

# Urban Computing Week 3 Assignment

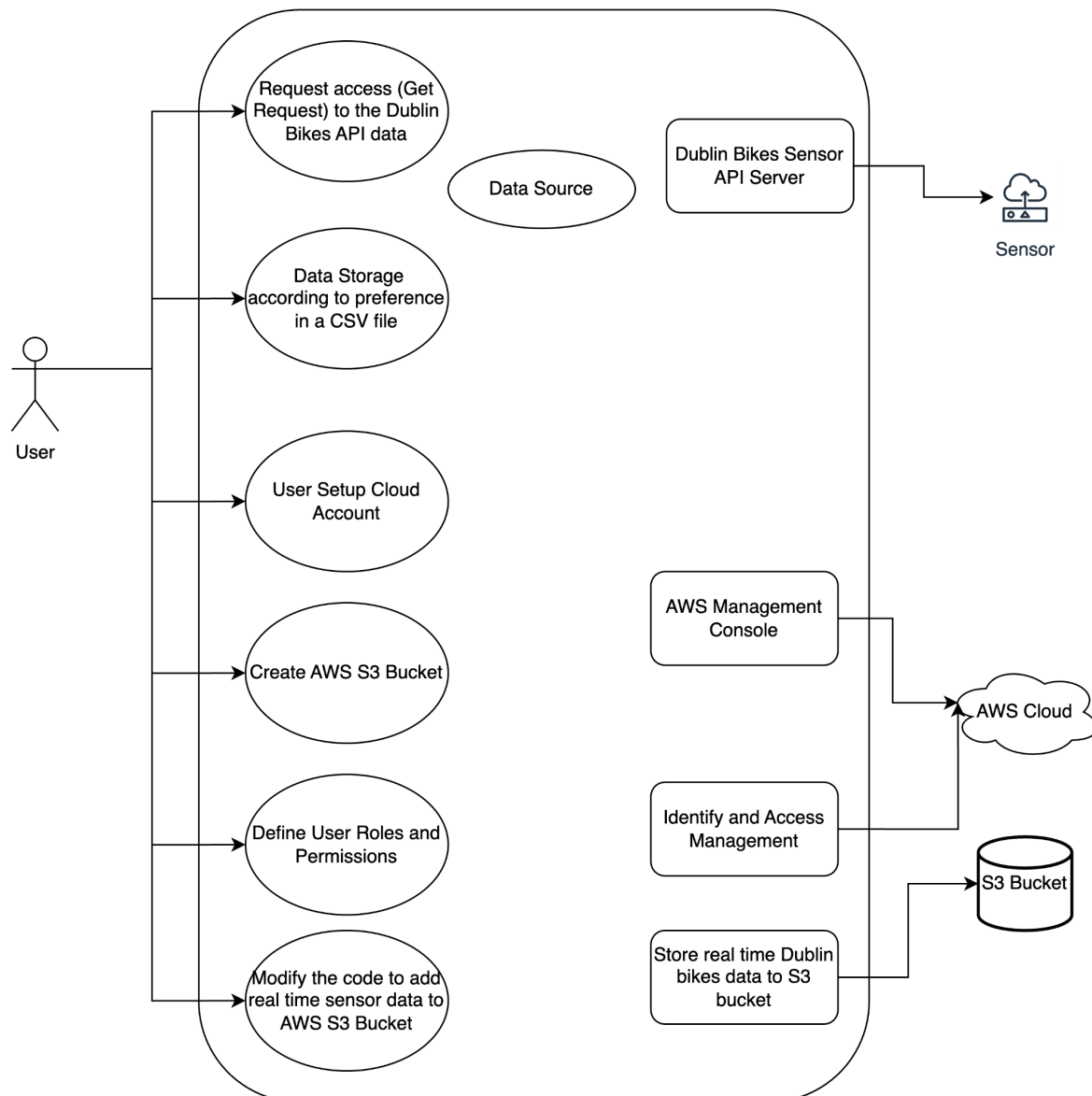
## Sensor Data Gathering

By: Sanat Paranjape, Student Id:23347997

Course Code:CS7NS4 , Stream: MSc CS Intelligent Systems

### Task 1 - Gathering your data

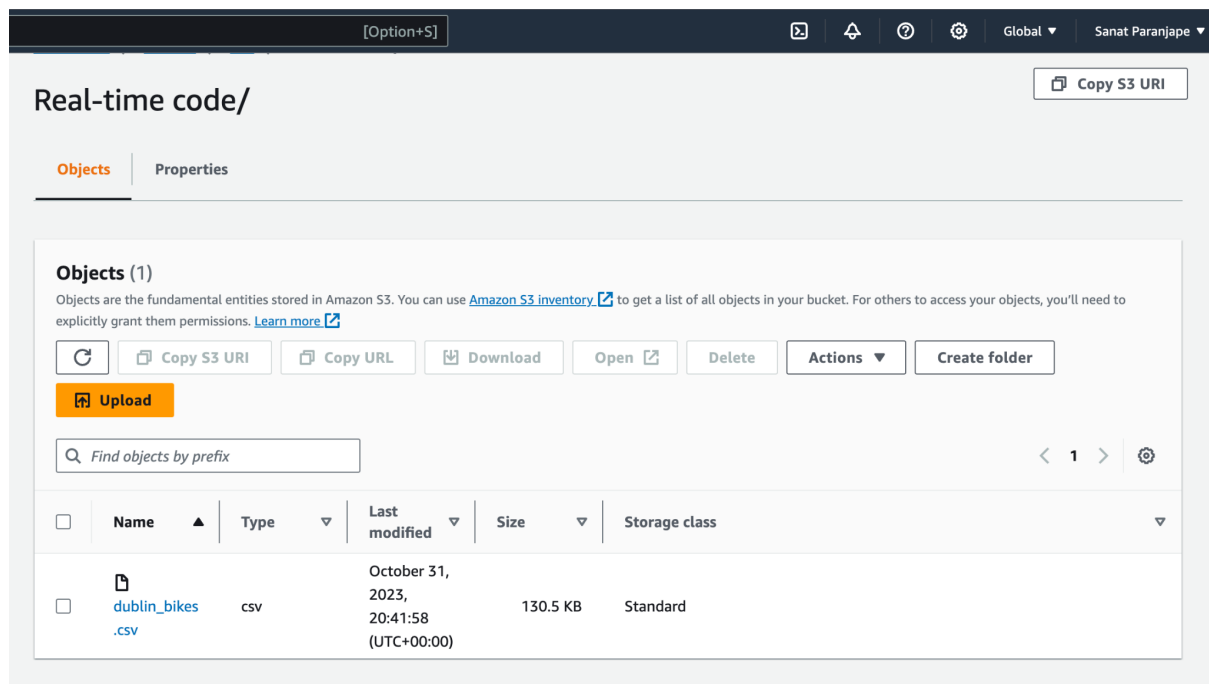
#### Technical Diagram :



This is the technical diagram to illustrate how my own data is submitted to the chosen cloud service i.e Amazon Web Service in real time. This diagram provides an overview of the data flow from the data source to the cloud.

## Uploading Own Data :

- **Source of Data** : In the previous assignment, I have successfully implemented fetching of data from Dublin Bikes API and saving the data in a csv format.
- **Cloud Setup** : To upload this dataset to a cloud service there are few prerequisite steps.
  - a. Create an AWS account to avail the cloud services. Once you're logged in, navigate to the AWS Management Console.
  - b. An S3 bucket is a container for storing your files. You can create a new bucket by navigating to 'Services' Selecting 'S3' under the storage section.
  - c. Click on the "Create bucket" button. Follow the instructions to set up the bucket.
  - d. You can directly upload the csv file from the upload option presented after creation of the bucket.
- **Integrating Cloud Service to the code** :
  - a. Integrating AWS S3 bucket with the data fetching code created in the previous assignment using Amazon SDK for python (Boto3).
  - b. Boto3 is used for managing the AWS S3 bucket and adding the fetched real time data directly to the cloud.
  - c. Using this library we can upload the csv file i.e Dublin Bikes API dataset to S3 bucket.



