

5/13/2018

Technical Report

Open Lake

Idriss Mohamed

x14110792

BSC (Honours) in computing

Data Analytics

2017/2018



National
College of
Ireland

Table of Contents

Executive Summary	3
1 Introduction.....	4
1.1 Purpose	4
1.2 Background.....	4
1.3 Aims.....	5
1.4 Technologies	6
1.5 Structure	8
1.6 Research:	8
1.6.1 Literature survey:.....	9
1.7 Acronyms, Definitions and Abbreviation	9
1.7.1 Acronyms	9
1.7.2 Definitions	10
1.8 System.....	10
1.8.1 Hardware and Software Requirements	10
1.8.2 Data Requirement	11
1.8.3 Functional requirements.....	11
1.8.4 Use case Diagram.....	11
1.8.5 Use Case Diagram	12
1.8.6 Use Case Diagram.....	13
1.9 Non-Functional Requirements	14
1.9.1 Security requirement.	14
1.9.2 User requirements.....	14
1.9.3 Environmental requirements	14
1.9.4 Usability requirements.....	14
1.9.5 Availability requirement	14
1.9.6 Research.....	15
1.10 Architecture Design	15
1.10.1 Logical view.....	16
1.10.2 Hardware Architecture.....	17
1.10.3 Software Architecture	17

1.11	Implementation	18
1.11.1	Python	18
1.11.2	Main python Methods interact with database	19
1.11.3	Name Node & Data Node (Hadoop).....	20
1.11.4	Hive Architecture	22
1.12	Low Level Architecture.....	24
1.13	System Design.....	24
1.14	Testing.....	25
1.15	Unit test.....	25
1.16	PostgreSQL	27
1.17	Evaluation	27
1.18	Conclusions	28
1.19	Further development or research.....	29
1.20	References	30
1.21	Appendix.....	31
1.22	Project Proposal	31
1.23	Objective	31
1.24	Technical Approach	32
1.25	The key aspect	33
1.26	Background.....	33
1.27	Special resources required	34
1.28	Evaluation	34
1.29	TECHNICAL DETAILS.....	34
1.30	Reference	36
1.31	Project Plan	37
1.32	Monthly Journal	38
1.32.1	September.....	38
1.32.2	October	39
1.32.3	November.....	40
1.32.4	December.....	41
1.32.5	January	42
1.32.6	February.....	43
1.32.7	March	44

Executive Summary

This project document gives an overview of a data lake system. The report is prepared in partial fulfilment of requirements towards obtaining BSc (Honors) in Computing- Program at the National college of Ireland.

This system will be used by several companies which need large data sets that are clean, complete and have business relevance. There are huge problems associated with growing public open source data. The report specifically illustrates the benefits of extracting data from open public source data sites and storing it in a data lake system. The data lake storage is cheap to build, and it can store a large amount of data.

There are massive open public datasets in Ireland and Europe. The system is going to store as much data as possible to data lake storage, extracted from such sites. Data transformations are required for standardization. Machine learning techniques can be employed to extract useful patterns from this data that could be used by several business units. The technical part of the report will focus on the process of building a data lake system.

The data extracted from different websites were in different formats such as .doc, .xls, .pdf, .xml,. json, px and various other extensions. A web crawler program in python will exhaustively search through all the child links in a website for data. The extraction queries were written in PostgreSQL and pgAdmin package. MS-EXCEL was used to store the list of target URLs and use that as input document for extraction program. The extracted files were stored in HDFS. Hive was used to query the Hadoop file system. The entire project was hosted in GitHub.

1 Introduction

1.1 Purpose

A data lake acts as a raw storage repository for big data analytics workloads. The data is organized by user defined patterns. It is able to store data in different formats. i.e. Both structured and unstructured formats. The nature of data could vary from Batch, Stream, Logs, IOT and other forms of large data sets. The data extracted is expected to have a large amount of data to serve various business verticals. e.g. Bank data, health data, tax data. It is possible for anyone to access the data stored in the data lakes using REST API's over HTTP. Lately the need to store unstructured data or big data for analytics is driving users to create Hadoop based data lake systems. There may also be Data lakes that could use relational data bases also. Data lakes not only act as data repositories like Data warehouses. They also enable analytics and so are owned by the teams which have DW set up with them.

The purpose of this project is to illustrate the requirements for building a data lake storage system and model a data lake for storing Public open source data in EU and Ireland.

1.2 Background

Most organizations are finding it difficult to capture, store and manage exploding quantities of data that are being produced by various business transactions. Simultaneously, the analysts are finding it difficult to find consolidated data at of this scale for exploration as well as KDD purposes. The data lake is the solution for storing persisting massive data volumes. It is characterized by support for structured, unstructured and diverse data types, different data sources and also time series and historical data. The structure and requirements of data are not known until the data is requested for analytics.

It looks like a Data warehouse serves the same purpose. However, there are some minor differences between a data lake and Data warehouse. Following points illustrate the differences.

1. Data warehouse has a data model and the incoming raw data, after pre-processing has to be structured according to the schema defined. The data lake on the other hand stores all the data in its raw form, be it structured, semi structured or unstructured.
2. Lot of pre-processing of data is required for constructing a DW. In a data lake the data is not processed until it is needed. Hence, its construction is cheap and faster.
3. Big data technologies like Hadoop have drastically reduced the cost of storing the data. HDFS is a low-cost commodity hardware based solution and hence storage costs of Data Lake are lot more cheaply than DW.
4. A Data warehouse is not agile for changes. There is a periodic downtime, when the refresh of the entire DW takes place. A data lake can be easily configured and reconfigured on the fly.
5. Data warehouses are very mature technologies. Due to the cost involved in DW construction, the security aspect is given very high importance compared to data lakes. Significant effort is going right now in the industry to make Data Lakes more secure.
6. The users of Data warehouse are not operational users. Even though data lakes are being built for all users like data explorers, the end users will always be analysts and data scientists

1.3 Aims

There is lot of open public source lying in various government websites. Many organizations are using this data as per the requirement basis. However, it would

save them lot of time and effort, if there is available, a common repository of cleaned and transformed data consolidated from different websites. The Meta data will help the users to figure out what data to choose and available data types. The motivation for taking up this project is to make such a repository available to public users and small-scale business units can benefit out of it.

Aim 1: Identify the target audience and target data large and complex enough to need a data lake system.

Aim 2: Extract data from websites (open Public Source websites from EU), and store it into HDFS as well as keeping track of the extraction process (e.g. by updating Boolean indicator(s) in a Postgres tables).

Aim 3: Construct Meta data and map the data tables.

Aim 4: The final aim is to evaluate the system for performance and user satisfaction. This is a continuous process. Once the project will go live, the user feedback will provide necessary ingredients to incorporate additional features.

1.4 Technologies

All the tools and Software's are open source. Ex: Python, PostGre2, Ubuntu OS, Hadoop, Hive etc. below is description for each technology have been use in the project.

PostgreSQL: An open source relational database system that uses and extends the SQL language. PostgreSQL was used to store the target URL links and extract the data into tables of HDFS.

pgAdmin: pgAdmin is free package which is supported on many computer platforms. Using PostgreSQL user interface, we can enter queries directly or execute them from a file. It was used to create tables and storing URL links into the table. DML queries were run on these tables using this interface.

Hadoop: Is a Map Reduce framework supported by a distributed file system HDFS (Hadoop Distributed File System). HDFS supports unstructured databases. This project requires us to store data of different file systems such as csv, html, px, xml, json and other several other data types. Majority of the open source data lake projects are using Hadoop.

Hive: Is built in on top of Hadoop to provide data summarization. Tables were created in Hive. Hive also provides querying feature.

Python: Is open source programming language. Python modules were used for creating the main engine for extracting different data from website and store it in HDFS and PostgreSQL. Python3 version was used which is required library to connect to HDFS and PostgreSQL.

Sublime text3: It is text editor that supports many programming languages and markup language. This was used for python coding and PostgreSQL queries to create tables.

GitHub: Is a data storage project versioning system. Several developers can parallel work on a soft ware project simultaneously. GitHub was used for coding in python.

Microsoft Excel: Used Microsoft excel to store all the target URL links. This file will act as an input for the scraping engine.

Linux: Is an open source operating system. Ubuntu is a Linux distribution based on Debian.

VMware: VMware, Inc. is a subsidiary of Dell Technologies that provides cloud computing and platform virtualization software and services. This virtualization platform was used to work with Hadoop.

1.5 Structure

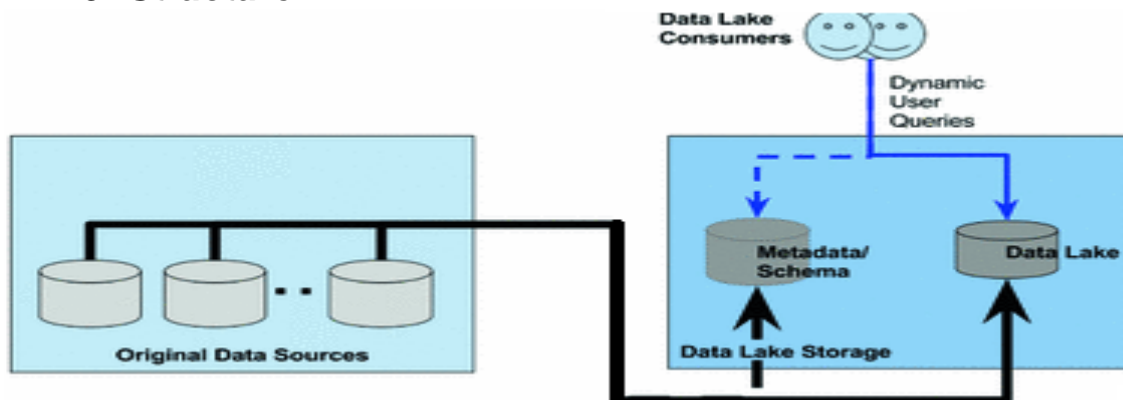


Figure 1: Knowledge discovery in in Data lake system

KDD: Knowledge discovery in the database is synonymous to data warehouse and data mining. Construction of Data Lake partially demonstrates the KDD process as shown in the figure above.

