High Availability (HA) Distributed File Storage

Document: Design document

Version 1.0

Publication date: 2-MAY-2016

Group name: Gryffindor

Group Members:

- ARJITH CHALASANI
- JITENDRA NEELAM
- URMILA JYOTHULA
- CHINNA BALAJI YALLA
- NITISH NAGABHAIRAVA
- PRANEEL REDDY PADALA
- SURYA TEJA BOLLIMPALLI
- SRI GANESH SAI GUNNAM
- SRI KASYAP KAPPAGANTULA
- SIRISHA MANASWINI BHAMIDI
- RAGHUVINAYAK RAO MEDISETTI
- MAHAMMAD SUHAIL ATCHUKATLA

Design Document

Contents:

1. Preface
   Release v1.1 on 2016-04-25
   Release v1.0 on 2016-04-18

2. Glossary and abbreviations

3. Model 1 Name
   3.1 Detailed Design
   3.2 Unit Test Plane
   - **Test:** identification alphanumeric string
   - **Purpose:** what will be tested
   - **Requirements:** requirement IDs that are connected to this test
   - **Environment:** settings that must be performed before the test can be run
   - **Operation:** step-by-step instructions how to execute the test
   - **Expected result:** what should happen after the operation is carried out
   - **Result:** success/failure (testers can ring in the result)
   - **Comment:** useful information filled in by testers

4. Model 2 Name

5. References

Design Document

1. Preface

The main concept of this project is to develop a secure file storage to the company SecureFile in the form of a distributed file storage system with high availability to the customers.

When a user uploads a file, the file is stored in a randomly chosen server. In this we are creating replicas for the file uploaded by the user and we use file transfer protocol for the transfer of data.

Service Developer: Gryffindor

Customer: Dragos llie

In this document we defined the technical terms and a short note on them, Design models, Unit test plan and references.

- **Release v1.1 on 2016-04-25**

  Preface modified

  Detailed design modified in model 1

  Purpose, operation and expected result changed

- **Release v1.0 on 2016-04-18**

  Initial Release

2. Glossary and abbreviations

- **HTTP: Hyper Text Transfer Protocol**

  It is a transfer of version data formats between server and client

  EX: plain txt, hyper txt, video and sound

- **FTPS: File Transfer Protocol Security**

  It is an extension for commonly used file transfer protocol(FTP) that adds support for the transfer layer security(TLS) and secure sockets layer (SSL)

- **Message digest: SHA-1**

  IT is a crypto graphic hash function which is consider practically impossible to invert *that is to recreate the input data from its hash value alone.*

  SHA-1: secure Hash algorithm.SHA-1 produces a 160bit (20 byte) hash value known as a message digest. SHA-1 advancements are SHA-2 and SHA-3

- **GUI: Graphical User Interface**

  It is a type of interface which helps in interaction with electronic devices through graphical icon and visual indicators.

- **SQL Server: Structured Query Language Server**

  SQL is used to store, query and manipulate data. It is used for manage data in a relational data base.

- **Restful API: Representation State Transfer**

  An architectural pattern to improve probability and scalability of a system.

3. Model 1 Name

Module 1: -FRONT END

In this module we described the web interface of the tool. First, the user needs to register into the server customer receives the conformation mail from the admin server. now customer can login into the server the information of the user is stored in the MySQL database.

3.1 Detailed design

We create a login page using HTML and CSS for the user in order to login into the database.

At first we create a registration page for the user to register into our database.

For the registered customers, a confirmation mail is sent through the admin server and this process is done using PHP.

HTML and CSS pages are connected through PHP.

F-T1:

Module: html, css, php

Purpose: Display Web GUI

Requirement: req_sys_1, req_nfsys_1

Environment: browser for rendering webpages

Operation: Here the user logins into the data base. In order to login user need to register in to data base where he receive a confirmation mail through snmp function then user can login into the database

Expected result: The user can access into the database. In order to login user needs to register and then user receives a confirmation mail. Confirmation mail is tested in this test

Comment: We use these languages for the creation of the webserver to the customers where they can login into the database.

