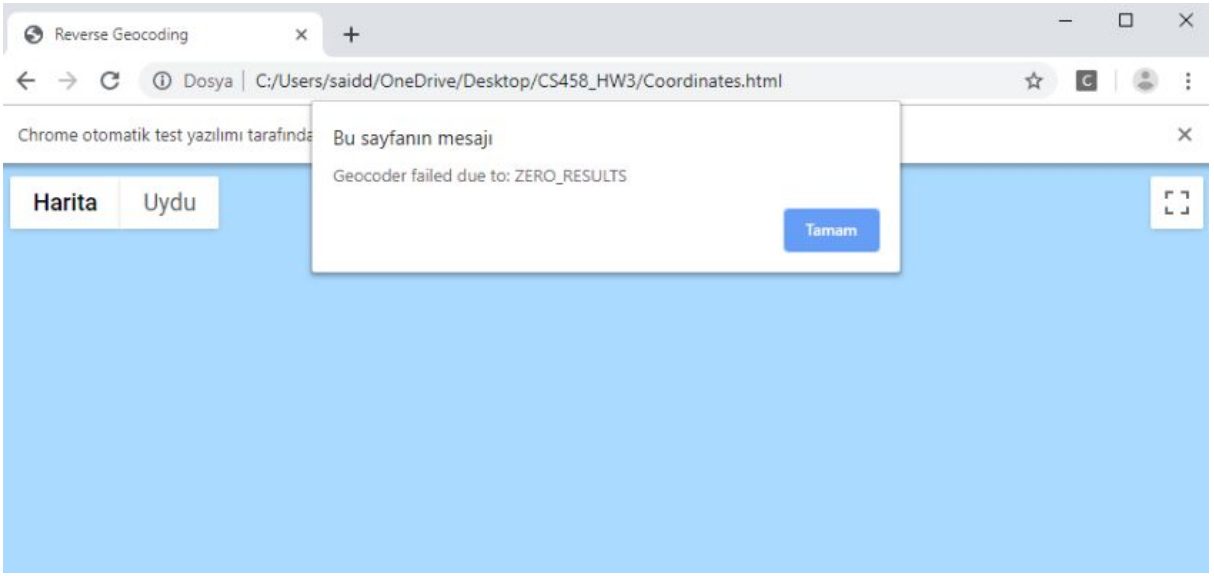
2. Out of boundaries: Enter more than 360 meridian and 180 latitude

- Test Code:

```
package homework3;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
		Thread.sleep(1000);
		WebElement searchBox;
		searchBox = driver.findElement(By.linkText("Enter Your
Coordinates and Find Your City!"));
		searchBox.click();
		Thread.sleep(1000);
		WebElement searchBox1;
		searchBox1 = driver.findElement(By.id("latlng"));
		searchBox1.clear();
		searchBox1.sendKeys("370,2500");
		Thread.sleep(1000);
		driver.findElement(By.id("submit")).click();
		Thread.sleep(5000);
		driver.quit();
	}
}
```

- Print Out of Code:

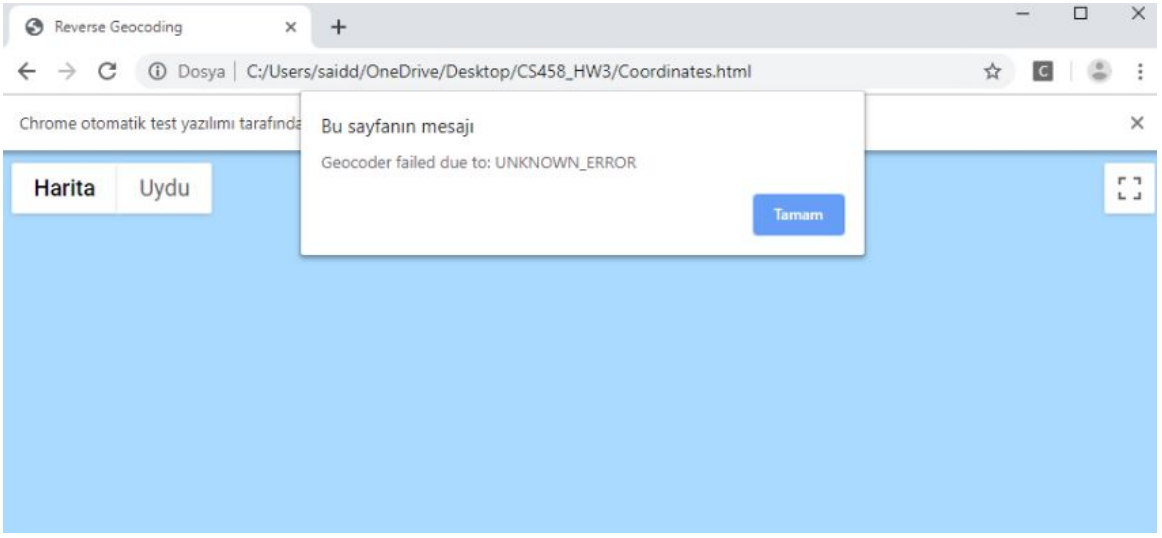


3. String value: Enter a string
    - Test Code:

```
package homework3;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
		Thread.sleep(1000);
		WebElement searchBox;
		searchBox = driver.findElement(By.linkText("Enter Your
Coordinates and Find Your City!"));
		searchBox.click();
		Thread.sleep(1000);
		WebElement searchBox1;
		searchBox1 = driver.findElement(By.id("latlng"));
		searchBox1.clear();
		searchBox1.sendKeys("haluk, altunel");
		Thread.sleep(1000);
		driver.findElement(By.id("submit")).click();
		Thread.sleep(5000);
		driver.quit();
	}
}
```

- Print Out of Code

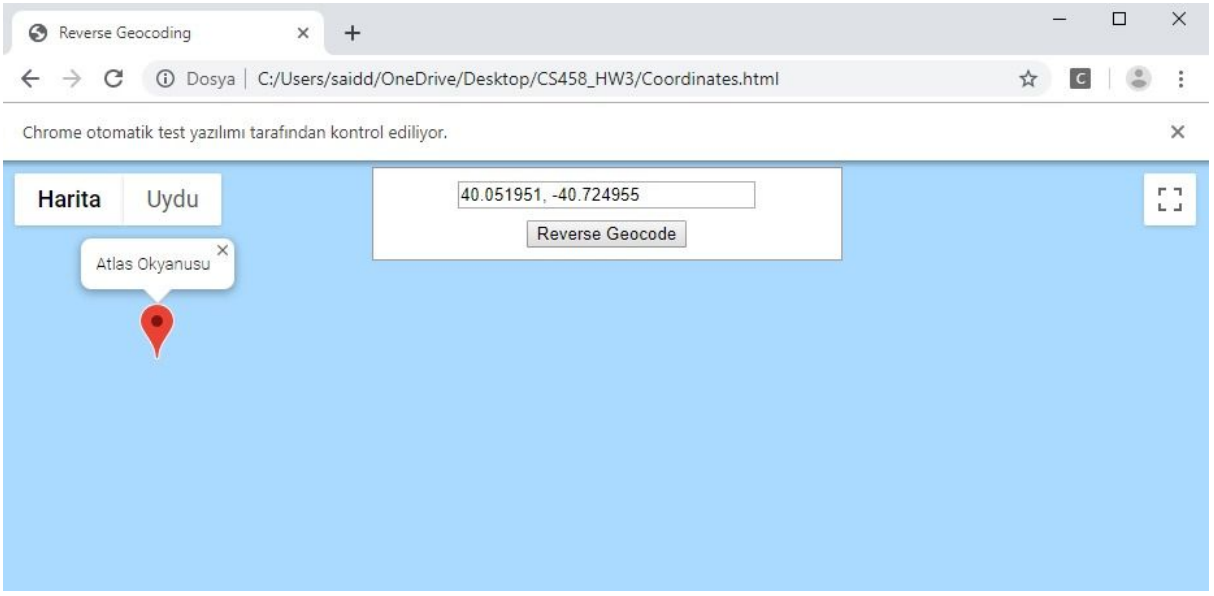


4. Enter somewhere in the sea.
    - Test Code:

```
package homework3;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
```

```
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
                    Thread.sleep(1000);
                    WebElement searchBox;
                    searchBox = driver.findElement(By.linkText("Enter Your
Coordinates and Find Your City!"));
                    searchBox.click();
                    Thread.sleep(1000);
                    WebElement searchBox1;
                    searchBox1 = driver.findElement(By.id("latlng"));
                    searchBox1.clear();
                    searchBox1.sendKeys("40.051951, -40.724955");
                    Thread.sleep(1000);
                    driver.findElement(By.id("submit")).click();
                    Thread.sleep(5000);
                    driver.quit();
            }
}
```

- Print Out of Code:



- **MAIN CODE**

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>Reverse Geocoding</title>
    <style>
      #map {
        height: 100%;
      }
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
```