Data Selection: The data selection stage requires us to study the sources of appropriate data, understand the data usage policies and final usability.

Target Data: The project used free and public open source data from government websites of Ireland and U.K. Hence all the issues of data selection have been addressed

1.6 Research:

The project requires the designers to do a wide research on existing data lake systems. Also understand the technologies to be used such that it's cheap to build it. Research was done by going through several websites such as www.datasciencecentral.com. Also, we had to make sure that a similar project did not exist already.

Azure (proprietary s/w) system architecture was studied to understand the various architectures available for data stores. We understood the nuances and expectations of application users and also the security measures that they expect. Looking at the technologies used in existing systems, we were able to choose corresponding open source tools to develop the system. There were also

advanced features like visualizations and performance tuning etc, which we could not implement due to constraints of time and technical know-how.

Another important document that helped is the “SAS Best practices Report on Data Lakes” by Dr Philip Russom. This provide a lot of insights into practical aspects of data lakes, their utilization reports, practical use cases etc.

1.6.1 Literature survey:

Referring to the white paper on Data Lakes - Best Practices report published in 2017, We identified the need for a data lake system and its usability is as high as 85% among enterprises. The benefits outperform the shortcomings. While a Data warehouse is extremely expensive to build and maintain, the Data Lake is quite cheap and quick to build. Already there are enterprise Data Lakes existing on Azure platform as well as Hadoop. For additional information refer to the white paper.

1.7 Acronyms, Definitions and Abbreviation

1.7.1 Acronyms

Acronyms	Meaning
<i>IOT</i>	<i>Internet of Things</i>
<i>REST</i>	<i>Representational state transfer</i>
<i>API</i>	<i>Application Program Interface</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>DW</i>	<i>Data Warehouse</i>
<i>URL</i>	<i>Universal Resource Locator</i>
<i>HDFS</i>	<i>Hadoop Distributed File System</i>
KDD	Knowledge Discovery from databases

1.7.2 Definitions

Knowledge discovery in the databases (KDD): It is the process of finding useful knowledge by processing the data bases as per the business requirement.

Data Lake: A data lake is a collection huge amount of structured and unstructured data organized by user-designed patterns.

Data Warehouse: A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data that could be used as decision support system.

Open Public source data: Transactional data or business data that is available in public domain for everyone use and is also free. This data could be used mainly to get knowledge and insights into the domain.

1.8 System

1.8.1 Hardware and Software Requirements

Technology	Version
<i>PostgreSQL</i>	<i>9.5.12</i>
<i>Linux</i>	<i>16.04.1</i>
<i>Ubuntu</i>	<i>16.04.1 LTS (Long Term Support)</i> <i>Memory 2.0 GiB</i> <i>OS type 64-bit</i>
<i>Laptop</i>	<i>Windows 10</i> <i>RAM 8.00 GB. 1 TB HDD storage</i> <i>System Type 64-bit Operating System.</i>
<i>VMware</i>	<i>VMware Workstation 14.1.1</i> <i>Developer VMware</i>
<i>Hadoop</i>	<i>Hadoop 2.7.2</i>
<i>Hive</i>	<i>Hive 2.3.3</i>
<i>Sublime Text</i>	<i>Sublime Text 3</i>

1.8.2 Data Requirement

Data exploration is the first step to construct a data lake. Depending on the category of the data lake, the expected data sets need to be fetched. We are using free data published by government websites in UK and Ireland to construct the system.

There was an elaborate exercise to segregate useful data from rest of the data sources. Data exploration also needed us to pick up the necessary attributes and establishing the meta data to describe the data sets. The data file types were varied in nature. Ex: json, .xls, .xml, .doc, .pdf and various other types.

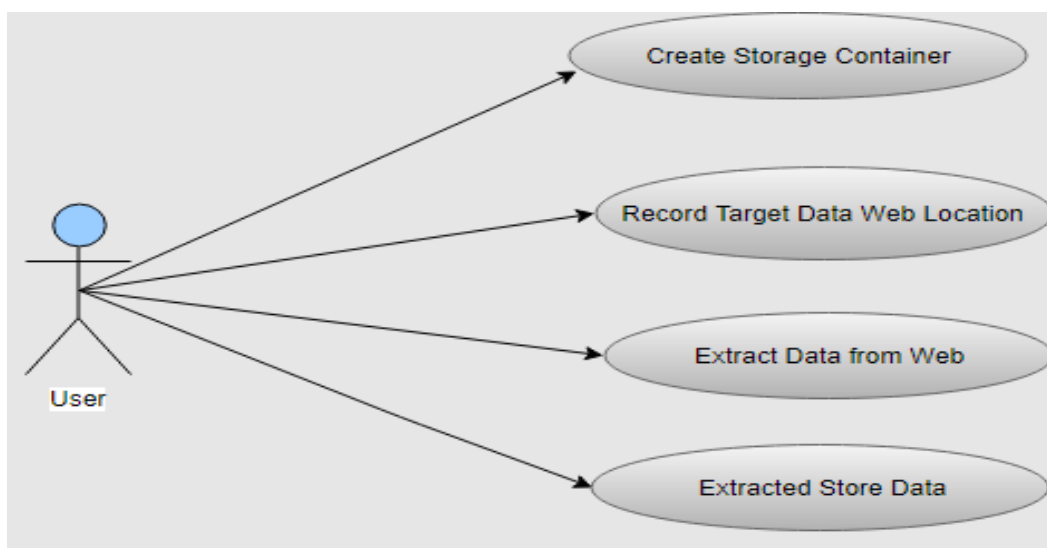
1.8.3 Functional requirements

The functional requirements are characterized by the KDD process. The requirements include the following

1. Users should be able to explore the data in the lake.
2. Understand the meta data to be able to define their use cases.
3. Users should be able to understand the data types and data queries that may be fired on the data lake to obtain the required data for analysis.

1.8.4 Use case Diagram

The following use case diagram provides an overview of the functional requirements



Creating Storage container would be considered a level 1 priority and be the first thing to do in the project. Create Storage is very importing required in the project.

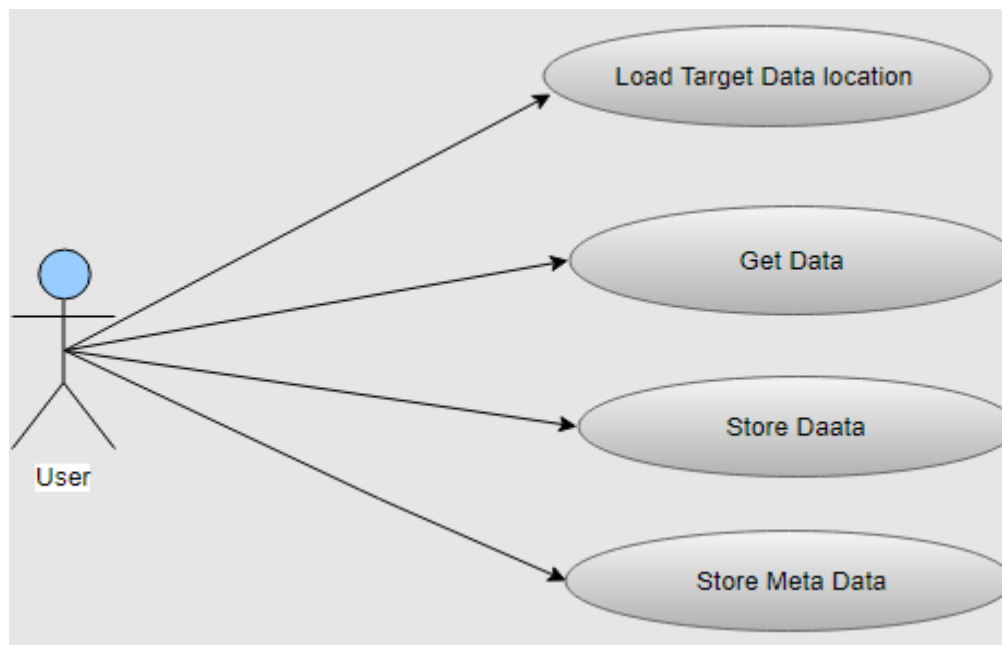
Use Case

The administrator accesses the application in order to create a Storage which will save all target data in order to allow the administrator loading into the Storage

Scope

The scope of this use case is to create a Storage using a database application in order to have a location from which I can access all URL and load Storage.

1.8.5 Use Case Diagram



Flow Description

Precondition

Dataset ready for processing.

