-1) Login authentication (Tested and Updated Response)

Note: Need to mandatorily pass only userName and password in the request

```
Endpoint: /validateunameandpassword
Request params:{
method:”get”,
headers:{

   “ContentType”:”application/json”
},
 url:”http://localhost:3000//validateunameandpassword/uname=’”+username+”’&”+password=”’+pwd+”’”,
}
```

Request:

```
{
   "userName": "May",
   "password": "OYC86ZKA5YB"
}
```

Response: (JSON Object)

```
{
          code: 200/500,
          message: String,
          data:
          {
             "userNameMessage": "Valid Username",
             "passwordMessage": "Valid Password"
          }
```

2) **Get All Users(For the Super-admin) (Tested and Updated Response)**

```
Endpoint: /user/getUsers
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/getUsers”,
```

Response: (JSON Object)

```
[
       {
        "userId": 1,
        "userName": "Charu",
        "password": "chs263",
        "fullName": "Charu Sreedharan",
        "memberType": 2,
        "userType": 2,
        "email": "chs263@pitt.edu",
        "contactNumber": "4127087401",
        "joinedOn": "2019-03-25T04:05:00.000+0000",
        "addressDetails": {
            "addressId": 101,
            "address": "6345 Street #1",
            "zipCode": 15123
        },
]
```

3) **Delete Users**

http://localhost:8080/user/deleteUsers/userName=XXX

```
Endpoint: /user/deleteUsers/userName=XXX
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/deleteUsers/userName=XXX”
```

4) **User registration(Tested and Updated Response)**

```
Endpoint: /user/addUser
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/addUser”,
data:
{
        "userName": "XXX",
        "password": "chs263",
        "fullName": "Charu Sreedharan",
        "memberType": 2,
        "userType": 2,
        "email": "chs263_4@pitt.edu",
        "contactNumber": "4127087403",
        "joinedOn": "2019-03-25T04:05:00.000+0000",
        "addressDetails": {
          "addressId": 101,
          "address": "6345 Street #1",
          "zipCode": 15123,
        }
}
```

Response: (Check status code-200/500)

```
{
        code: 200/500,
        message: String,
}
```

Role - Restaurant Owner

1) Get Restaurants By User Name

Endpoint: /restaurant/getRestaurantByUserName

```
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/restaurant/getRestaurant/userName=”XXXX””,
 e.g: http://localhost:8080/restaurant/getRestaurant/uname=geetha
```

Response: (JSON Object)

```
        [
          {
            "restaurantId": 1,
            "userId": 3,
            "name": "Aaryas",
            "city": "PA",
            "zipCode": "15234"
          },
          {
            "restaurantId": 3,
            "userId": 3,
            "name": "Pizza Hut",
            "city": "Boston",
            "zipCode": "45678"
        },
        {
            "restaurantId": 6,
            "userId": 3,
            "name": "KFC",
            "city": "dubai",
            "zipCode": "14289"
        }
    ]
    OR
    NULL OBJECT
```

1) Create Cuisines

```
Endpoint: /restaurant/{restId}/cuisine/addCuisine
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/restaurant/{restId}/cuisine/addCuisine”,
data:
{

    "cuisineName": "American"

}

Response: (Check status code-200/500)
{
    code: 200/500,
    message: String,
}
```

View Cuisines:

```
Endpoint: /restaurant/{restId}/cuisine/getCuisine
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/restaurant/{restId}/cuisine/getCuisine,
```

Response: (JSON Object):

```
[
   {
      "cuisineId": 4,
      "cuisineName": "Continental"
   },
   {
      "cuisineId": 5,
      "cuisineName": "Japanese"
   },
   {
      "cuisineId": 6,
      "cuisineName": "English"
   }
]
```

Update Cuisines

Endpoint: http://localhost:8080/cuisine/updateCuisine/cuisineID=XXX

```
Request params:{
method:”put”,
 headers:{
   “ContentType”:”application/json”
},
 url:http://localhost:8080/cuisine/updateCuisine/cuisineID=XXX

 Data:
 {

         "cuisineName": "Italian"

 }
```

