



School of Computing | Diploma in Information Technology

ST0280 Cloud and Service Development

Continuous Assessment 1

Done by:

Zachary Tan Chien Loong (p1829176)  
Lee Wei Xian(p1828878)

[LINK TO GITHUB REPO](#)

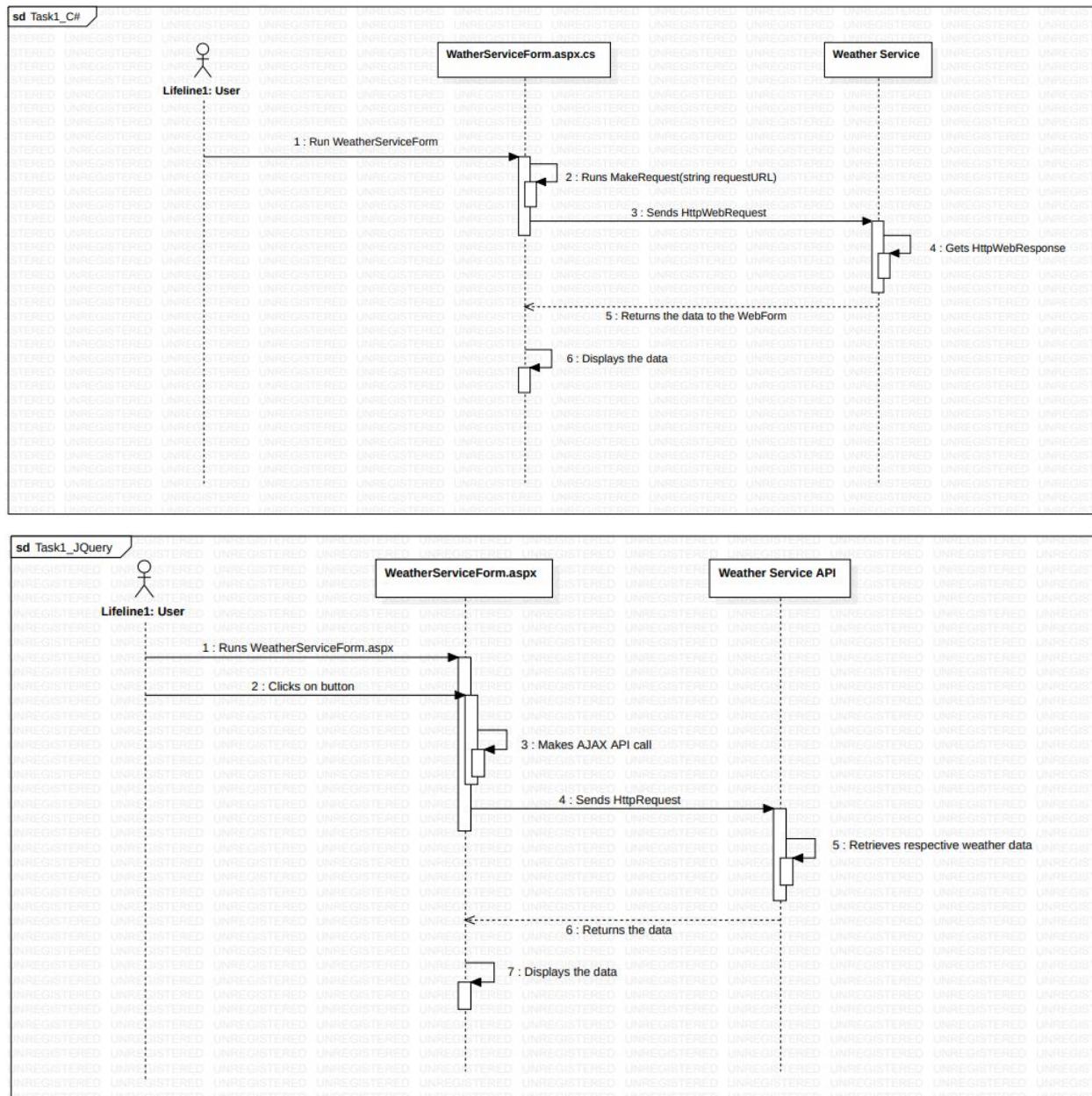
# Task1

Web-API Documentation:

## 1. Get weather service

METHOD	GET
ROUTE	<a href="https://api.data.gov.sg/v1/environment/2-hour-weather-forecast">https://api.data.gov.sg/v1/environment/2-hour-weather-forecast</a>
OPTIONAL PARAMETERS	date_time: String date: String
EXAMPLE	GET <a href="https://api.data.gov.sg/v1/environment/2-hour-weather-forecast">https://api.data.gov.sg/v1/environment/2-hour-weather-forecast</a>
RESPONSE	<pre>"area_metadata": [     {         "name": "Ang Mo Kio",         "label_location": {             "latitude": 1.375,             "longitude": 103.839         }     },     {         "name": "Bedok",         "label_location": {             "latitude": 1.321,             "longitude": 103.924         }     },     {         "name": "Bishan",         "label_location": {             "latitude": 1.350772,             "longitude": 103.839         }     } ],</pre>

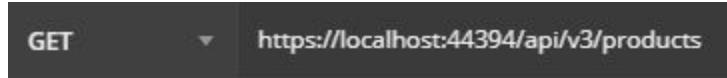
## Sequence Diagram:



## Task2

Web-Api Documentation:

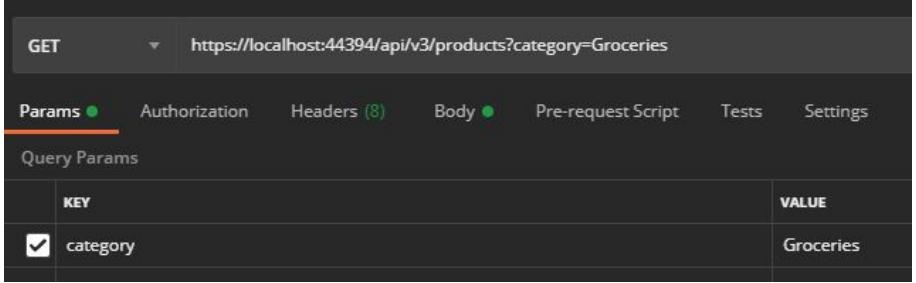
1. Get all products

METHOD	GET
ROUTE	/api/v3/products
EXAMPLE	
RESPONSE	 <pre>[{"Id": 1, "Name": "Tomato soup", "Category": "Groceries", "Price": 1.39}, {"Id": 2, "Name": "Yo-yo", "Category": "Toys", "Price": 3.75}, {"Id": 3, "Name": "Hammer", "Category": "Hardware", "Price": 16.99}, {"Id": 5, "Name": "Pokeball", "Category": "Level 99", "Price": 99.0}]</pre>

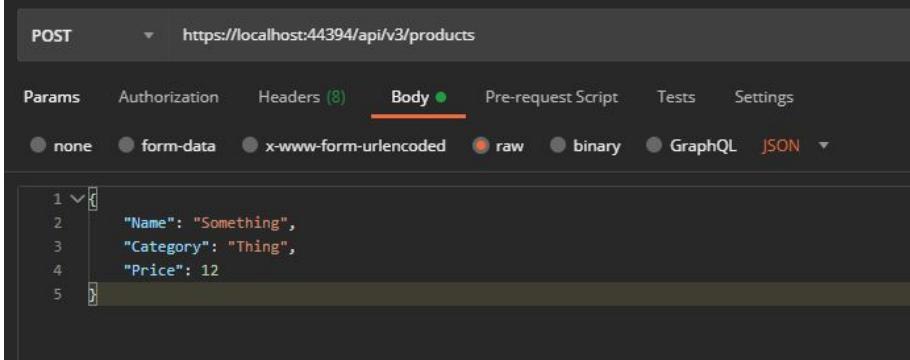
## 2. Get product by id

METHOD	GET
ROUTE	/api/v3/products/{id:INT:min(1)}
EXAMPLE	GET <a href="https://localhost:44394/api/v3/products/1">https://localhost:44394/api/v3/products/1</a>
RESPONSE	<pre>[{"Id": 1, "Name": "Tomato soup", "Category": "Groceries", "Price": 1.39}]</pre>

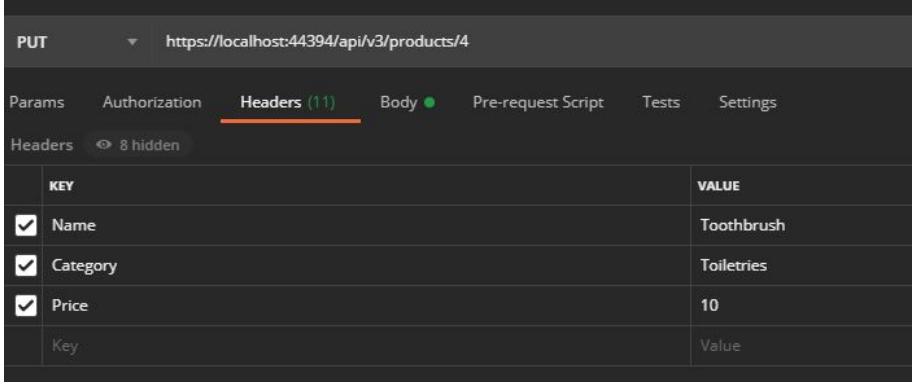
### 3. Get product by category

METHOD	GET
ROUTE	/api/v3/products?category={categoryName}
EXAMPLE	
RESPONSE	<pre>{     "Id": 1,     "Name": "Tomato soup",     "Category": "Groceries",     "Price": 1.39 }</pre>

#### 4. Create new product

METHOD	POST
ROUTE	/api/v3/products
EXAMPLE	 <pre>1 &lt;pre&gt;{ 2   "Name": "Something", 3   "Category": "Thing", 4   "Price": 12 5 }&lt;/pre&gt;</pre>
RESPONSE	 <pre>1 &lt;pre&gt;{ 2   "Id": 5, 3   "Name": "Something", 4   "Category": "Thing", 5   "Price": 12.0 6 }&lt;/pre&gt;</pre>