Activation

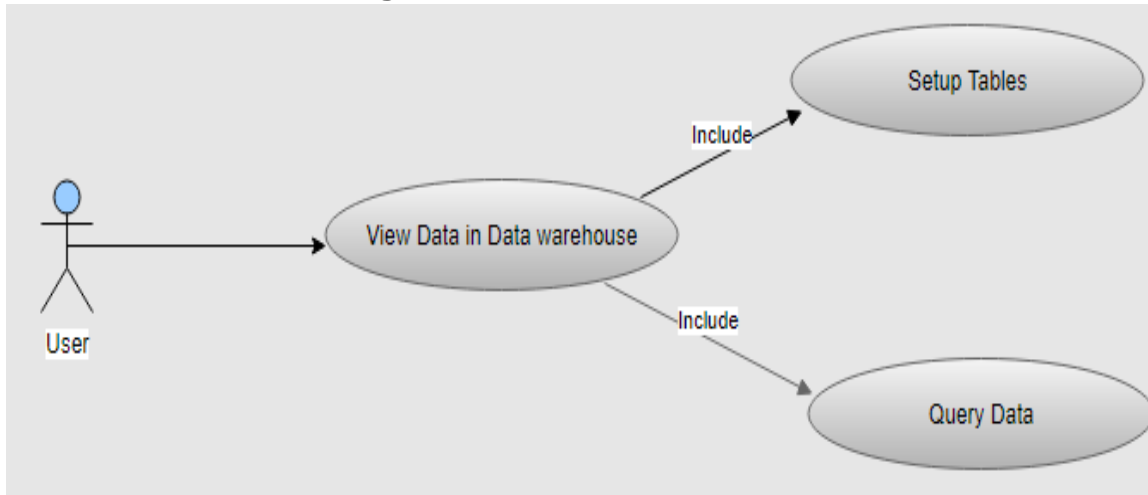
This use case starts when an <Admin>Access the target data.

Main flow

1. The <Admin> Load target data location.

2. The <Admin> Get the target data.
3. The <Admin> Store the target data.
4. The <Admin> store metadata.

1.8.6 Use Case Diagram



Flow Description

Precondition

The database must be accessible if the Storage is to be created by the administrator

Flow Description

Precondition

The database must be accessible if the Storage is to be created by the

Main flow

1. The <Admin> view data in data warehouse.
2. The <Admin> create tables.
3. The <Admin> imports a dataset from Storage using Programming.

1.9 Non-Functional Requirements

1.9.1 Security requirement.

No security required at this stage

1.9.2 User requirements

Apart from that identify the hardware and software requirements that users would need to have access to and fire queries to the data lake.

This project is built with open source tools and REST API's. The data is also public open source data. Hence the users can directly access the data lake system through the HTTP request provided.

1.9.3 Environmental requirements

There are no particular environmental factors to be considered due to the scale of the system. If the data lake size is enormous, then a cloud based data lakes could be deployed.

1.9.4 Usability requirements

No particular usability requirements have been defined, since the project is just a working prototype. In general usability requirements may include security requirements, performance requirements (efficiency of a query) etc.

1.9.5 Availability requirement

The old data will be remaining in the storage system. The system will continue to be fed from any new updated data from the same website or any other open source data bases. If the data lake is made online the server must be powerful enough to handle data requests by data scientists from the data lake.

1.9.6 Research

Following research was conducted in each stage of project development, as the time line progressed.

1. Know the data sources: Identify the data first and schema later. Keep discovering for unknown sources
2. Study the data sets: Analyse the data samples to know their characteristics and attributes. Correlate the properties with functional knowledge. Discard redundant data obtained from multiple sources.
3. Improve the relationships between data sets: Identifying the business type attributes in the individual datasets is the key to establishing the relationship between them.
4. Measure and Monitor: As the data lake is being populated, keep checking for new arrivals / removals and edits.
5. Do not over build: Data Lake is meant to be flexible and easy for analytics and there is no need to focus much on data quantum. The evolution of Data Lake happens gradually, both structure and storage.

1.10 Architecture Design

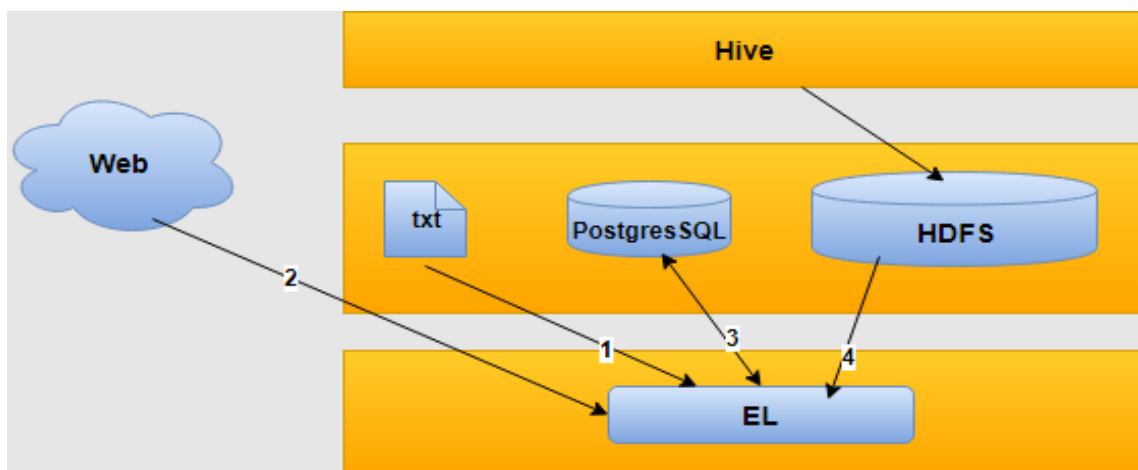


Figure 2: High end System Architecture

There can be relational Data Lakes as well as Hadoop-based Data Lakes. We are interested to create a read-only functionality; hence it's preferred to have a

Hadoop-based architecture. This also resolves the restrictions on cost and scalability. More than 60% of Data Lakes are constructed purely on Hadoop systems. Hadoop also has inbuilt support for all the data types and also parallel processing.

Apache Hive is the software built on top of Hadoop architecture. It facilitates data summarization, query and analysis. Hive provides an interface similar to SQL, to query data stored in file systems that are integrated with Hadoop.

Following are the sample URL from web, that contained the data files required for Data Lake.

Id	URL	Data Type	Web Name
1	https://api.oireachtas.ie/v1/swagger.json	json	data.gov.ie
2	http://data.europa.eu/euodp/repository/ec/dg-grow/mapps/	Csv	europ.eu
3	http://www.cso.ie/StatbankServices/StatbankServices.html	Html	cso.ie
4	http://www.cso.ie/StatbankServices/StatbankServices.xml	xml	Cso.ie

Steps to access the web links

1. Read URL from Text File using python Code.
2. Access the URL from web using python code.
3. Store URL in PostgreSQL tables.
4. Load data into Hadoop File system.
5. The code will create a file for each dataset in Hadoop file system
6. Create table hive to store data to be ready for analysis.

1.10.1 Logical view

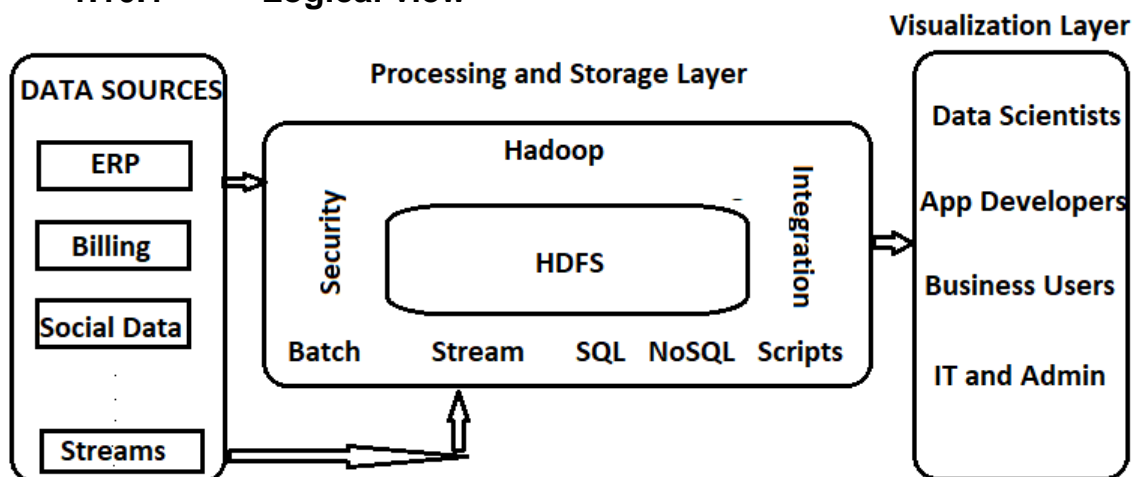


Figure 3: Data Lake – Logical View

The above logical view specifies the functional requirements of the Data Lake system. The main artefact of the logical view is the design model. The design model gives a concrete description of the functional behaviour of the system.

As seen in Figure 3, there are 3 layers of the system. The data source layer consists of various kinds of data sources. In our case, we just used web data, where the data files can be downloaded through URL links.

The middle layer is made up of Data model and querying facilities. In our case we used HDFS to store the data and Hive to write queries for data base creation as well as retrieval.

The Third layer is integrating the Data Lake to the End user interface. The User interface will contain various facilities to visualize and extract the required data. Currently the project is restricted at this phase.

1.10.2 Hardware Architecture

Laptop and Desktop machines were used to complete this project. The systems had 8 GB RAM and a terabyte of Hard disk space to simulate a data lake and HDFS. High speed Wi-Fi and cable lab were used for data retrieval.