Response: (Check status code-200/500)

```
{
         code: 200/500,
         message: String,
}
```

Delete Cuisines

Endpoint: /cuisine/deleteCuisine/cuisineID=XXX

```
Request params:{
method:”delete”,
 headers:{
   “ContentType”:”application/json”
},
 url:http://localhost:8080/cuisine/deleteCuisine/cuisineID=XXX

Response: (Check status code-200/500)
{
            code: 200/500,
            message: String,
}
```

2) Create Items

```
Endpoint: /restaurant/{restId}/cuisine/{cuisineId}/item/addItem
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/restaurant/{restId}/cuisine/{cuisineId}/item/addItem”,

data:
    {
        "itemName": "Tikka Masala",
        "description": "Tikka Masala",
        "price": 75
    }

Response: (Check status code-200/500)
{
            code: 200/500,
            messa/restaurant/{restId}/cuisine/{cuisineId}/item/getItemge: String,
}
```

View all items under a cuisine and a restaurant:

```
Endpoint:
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/cuisine/{cuisineId}/item/getItem,
```

Response: (JSON Object)

```
[
   {
       "itemId": 6,
       "cuisineId": 2,
       "itemName": "Noodles",
       "description": "Noodles",
       "price": 20
   },
   {
       "itemId": 9,
       "cuisineId": 2,
       "itemName": "Idli",
       "description": "Idli",
       "price": 15
   },
   {
       "itemId": 10,
       "cuisineId": 2,
       "itemName": "Dosa",
       "description": "Dosa",
       "price": 7
   }
]
```

Update Items

Endpoint: /item/updateItem/itemID=XXX

```
Request params:{
method:”put”,
 headers:{
   “ContentType”:”application/json”
},
data:
{
        "itemName": "MeatBalls",
        "description":"MeatBalls",
        "price": 40
```

}

url:http://localhost:8080/item/updateItem/itemID=XXX

Response: (Check status code-200/500)
{
	code: 200/500,
	message: String,
}

Delete Items

Endpoint: /items/deleteItem/itemID=XXX
Request params:{
method:”delete”,
 headers:{
   “ContentType”:”application/json”
},
 url:http://localhost:8080/items/deleteItem/itemID=XXX

Response: (Check status code-200/500)
{
	code: 200/500,
	message: String,
}

User Role:

Fetch User Details By User Name (Tested and Updated Response)

Endpoint: /user/getUserDetails/userName={userName}
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 Url: “http://localhost:8080/user/getUserDetails/userName={userName}”

Response: (JSON Object)

```
{
   "userId": 1,
   "userName": "Charu",
   "password": "chs263",
   "fullName": "Charu Sreedharan",
   "memberType": 2,
   "userType": 2,
   "email": "chs263@pitt.edu",
   "contactNumber": "4127087401",
   "joinedOn": "2019-03-25T04:05:00.000+0000",
   "addressDetails": {
       "addressId": 101,
       "address": "6345 Street #1",
       "zipCode": 15000
   },
   "loginStatus": 0
}
```

Update User Details (Tested and Updated Response)

Endpoint: /user/updateUserDetails/userName={userName}

```
Request params:{
method:”put”,
 headers:{
   “ContentType”:”application/json”
},
data:
 {
     "userId": 1,
     "userName": "Charu",
     "password": "chs263",
     "fullName": "Charu Sreedharan",
     "memberType": 2,
     "userType": 2,
     "email": "chs263@pitt.edu",
     "contactNumber": "4127087401",
     "joinedOn": "2019-03-25T04:05:00.000+0000",
     "addressDetails": {
         "addressId": 101,
         "address": "6345 Street #1",
         "zipCode": 15000
     },
     "loginStatus": 0
 }
 url:http://localhost:8080/items/updateItem/itemID=XXX
```