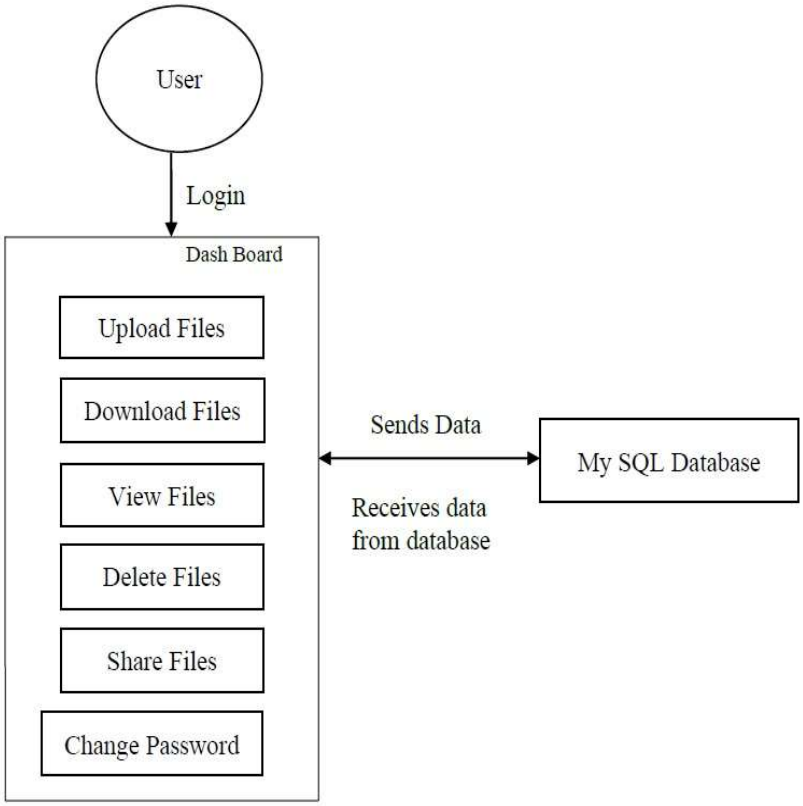

Fig 1: Front End

Module 2: database management

This module contains the database management portion of the product it also includes how the database interacts with the customers
In here we use MySQL for creation of the database.it is used as an intermediate to interact with the front end the back end of the project.

4.1 detailed design:

This module provides the vivid explanation of the database management in the project. we create a database to the customer where customer can store the information and files. The customer is assigned with random server and file is uploaded. we just use database to allocate data to the customer the customer is provided with a limited data of 9 Gb.
The database is created using MYSQL. Through MySQL we create tables where the data is stored by the servers.

PHP- MySQL I

Is a relational database driver that provides an interface between the MySQL and php .it includes functions like establishing connection in the database and functions like manipulating the database (create, delete, modify etc) the admin server can create a database to the customer or also remove him from the database.

Python – MySQL: -it is a MySQL driver for the python interface. Providing an interface between the python programming language and the MySQL relational database management system

D-T1: Module: MySQL

PURPOSE: for creation of database to the customer

REQUIREMENT: REQ-SYS-2

ENVIORNMENT: this Module is used in the PHP script

OPERATION: creation of database is done by using mysql through using command "mysql -u username -p" in the terminal. in order to view the database, we use the commands "mysql -u username-p" and then we type mysql>show databases;

EXPECTED RESULT: the database to the users is created by using the mysql. The tables are created in order to store the data which is uploaded

COMMENT: it updates the database periodically whenever a new customer is added into the database.

D-T2: Module: PHP: MySQL

Purpose: creating an access to the customer to store the information in MySQL
Requirement: REQ-SYS-2
Environment: module must include in the PHP script
Operation: creation of the user in the MySQL is done through 'create user 'username'@'localhost'"; without quotes than he is registered into MySQL. through php a customer is registered into the database

Expected result: creation of 'MySQL user' to the customers is done. The user is granted permission to store the data
Result:
Comment: interface between MySQL and PHP script

D-T3 MODULE: python: MySQL

Purpose: for creation of replicas and storage of the replicas in the rest of the servers (MySQL tables)

Requirement: REQ-SYS-2, REQ-SYS-3
Environment: module needs to be included in the python script
Operation: the creation of the replicas is done in python. We create a cluster and it is used to decide master and slave for performing the operation of storage and retrieving the data from database

