ASSIGNMENT NO. B-C2

Problem Statement:

Execute at least three commands related to the Storage organization of the cloud. Create necessary GUI using python.

Learning Objectives:

1. To Study and Understand the concepts of storage organisation of the cloud.
2. To learn implementation of a GUI using python.

Learning Outcomes:

Learnt about the cloud storage organisation and handling using a python GUI.

Software and Hardware Requirements:

1. 64-bit operating System(Linux)
2. Text Editor-gedit
3. Terminal
4. python 2.7
5. ubidots
6. Modelio (version-3.6)

Theory

Ubidots

Ubidots offers a platform for developers that enables them to easily capture sensor data and turn it into useful information. Use the Ubidots platform to send data to the cloud from any Internet-enabled device. You can then Configure actions and alerts based on your real-time data and unlock the value of your data through visual tools. Ubidots offers a REST API that allows you to read and write data to the resources available: data sources, variables, values, events and insights. The API supports both HTTP and HTTPS and an API Key is required.

Python Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

Mathematical Model

```
Let S be the solution for class QuickSort
S = {s, e, i, o, f, DD, NDD, success, failure}
s = Initial state
e = End state

i = Input of the system {b1,b2,b3,b4 where n = Button }\newline
b1 = create datasource {'ds_name','ds_parameters'}\newline
b2 = create variable {'var_name','var_parameters'}\newline
b3 = set values {'v1,v2...', v = I,string}\newline
b4 = get values {'v1,v2...', v = I,string}\newline

o = Output of the system {logs.txt, alerts, memory_organization}


 DD = Deterministic data: Input to set values

NDD = Non deterministic data: Space available or required on the cloud platform

f = {create_datasource, create_variable, set_values, get_values}


Success - Required Ouptut is generated i.e. elements from xml file get
  sorted
Failure - Incorrect Output is generated
```