Response: (Check status code-200/500)

```
{
        code: 200/500,
        message: String,
}
```

Log out User (Tested and Updated Response)

```
Endpoint: /user/logOutUser/userName={userName}
Request params:{
method:”get”,
 headers:{
  “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/logOutUser/userName={userName}
```

```
Response: (Check status code-200/500)
{
        code: 200/500,
        message: String,
}
```

GetDistinctCityNames((Tested and Updated Response)

```
Endpoint: /restaurant/city/getDistinctCityNames
Request params:{
method:”get”,
 headers:{
  “ContentType”:”application/json”
},
 url:”http://localhost:8080/city/getDistinctCityNames,
```

Response: (JSON Object)

```
[
   "Cranbrook",
   "Champlain",
   "Capena",
   "Zamora",
   "Lithgow",
   "Cimitile",
   "Lac Ste. Anne"
]
```

Get RestaurantsByCityName(Tested and Updated Response)

```
Endpoint: /restaurant/getRestaurantsByCityName/cityName={cityName}
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080//restaurant/getRestaurantsByCityName/cityName={cityName} ,
```

Response: (JSON Object)

```
[
   {
      "restaurantId": 10,
      "userId": 24,
      "name": "Et Arcu Imperdiet PC",
      "city": "Montreal",
      "zipCode": 54963,
      "totalPrice": 0
   }
]
```

1) <u>Get Restaurants </u>(Tested and Updated Response)

```
Endpoint: /restaurant/getAllRestaurants
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/getRestaurants,
```

Response: (JSON Object)

```
[
   {
      "restaurantId": 1,
      "userId": 384,
      "name": "Vivamus Euismod Urna Associates",
      "city": "Kalken",
      "zipCode": 8115,
      "totalPrice": 0
   },
   {
      "restaurantId": 2,
      "userId": 137,
      "name": "Sit Amet LLC",
      "city": "Kelowna",
      "zipCode": 95577,
      "totalPrice": 0
   },
   {
      "restaurantId": 3,
      "userId": 387,
      "name": "Mauris Ut Associates",
      "city": "Hallein",
      "zipCode": 58859,
      "totalPrice": 0
   }
]
```

2) Get cuisines and items by Restaurant ID (Tested and Updated Response)

```
Endpoint: /restaurant/{restId}/cuisine/getCuisinesAndItemsByRestID
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/getCuisinesItemsByRestID/RestID=”XXXX””,
```

Response: (JSON Object)

```
[
   {
      "cuisineId": 3,
      "cuisineName": "Thai",
      "itemDetailsList": [
         {
            "itemId": 35,
            "cuisineId": 3,
            "itemName": "Satay",
            "description": "Choice of meat on skewer, BBQ beef served with peanut sauce and
cucumber salad.",
            "price": 8.95
         },
      {
        "itemId": 36,
        "cuisineId": 3,
        "itemName": "Thai Dumpling\r\n",
        "description": "Served with shrimp, ground chicken, vegetable and served with black
soy sauce.",
        "price": 6.95
      },
      {
        "itemId": 37,
        "cuisineId": 3,
        "itemName": "Mee Krob\r\n",
        "description": "Sweet crispy noodles with shrimp, chicken and bean sprouts.",
        "price": 7.95
      },
     },
  {
    "cuisineId": 5,
    "cuisineName": "Italian",
    "itemDetailsList": [
      {
        "itemId": 70,
        "cuisineId": 5,
        "itemName": "Mussels di Johnny ",
        "description": "New Zealand mussels on the half shell in a delightful garlic & butter
sauce. \r\n",
        "price": 5.99
      },
      {
        "itemId": 71,
        "cuisineId": 5,
        "itemName": "Calamari Antipasto ",
        "description": "Fresh Calamari steamed and served chilled, mixed with balsamic
vinaigrette, artichoke hearts, roasted red peppers, kalamata olives, diced almonds, and dried
cranberries. \r\n",
        "price": 5
      }
    ]
  },
  {
    "cuisineId": 6,
    "cuisineName": "American",
    "itemDetailsList": [
            {
                "itemId": 80,
                "cuisineId": 6,
                "itemName": "lobster roll",
                "description": "LOBSTER, BIBB LETTUCE, TARRAGON AIOLI, BRIOCHE, OLD BAY
HOUSE CHIPS",
                "price": 21
            },
            {
                "itemId": 81,
                "cuisineId": 6,
                "itemName": "tuna bowl",
                "description": "ARRO GRAIN, PEA TENDRILS, CARROTS, KIMCHI CUCUMBER,
SOFT BOILED EGG, AVOCADO, SHOYU",
                "price": 21
            }
        ]
    },
    {
        "cuisineId": 8,
        "cuisineName": "Korean",
        "itemDetailsList": []
    },
    {
        "cuisineId": 10,
        "cuisineName": "Greek",
        "itemDetailsList": []
    }
]
```