Expected result: creation of replicas of the files and storage of these files in the rest of the servers {MySQL tables)

Result:

Comment: interface between MySQL and python

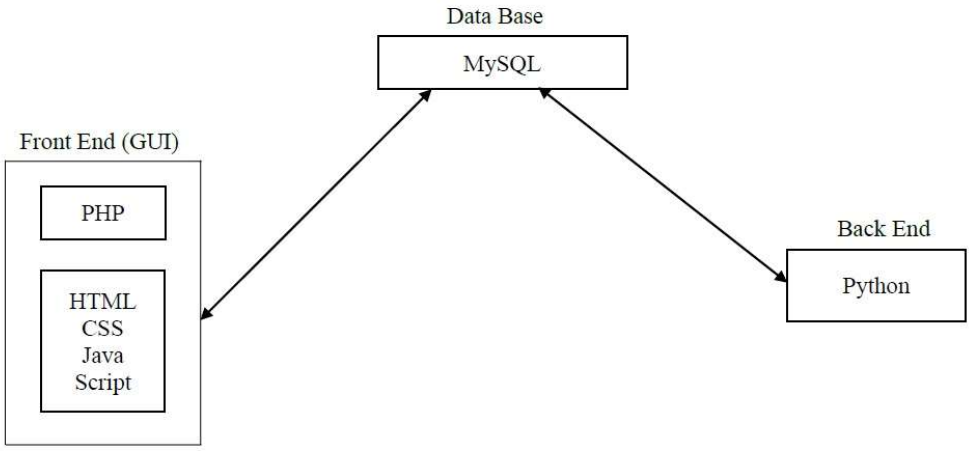

Fig 2: Data Base

Module 3: -back end

The back end of the project is performed over the python language. We use the back end for the creation of replicas of the files

PYTHON is used to perform two main operations:

1) Pinging the server.
2) Replication of files.

5.1 detailed design

The back end is used for the creation of replicas to the files at first the file is randomly uploaded into the server and we create replicas to those files and these are stored in the other servers. The connection to the servers are checked by constant pinging performed by the back end. If the connection is lost with the server, it is informed to the admin server he removes the server and manually repairs it later the verification of the server takes place and see that all the files are present and conformation is done.

5.2-unit test plan

B-T1: **module:** FTPS

Purpose: it provides a transfer of files from one server to another server

Requirement: REQ-SYS_3

Environment: module must be included in python script

Operation: it is executed from the directory, through this transfer of files between a user and server on a computer network takes place
To install Ftps - sudo apt-get install vsftpd
creating ftp user - sudo adduser ftpuser1

Expected result: The transfer of file takes place

Result:

Comment: it is used to transfer files from a server to another server .it is used to download files from the server.

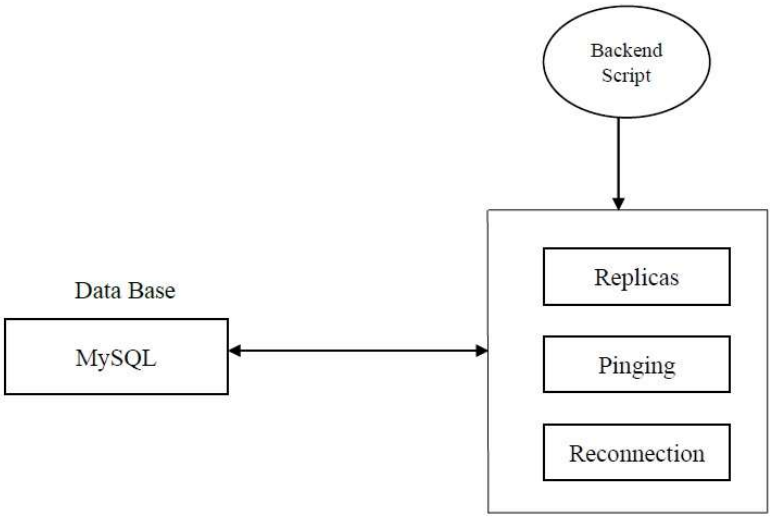

Fig 3: Back End

6. References:

- Sommerville, Ian. Software Engineering, 9th ed. Addison-Wesley, 2011