## Data Uploaded in Real-Time :

The data is uploaded in **real-time** to ensure that it reflects the most current information. This **real-time** functionality is also included in the source code(attached).

## Sample of data :

```
trial.py > ...
1 import requests
2 import csv
3
4 # URL of the CSV file
5 url = "https://uc3.s3.eu-west-1.amazonaws.com/Real-time+code/dublin_bikes.csv"
6
7 try:
8
9     response = requests.get(url)
10
11
12     if response.status_code == 200:
13
14         csv_data = response.text
15
16         reader = csv.reader(csv_data.splitlines())
17
18         #first 20 entries
19         for i, row in enumerate(reader):
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

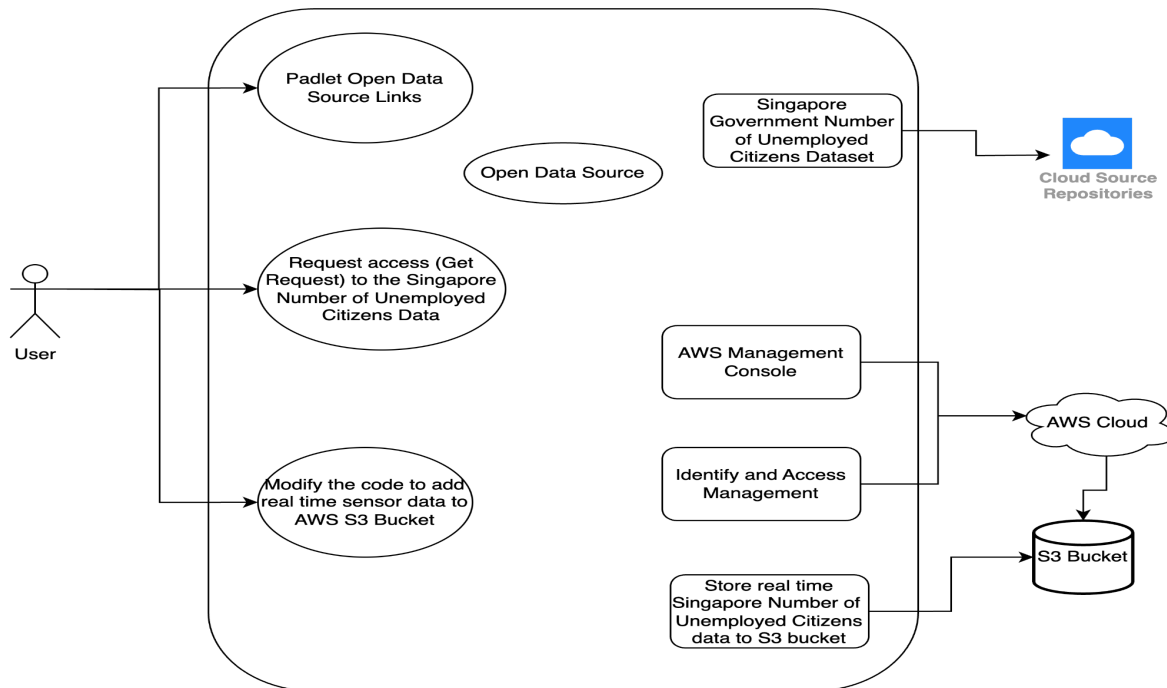
```

• sanatanparanjape@Sanats-MacBook-Air SanatParanjape_23347997_UC Assignment 2_Code % python3 trial.py
['id', 'harvest_time', 'station_id', 'available_bike_stands', 'bike_stands', 'available_bikes', 'banking', 'bonus', 'last_update', 'status', 'address', 'name', 'latitude', 'longitude']
['66682200', '2023-10-31T20:40:02', '1', '23', '31', '8', 'False', 'False', '2023-10-31T20:35:24', 'OPEN', 'Clarendon Row', 'CLARENDON ROW', '53.3409', '-6.2625']
['66682267', '2023-10-31T20:40:02', '2', '15', '20', '5', 'False', 'False', '2023-10-31T20:35:53', 'OPEN', 'Blessington Street', 'BLESSINGTON STREET', '53.3568', '-6.26814']
['66682208', '2023-10-31T20:40:02', '3', '18', '20', '2', 'False', 'False', '2023-10-31T20:37:42', 'OPEN', 'Bolton Street', 'BOLTON STREET', '53.3512', '-6.26986']
['66682228', '2023-10-31T20:40:02', '4', '19', '20', '1', 'False', 'False', '2023-10-31T20:38:26', 'OPEN', 'Greek Street', 'GREEK STREET', '53.3469', '-6.27298']
['66682223', '2023-10-31T20:40:02', '5', '4', '40', '36', 'False', 'False', '2023-10-31T20:35:50', 'OPEN', 'Charlemont Street', 'CHARLEMONT PLACE', '53.3307', '-6.26018']
['66682198', '2023-10-31T20:40:02', '6', '19', '20', '1', 'False', 'False', '2023-10-31T20:35:23', 'OPEN', 'Christchurch Place', 'CHRISTCHURCH PLACE', '53.3434', '-6.27012']
['66682231', '2023-10-31T20:40:02', '7', '28', '29', '1', 'False', 'False', '2023-10-31T20:36:10', 'OPEN', 'High Street', 'HIGH STREET', '53.3436', '-6.27507']
['66682243', '2023-10-31T20:40:02', '8', '18', '30', '12', 'False', 'False', '2023-10-31T20:29:18', 'OPEN', 'Custom House Quay', 'CUSTOM HOUSE QUAY', '53.3479', '-6.24805']
['66682218', '2023-10-31T20:40:02', '9', '22', '24', '1', 'False', 'False', '2023-10-31T20:32:15', 'OPEN', 'Exchequer Street', 'EXCHEQUER STREET', '53.343', '-6.26358']
['66682275', '2023-10-31T20:40:02', '10', '11', '16', '5', 'True', 'False', '2023-10-31T20:29:51', 'OPEN', 'Dame Street', 'DAME STREET', '53.344', '-6.2668']
['66682209', '2023-10-31T20:40:02', '11', '30', '30', '0', 'False', 'False', '2023-10-31T20:34:32', 'OPEN', 'Earlsfort Terrace', 'EARLSFORT TERRACE', '53.3343', '-6.258']
```

## • Task 2 - Gathering open data

For this task, I have integrated open data from a chosen source provided in the assignment instructions. I have used the link ([https://padlet.com/melanie\\_bouroche\\_tcd/zzxg5b9r01kf](https://padlet.com/melanie_bouroche_tcd/zzxg5b9r01kf)) to fetch open source data about Singapore and selected the (Number of Unemployed Citizens) using the link ([https://beta.data.gov.sg/datasets/d\\_832d446094a3f15f813c6755cdc0650f/view](https://beta.data.gov.sg/datasets/d_832d446094a3f15f813c6755cdc0650f/view)) provided in the padlet.

## Technical Diagram :



I have successfully incorporated open data from the open data source on the padlet to the AWS S3 bucket. This is the provided technical diagram which shows the approach to gather open data and store it on the cloud.

### Open Data on the cloud :

- **Source of Open Data** : The open data source is the link provided in the padlet link ([https://padlet.com/melanie\\_bouroche\\_tcd/zzxg5b9r01kf](https://padlet.com/melanie_bouroche_tcd/zzxg5b9r01kf)). Number of Unemployed Citizens data in Singapore has been gathered from the above link followed by ([https://beta.data.gov.sg/datasets/d\\_832d446094a3f15f813c6755cdc0650f/view](https://beta.data.gov.sg/datasets/d_832d446094a3f15f813c6755cdc0650f/view)) provided in the Team Singapore section.
- **Cloud Setup** : To upload this dataset to a cloud service there are few prerequisite steps.
  - a. Create an AWS account to avail the cloud services. Once you're logged in, navigate to the AWS Management Console.
  - b. An S3 bucket is a container for storing your files. You can create a new bucket by navigating to 'Services' Selecting 'S3' under the storage section.
  - c. Click on the "Create bucket" button. Follow the instructions to set up the bucket.
  - d. You can directly upload the csv file from the upload option presented after creation of the bucket.
- **Integrating Cloud Service to the code** :
  - a. Integrating AWS S3 bucket with the data fetching code created in the previous assignment using Amazon SDK for python (Boto3).
  - b. Boto3 is used for managing the AWS S3 bucket and adding the fetched real time data directly to the cloud.
  - c. Using this library we can upload the csv file i.e Singapore Unemployment dataset to S3 bucket.

The screenshot shows the Amazon S3 console interface. At the top, the breadcrumb navigation is 'Amazon S3 > Buckets > uc3 > Open data code/'. The main heading is 'Open data code/' with a 'Copy S3 URI' button. Below the heading are two tabs: 'Objects' (selected) and 'Properties'. The 'Objects' section shows 'Objects (1)' and a description: 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)'. Below this is a toolbar with buttons: Refresh, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar 'Find objects by prefix' is also present. Below the search bar is a table with columns: Name, Type, Last modified, Size, and Storage class. The table contains one row for 'file.csv', which is a 'csv' file, last modified on 'October 31, 2023, 20:02:14 (UTC+00:00)', with a size of '3.5 KB' and 'Standard' storage class.

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	file.csv	csv	October 31, 2023, 20:02:14 (UTC+00:00)	3.5 KB	Standard

Sample of data :

```
trial.py > ...
1 import requests
2 import csv
3
4 # URL of the CSV file
5 #url = "https://uc3.s3.eu-west-1.amazonaws.com/Real-time+code/dublin_bikes.csv"
6 url = "https://uc3.s3.eu-west-1.amazonaws.com/Open+data+code/file.csv"
7
8 try:
9
10     response = requests.get(url)
11
12
13     if response.status_code == 200:
14
15         csv_data = response.text
16
17
18         reader = csv.reader(csv_data.splitlines())
19
20         #first 20 entries
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS AZURE

```
● sanatparanjape@Sanats-MacBook-Air SanatParanjape_23347997_UC Assignment_2_Code % python3 trial.py
['period', 'residential_status', 'non-seasonally_adj_unemp', 'seasonally_adj_unemp']
['1992-03', 'citizen', '25200', '30400']
['1992-06', 'citizen', '39100', '29500']
['1992-09', 'citizen', '26900', '32300']
['1992-12', 'citizen', '27200', '26900']
['1993-03', 'citizen', '21400', '25800']
['1993-06', 'citizen', '38700', '29100']
['1993-09', 'citizen', '26000', '31500']
['1993-12', 'citizen', '29100', '28700']
['1994-03', 'citizen', '24600', '28700']
['1994-06', 'citizen', '38600', '29100']
['1994-09', 'citizen', '24500', '32000']
['1994-12', 'citizen', '33600', '32000']
['1995-03', 'citizen', '26600', '31000']
['1995-06', 'citizen', '42800', '32300']
['1995-09', 'citizen', '22000', '28700']
['1995-12', 'citizen', '33800', '32100']
['1996-03', 'citizen', '28600', '33300']
['1996-06', 'citizen', '44300', '33500']
```

I did not make use of AI tools in the preparation of this assignment.