// total_price info provided from the **Frontend** (quantity*item_price)
3) **AddtoCart (Tested and Updated Response) -> Changed to userName from userId**

Endpoint: /cart/addToCart
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/cart/addToCart”,
data:

```
[
    {
        "userName": "May",
        "restaurantId": 3,
        "cuisineId": 5,
        "itemId": 70,
        "quantity": 2,
        "totalPrice": 10,
        "spiceLevel": 5,
        "comment": "Good"
    },
    {
        "userName": "Quinn",
        "restaurantId": 7,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    }
]
```

Response:

```
[
    {
        "suborderId": 59,
        "userId": 7,
        "userName": "May",
        "restaurantId": 3,
        "cuisineId": 5,
        "itemId": 70,
        "quantity": 2,
        "totalPrice": 10,
        "spiceLevel": 5,
        "comment": "Good"
    },
    {
        "suborderId": 60,
        "userId": 8,
        "userName": "Quinn",
        "restaurantId": 7,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    }
]
```

4) ViewCartBySuborderIds (Tested and Updated Response)

Endpoint: /cart/viewCartBySuborderIds

```
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/cart/viewCartBySuborderIds
 ”,
Data:
[
            {"id": 1},
            {"id": 2}
]
```

Response: (Check status code-200/500)

```
[
    {
        "suborderId": 1,
        "userId": 1,
        "restaurantId": 3,
        "cuisineId": 5,
        "itemId": 70,
        "quantity": 2,
        "totalPrice": 10,
        "spiceLevel": 5,
        "comment": "Good"
    },
    {
        "suborderId": 2,
        "userId": 2,
        "restaurantId": 5,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    }
]
```

5) View Cart by User Name (Tested and Updated Response)

```
Endpoint: cart/viewCartByUserName/username={userName}
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/cart/viewCartByUserName/username={userName}”,
```

Response:

```
[
    {
        "suborderId": 2,
        "userId": 2,
        "restaurantId": 5,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    },
    {
        "suborderId": 14,
        "userId": 2,
        "restaurantId": 5,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    },
    {
        "suborderId": 16,
        "userId": 2,
        "restaurantId": 5,
        "cuisineId": 7,
        "itemId": 36,
        "quantity": 5,
        "totalPrice": 100,
        "spiceLevel": 10,
        "comment": "Nice"
    }
]
```

6) DeleteCart: (Tested and Updated Response)

```
Endpoint: /cart/deleteCart
Request params:{
method:"delete",
 headers:{
   "ContentType":"application/json"
},
 url:http://localhost:8080/items/deleteItem/itemID=XXX
```

```
Data:
 [
        {"id": 6},
        {"id": 2},
        {"id": 3},
        {"id": 4}
 ]
```

```
Response: (Check status code-200/500)
{
        code: 200/500,
        message: String,
}
```

// TotalPrice info and deliveryFlag populated in server side. Don't provide them.