1.10.3 Software Architecture

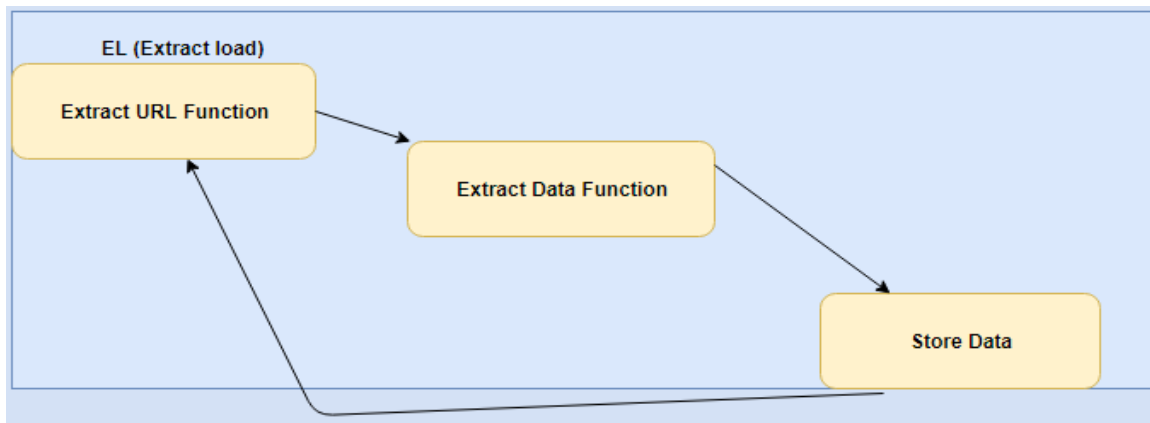


Figure 4 Software Architecture

The figure 4 show how the software architecture for the project. The process started from Extract URL function which is extracts the URL then past it to other

function which will extract the actual data from web then load it into the Hadoop file system.

1.11 Implementation

The implementation was done view technologies Hadoop file system, PostgreSQL, Python programming language and Hive queries. There are more than 10 data types stored in HDFS which were extracted from more than 8 different public open source data sites from EU.

1.11.1 Python

There is extensible library function in python to open the web URLs

urllib.request import urlopen

Psycopg2 is a fairly mature driver for interacting with PostgreSQL from the Python scripting language.

import psycopg2

Following are the basic steps using python programming.

The code 'read URL link' from PostgreSQL loads the data into HDFS using webHDFS library.

The code 'Create unique name' loads the data set with unique name using (filename = str(uuid.uuid4())+"."+data_type).

After testing this python code to read and write a text file, the program was extended to load several data types such as. json, .xml, .html, .csv etc.

Hive was used to create tables in the HDFS, where the extracted data was stored.

Meta data also was created using Hive. Meta data will help people to understand what to expect from the data tables of HDFS.

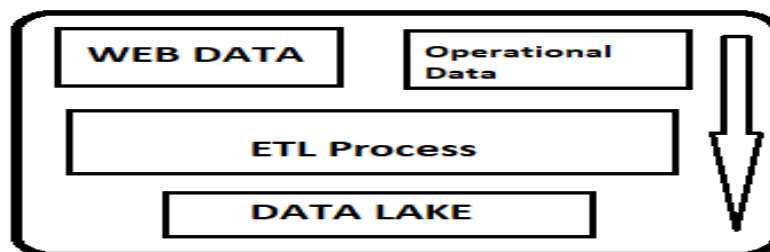


Figure 5: Data flow from open source to data lake

1.11.2 Main python Methods interact with database

Methods	Interaction
connect(info)	Are the first methods we use when the system wanted to talk to database, it takes user and password, and some time take user password and database name, and will learn to take the database name.
cursor ()	Get a cursor object ready to execute queries. It returns a new curse object that allow the executing of the queries and holds the temporary data.
execute(sql)	Execute methods belong to the cursor methods and execute a single SQL statement
fetchone()	Is also belong the cursor. Is grabbing the first row of data that isn't in the current query result.
fetchall ()	It returns list of lists of the current query result.
commit()	Is belong to the connection object. It saves any changes made to the database in preceding queries.
rollback()	It rolls back any temporary changes in preceding query, is the opposite of the commit method.
close ()	Is belong to connection object. It safely closes the connection.

1.11.3 Name Node & Data Node (Hadoop)

Hadoop architecture consists of two parts. HDFS – This is a distributed data storage system and Map Reduce – This is a distributed data processing system.

Name Node: This node only stores the metadata of Hadoop file system. It contains the directory tree of all files in the file system, and tracks the files across the cluster.

Data Node: The data is actually stored in the Data Nodes. Name node just maintains an index the data nodes as shown in the figure below. [5].

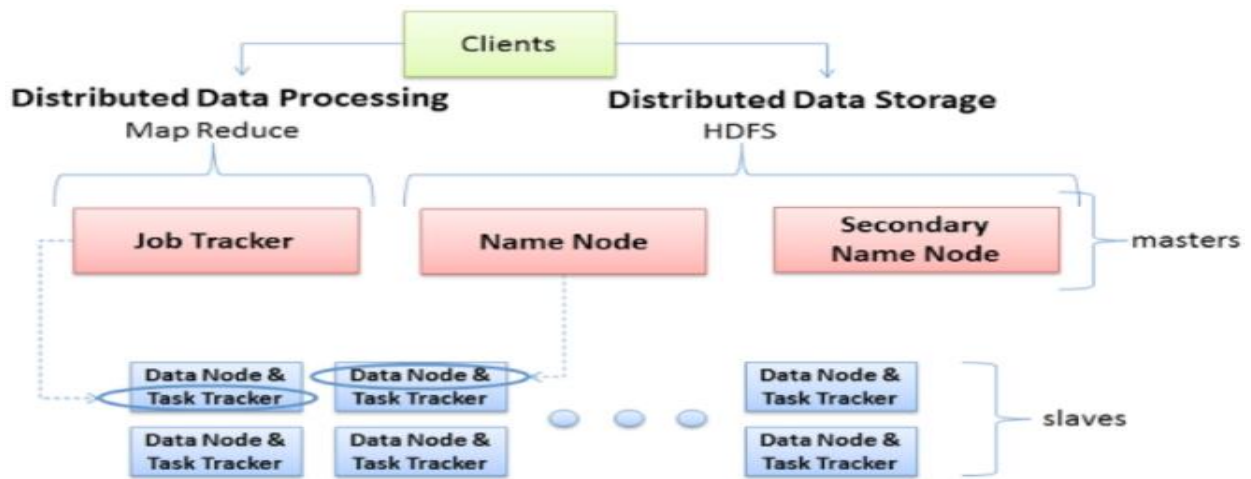


Figure 6: Name Node and Data Node

Following Figures shows a sample dataset stored in Hadoop file System. We can use localhost:50070 to lists files through browser directory. Also, there is other way to list the files in Hadoop File System and that through command by typing, `hadoop fs -ls /`

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/json Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	hduser	supergroup	5.01 KB	4/27/2018, 3:30:03 PM	1	128 MB	007a5cdf-5bf1-4eaf-8ea9-7190c232c82f.json
-rwxr-xr-x	hduser	supergroup	5.01 KB	5/2/2018, 2:07:37 PM	1	128 MB	01fa0b4b-36ea-4401-ab90-f2e4a08c08c3.json
-rwxr-xr-x	hduser	supergroup	5.01 KB	5/3/2018, 6:18:56 PM	1	128 MB	0211d276-6977-48f8-9376-e56a07212cff.json
-rwxr-xr-x	hduser	supergroup	10.63 KB	4/29/2018, 2:20:47 PM	1	128 MB	050f6b53-54e8-491b-abb5-b44967d77d96.json

Figure 7: json data in Hadoop file system

Browse Directory

/csv Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	hduser	supergroup	2.4 KB	5/2/2018, 2:11:46 PM	1	128 MB	027e3d8d-1cad-4b96-8369-7463efd3fd30.csv
-rwxr-xr-x	hduser	supergroup	3.38 KB	5/2/2018, 2:24:02 PM	1	128 MB	2721b4a8-ebcf-4e6b-925e-19b5f305c73c.csv
-rwxr-xr-x	hduser	supergroup	2.4 KB	5/2/2018, 2:17:20 PM	1	128 MB	369e99c8-4849-47f2-9e87-68a82bdf079c.csv
-rwxr-xr-x	hduser	supergroup	2.4 KB	5/2/2018, 2:15:56 PM	1	128 MB	3957b70e-19a0-4768-a2ed-16ddcc011690.csv
-rwxr-xr-x	hduser	supergroup	2.4 KB	5/2/2018, 2:07:41 PM	1	128 MB	5cff3d4d-f633-4c5f-9a5e-109803129cc6.csv
-rwxr-xr-x	hduser	supergroup	3.38 KB	5/2/2018, 2:15:56 PM	1	128 MB	5e0b3890-5762-40f5-9157-fc018fb88cbc.csv
-rwxr-xr-x	hduser	supergroup	2.4 KB	5/2/2018, 2:11:46 PM	1	128 MB	76cc6645-9c20-4293-9798-ce7eaef4abd6.csv

Figure 8: csv data in Hadoop file system

```
cd /usr/local/hadoop
sbin/start-dfs.sh
jps

6563 Jps
6181 DataNode
6346 SecondaryNameNode
6063 NameNode
```

Figure 7: Hadoop command

In figure 9 it shows the command for starting Hadoop file system, this command must be first step to do when you want to access the HDFS and it shows the Jps, DataNode, SecondaryNameNode, NameNode.