## 5. Update product by id

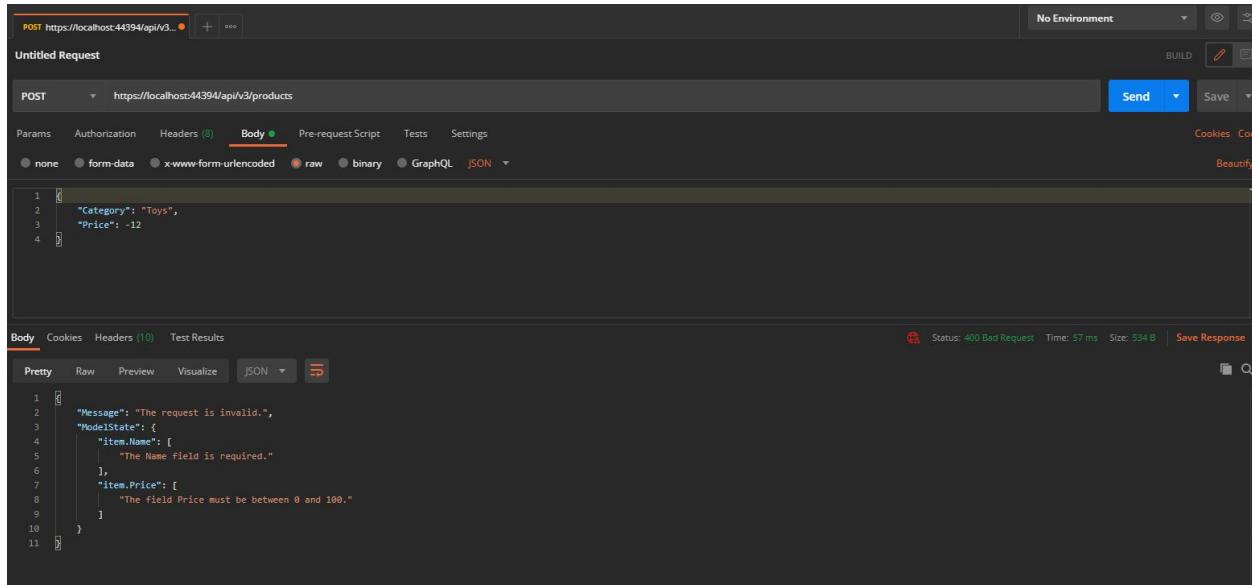
METHOD	PUT								
ROUTE	/api/v3/products/{id:INT}								
EXAMPLE	 <p>The screenshot shows the Postman interface with a PUT request to https://localhost:44394/api/v3/products/4. The Headers tab is selected, displaying three key-value pairs:</p> <table border="1"><thead><tr><th>KEY</th><th>VALUE</th></tr></thead><tbody><tr><td>Name</td><td>Toothbrush</td></tr><tr><td>Category</td><td>Toiletries</td></tr><tr><td>Price</td><td>10</td></tr></tbody></table>	KEY	VALUE	Name	Toothbrush	Category	Toiletries	Price	10
KEY	VALUE								
Name	Toothbrush								
Category	Toiletries								
Price	10								

## 6. Delete product

METHOD	DELETE	
ROUTE	/api/v3/products/{id:INT}	
EXAMPLE	<b>DELETE</b> ▾ <a href="https://localhost:44394/api/v3/products/1">https://localhost:44394/api/v3/products/1</a>	

## Screenshots:

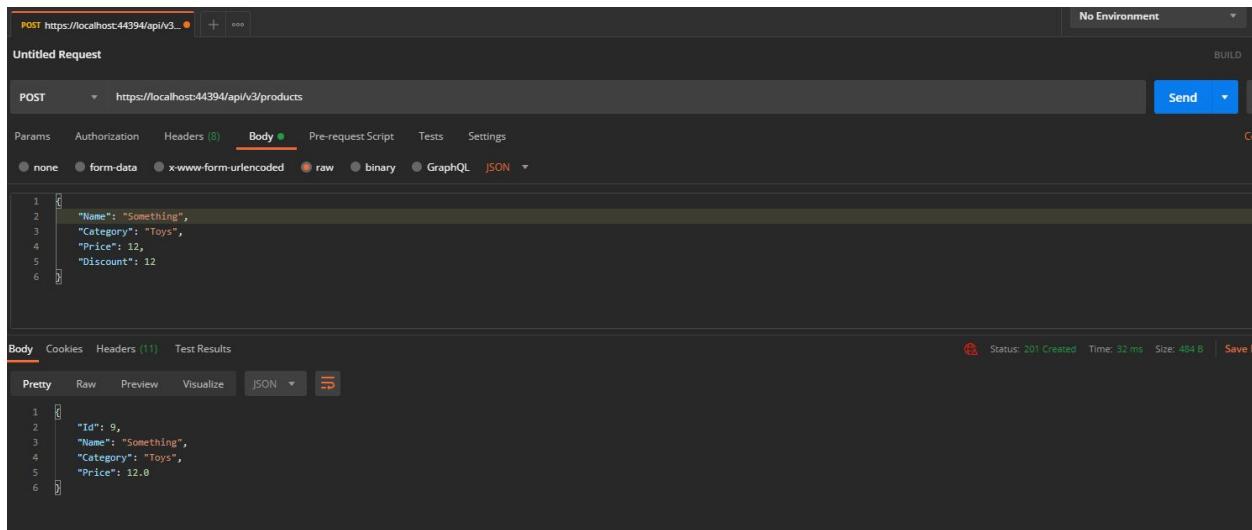
*Under-posting:* Situation whereby the client leaves out some properties when sending request(s) to the controller.



The screenshot shows a POST request to `https://localhost:44394/api/v3/products`. The request body is JSON with only `"Category": "Toys"` and `"Price": -12`. The response status is 400 Bad Request, with the message: `"Message": "The request is invalid.", "ModelState": { "item.Name": [ "The Name field is required." ], "item.Price": [ "The field Price must be between 0 and 100." ] }`.

[This happens when there is a case of under-posting]

*Over-posting:* Situation whereby the client sends more data than expected to the controller.



The screenshot shows a POST request to `https://localhost:44394/api/v3/products`. The request body is JSON with `"Name": "Something", "Category": "Toys", "Price": 12, "Discount": 12`. The response status is 201 Created, with the message: `"Id": 9, "Name": "Something", "Category": "Toys", "Price": 12.0`.