```
    #floating-panel {
      position: absolute;
      top: 10px;
      left: 25%;
      z-index: 5;
      background-color: #fff;
      padding: 5px;
      border: 1px solid #999;
      text-align: center;
      font-family: 'Roboto','sans-serif';
      line-height: 30px;
      padding-left: 10px;
    }
    #floating-panel {
      position: absolute;
      top: 5px;
      left: 50%;
      margin-left: -180px;
      width: 350px;
      z-index: 5;
      background-color: #fff;
      padding: 5px;
      border: 1px solid #999;
    }
    #latlng {
      width: 225px;
    }
  </style>
</head>
<body>
  <div id="floating-panel">
    <input id="latlng" type="text" value="0,0">
    <input id="submit" type="button" value="Reverse Geocode">
  </div>
  <div id="map"></div>
  <script>
    function initMap() {
      var input = document.getElementById('latlng').value;
      var latlngStr = input.split(',', 2);
      //var latlng = {lat: parseFloat(latlngStr[0]), lng: parseFloat(latlngStr[1])};
      var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 8,
        center: {lat: parseFloat(latlngStr[0]),lng: parseFloat(latlngStr[1])}
      });
      var geocoder = new google.maps.Geocoder;
      var infowindow = new google.maps.InfoWindow;
      document.getElementById('submit').addEventListener('click', function() {
        geocodeLatLng(geocoder, map, infowindow);
      });
    }
```

```
    function geocodeLatLng(geocoder, map, infowindow) {
      var input = document.getElementById('latlng').value;
      var latlngStr = input.split(',', 2);
      var latlng = {lat: parseFloat(latlngStr[0]), lng: parseFloat(latlngStr[1])};
      geocoder.geocode({'location': latlng}, function(results, status) {
        if (status === 'OK') {
          if (results[0]) {
            map.setZoom(11);
            var marker = new google.maps.Marker({
              position: latlng,
              map: map
            });
            infowindow.setContent(results[0].formatted_address);
            infowindow.open(map, marker);
          } else {
            window.alert('No results found');
          }
        } else {
          window.alert('Geocoder failed due to: ' + status);
        }
      });
    }
  </script>
  <script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCQ0zQ77DEqkokOvnCs6
IPlII9Xcb35bUM&callback=initMap">
  </script>
 </body>
</html>
```

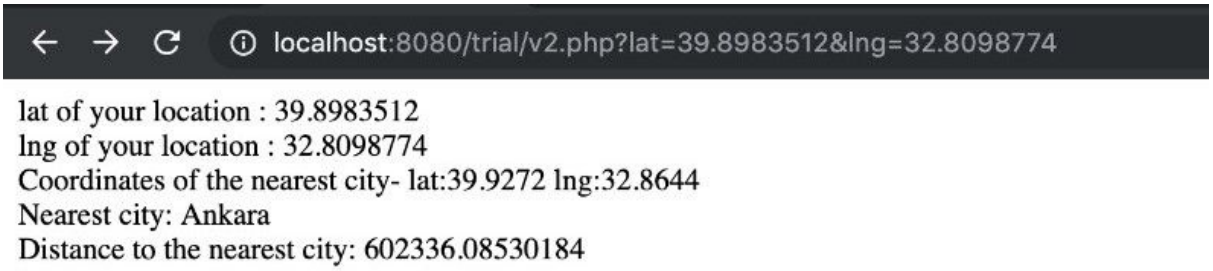
b. It gets GPS coordinates of your device automatically and it shows your distance to the nearest city center.

1. True Case: It gets your location normally.
    - Test Code:

```
package homework3;
import java.util.HashMap;
import java.util.Map;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
public class hw3 {
        public static void main(String[] args) throws InterruptedException {
                Map<String, Object> prefs = new HashMap<String, Object>();
                prefs.put("profile.default_content_setting_values.notifications", 1);
                ChromeOptions options = new ChromeOptions();
                options.setExperimentalOption("prefs", prefs);
                System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
                WebDriver driver = new ChromeDriver(options);
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
```

```
                Thread.sleep(1000);
                WebElement searchBox;
                searchBox = driver.findElement(By.linkText("Find Nearest City
Center!"));
                searchBox.click();
                Thread.sleep(5000);
                driver.quit();
        }
}
```

- Print Out of Code:



2. Without wi-fi: Enter without wi-fi
    - Test Code:
      The code here is the same as the true case only output is different
    - Print Out of Code:

3. Don’t accept website to reach your location
    - Test Code:
      The code here is the same except:
      prefs.put("profile.default_content_setting_values.notifications", 2);
    - Print Out of Code:

- **MAIN CODE**