1. `hadoop fs -mkdir /csv`
2. `hadoop fs -ls`
3. `hadoop fs -ls /csv`

Figure 8: Hadoop command

The first line in the figure 10 Create csv folder in Hadoop File System second command lists all the file are in Hadoop File System, the third commend list list all file in csv folder.

1.11.4 Hive Architecture

Apache Hive supports analysis and querying of large datasets stored in HDFS. Just like SQL, Hive also provides a querying facility using a language called HiveQL. By default, Hive stores metadata in an embedded Apache Derby database. Client/server databases like MySQL can also be used. [4]

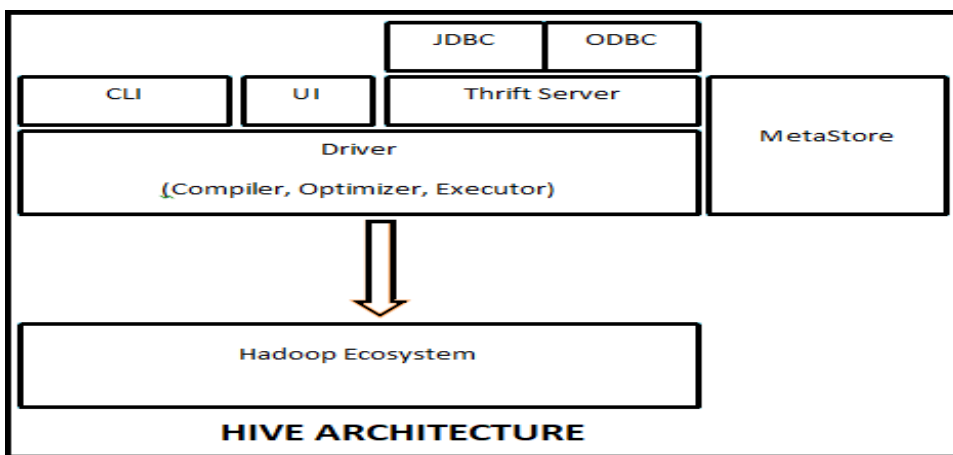


Figure 9: Hive Architecture

Metastore: The schema and meta data of each table is stored here.

Driver: It is like an execution engine for Hive queries.

Compiler: Compiles the HiveQL queries. Execution plans are derived from compilation process.

Optimizer: This optimizes the execution plan to create DAG(Directed Acyclic graph). This is needed to speed up the query execution.

Executor: It executes the tasks by interacting with job tracker to schedule the jobs.

CLI, UI, and Thrift Server: Thrift server allows external clients to interact with Hive over a network, similar to the JDBC or ODBC protocols.

Following figures show the commands to create tables using Hive.

```
hive> CREATE TABLE population(
  > CODE int,
  > ED_NAME string,
  > COUNTY string,
  > Pop_By_Place_Of_Birth_Ireland_2006 int,
  > Pop_By_Place_Of_Birth_Ireland_2011 int,
  > Pop_By_Place_Of_Birth_UK_2006 int,
  > Pop_By_Place_Of_Birth_UK_2011 int,
  > Pop_By_Place_Of_Birth_Poland_2006 int,
  > Pop_By_Place_Of_Birth_Poland_2011 int,
  > Pop_By_Place_Of_Birth_Lithuania_2006 int,
  > Pop_By_Place_Of_Birth_Lithuania_2011 int,
  > Pop_By_Place_Of_Birth_Other_EU_27_2006 int,
  > Pop_By_Place_Of_Birth_Other_EU_28_2011 int,
  > Pop_By_Place_Of_Birth_Rest_Of_World_2006 int,
  > Pop_By_Place_Of_Birth_Rest_Of_World_2011 int,
  > Pop_By_Place_Of_Birth_Total_2006 int,
  > Pop_By_Place_Of_Birth_Total_2011 int,
  > Perc_Pop_By_Place_Of_Birth_Ireland_2006 int
  > )
  > ROW FORMAT DELIMITED
  > FIELDS TERMINATED BY ','
  > ;
OK
Time taken: 0.449 seconds
hive> show tables;
OK
population
test
Time taken: 0.474 seconds, Fetched: 2 row(s)
hive> select * from population;
OK
Time taken: 5.637 seconds
hive> LOAD DATA LOCAL INPATH '/home/hdusr/test1.txt' OVERWRITE INTO TABLE population;
```

Figure 10: Create hive table and load the data into the table

Following figure shows the data in the population tables

```
hive> LOAD DATA LOCAL INPATH '/home/hdusr/test1.txt' OVERWRITE INTO TABLE population;
FAILED: SemanticException Line 1:23 Invalid path ''/home/hdusr/test1.txt': No files matching path file:/home/hdusr/test1.txt
hive> LOAD DATA LOCAL INPATH '/home/hduser/test1.txt' OVERWRITE INTO TABLE population;
Loading data to table default.population
OK
Time taken: 3.843 seconds
hive> select * from population;
OK
1      001 Carlow Urban      Carlow 3657  3236  186   183   210   242   26    33   149   133   193   243   4421
4070   NULL
2      002 Graigue Urban      Carlow 1226  1068  64    57    64    71    10    1    46    23    80    40    1490
1260   NULL
3      003 Clonmore      Carlow 480    518   35    29    0     0     02    2    2    10    8    527   559   NULL
4      004 Hacketstown Carlow 960    1010  47    41    28    13    21    14    7    9    11    1060  1083  NULL
5      005 Haroldstown Carlow 252    246   12    11    1     0     00    0    0    2    2    267   259   NULL
6      006 Kineagh      Carlow 285    303   23    21    0     0     00    0    2    1    1    309   327   NULL
7      007 Rahill       Carlow 533    605   32    36    1     5     50    3    5    11    11    585   662   NULL
8      008 Rathvilly    Carlow 734    756   52    49    1     13    02    6    10   7    7    800   837   NULL
9      009 Tiknock     Carlow 297    313   24    14    6     2     00    1    0    3    6    331   335   NULL
10     010 Williamstown Carlow 243    272   19    19    2     00    0    0    4    0    3    6    271   297
NULL
11     011 Agha        Carlow 306    361   9     9     0     0     03    5    4    2    5    322   382   NULL
12     012 Ballinacarrig Carlow 879    911   33    51    1     47    4    4    5    11   7    13    932   994
```

Figure 11: Read data from Hive table

1.12 Low Level Architecture

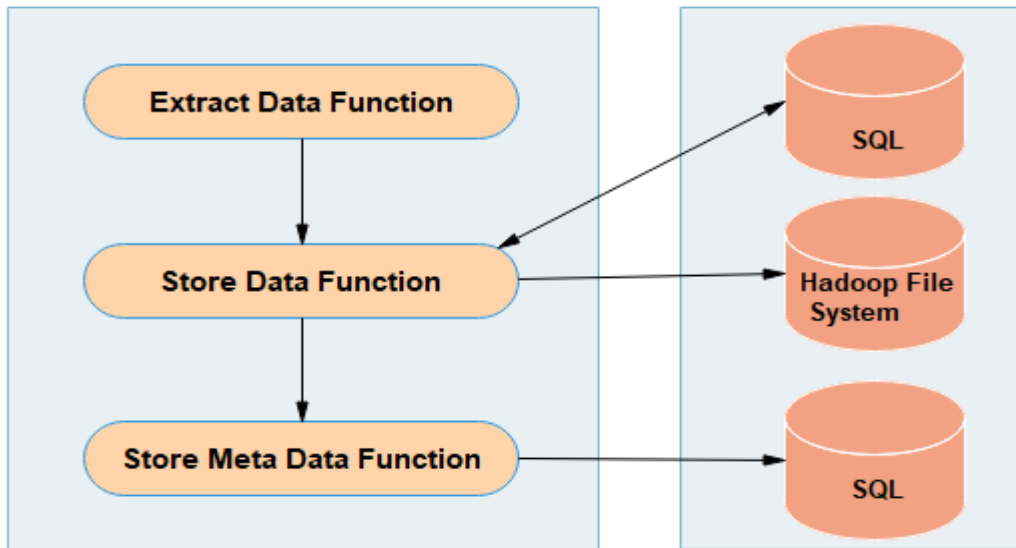


Figure 12: showing the Low Level Architecture

The -EL Agent (Extract and Load) describes the low-level architecture.

1. Extract function does the extraction from web then passes it to 'store data' function. From store function the data is sent to HDFS.
2. Store metadata function stores the metadata into PostgreSQL

1.13 System Design

The System Design within the project were achieved by adopting and applying the Knowledge Discovery in Databases (KDD) methodology. The fundamental stages when creating a Data lake system project include; data selection and data Target EL (Extract, Load).

1.14 Testing

Function	Purpose	Expected	Actual	Pass/Fail
Load same data type data into Postgres to Hadoop file system.	To extract data from web with the identical data type and store into HDFS as well update the Boolean indicator in Postgres table.	After execution the code, similar data type, data should store Hadoop File system and update the Boolean indicator from FALSE to TRUE.	As Expected	Pass
Load different data type data into Postgres to Hadoop file system.	To extract data from web with the different data type and should not store into HDFS as well update the Boolean indicator in Postgres table.	After execution the code, different data type, data should not store Hadoop File system as well should not update the Boolean indicator, it remains FALSE	As expected	Load different data type data into Postgres to Hadoop file system.

1.15 Unit test.

Testing is the very important phase in the Software Development Life Cycle, to check the working functionality of the software is working as per the expected. In this project, Unit Testing has been to check the working functionality of the 'json' data type. Which is implemented in Python code.