Use-case diagram

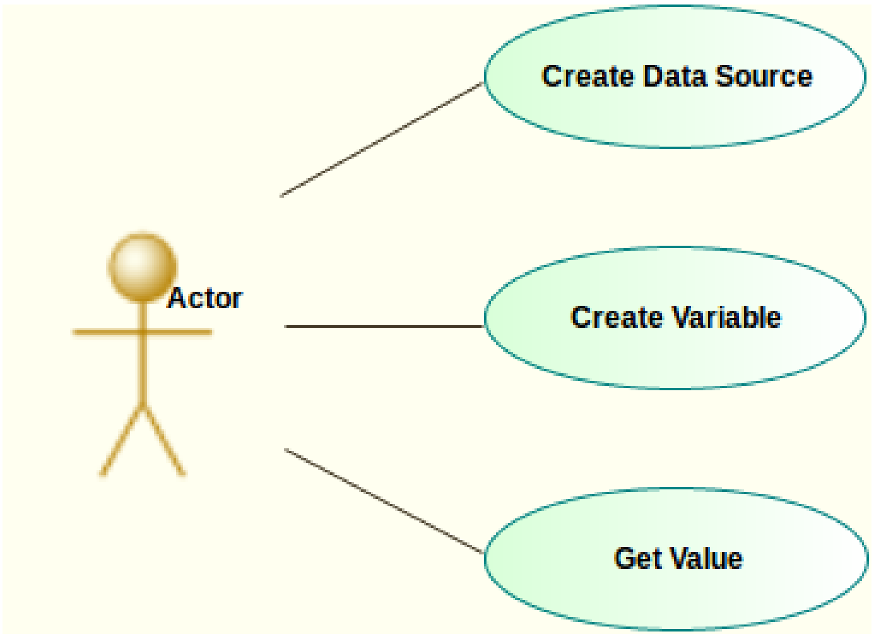
Class diagram

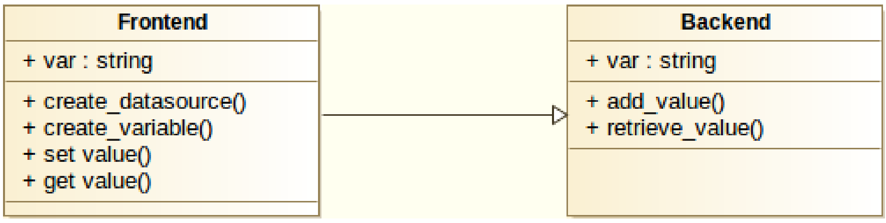
Sequence diagram

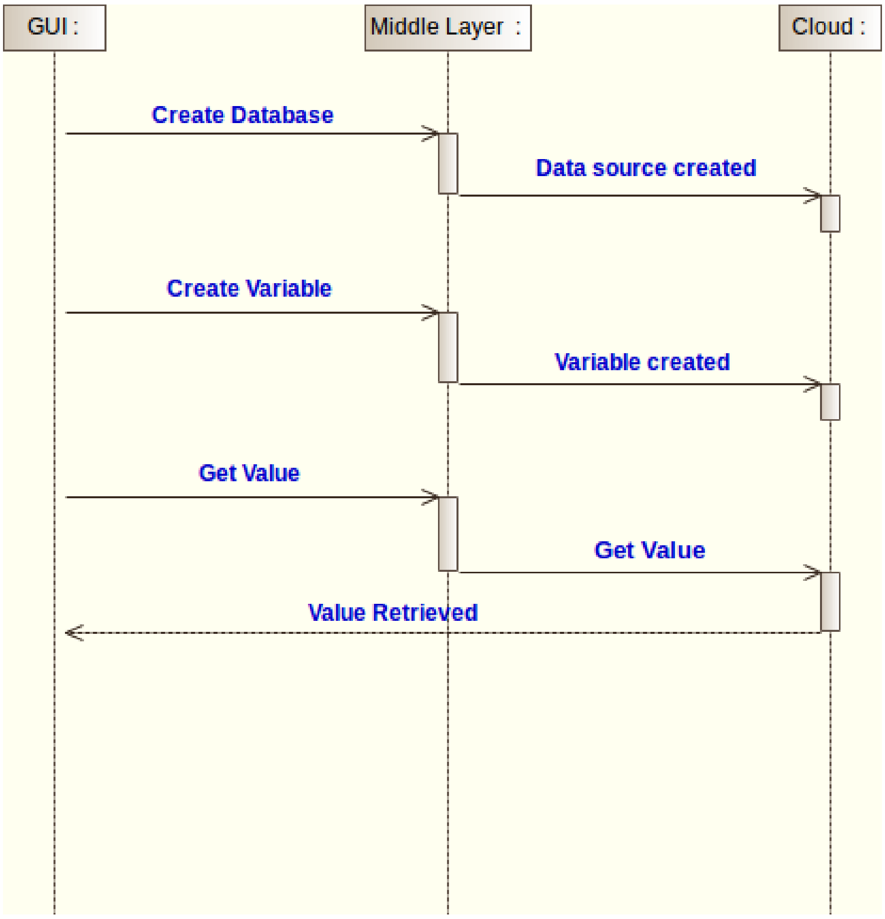
State diagram

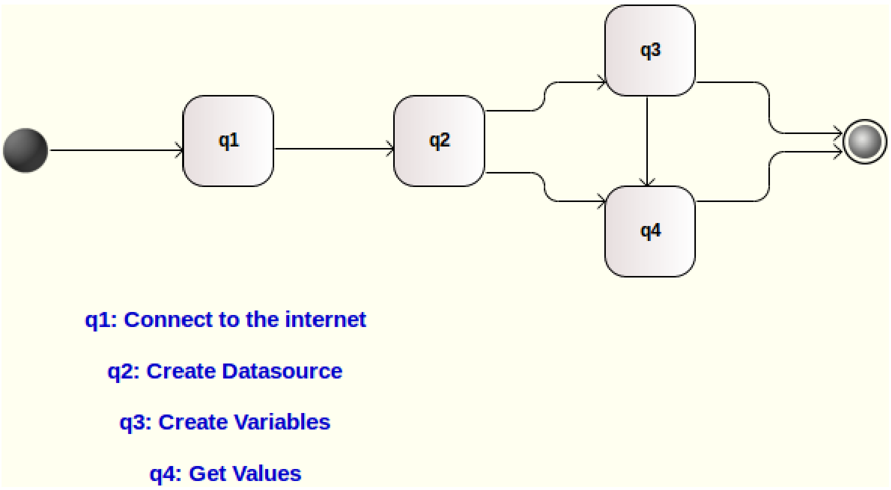
Algorithm

1. Get an instance of the cloud platform.
2. Create a datasource.
3. Create a variable.
4. Set values to the variable
5. Get values from the variable


Create Data Source
Actor
Create Variable
Get Value

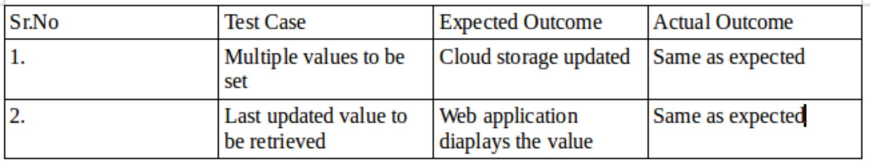

Frontend
+ var : string
+ create_datasource()
+ create_variable()
+ set value()
+ get value()
Backend
+ var : string
+ add_value()
+ retrieve_value()


GUI :
Middle Layer :
Cloud :
Create Database
Data source created
Create Variable
Variable created
Get Value
Get Value
Value Retrieved


q1
q2
q3
q4
q1: Connect to the internet
q2: Create Datasource
q3: Create Variables
q4: Get Values

Positive Testing

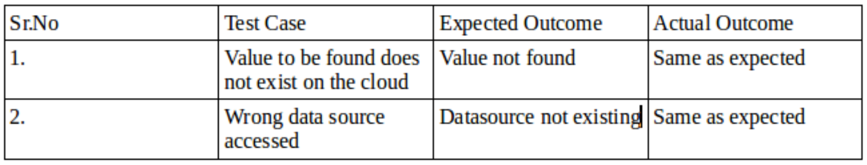
Positive testing is a testing technique to show that a product or application under test does what it is supposed to do. Positive testing verifies how the application behaves for the positive set of data.



Sr.No	Test Case	Expected Outcome	Actual Outcome
1.	Multiple values to be set	Cloud storage updated	Same as expected
2.	Last updated value to be retrieved	Web application diaplays the value	Same as expected

Negative Testing

Negative testing ensures that your application can specifically handle invalid input or unexpected user behavior.



Sr.No	Test Case	Expected Outcome	Actual Outcome
1.	Value to be found does not exist on the cloud	Value not found	Same as expected
2.	Wrong data source accessed	Datasource not existing	Same as expected

Conclusion

We have successfully implemented the storage organization on a cloud platform.