[This happens when there is a case of over-posting]

## Task3

Web-Api Documentation:

1. Register an account

METHOD	POST
ROUTE	/api/Account/Register
PARAMETERS REQUIRED	email: String, password: String, confirmPassword: String, reCaptcha Site-Key: String

2. Login to an account

METHOD	POST
ROUTE	/api/Token
PARAMETERS REQUIRED	username: String, password: String, 'password': grant_type

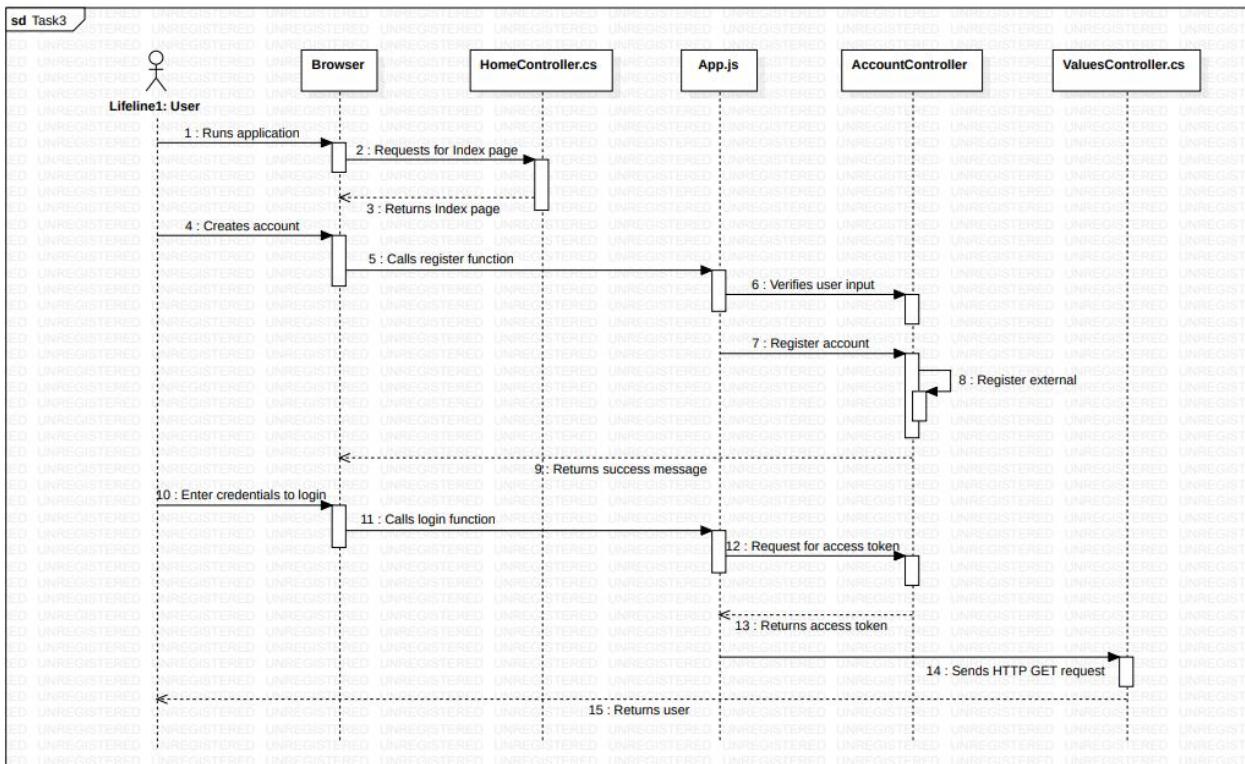
3. Logout of an account

METHOD	POST
ROUTE	/api/Account/Logout

#### 4. Get Values(Authorization)

METHOD	GET
ROUTE	/api/values

Sequence Diagram:



## Screenshots:

### Register Successful:

The screenshot shows the Postman application interface. The top bar includes 'File', 'Edit', 'View', 'Help', and a 'Restart' button. The main workspace is titled 'My Workspace' with an 'Invite' button. A message at the top left says 'An update has been downloaded for Postman. Restart now to install the update.' Below this, the 'History' tab is selected, showing a list of recent requests. An 'Untitled Request' is currently selected, showing a POST request to 'https://localhost:44320/api/Account/Register'. The 'Body' tab is active, showing form-data parameters: 'confirmationToken' with value 'Ntuclhub0310!' and 'RecaptchaSiteKey' with value '6LfffwgaAAAAADFoQljbkd0XJV6oys6ygcPa77wd'. The response status is 200 OK, time 299 ms, size 328 B. The body of the response is a single character '1'.