Passed the json data and checked the status after execution of the code, with the help of the Boolean indicator; which was True, and then data stored into Hadoop file system.

```

FOLDERS
└─ Open_Lake_Code
  └─ __pycache__
  └─ create_table_DB
  └─ import_testingpg
  └─ main.py
  └─ store_csv.py
  └─ store_html.py
  └─ store_json.py
  └─ update_record.py

main.py
23 def perform_WebCrawler(conn, links):
24     global extracting
25     baseHDFSurl="http://localhost:50070/webhdfs/v1/"
26     HDFSuser="hduser"
27     HDFSop="CREATE"
28     for id, url, data_type, website_name, IsExtracted, HDFS_File in rows:
29         # get the actual data from web
30         page_data = urlopen(url).read()
31         print(id)
32         print("url:",url)
33         print("data_type:", data_type)
34         print(website_name)
35         print("IsExtracted:", IsExtracted)
36         print("HDFS_File:",HDFS_File)
37         print(page_data)
38         #read in the length of the page
39         print("page data length = %s" % len(page_data))
40
41         if data_type == 'json':
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Importing json methods
Importing csv methods
Importing html methods
Importing update record methods
/home/hduser/.local/lib/python3.5/site-packages/psycogp2/_init_.py:144: UserWarning: The psycogp2 wheel package will be renamed from release 2.8; in order to keep installing from binary please use 'pip install psycogp2-binary' instead. For details see: <http://initd.org/psycogp/docs/install.html#binary-install-from-pypi>
***
url: http://www.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/responseinstance/EP001
data_type: json
data.gov.ie
IsExtracted: True
HDFS_File: c6faffc4-b416-44e4-bc00-89659172b312.json
None
http://localhost:50070/webhdfs/v1/json/29eaa91d-fe81-439b-a5b5-3b5bcc38a00c.json?user.name=hduser&op=CREATE
<Response [201]>
[Finished in 3.6s]

```

Figure 13: Code testing json data

In next test, done the same process for the csv data type, and verified the result; which was False, and data was not stored into Hadoop file system, which is highlighted in the screen shot.

```

install.html#binary-install-from-pypi>
"""
1
url: http://www.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/responseinstance/EP001
data_type: json
data.gov.ie
IsExtracted: True
HDFS_File: 29eaa91d-fe81-439b-a5b5-3b5bcc38a00c.json
page_data length = 5128
None
http://localhost:50070/webhdfs/v1/json/d7be6425-3992-4593-ad7b-b139fa66936b.json?user.name=hduser&op=CREATE
<Response [201]>
2
url: http://data.dublinked.ie/cgi-bin/rtpi/busstopinformation?stopid=184&format=xml
data_type: csv
data.gov.ie
IsExtracted: False
HDFS_File:
page_data length = 015
[Finished in 3.3s]
Line 34, Column 19

```

Figure 14: showing the result for the first test

Same results have been checked with PostgreSQL; For both tests the output is True for json and False for the csv, data type.

url	data_type	website_name	IsExtracted	HDFS_FileName
character varying(255)	character varying(10)	character varying(25)	boolean	character varyi
http://data.dublinked.ie/cgi-bin/rtpi/busstopinformation?stopid=184&format=xml	csv	data.gov.ie	FALSE	' '
http://www.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/responseinstance/EP001	json	data.gov.ie	TRUE	d7be6425-3992

Figure 15. Boolean indicator in PostgreSQL

1.16 PostgreSQL

Figure 16 below shows the data extracted from web and stored as text file in first stage of the project before moving the storage into Hadoop file system

Data Output		
	type	content
	character varying (255)	text
1	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
2	csv	\xefbbbf446f632c48656164696e672c5265662c646573632c496e632c50592c4f424a45435449440a5461626c65442c312c35...
3	html	<!DOCTYPE html>
4	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
5	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
6	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
7	csv	\xefbbbf446f632c48656164696e672c5265662c646573632c496e632c50592c4f424a45435449440a5461626c65442c312c35...
8	html	<!DOCTYPE html>
9	csv	\xefbbbf446f632c48656164696e672c5265662c646573632c496e632c50592c4f424a45435449440a5461626c65442c312c35...
10	html	<!DOCTYPE html>
11	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
12	html	<!DOCTYPE html>
13	json	{"dataset":{"dimension":{"Sex":{"label":"Sex","category":{"index":{"-":0,"1":1,"2":2},"label":{"-":"Both sexes","1":"Male","2":"F...
14	html	<!DOCTYPE html>

Figure 16: shows table in PostgreSQL

1.17 Evaluation

The output of the system is in the form of data sets stored in HDFS system, which can be queried using Hive queries. Sample data sets of different data structures were also shown. The system will be mainly evaluated by the end user experience. Visualization features were not implemented. Hence the evaluation is purely based on quality and quantity of data available for end user analysis for their area of interest. Since HDFS is being used, the system is scalable.

1.18 Conclusions

Data lakes have both analytical and operational purpose also. Unlike Data warehouses, the data lakes can be created using both relational databases as well as unstructured data models. It is able to support all the different types of data structures. Hence Data Lake is slowly growing as an extension to enterprise data analysis systems. There are different kinds of Data lakes such as

- Analytical
- Marketing
- Sales Performance
- Health care
- Financial fraud detection

The data lake also has advantage of early ingestion and slow and steady data inclusion. Data lakes are new technology and users are concerned about the trade off before constructing them. users consider Data lakes are opportunity and find it a burden because a data lake is hard to secure and govern. Also, it requires skills on Hadoop etc. Following can be summarized as benefits of data lakes.

- Advanced analytic facility
- Emergence of new data driven practices
- Gain business value from Big data.
- Extension of Data Warehouse.
- Supports diverse data structures.
- Quick access to data.
- Others such as scalability, Low cost H/W and S/W etc.

The leading barriers are governance, integration, lack of experience, privacy issues, and immature tech and practices. Other limitations are w.r.to people's reluctance to change. i.e. to update technology and learn new tools.

We could conclude that a data lake system will be successful if it can align with both short term goals as well as long term strategic goals.

1.19 Further development or research

The Data Lake system constructed as a part of this academic project is a naïve system and cannot be operational, until the following features are embedded.

Data integration: Identify specific business use case. Then we need to collect and integrate the entire subject oriented data. Also allow data movement, in-database processing.

Data Quality: The data quality needs to improve. There is a need to Cleanse, standardize, and enrich data in real time.

Self-service big data preparation: Construct Business users profile, cleanse, and transform data on Hadoop without writing code.

Business glossary and metadata management: Track lineage, business rules, descriptive details, and workflow for improved governance of the data assets.

Event stream processing: The system should be able to analyse real-time streams for better decisions.

Data virtualization: System should provide blended, secure views of the data without moving it.

Hadoop support: The system should access, deliver, and process data inside Hadoop across both the data management and analytics life cycle.

Visualization and advanced analytics: Data lake should be able to deliver cutting-edge visualization and analysis capabilities without requiring analytical skills.

1.20 References

1. Hadoop.apache.org. (2018). *Welcome to Apache™ Hadoop®!*. [online] Available at: <http://hadoop.apache.org/index.html> [Accessed 4 Apr. 2018].
2. A4academics.com. (2018). *Hadoop Hive Architecture, Data Modeling & Working Modes*. [online] Available at: <http://a4academics.com/tutorials/83-hadoop/836-hadoop-hive> [Accessed 3 May 2018].
3. Docs.python.org. (2018). *21.6. urllib.request — Extensible library for opening URLs — Python 3.6.5 documentation*. [online] Available at: <https://docs.python.org/3/library/urllib.request.html> [Accessed 19 Mar. 2018].
4. En.wikipedia.org. (2018). *Apache Hive*. [online] Available at: https://en.wikipedia.org/wiki/Apache_Hive#/media/File:Hive_architecture.png [Accessed 1 May 2018].
5. Harishshan.blogspot.ie. (2018). *Hadoop*. [online] Available at: <http://harishshan.blogspot.ie/2014/09/hadoop.html> [Accessed 24 Mar. 2018].
6. YouTube. (2018). *Introduction to Linux and Basic Linux Commands for Beginners*. [online] Available at: <https://www.youtube.com/watch?v=IVquJh3DXUA> [Accessed 15 Apr. 2018].
7. DATAVERSITY (2018). *Data Lake Architecture*. [online] Slideshare.net. Available at: <https://www.slideshare.net/Dataversity/data-lake-architecture> [Accessed 4 May 2018].
8. Kdnuggets.com. (2018). *Data Lake vs Data Warehouse: Key Differences*. [online] Available at: <https://www.kdnuggets.com/2015/09/data-lake-vs-data-warehouse-key-differences.html> [Accessed 29 May 2018].
9. Search Technologies. (2018). *A Data Lake Architecture with Hadoop and Open Source Search Engines*. [online] Available at: <https://www.searchtechnologies.com/blog/search-data-lake-with-big-data> [Accessed 13 Mar. 2018].
10. Search Technologies. (2018). *A Data Lake Architecture with Hadoop and Open Source Search Engines*. [online] Available at: <https://www.searchtechnologies.com/blog/search-data-lake-with-big-data> [Accessed 13 Mar. 2018].
11. Knowledgent. (2018). *How to Design a Successful Data Lake - Knowledgent*. [online] Available at: <https://knowledgent.com/whitepaper/design-successful-data-lake/> [Accessed 15 Apr. 2018].