```
<!DOCTYPE html>
<?php
session_start();
$connection = mysqli_connect('dijkstra.ug.bcc.bilkent.edu.tr', 'begum.tascioglu',
'3Y2HxbNq', 'begum_tascioglu');
$_SESSION['apply']= "false";
if(! $connection)
{
   die('Connection Error!!! ' . mysqli_error());
}
?>
<html>
 <head>
  <title>Geolocation</title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
   #map {
    height: 100%;
   }
   html, body {
    height: 100%;
    margin: 0;
    padding: 0;
   }
  </style>
 </head>
 <body>
  <div id="map"></div>
  <script>
   var map, infoWindow;
   function initMap() {
    map = new google.maps.Map(document.getElementById('map'), {
     center: {lat: -34.397, lng: 150.644},
```

```
          zoom: 6
        });
        infoWindow = new google.maps.InfoWindow;
        if (navigator.geolocation) {
          navigator.geolocation.getCurrentPosition(function(position) {
            var pos = {
              lat: position.coords.latitude,
              lng: position.coords.longitude
            };
            infoWindow.setPosition(pos);
            infoWindow.setContent('Location found.');
            infoWindow.open(map);
            map.setCenter(pos);
            window.location.href = "http://localhost:8080/trial/v2.php?lat=" +
position.coords.latitude + "&lng=" + position.coords.longitude;
          }, function() {
            handleLocationError(true, infoWindow, map.getCenter());
          });
        } else {
          handleLocationError(false, infoWindow, map.getCenter());
        }
      }
      function handleLocationError(browserHasGeolocation, infoWindow, pos) {
        infoWindow.setPosition(pos);
        infoWindow.setContent(browserHasGeolocation ?
                              'Error: The Geolocation service failed.' :
                              'Error: Your browser doesn\'t support geolocation.');
        infoWindow.open(map);
      }
    </script>
    <script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCQ0zQ77DEqkokOvnCs6
lPlII9Xcb35bUM&callback=initMap">
    </script>
  </body>
</html>
-----------------------------------------------------------------------------------------------------
<?php
session_start();
$connection = mysqli_connect('dijkstra.ug.bcc.bilkent.edu.tr', 'begum.tascioglu',
'3Y2HxbNq', 'begum_tascioglu');
$_SESSION['apply']= "false";
if(! $connection)
{
   die('Connection Error!!! ' . mysqli_error());
}

function haversineGreatCircleDistance(
  $latitudeFrom, $longitudeFrom, $latitudeTo, $longitudeTo, $earthRadius = 6371000)
  {
```

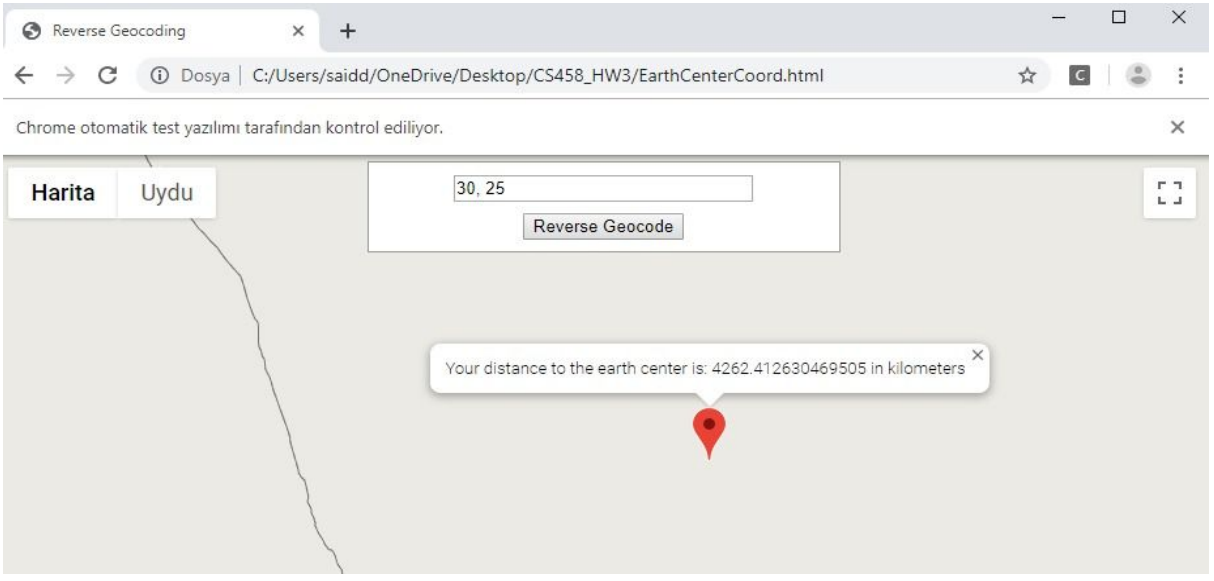
```
    // convert from degrees to radians
    $latFrom = deg2rad($latitudeFrom);
    $lonFrom = deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo = deg2rad($longitudeTo);

    $latDelta = $latTo - $latFrom;
    $lonDelta = $lonTo - $lonFrom;

    $angle = 2 * asin(sqrt(pow(sin($latDelta / 2), 2) +
    cos($latFrom) * cos($latTo) * pow(sin($lonDelta / 2), 2)));
    return $angle * $earthRadius;
  }
   $i =1;
   $min = 1000000;
   $min_city_name = "";
   $query = "SELECT * FROM City";
   $result = $connection-> query($query);
   $loc_lat = $_GET['lat'];
   $loc_lng = $_GET['lng'];
   echo "lat of your location  : ".$loc_lat."<br>";
   echo "lng of your location : ".$loc_lng."<br>";
   $query = "SELECT * FROM City";
   $result = $connection-> query($query);
   $mincity_lat =0;
   $mincity_lng =0;
   if($result -> num_rows > 1)
   {
     while($row=$result->fetch_assoc()){
      $center_lat = $row['lat'];
      $center_lng = $row['lng'];
      $min_mayb =
haversineGreatCircleDistance($loc_lat,$loc_lng,$center_lat,$center_lng,$earthRadiu
s = 6371000);
      if ($min_mayb < $min ){
        $min = $min_mayb;
        $min_city_name = $row['name'];
        $mincity_lat =$row['lat'];
        $mincity_lng =$row['lng'];
      }
     }
      echo  "Coordinates of the nearest city- lat:".$mincity_lat." lng:".
$mincity_lng."<br>";
     echo  "Nearest city: ".$min_city_name."<br>";
     echo "Distance to the nearest city: ".$min_mayb;
   }
   else{
     echo "error table not found";
   }
  ?>
```

c. You enter your coordinates, then it shows your distance to the earth center.

1. True Case:Enter valid location
    - Test Code:

```
package homework3;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
        public static void main(String[] args) throws InterruptedException {
                System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
                Thread.sleep(1000);
                WebElement searchBox;
                searchBox = driver.findElement(By.linkText("Find Distance to
the Earth Center via Coordinates!"));
                searchBox.click();
                Thread.sleep(1000);
                WebElement searchBox1;
                searchBox1 = driver.findElement(By.id("latlng"));
                searchBox1.clear();
                searchBox1.sendKeys("30, 25");
                Thread.sleep(1000);
                driver.findElement(By.id("submit")).click();
                Thread.sleep(10000);
                driver.quit();
        }
}
```

    - Print Out of Code:

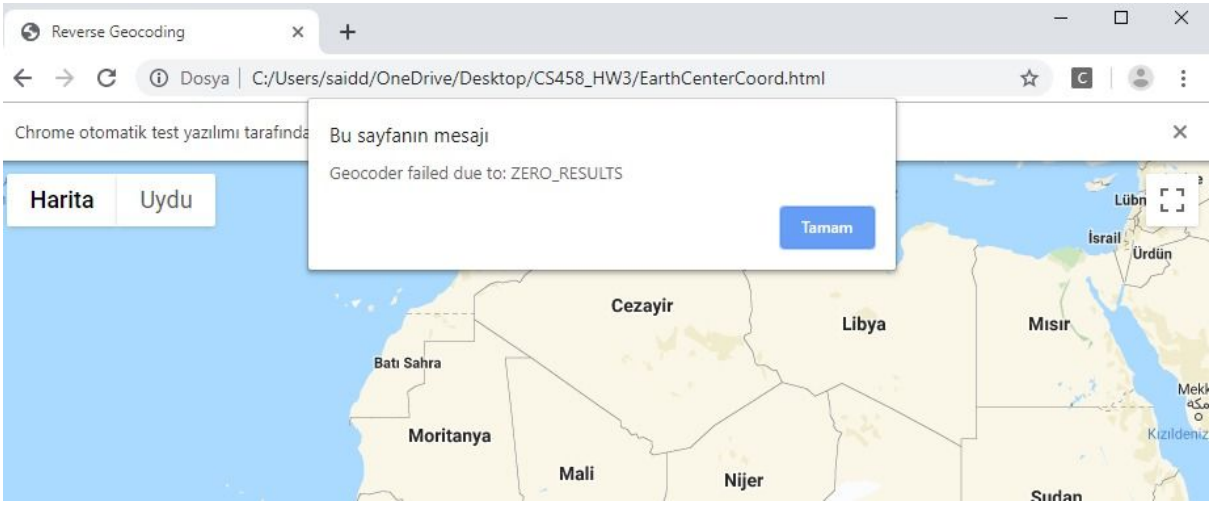


2. Out of boundaries: Enter more than 360 meridian and 180 latitude.
    - Test Code:

```
package homework3;
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
		WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
		Thread.sleep(1000);
		WebElement searchBox;
		searchBox = driver.findElement(By.linkText("Find Distance to
the Earth Center via Coordinates!"));
		searchBox.click();
		Thread.sleep(1000);
		WebElement searchBox1;
		searchBox1 = driver.findElement(By.id("latlng"));
		searchBox1.clear();
		searchBox1.sendKeys("370, 2500");
		Thread.sleep(1000);
		driver.findElement(By.id("submit")).click();
		Thread.sleep(10000);
		driver.quit();
	}
}
```

- Print Out of Code:



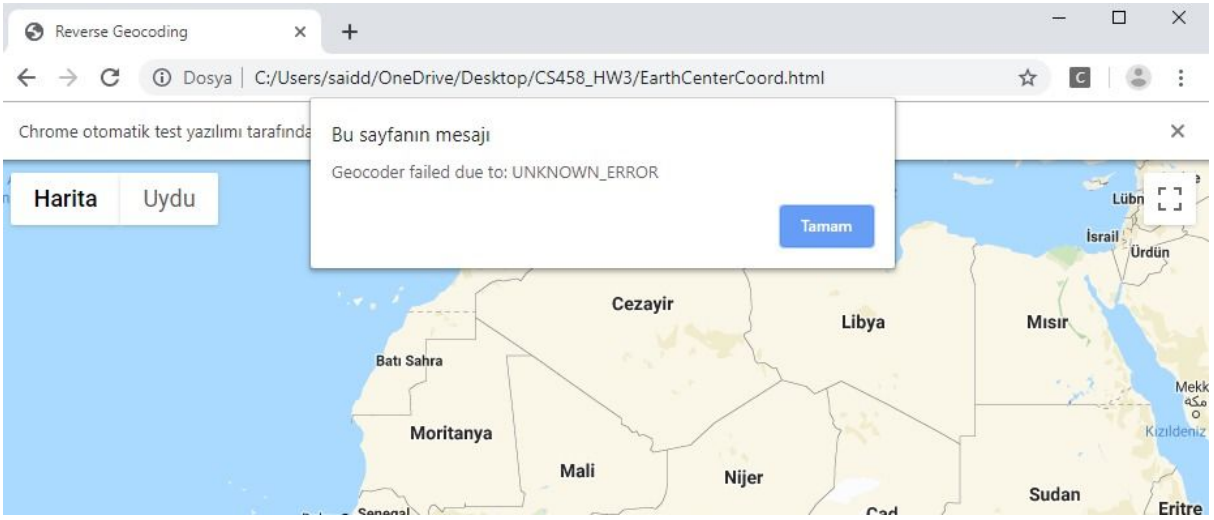
3. String value: Enter a string

- Test Code:

```
package homework3;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class hw3 {
	public static void main(String[] args) throws InterruptedException {
```

```
                System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
                Thread.sleep(1000);
                WebElement searchBox;
                searchBox = driver.findElement(By.linkText("Find Distance to
the Earth Center via Coordinates!"));
                searchBox.click();
                Thread.sleep(1000);
                WebElement searchBox1;
                searchBox1 = driver.findElement(By.id("latlng"));
                searchBox1.clear();
                searchBox1.sendKeys("furkan, hüseyin");
                Thread.sleep(1000);
                driver.findElement(By.id("submit")).click();
                Thread.sleep(10000);
                driver.quit();
        }
}
```

- Print Out of Code:



- **MAIN CODE**

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>Reverse Geocoding</title>
    <style>
      #map {
        height: 100%;
      }
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
```

```
    }
    #floating-panel {
      position: absolute;
      top: 10px;
      left: 25%;
      z-index: 5;
      background-color: #fff;
      padding: 5px;
      border: 1px solid #999;
      text-align: center;
      font-family: 'Roboto','sans-serif';
      line-height: 30px;
      padding-left: 10px;
    }
    #floating-panel {
      position: absolute;
      top: 5px;
      left: 50%;
      margin-left: -180px;
      width: 350px;
      z-index: 5;
      background-color: #fff;
      padding: 5px;
      border: 1px solid #999;
    }
    #latlng {
      width: 225px;
    }
  </style>
</head>
<body>
  <div id="floating-panel">
    <input id="latlng" type="text" value="0,0">
    <input id="submit" type="button" value="Reverse Geocode">
  </div>
  <div id="map"></div>
  <script>
    function initMap() {
      var input = document.getElementById('latlng').value;
      var latlngStr = input.split(',', 2);
      //var latlng = {lat: parseFloat(latlngStr[0]), lng: parseFloat(latlngStr[1])};
      var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 8,
        center: {lat: parseFloat(latlngStr[0]),lng: parseFloat(latlngStr[1])}
      });
      var geocoder = new google.maps.Geocoder;
      var infowindow = new google.maps.InfoWindow;

      document.getElementById('submit').addEventListener('click', function() {
        geocodeLatLng(geocoder, map, infowindow);
      });
```

```
}

function geocodeLatLng(geocoder, map, infowindow) {
  var input = document.getElementById('latlng').value;
  var latlngStr = input.split(',', 2);
  var latlng = {lat: parseFloat(latlngStr[0]), lng: parseFloat(latlngStr[1])};
  geocoder.geocode({'location': latlng}, function(results, status) {
    if (status === 'OK') {
      if (results[0]) {
        map.setZoom(11);
        var marker = new google.maps.Marker({
          position: latlng,
          map: map
        });
        infowindow.setContent(results[0].formatted_address);
        infowindow.open(map, marker);
      } else {
        window.alert('No results found');
      }
    } else {
      window.alert('Geocoder failed due to: ' + status);
    }
  });
}
</script>
<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCQ0zQ77DEqkok
OvnCs6IPIII9Xcb35bUM&callback=initMap">
  </script>
 </body>
</html>
```

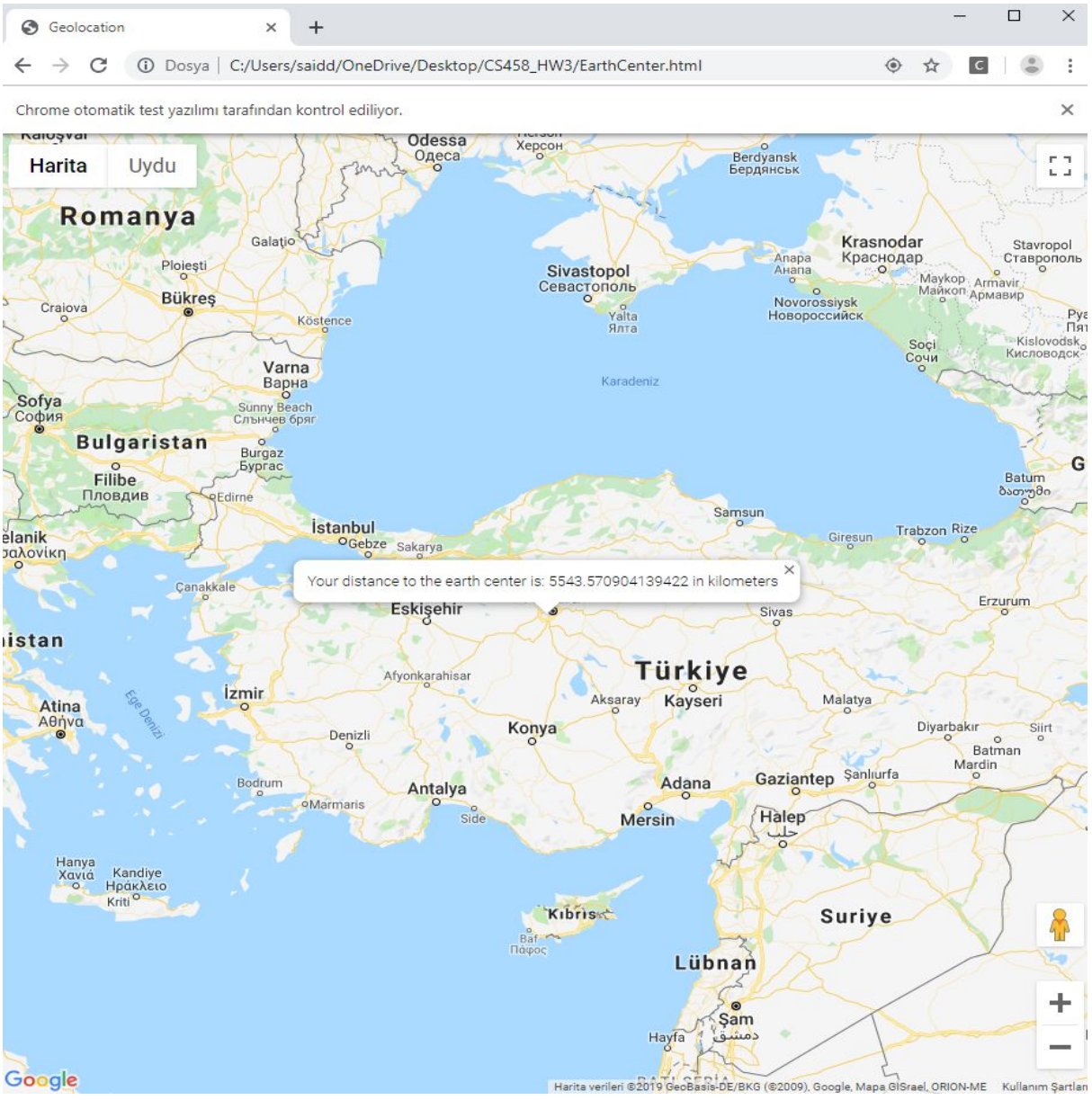
d. It gets GPS of your device, then it shows your distance to the earth center.

1. True Case: It gets your location normally.
    - Test Code:

```
package homework3;
import java.util.HashMap;
import java.util.Map;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
public class hw3 {
        public static void main(String[] args) throws InterruptedException {
                Map<String, Object> prefs = new HashMap<String,
Object>();
        prefs.put("profile.default_content_setting_values.notifications", 1);
                ChromeOptions options = new ChromeOptions();
```

```
                options.setExperimentalOption("prefs", prefs);
            System.setProperty("webdriver.chrome.driver",
"C:\\Users\\saidd\\OneDrive\\Desktop\\chromedriver.exe");
                WebDriver driver = new ChromeDriver(options);
driver.get("C:\\Users\\saidd\\OneDrive\\Desktop\\CS458_HW3\\Main.html");
                Thread.sleep(1000);
                WebElement searchBox;
                searchBox = driver.findElement(By.linkText("Find Distance to
the Earth Center via GPS!"));
                searchBox.click();
                Thread.sleep(2000);
                driver.quit();
            }
}
```

- Print Out of Code



2. Without wi-fi: Enter without wi-fi
    - Test Code:
      The code here is the same as the true case only output is different
    - Print Out of Code:

3. Don’t accept website to reach your location
    - Test Code:
    - The code here is the same except:
      prefs.put("profile.default_content_setting_values.notifications", 2);
    - Print Out of Code

2. Refactor your code and explain what you have done.

Before reading this step, we recommend you to read our answer to the 3rd question.

After the green lines, we've started to work on refactoring code. Thinking about the ways that we can enhance both the performance of the system and the aesthetic visuality of the system.

For the performance, we've focused on optimizing the code. First we've found and deleted the dead codes. Then Found the earth's center distance with mathematical formulas which are presented in our code. That way, instead of calling a library function (which would be inefficient since function calls costly), we handle it ourselves. Most importantly, for locating the nearest city center, instead of comparing the current location with the center of every city on earth, we created our own data set (which is attached in appendix) that contains every city's location in Turkey. That way we don't need to check every city on earth and do not need to call a library function which would decrease our efficiency. Since we only run this program within Turkey boundaries, this approach didn't affect the correctness of our program.

For the visualization, we've refactored the code that when it shows the distance to earth's center, it also shows your location or the location that you've entered instead of just showing the distance. Also at this step, instead of showing the distance value in a pop-up which is at the top of the window, we displayed it at the pop-up screen at the top of the current location. We also changed our zoom value give better results. At the error messages, we refactored it so that it gives informal errors which helps the user to see the cause of the problem.

3. Evaluate your TDD experience.

Working on Test Driven Development style was an unusual experience. It felt like we're doing things reverse this time since we were writing test cases first and developing the system according to it. There was always something to do since TDD relies on very short development cycles which made the development process fun.

Our steps can be briefly explained like this: First we've created the test cases and wrote the test case codes accordingly. And of course, since we don't have an implemented system, our test cases failed. This was our red line. Then we started to implement the system which the skeleton (interface) was already defined in our test cases. This was our green line. When the code was finally working and we can see the test cases are succeeding, our green line has ended and we start to think about ways to optimize our code both on performance and visuality which was the starting point of refactoring. We've made optimizing, removal of dead code, optimization, some aesthetic enhancements for visualization etc. While working on refactoring the code, we didn't forget to re-test the system to see if we made any changes that affected the functionality of the code which was our regression test (even if this is not a main part of TDD).

TDD is a very different way to approach to development but we see that it was quite useful for planning since we can see that should be in our system, which is to say, we can see the interface of the system before developing. We know what kind of components will be used in system and while developing we just filling the blanks. It’s really like solving a puzzle. One of the best parts is the refactoring part because while developing a system in the normal way, we usually tend to forget to enhance our program, we do not usually check if there is a way to optimize our code properly but since it is a mandatory step at TDD, we really had to think about ways to improve the system.