### Register(email invalid!):

This screenshot shows the same Postman interface after an attempt to register with an invalid email. The 'History' tab shows previous successful registrations. The current 'Untitled Request' is a POST to 'https://localhost:44320/api/Account/Register'. The 'Body' tab shows parameters: 'email' with value 'alfred@' and 'password' with value 'Ntuclhub0310!'. The response status is 400 Bad Request, time 7.89 s, size 472 B. The JSON response body is as follows:

```
1  [
2      "Message": "The request is invalid.",
3      "ModelState": {
4          "":
5              [
6                  "Email 'alfred@' is invalid."
7              ]
8          ]
9      }
10 }
```

## Register(password invalid):

The Postman interface shows a failed POST request to `https://localhost:44320/api/Account/Register`. The response status is 400 Bad Request, indicating the password is invalid. The JSON response body is:

```
1 "Message": "The request is invalid.",  
2 "ModelState": {  
3     "": [  
4         "Passwords must have at least one non letter or digit character. Passwords must have at least one digit ('0'-'9')."  
5     ]  
6 }
```

## Register(password do not match):

The Postman interface shows a failed POST request to `https://localhost:44320/api/Account/Register`. The response status is 400 Bad Request, indicating the password and confirmation password do not match. The JSON response body is:

```
1 "Message": "The request is invalid.",  
2 "ModelState": {  
3     "model.ConfirmPassword": [  
4         "The password and confirmation password do not match."  
5     ]  
6 }
```

## Login(Successful):

The screenshot shows the Postman application interface. In the center, there is a "POST" request to "https://localhost:44320/Token". The "Body" tab is selected, showing the JSON payload: {"username": "jilin12@gmail.com", "password": "Ntuchub0310!"} with the "x-www-form-urlencoded" option checked. The response status is 200 OK, and the response body is a JSON token object:

```
2 "access_token": "Lm5FkRn50mrWh-28XnjyZ8pU3fNv5u1w0hQ9nQ7_Pw5UgIppBqY39x2F0xF4AZF-B0vsutb3aj4Y3FxY5wG6_fQeyuwmXib-wZPfVa_pZt0ingK1Hyz4nKHG7sm-N1zj0WhdwyVm8PfukGh0v45lHxy-S7Bw9BAHq9qRrB8M5lx4X-17zJwr-8ukpgIzrvncpFYrCe1RxlFsHyotDK_gukqiMrgb59c2Scn5A8Cg-XV8cefUDKfrRE9mqbfhNL_VBRGhbME0g72RpXbnet5C1-C-KaJroL39g1gKoLSsydQzD89R-Sbw_Vw_j7b9uauqszp557IK030NPMPad4FccExnc07K7YXH0e0-7kIzWSRm5iisiEzASvt8YE8EgI-JRZUu5719LBbRK7ycVN7zuPUmaQahtisbKGnfDXI_Q6U20YzirRnan2VyyiiyIzPX-LQK16_U0m06_R9sXIahMuor7P696xE", 3 "token_type": "bearer", 4 "expires_in": 1209599, 5 "userName": "jilin12@gmail.com", 6 "issued": "Sun, 13 Dec 2020 12:45:06 GMT", 7 "expires": "Sun, 27 Dec 2020 12:45:06 GMT"
```

## Login Unsuccessful:

The screenshot shows the Postman application interface. In the center, there is a "GET" request to "https://localhost:44320/Token". The "Body" tab is selected, showing the JSON payload: {"username": "alfredwdwdwdwdTan@gmail.com", "password": "Ntuchub0310!"} with the "x-www-form-urlencoded" option checked. The response status is 400 Bad Request, and the response body is a JSON error object:

```
1 { 2 "error": "invalid_grant", 3 "error_description": "The user name or password is incorrect." 4 }
```

## GetValues(Successful):

The screenshot shows the Postman application interface. In the center, there is an 'Untitled Request' panel for a GET request to `https://localhost:44320/api/values/1`. The 'Authorization' tab is selected, showing a dropdown set to 'Bearer Token'. A note says: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables.' Below it, a 'Token' field contains a long string of characters. The 'Body' tab shows the response body: `1 "Hello, jjlin12@gmail.com."`. At the bottom right of the panel, status information is displayed: Status: 200 OK, Time: 17 ms, Size: 375 B. The bottom of the screen shows the Windows taskbar with various pinned icons.

## Unsuccessful:

The screenshot shows the Postman application interface. In the center, there is an 'Untitled Request' panel for a GET request to `https://localhost:44320/api/values`. The 'Authorization' tab is selected, showing a dropdown set to 'Bearer Token'. A note says: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables.' Below it, a 'Token' field contains a long string of characters. The 'Body' tab shows the response body: `1 { 2 [ "Message": "Authorization has been denied for this request." ] } 3 ]`. At the bottom right of the panel, status information is displayed: Status: 401 Unauthorized, Time: 505 ms, Size: 457 B. A message at the top left says: 'An update has been downloaded for Postman. Restart now to install the update.' The bottom of the screen shows the Windows taskbar with various pinned icons.

## AccountLogout:

The screenshot shows the Postman application interface. At the top, there's a toolbar with 'File', 'Edit', 'View', and 'Help' options, along with a 'Restart' button. Below the toolbar, a banner indicates an update is available. The main workspace is titled 'Untitled Request'. A POST request is being made to 'https://localhost:44320/api/Account/Logout'. In the 'Authorization' tab of the request settings, a 'Bearer Token' is selected, and a token value is shown in the input field. The response status is 200 OK with a time of 44 ms and a size of 401 B. The bottom navigation bar includes 'Bootcamp', 'Build', 'Browse', and other icons.