12. SearchDataManagement. (2018). *What is Hadoop data lake? - Definition from Whatls.com.* [online] Available at: <https://searchdatamanagement.techtarget.com/definition/Hadoop-data-lake> [Accessed 2 May 2018].
13. Datasciencecentral.com. (2018). Demystifying Data Lake Architecture. [online] Available at: <https://www.datasciencecentral.com/profiles/blogs/demystifying-data-lake-architecture> [Accessed 10 Dec. 2018].

1.21 Appendix

1.22 Project Proposal

1.23 Objective

The objective of taking up this project is to create a cheap and working prototype of a Data Lake system, using free and open source tools as well as data acquired from govern websites that are available freely for public consumption.

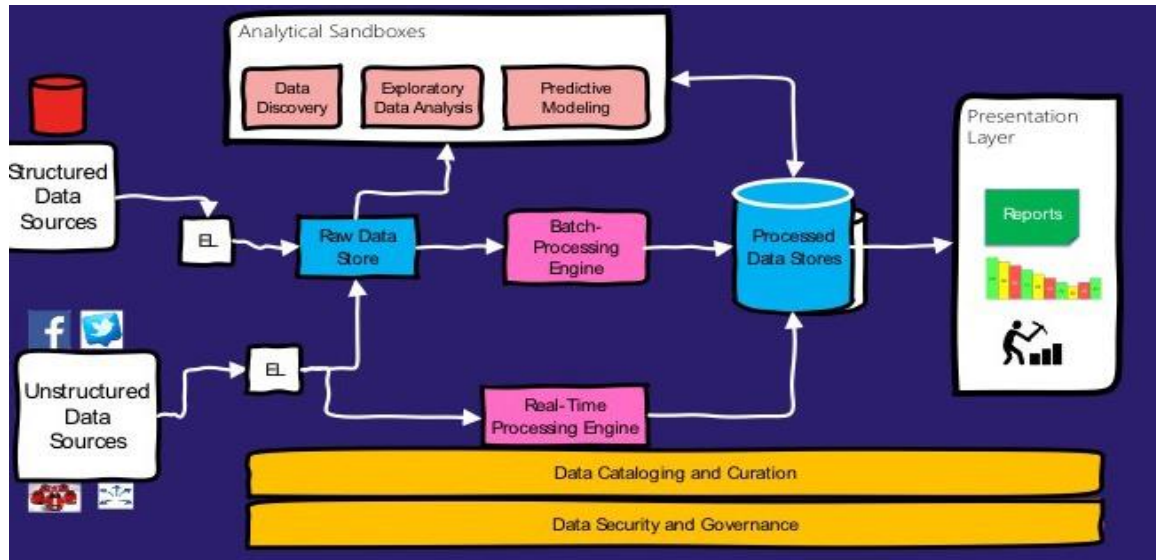
The system will also have a meta data that will guide the users on the data structures and their relationships.

The final goal is to help the small business establishments in EU and Ireland to perform a market analysis, as well as help students and analysts to test their Machine learning algorithms on the data sets available in the data lake.

Data lake scaling is a continuous process and the usability is expected to grow as we might provide additional facilities like security, visualization etc.

1.24 Technical Approach

KDD (knowledge Discovery in database) has phases starting from data selection to visualization. This project will provide data sources to implement KDD processes. [13]



Conceptual Data lake architecture

Selection: Identify all the structured and unstructured data sources.

Extract and load: Use appropriate APIs or web scrapping techniques to extract and load the raw data into the local storage system.

Transformation: Clean and transform the extracted data using a batch processing method. Establishing a batch process will be useful during periodic refresh of the data lake.

Schema development: Identify / design appropriate schema to store the data extracted from sources into the data lake.

Visualization: The end uses of this data should be provided with appropriate visualization tools so that it will be easy for them to choose the task relevant data for their KDD process.

1.25 The key aspect

Aim 1: Identify datasets from open public sources in Ireland and Europe. The dataset collection will be on all data related to banks, health, sports, housing exc. The data lake storage will contain a massive dataset to be used by data scientists of various verticals.

Aim 2: Clean the dataset. This process will be after identifying the target users of the data lake. This data will come with some errors and missing values, there is a need to work laboriously on this time taking step.

Aim 3: Identify the relevant attributes and store them in appropriate format / schema.

Aim 4: Provide facilities for visualizations such as graphs, charts etc. using R-Studio or any other analysis tools.

Aim 5: Test the data sets for usability using any of the existing machine learning algorithms.

1.26 Background

During my internship in AIB bank, I have moved to few locations inside the bank and I started to work with IT Audit. Later I moved to data analytics that helped me a lot to decide my final year project. I decided to do data lake system after understanding the real-world customer requirements as the.

I decided to build data lake storage system because is cheap to build a massive storage for any type of dataset. Other reason to build data lake storage is that is there are many open public source data from government websites. These datasets might not be used to their optimum level due to difficulties in extraction and cleaning of the data. The data lake that is being constructed will be an authentic and clean source of data for data scientists and machine learning experts.

1.27 Special resources required

Paid cloud storage might be in the later stage of the project dependent on the growing of the data the system.

1.28 Evaluation

After the project is completed the data lake system will be bench marked with any of the existing similar data lakes. Also, the quality of the data will be checked by testing the data with any of the existing machine learning algorithms.

Visualization is very important for the user of the data lake to help him choose the appropriate attributes for his data mining task. Hence various visualization features will be make available on the data lake.

1.29 TECHNICAL DETAILS

Java: If there is a need to use APIs to extract the data from data sources, java class programs might be written to call the APIs.

Excel: Use excel sheet to insert links to the websites from where the data needs to be extracted.

SQL / MySQL: Used for data storage and schema design.

R-Studio: One of the popular open source statistic tools, it provides features such as user's report, construction of graphs from the dashboard.

Tableau: Is one of interactive data visualization tools. It allows users to create a graph from the dashboard. I will be using Tableau to provide a graphic report.

































MapReduce: In case the data is very large we may have to use HDFS architecture for storage and analysis of the data.

1.30 Reference

Bibliography

- Anon., 2016. *7 Fundamental Steps to Complete a Data Project*. [Online]
Available at: <https://blog.dataiku.com/2016/07/06/fundamental-steps-data-project-success>
[Accessed 27 Novemebr 2017].
- Anon., 2017. *Big Data Analytics*. [Online]
Available at: <https://www.edureka.co/blog/what-is-tableau/>
[Accessed 28 November 2017].
- Anon., n.d. *API Overview*. [Online]
Available at: <https://docs.webhose.io/docs>
[Accessed 24 November 2017].
- Anon., n.d. *en.wikipedia..* [Online]
Available at: https://en.wikipedia.org/wiki/Machine_learning
[Accessed 30 November 2017].
- Anon., n.d. *MySQL*. [Online]
Available at: <https://www.siteground.com/tutorials/php-mysql/mysql/>
[Accessed 29 November 2017].
- Anon., n.d. *SAS*. [Online]
Available at: https://www.sas.com/en_ie/insights/big-data/hadoop.html
[Accessed 22 November 2017].
- Brownlee, D. J., 2016. *Your First Machine Learning Project*. [Online]
Available at: <https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>
[Accessed 22 November 2017].
- Ko, D., n.d. *webhos.* [Online]
Available at: <https://webhose.io/web-content-api>
[Accessed 18 Novemebr 2017].
- Marr, B., 2017. *Big Data*. [Online]
Available at: <https://www.bernardmarr.com/default.asp?contentID=1080>
[Accessed 20 November 2017].

1.31 Project Plan

	 Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish
1		▾ Project Proposal Stage	170 days	Tue 03/10/17	Sat 26/05/18
2		Brainstorming Project Ideas	3 days	Tue 03/10/17	Thu 05/10/17
3		Supervisor feedback on Idea	1 day	Thu 05/10/17	Thu 05/10/17
4		Project Pitch	1 day	Thu 05/10/17	Thu 05/10/17
5		Project Proposal preparation	6 days	Wed 11/10/17	Wed 18/10/17
6		Meeting with supervisor	1 day	Fri 20/10/17	Fri 20/10/17
7		Project Proposal submission	1 day	Thu 26/10/17	Thu 26/10/17
8		▾ Technial Research	27 days	Thu 26/10/17	Fri 01/12/17
9		Search Target data	3 days	Fri 27/10/17	Tue 31/10/17
10		Research for more detail about project	3 days	Tue 31/10/17	Thu 02/11/17
11		Research the programming language	4 days	Tue 03/10/17	Fri 06/10/17
12		Complete and review project proposal	4 days	Sun 08/10/17	Wed 11/10/17
13		▾ Requirements Specification	11 days	Sat 11/11/17	Fri 24/11/17
14		▾ Use Case Design	11 days	Sun 12/11/17	Fri 24/11/17
15		▸ Use Cases	1 day	Mon 13/11/17	Mon 13/11/17
18		Use Case Diagram	1 day	Tue 14/11/17	Tue 14/11/17
19		Technologies Used	3 days	Wed 15/11/17	Fri 17/11/17
20		Funtional & Non Funtional requirement	6 days	Wed 15/11/17	Wed 22/11/17
21		Requirement Specification submission	1 day	Fri 24/11/17	Fri 24/11/17
22		▸ Prototype Preparation	17 days	Sat 25/11/17	Mon 18/12/17
25		Mid Point Presentation & Prototype	5 days	Wed 29/11/17	Tue 05/12/17
26		Implementing	100 days	Wed 06/12/17	Tue 24/04/18
27		▾ Result	5 days	Wed 25/04/18	Tue 01/05/18
28		Coding	2 days	Thu 26/04/18	Fri 27/04/18
29		more coding	1 day	Fri 27/04/18	Fri 27/04/18
30		Design and Testing	1 day	Sat 28/04/18	Sat 28/04/18
31		▾ Report Writing	14 days	Sun 29/04/18	Wed 16/05/18
32		Complete the final documentation report	5 days	Sun 29/04/18	Thu 03/05/18
33		Final Software and Document Upload	2 days	Sat 05/05/18	Mon 07/05/18
34		▸ Testing & Implementation	14 days	Sun 06/05/18	Wed 23/05/18
36		Preparation for Project Presentation	4 days	Wed 23/05/18	Sat 26/05/18

1.32 Monthly Journal

1.32.1 September

My Achievements

This month I started my final year BSHC in computing course. I chose to do specialization in Data Analytics. I had spent 8 months working in AIB bank in two different areas namely, IT Audit and Data analytics. It was part of my internship. I had opportunity to work with more than one team. I came to college and I am hoping to finish this year with good scores and walk proudly out of NCI accumulating good knowledge.