7) CreateOrder: (Tested and Updated Response) ->

Note: Need to mandatorily pass only suborderid and totalPrice inside suborderDetailsVO in the request

Endpoint: /order/createOrder

```
Request params:{
method:”post”,
headers:{
   “ContentType”:”application/json”
},

 Url: “http://localhost:8080/order/createOrder”,


data:

{
   "suborderDetailsVO": [
      {
         "suborderId": 7,
         "totalPrice": 100
      },
      {
         "suborderId": 8,
         "totalPrice": 200
      },
      {
         "suborderId": 9,
         "totalPrice": 300
      }
   ],
   "userId": 1,
   "restaurantId": 1,
   "deliveryAddressDetails": {

      "address": "Ap #228-2542 Non Rd.",
      "zipCode": 15260
   }
}
```

Response: (Check status code-200/500)

```
{
         code: 200/500,
         message: String,
         }
}
```

8) Get All Orders (Tested and Updated Response)

```
Endpoint: /order/getAllOrders
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/order/getAllOrders
```

Response: (JSON Object)

```
[
    {
        "orderId": 7,
        "suborderDetailsVO": [
            {
                "suborderId": 10,
                "userId": 1,
                "restaurantId": 3,
                "cuisineId": 5,
                "itemId": 70,
                "quantity": 2,
                "totalPrice": 10,
                "spiceLevel": 5,
                "comment": "Good"
            },
            {
                "suborderId": 12,
                "userId": 1,
                "restaurantId": 3,
                "cuisineId": 5,
                "itemId": 70,
                "quantity": 2,
                "totalPrice": 10,
                "spiceLevel": 5,
                "comment": "Good"
            },
            {
                "suborderId": 16,
                "userId": 2,
                "restaurantId": 5,
                "cuisineId": 7,
                "itemId": 36,
                "quantity": 5,
                "totalPrice": 100,
                "spiceLevel": 10,
                "comment": "Nice"
            }
        ],
        "userId": 1,
        "restaurantId": 1,
        "totalPrice": 45,
        "deliveryAddressDetails": {
            "addressId": 101,
            "address": "Ap #228-2542 Non Rd.",
            "zipCode": 99073
        },
        "estimatedDeliveryTime": "2019-04-16T04:00:00.000+0000",
        "deliveryFlag": 1
    },
    {
        "orderId": 8,
        "suborderDetailsVO": [
            {
                "suborderId": 10,
                "userId": 1,
                "restaurantId": 3,
                "cuisineId": 5,
                "itemId": 70,
                "quantity": 2,
                "totalPrice": 10,
                "spiceLevel": 5,
                "comment": "Good"
            }
        ],
        "userId": 2,
        "restaurantId": 5,
        "totalPrice": 50,
        "deliveryAddressDetails": {
            "addressId": 102,
            "address": "2614 Sed Avenue",
            "zipCode": 80068
        },
        "estimatedDeliveryTime": "2019-04-16T04:00:00.000+0000",
        "deliveryFlag": 1
    }
]
```

9) Get Order Details from User Name(Tested and Updated Response)

Endpoint: /order/getOrdersByUserName/uname={uname}
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/order/getOrdersByUserName/uname={uname}

Response: (JSON Object)

```
[
   {
       "orderId": 7,
       "suborderDetailsVO": [
          {
              "suborderId": 10,
              "userId": 1,
              "restaurantId": 3,
              "cuisineId": 5,
              "itemId": 70,
              "quantity": 2,
              "totalPrice": 10,
              "spiceLevel": 5,
              "comment": "Good"
          },
          {
              "suborderId": 16,
              "userId": 2,
              "restaurantId": 5,
              "cuisineId": 7,
              "itemId": 36,
              "quantity": 5,
              "totalPrice": 100,
              "spiceLevel": 10,
              "comment": "Nice"
          }
       ],
       "userId": 1,
       "restaurantId": 1,
       "totalPrice": 45,
       "deliveryAddressDetails": {
            "addressId": 101,
            "address": "Ap #228-2542 Non Rd.",
            "zipCode": 15260
        },
        "estimatedDeliveryTime": "2019-04-16T04:00:00.000+0000",
        "deliveryFlag": 1
    },
    {
        "orderId": 17,
        "suborderDetailsVO": [
            {
                "suborderId": 10,
                "userId": 1,
                "restaurantId": 3,
                "cuisineId": 5,
                "itemId": 70,
                "quantity": 2,
                "totalPrice": 10,
                "spiceLevel": 5,
                "comment": "Good"
            },
            {
                "suborderId": 16,
                "userId": 2,
                "restaurantId": 5,
                "cuisineId": 7,
                "itemId": 36,
                "quantity": 5,
                "totalPrice": 100,
                "spiceLevel": 10,
                "comment": "Nice"
            }
        ],
        "userId": 1,
        "restaurantId": 1,
        "totalPrice": 120,
        "deliveryAddressDetails": {
            "addressId": 101,
            "address": "Ap #228-2542 Non Rd.",
            "zipCode": 15260
        },
        "estimatedDeliveryTime": "2019-04-15T23:12:44.000+0000",
        "deliveryFlag": 0
    }
]
```

Review:

10) Create Review (Tested and Updated Response)

```
Endpoint: /review/createReview
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/review/createReview”,
data:
{
    "userId": 5,
    "restaurantId": 7,
    "stars": 1,
    "description": "Bad"
}

Response: (Check status code-200/500)
{
            code: 200/500,
            message: String,
            }
}
```

11) getAllReviews (Tested and Updated Response)

```
Endpoint: /review/getAllReviews
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080//review/getAllReviews

Response: (JSON Object)
[
    {
        "reviewId": 1,
        "userId": 272,
        "restaurantId": 441,
        "stars": 3,
        "description": "VZG97FIT0LZ"
    },
    {
        "reviewId": 2,
        "userId": 53,
        "restaurantId": 314,
        "stars": 6,
        "description": "OYB72RYU1HD"
    },
    {
        "reviewId": 3,
        "userId": 315,
        "restaurantId": 110,
        "stars": 9,
        "description": "UQI23YHN0JP"
    }
]
```

12) getReviewsByUserName(Tested and Updated Response)

```
Endpoint: /getReviewsByUserName/uname={uname}
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/getReviewsByUserName/uname={uname}
```

Response: (JSON Object)

```
[
    {
        "reviewId": 4,
        "userId": 50,
        "restaurantId": 195,
        "stars": 8,
        "description": "GOY64KZI3KI"
    }
]
```

13) Update Review((Tested and Updated Response)

Allow only stars and description to be modified

Endpoint: /review/updateReview/id={id}
Request params:{
method:”put”,
 headers:{
   “ContentType”:”application/json”
},
 Url: http://localhost:8080/review/updateReview/id={id}

 Data:
{
     "stars": 1,
     "description": "Bad"
 }

Response: (Check status code-200/500)
{
            code: 200/500,
            message: String,
}

14) Delete Review(Tested and Updated Response)

Endpoint: /review/deleteReview/id={id}
Request params:{
method:”delete”,
 headers:{
   “ContentType”:”application/json”
},
 url:http://localhost:8080/review/deleteReview/id={id}

Response: (Check status code-200/500)
{
            code: 200/500,
            message: String,
}

For Super-Admin(Not implemented yet)

Endpoint: /user/addRestaurant

```
Request params:{
method:”post”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/user/addCuisine”,
data:
{
        "cuisineName": "XXX"

}
}
```

Response: (Check status code-200/500)

```
{
            code: 200/500,
            message: String,
}
```

- Change User Registration

Agg1: (Tested and Updated Response)

- Sum up the Orders by a restaurant(sales) - restaurant owner home page

Endpoint: /restaurant/getRestaurantsSortedBySales/uname={uname}

```
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},
 url:”http://localhost:8080/restaurant/getRestaurantsBySales/uname={uname}
```

Response: (JSON Object)

```
[
    {
        "restaurantId": 501,
        "userId": 384,
        "name": "Hooters",
        "city": "Pittsburgh",
        "zipCode": 15209,
        "totalPrice": 1000
    },
    {
        "restaurantId": 502,
        "userId": 384,
        "name": "Primanti Bros",
        "city": "Pittsburgh",
        "zipCode": 12345,
        "totalPrice": 500
    },
    {
        "restaurantId": 1,
        "userId": 384,
        "name": "Vivamus Euismod Urna Associates",
        "city": "Kalken",
        "zipCode": 81151,
        "totalPrice": 215
    },
    {
        "restaurantId": 1,
        "userId": 384,
        "name": "Vivamus Euismod Urna Associates",
        "city": "Kalken",
        "zipCode": 81151,
        "totalPrice": 215
    },
    {
        "restaurantId": 1,
        "userId": 384,
        "name": "Vivamus Euismod Urna Associates",
        "city": "Kalken",
        "zipCode": 81151,
        "totalPrice": 215
    }
]
```

Agg2:(Tested and Updated Response)

Top cuisines sold by a restaurant: User home page. Top cuisine(button) based on the number of orders - User Role

Endpoint: /restaurant/{restId}/cuisine/getTopCuisinesByRestID
Request params:{
method:"get",

```
 headers:{
   “ContentType”:”application/json”
},
 Url: “http://localhost:8080/restaurant/10/cuisine/getTopCuisinesByRestID”
```

Response: (JSON Object)

```
[
   {
       "cuisineId": 1,
       "cuisineName": "Indian",
       "itemDetailsList": []
   },
   {
       "cuisineId": 2,
       "cuisineName": "Chinese",
       "itemDetailsList": []
   },
   {
       "cuisineId": 3,
       "cuisineName": "Thai",
       "itemDetailsList": []
   }
]
```

Agg3 (Tested and Updated Response)

Top-rated restaurants in a city. Search by city - User Role

Endpoint: /restaurant/getTopRatedRestaurantsByCityName/cityName={cityName}

```
Request params:{
method:”get”,
 headers:{
   “ContentType”:”application/json”
},

 Url: “http://localhost:8080/restaurant/getTopRatedRestaurantsByCityName/cityName={cityName}”
```

Response: (JSON Object) ->

```
[
   {
       "restaurantId": 504,
       "userId": 5,
       "name": "Siachen",
       "city": "Kalken",
       "zipCode": 55667,
       "totalPrice": 0
   },
   {
       "restaurantId": 503,
       "userId": 50,
       "name": "Kalken Mahal",
       "city": "Kalken",
       "zipCode": 695581,
       "totalPrice": 0
   },
   {
       "restaurantId": 1,
       "userId": 384,
       "name": "Vivamus Euismod Urna Associates",
       "city": "Kalken",
       "zipCode": 81151,
       "totalPrice": 0
   }
]
```

Search1: (Service above)(Tested and Updated Response above)

Searchrestaurantbycity(cityName)
and
getAllDistinctRestaurantCities(cityName)

Search2: (Tested and Updated Response)

Most economical items for a restaurant under a price:

Endpoint: /restaurant/getMostEconomicalItemsByRestId/restId={restId}&price={price}
Request params:{
method:”get”,
 headers:{
  “ContentType”:”application/json”
},

Url: “http://localhost:8080/item/getMostEconomicalItemsByRestId/restId={restId}&price={price}”

Response: (JSON Object) ->

```
[
    {
        "itemId": 138,
        "cuisineId": 10,
        "itemName": "Dolmades",
        "description": "Grape Leaves Stuffed with Rice, Pine Nuts and Raisins, EVOO.",
        "price": 13
    },
    {
        "itemId": 136,
        "cuisineId": 10,
        "itemName": "Garides Saganaki",
        "description": "Braised Jumbo Shrimp in Fresh Tomato Sauce, Melted Greek Feta Cheese, Organic Herbs.",
        "price": 17
    }
]
```