## Task4

### Web-API Documentation:

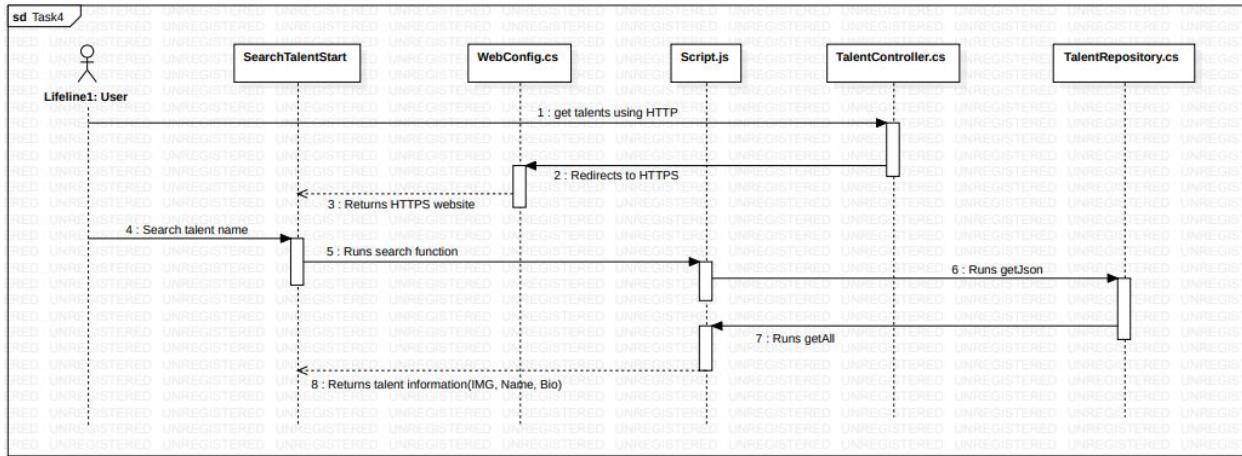
#### 1. Retrieve all talents

METHOD	GET
ROUTE	/api/talents

#### 2. Retrieve talent by id

METHOD	GET
ROUTE	/api/values/{id:INT}

## Sequence Diagram:



## Screenshots:

### Get Api/Talents:

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** http://localhost:54602/api/talents
- Status:** 200 OK
- Time:** 283 ms
- Size:** 2.43 KB
- Body (Pretty):**

```

1 {
2   "Id": 0,
3   "Name": "Barot Bellingham",
4   "ShortName": "Barot_Bellingham",
5   "Reknown": "Royal Academy of Painting and Sculpture",
6   "Bio": "Barot has just finished his final year at The Royal Academy of Painting and Sculpture, where he excelled in glass etching
      paintings and portraiture. Hailed as one of the most diverse artists of his generation, Barot is equally as skilled with
      watercolors as he is with oils, and is just as well-balanced in different subject areas. Barot's collection entitled \"The
      Un-collection\" will adorn the walls of Gilbert Hall, depicting his range of skills and sensibilities - all of them, uniquely
      Barot, yet undeniably different"
7 },
8 },
9 {
10   "Id": 1,
11   "Name": "Jonathan G. Ferrar II",
12   "ShortName": "Jonathan_Ferran"
13 }

```

## Get Api/Talents/{id}:

Untitled Request

GET http://localhost:54602/api/talents/1

Status: 200 OK Time: 41 ms Size: 871 B Save Response

```

1 {
2   "Id": 1,
3   "Name": "Jonathan G. Ferrar II",
4   "ShortName": "Jonathan_Ferrar",
5   "Rekrown": "Artist to Watch in 2012",
6   "Bio": "The Artist to Watch in 2012 by the London Review, Johnathan has already sold one of the highest priced-commissions paid to an art student, ever on record. The piece, entitled Gratitude Resort, a work in oil and mixed media, was sold for $750,000 and Jonathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations"
7 }