During my first week in college I got to know my modules and had introduction about each subject and what I will be studying in first semester. I was not quite sure why we were studying 6 Moodle's during my first semester. Now I got to know why. Also, I was very confused about my project. I had few ideas and I thought these ideas were great from my understanding. But when I talked to Mr. Michael Bradford I realized that I do not have enough data to implement all those ideas, since it was based on fraud and risk analysis. So, I had to look for another idea. After this meeting I had done some research for find good project Ideas.

In my second week I started to setup my group for Web service API project. It was the same group which worked during previous years. Our first project was to do Investigation on Facebook design. My part is to do research on Facebooks architecture. This month also were supposed to do the CA individually for the course Introduction to Artificial Intelligence. We are supposed to do research and summaries three AI strategies that have been used in developing solutions for Chess Game. Also, I must start this moth strategic management CA where I have to do SWOT analysis and PEST analysis in AIB Bank. This is part of my internship work. During this month also, I will be doing my project pitch. Overall, I can say this month has not been a very busy month as we went through the subject and got to know my lectures.

My Reflection

I didn't feel I have anything to change since this is my first month

Intended Changes

From Next month, I shall plan for each subject individually.

Supervisor Meetings

No Supervisors had been assigned till now.

1.32.2 October

. My Achievements

This month has been quite a busy one. I had to complete the CA's, report, practical AI and research on project ideas. I started in September and planned to submit on time. Also, I was busy learning for weekly statistic CA that is on every Friday. In this month most my time was spent searching about my project idea and trying to figure out approach for my final year project. I have been submitting all my CA's on time. Most of the time went in figuring out what is to be done. Also, I had few meetings with my previous manager in AIB regarding secondary research on my SWOT and PEST analysis. So I went down to the bank and had some chat with few managers. I felt nice meeting them and I came up with lot of knowledge to incorporate into my report.

I had my report on Facebook architecture finished and also other CA was submitted on time. I feel the pressure about my final semesters and the amount of work to be done. So, I created my study plan for first semester, which will help me a lot in managing my time. I am trying to finish the web service API project. There are lot of discussions happening on it.

My Reflection

I understood that when I tend to spend too much time on one subject, the other subjects are lagging behind. Also, I felt that I had to give more time to my project and to spent time learning about the tools that I could use in the project.

Supervisor Meetings

Date of Meeting: 20/11/2017

Items discussed: I had a meeting with my supervisor, Mr. Michael Bradford about project design and project plan. He gave me few tips about choice of tools and

dataset. I will use his tips for sure. Also we will be meeting every two weeks to have more discussions on the projects. There isn't much time left for the review.

1.32.3 November

My Achievements

This month my focus was on completing two documents, namely Requirements specification and the Technical report. This took most of the time. This month I started to dig deeper in to the project. I did some research on the database that I should be using in my project. I found few government websites that contain free datasets. These datasets need to be analyzed before extraction. Also, I was working on my requirements specification and updated the project proposal. While writing my proposal and requirements specification I did a lot of research on the tools, programming language and technical architecture for the project.

I was very busy preparing for few CA's, Data Application Development, Web Service API and Chess game demonstration.

My Reflection

I felt that this month was the toughest. I had a lot of pressure to complete the Project proposal, requirement specifications, Chess game presentation, two CA's for Data Devolvement Application, and Web Service API. I found it tough to manage all these together and it affected my project research.

Intended Changes

I realized that I need to learn more about the tools and data extraction process. I set a plan to do that in my free time out of college. Next month I am going to focus on my exam preparation and completing other projects such as Data Development Application and Web Service API.

Supervisor Meetings

Date of Meeting: 03,10, 17 Of October 10/2017

Items discussed:

Each day we had different topics to discuss about. At first we talked about the project design and architecture. Second time we talked about project use case and how it will be related to the project. It was a bit confusing initially. On 17th Nov I discussed about the project proposal and requirement specification. In general, I'm happy with my supervisor's guidance.

Action Items:

I need to complete my project proposal and requirement specification. I still have some time left to update those documents.

1.32.4 December

My Achievements

In this month I completed the documentation and also the prototype that was due for the midpoint presentation. I also attended the mid-point presentation. This month I was still working on my Data application project and the project on web services API. Each one was worth 50%. so I chose to focus on both. I was constantly in touch with my group members, Jon and Alex. We all together set a plan to complete the web service API and go through our group project. Web service API took time to do and few things might go wrong during coding. So we submitted it little before time.

After we submitted both thhe projects in this month I started my exam preparation that starts in the month of January. I have following course exams to give this semester. They are Introduction to Artificial Intelligence, Strategic Management and Business Data Analysis. I had only few days to go for the exam. So I was very busy during Christmas studying and managing other staff outside college.

My Reflection

Took a big step and completed the software projects documentation.

Intended Changes

Next month, I intend to continue working on my software project following the project plan. After receiving the feedback from Mr Michael, my supervisor during the mid-point presentation, I ensured to complete the project on time.

Supervisor Meetings

Date of Meeting:01/12/2017

During the only chance that I got to meet my supervisor , Mr Michael, he has helped me to understand the requirements for my midpoint presentation. He advised me to change few things in my requirement specifications, such as use case and design architecture. Some of Items discussed in this meeting happened to be the end of Action Items:

1.32.5 January

My Achievements

This month I had received feedback from my supervisor about the mistakes I did in my project documentation during midpoint presentation. Michael pointed few things and should be changed and about my understanding of creating a data lake. I spent a lot of time in correcting my mistakes and showed it to him after two weeks. During this month, I gave three exams. They are, Business Data Analysis, Introduction to Artificial Intelligence and Strategic Management. It was quite tough to prepare for those exams especially because preparation started after a busy month. After exams we had two weeks break. In these two weeks, I was learning extraction of data from website to be used in my project. The college starts at end of January for second semester. In this semester we have two subjects namely, Data and web mining and advance business analytics. I intend to focus mainly on the software project.

My Reflection

I felt I spent a lot of time learning theoretical concepts for my software project and I need to do more in coding part also as much as possible.

Intended Changes

Next month, I intended to into the technicalities of the software project.

Supervisor Meetings

Date of Meeting: Thursday 25/01/2018

Items discussed: Most of our discussions are about the feedback and plan of action for my project completion on time.

Action Items:

Change few things in my software project documentation and correct my mistakes.

1.32.6 February

My Achievements

In this month I spent most of the time focused on designing the code for the extraction of the data using python and PostgreSQL database. In the beginning of the month I familiarized myself with python and PostgreSQL. After that I started coding a python library to insert data into my dataset. Also, I have been doing my project analysis design documentation. Apart from that, I completed research on how to store data into the database using python. In the other modules, I finished my data mining project and I started the data mining group project.

My Reflection

I tried storing some data into the database to test my project implementation. I was also able to store some URL links to an excel sheet. These URLs will be used to populate the data lake.

Intended Changes

Next month, I will try to do some more coding. The next part of the project is going to get much more difficult and I realised that I need to spend more time coding. Also, I uploaded my test code to GitHub to make sure everything has backup. It gave me feedback on the right choice of data base to use.

Supervisor Meetings

Date of Meeting: 01,08,15 of February

Items discussed: kept meeting Michael to guide me through of the coding part as I found it a bit difficult. I am happy that he answered all my questions related to the coding part and also other questions on the projects.

1.32.7 March

My Achievements

In this month I have been working on the coding part. I made good progress and I have stored data of different formats to the PostgreSQL database stage and also, I started to get familiar myself with Hadoop distributed System which I will be using to store all my dataset I will extract. Also, I am in the final stages of writing my showcase project description. I was quite lost when I started coding last month because, I have been using java (NetBeans). Later, I changed my mind after I spoke with my supervisor. He guided me very well and suggested me to use python instead. I still have a lot of research left to do. I had few meetings with my data mining group project team mates. We have been working on the dataset that we got from Kaggle.

My Reflection

I completed the necessary part of the project that was due for the next month. I have some coding to do in the coming weeks.

Intended Changes

Next month, I intend to continue working on the code and research on new technologies for my project such as Hadoop, Hive, Pig, Spark. Once I am able to store dataset completely, I intend to test one of these tools on it.

Supervisor Meetings

Date of Meeting: 01,08,15,22,29 of March

Items discussed: This month I met my supervisor five times. Most of the discussions were about the code and the errors. Mr. Michael guided me to fix some of the errors that I had during installation of pip and database. Thus, meeting my supervisor helped me a lot.