```

## Browser HTTPS:

localhost:44339/api/talents

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<ArrayOfTalent xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/SearchTalent.Models">
  <Talent>
    <Bio>Barot has just finished his final year at The Royal Academy of Painting and Sculpture, where he excelled in glass etching paintings and portraiture. Hailed as one of the most diverse artists of his generation, Barot is equally as skilled with watercolors as he is with oils, and is just as well-balanced in different subject areas. Barot's collection entitled "The Un-Collection" will adorn the walls of Gilbert Hall, depicting his range of skills and sensibilities - all of them, uniquely Barot, yet undeniably different.</Bio>
    <Id>0</Id>
    <Name>Barot Bellingham</Name>
    <Rekrown>Royal Academy of Painting and Sculpture</Rekrown>
    <ShortName>Barot_Bellingham</ShortName>
  </Talent>
  <Talent>
    <Bio>The Artist to Watch in 2012 by the London Review, Johnathan has already sold one of the highest priced-commissions paid to an art student, ever on record. The piece, entitled Gratitude Resort, a work in oil and mixed media, was sold for $750,000 and Jonathan donated all the proceeds to Art for Peace, an organization that provides college art scholarships for creative children in developing nations</Bio>
    <Id>1</Id>
    <Name>Jonathan G. Ferrar II</Name>
    <Rekrown>Artist to Watch in 2012</Rekrown>
    <ShortName>Jonathan_Ferrar</ShortName>
  </Talent>
  <Talent>
    <Bio>Hillary is a sophomore art sculpture student at New York University, and has already won all the major international prizes for new sculptors, including the Divinity Circle, the International Sculptor's Medal, and the Academy of Paris Award. Hillary's CAC exhibit features 25 abstract watercolor paintings that contain only water images including waves, deep sea, and river.</Bio>
    <Id>2</Id>
    <Name>Hillary Hewitt Goldwynn-Post</Name>
  </Talent>
</ArrayOfTalent>

```

Request URL: http://localhost:54602/api/talents  
 Request Method: GET  
 Status Code: 302 Found  
 Remote Address: [::1]:54602  
 Referrer Policy: strict-origin-when-cross-origin

Response Headers

Cache-Control: no-cache  
 Content-Length: 32  
 Content-Type: text/plain; charset=utf-8  
 Date: Sun, 03 Jan 2021 13:44:43 GMT  
 Expires: -1  
 Location: https://localhost:44339/api/talents  
 Pragma: no-cache  
 Server: Microsoft-IIS/10.0  
 X-AspNet-Version: 4.0.30319  
 X-Powered-By: ASP.NET  
 X-SourceFiles: =2017-08-20T00:27:31Z\bin\IIS\10.0\wwwroot\wwwroot\index.html  
 X-UA-Compatible: IE=Edge,chrome=1

Request Headers

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
 Accept-Encoding: gzip, deflate  
 Accept-Language: en-US,en;q=0.9  
 Host: localhost:44339  
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36

## LiveSearch:

The screenshot shows a web browser window with the title 'Live Search'. The URL in the address bar is 'localhost:44339/searchTalentStart/index.html'. The main content area has a header 'LIVE SEARCH' with a sub-instruction 'Enter the name or info about a speaker' and a search input field containing 'name or info'. Below this, there are four search results, each with a small profile picture and a name. The first result is 'Barot Bellingham', described as having just finished his final year at The Royal Academy of Painting and Sculpture, excelling in glass etching, portraiture, and watercolors. The second result is 'Jonathan G. Ferrar II', described as the Artist to Watch in 2012 by the London Review, having sold one of the highest priced commissions to an art student ever. The third result is 'Hillary Hewitt Goldwynn-Post', described as a sophomore art sculpture student at New York University who has won major international prizes. The fourth result is 'Hassum Harrod', described as having been sponsored on scholarship for his entire undergraduate career at The Art College in New Delhi.

## Task5

### Web-API Documentation:

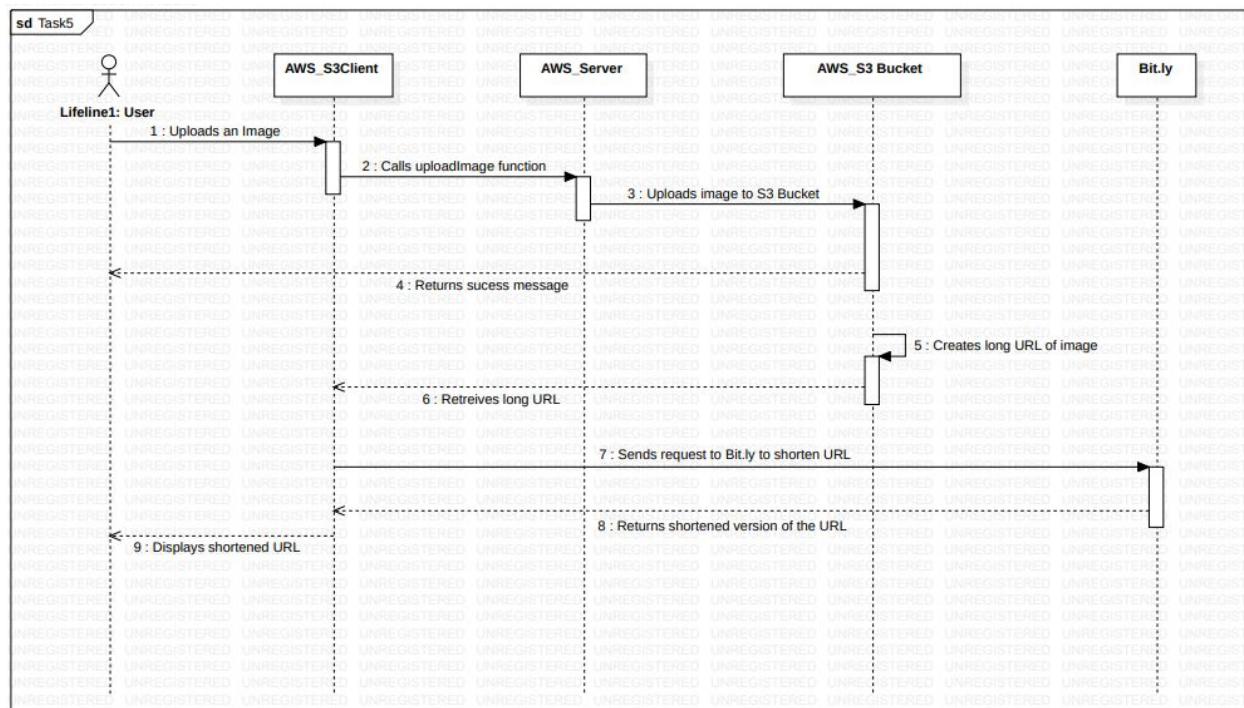
1. Upload image to S3

METHOD	POST
--------	------

2. Convert long URL to Bit.ly URL

METHOD	POST
ROUTE	<a href="https://api-ssl.bitly.com/v4/shorten">https://api-ssl.bitly.com/v4/shorten</a>

## Sequence Diagram:



## Screenshots:

### AWS S3 Bucket:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Barot_Bellingham_tn.jpg	jpg	December 27, 2020, 01:56:51 (UTC+08:00)	12.5 KB	Standard
<input type="checkbox"/>	booked.PNG	PNG	January 3, 2021, 12:12:13 (UTC+08:00)	106.9 KB	Standard
<input type="checkbox"/>	Bowlofrice.PNG	PNG	January 3, 2021, 11:48:09 (UTC+08:00)	59.8 KB	Standard
<input type="checkbox"/>	Burger.PNG	PNG	December 31, 2020, 00:00:05 (UTC+08:00)	17.0 KB	Standard
<input type="checkbox"/>	connect.PNG	PNG	January 3, 2021, 11:50:50 (UTC+08:00)	624.0 KB	Standard
<input type="checkbox"/>	Constance_Smith_tn.jpg	jpg	December 28, 2020, 23:11:02 (UTC+08:00)	12.5 KB	Standard
<input type="checkbox"/>	drs.JPG	JPG	January 3, 2021, 11:53:40 (UTC+08:00)	50.5 KB	Standard

## Task6

Web-API Documentation:

1. One time purchase

METHOD	POST
ROUTE	/purchase
PARAMETERS REQUIRED	ID: Integer, Items: Item Object

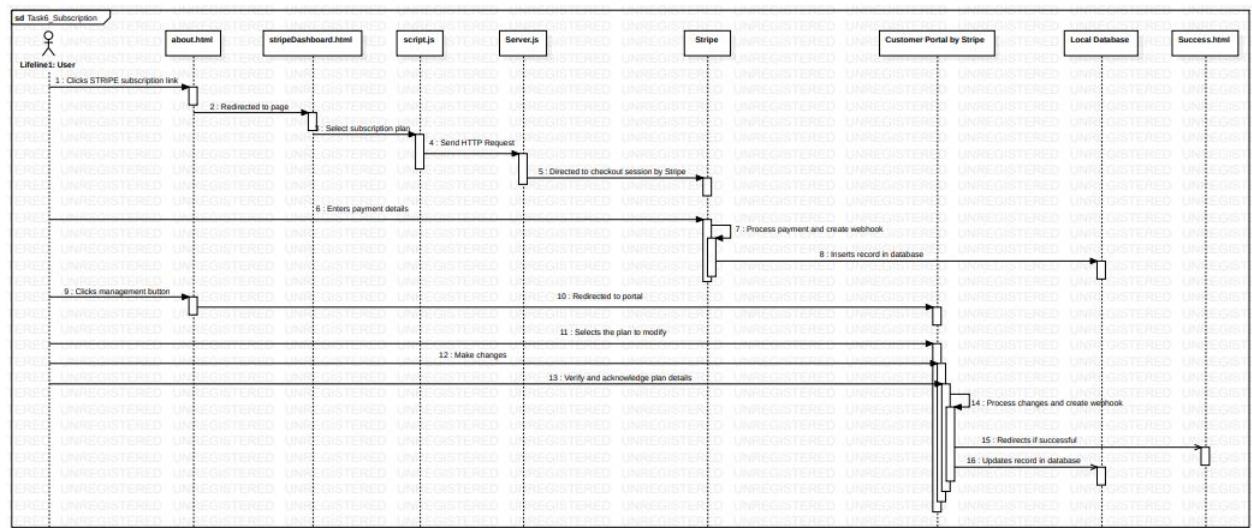
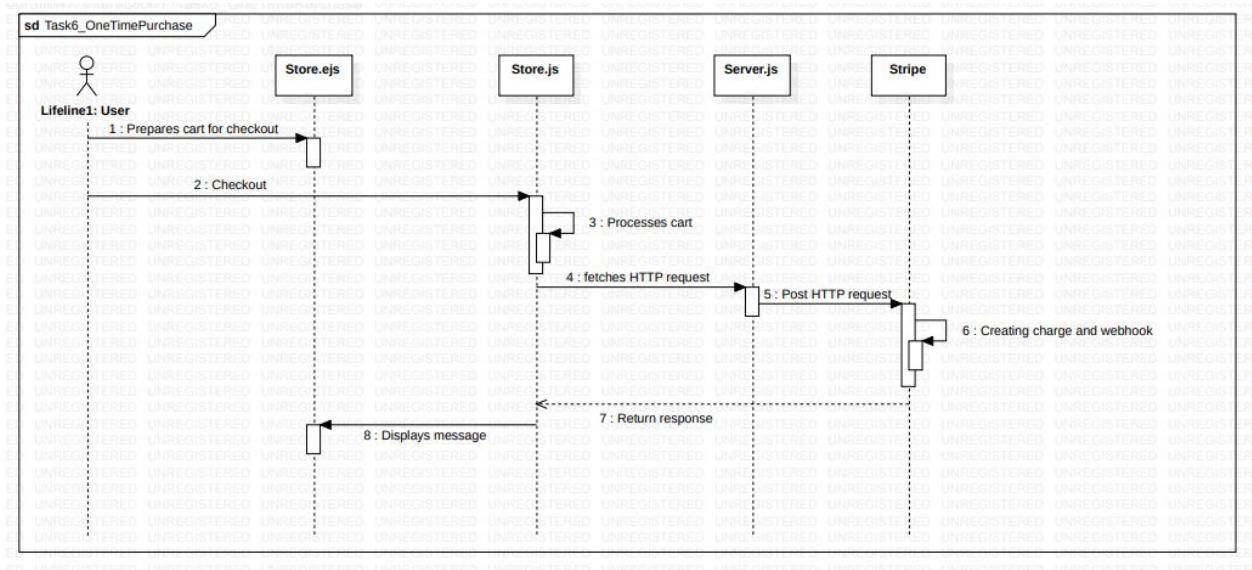
2. Create checkout session

METHOD	POST
ROUTE	/create-checkout-session
PARAMETERS REQUIRED	priceld: String

3. Webhook

METHOD	POST
ROUTE	/webhook
PARAMETERS REQUIRED	event: String

## Sequence Diagram:



# Task7

Sequence Diagram:

