**Acceptance test plan**

**High Availability (HA) Distributed File Storage**

**Document: Acceptance test plan**

**Version 1.0**

**Publication date: 13-MArch-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**

**Acceptance test plan**

**Contents:**

**1. Preface**

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

1. **Glossary and abbreviations**
2. **Acceptance test plan**

**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**

1. **References**

**Acceptance test plan**

**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.**

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

**Service Developer: Gryffindor**

**Customer: Dragos llie**

1. **In this document we defined the technical terms, Acceptance test plan and references.**

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

**Acceptance test plan**

**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 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. Acceptance test plan**

**FRONT END**

**1) F-T1: AUTHENICATION**

**TEST: Login page test**

**Purpose: TO create an access to the customer to login into the dashboard.it is also used to prevent unauthorised users to login into the page**

**Requirements: REQ-SYS\_1, REQ-SYS\_2, REQ-SYS\_3, REQ-USR\_2 Environment: Browser for rendering the webpages, php**

**Preinstalled PHPOpen terminal**

**Run command: sudo apt-get install php5-server Pre-installed apache server**

**Run command: sudo apt-get install apache2**

**Acceptance test plan**

**Pre-installed MYSQL database**

**Run command: sudo apt -get update Run command: sudo apt-get upgrade**

**Run command: sudo apt-get install mysql-server Database must be created for the users to store data**

**Operations:**

**Open web browser**

**Go to login html page from local host and enter user credentials If correct credentials are entered, it will direct to dashboard**

**If incorrect credentials are entered, alter message is shown and access will be not granted**

**Expected result: Displays the dashboard to the users and where the user can upload the files**

**Result:**

**Comment: Before rendering the webpage apache2 status must be checked. “service apache2” status on the terminal**

1. **F-T2: Receive update on status and uptime of service Test: Notification test**

**Purpose: To check the automatic update of service and service uptime Requirements: REQ-USR\_3, REQ-SYS\_2, REQ-SYS\_3 Environment: Web browser, apache server**

**preinstalled php5open terminal**

**run command: sudo apt-get install php5 preinstalled apache server**

**run command: sudo apt-get install apache2 preinstalled MySQL database**

**run command: sudo apt-get install MySQL-server**

**database must be created with tables containing fields for status and uptime for each service**

**it shows the service we provide and also the status of each server are kept in this tables**

|  |  |  |
| --- | --- | --- |
| **Operation:** |  |  |
|  | **Open web browser** |  |
|  | **Enter the dash board through the login page** |  |
|  | **On the home page to the right we show all the services we provide** |  |
|  | **for uploading the files and altering the files** |  |
| **When the server is down the ping is continuously done and when** |  |
|  | **server is back we inform it to the admin server** |  |

**Acceptance test plan**

**On the statistics page the uptime of each server are kept and we see if the server is online or not if it is down we note the time and repair it manually**

**Expected result: Status changes from running to ERROR when a service stops. When a service is restarted, the uptime is updated and reset**

**Result:**

1. **F-T3: Uploading the files Test: Upload of files**

**Purpose: To check whether the files are uploaded into the dashboard Requirements: REQ-USR\_4, REQ-SYS\_3**

**Environment: Web browser, MySQL Preinstalled php5**

**Open terminal**

**Run command: sudo apt-get install php5 Preinstalled MYSQL database**

**Run command: sudo apt-get install mysql server**

**Database must be created to upload the files for the customer Files must be uploaded into his dashboard**

**Operations:**

**Open web browser**

**Enter the dashboard through the login page**

**Upload the files by pressing over the upload option. The user space is limited**

**Expected result: The files are uploaded into the user dashboard and these files show the space occupied in the dashboard**

**Result:**

**DATA BASE**

**1) D-T1: MySQL database contains user credentials**

**Module: php, MySQL**

**Purpose: For interaction between php script and MySQL**

**Requirements: REQ-SYS\_2**

**Environment: MySQL database contains a table in the database for the user credentials**

**Pre-installed php5**

**Open terminal**

**Request command: sudo apt-get install php5 Preinstalled MySQL database**

**Run command: sudo apt-get install MySQL server**

**Acceptance test plan**

**Database must be created with tables containing fields for status and uptime of server**

**Database provides each customer a space to store the files with a limited space**

**Operations:**

**Open web browser**

**Enter the tool dashboard click on register new user. enter the requested credentials. a new user is created**

**Open MYSQL in terminal by entering” MySQL -u user -p” Check users table in the tools database for the registered user credentials**

**Expected result: The table contains user credentials as created from the front end**

**Result:**

**DATA BASE**

**1) D-T1: MySQL database contains user credentials**

**Module: php, MySQL**

**Purpose: For interaction between php script and MySQL**

**Requirements: REQ-SYS\_2**

**Environment: MySQL database contains a table in the database for the user credentials**

**Pre-installed php5**

**Open terminal**

**Request command: sudo apt-get install php5 Preinstalled MySQL database**

**Run command: sudo apt-get install MySQL server**

**Acceptance test plan**

**Database must be created with tables containing fields for status and uptime of server**

**Database provides each customer a space to store the files with a limited space**

**Operations:**

**Open web browser**

**Enter the tool dashboard click on register new user. enter the requested credentials. a new user is created**

**Open MYSQL in terminal by entering” MySQL -u user -p” Check users table in the tools database for the registered user credentials**

**Expected result: The table contains user credentials as created from the front end**

**Result:**

**DATA BASE**

**1) D-T1: MySQL database contains user credentials**

**Module: php, MySQL**

**Purpose: For interaction between php script and MySQL**

**Requirements: REQ-SYS\_2**

**Environment: MySQL database contains a table in the database for the user credentials**

**Pre-installed php5**

**Open terminal**

**Request command: sudo apt-get install php5 Preinstalled MySQL database**

**Run command: sudo apt-get install MySQL server**

**Acceptance test plan**

**Database must be created with tables containing fields for status and uptime of server**

**Database provides each customer a space to store the files with a limited space**

**Operations:**

**Open web browser**

**Enter the tool dashboard click on register new user. enter the requested credentials. a new user is created**

**Open MYSQL in terminal by entering” MySQL -u user -p” Check users table in the tools database for the registered user credentials**

**Expected result: The table contains user credentials as created from the front end**

**Result:**

1. **D-T3: MYSQL database for each user**

**Module: MYSQL**

**Purpose: For creation of database for the user**

**Requirements: REQ-USR\_4**

**Environment: MYSQL database creation of customer database**

**Acceptance test plan**

**Preinstalled MYSQL server**

**Run command: sudo apt-get install MySQL server To create a separate database for each customer All the files are stored in his personal database**

**Operation:**

**Open web browser**

**Enter the login details of the customer and login into dashboard At the right side all the options are present to upload files**

**Expected result: The user is allocated with a database to store the information.**

**Result:**

**BACK END**

**1) B-t1: Ensure establishment of secure connection between user and server**

**Module: Python**

**Purpose: To execute commands and create a connection between user and server**

**Requirements: REQ-NFSYS\_2**

|  |  |  |
| --- | --- | --- |
| **Environment:** | |  |
| **Operation:** | |  |
|  | **It is used to establish a connection between user and server** |  |
|  | **When the user logins into the server through login page the connection or** |  |
|  | **selection of server is random process** |  |
| **The user can upload the files into the servers** |  |

**Expected result: It is used to select a random server to the user and establish a connection between them**

**Result**

**2) B-T2: Status and uptime of servers are to be known through pinging**

**Module: Python: MySQL**

**Purpose: To insert status and uptime in MySQL database. Later the server that is down is manually repaired**

**Requirements: REQ-SYS\_2, REQ-NFSYS\_2**

**Environment:**

**Python connection**

**Establish connection between backend and database MYSQL database containing tables for each server**

**Acceptance test plan**

**OPERATIONS:**

**To check that the servers are online or not**

**The constant pinging process is done through the python coding**

**Expected result: Establishment of secure connection between database and backend. the servers working is known through constant pinging of the server**

**Result:**

**3) B-T3: Creation of the replicas**

**Module: Restful API**

**Purpose: To create the replicas of the files and save them in main server and their links are to be kept in other servers**

**Requirements: REQ-SYS\_3**

|  |  |  |
| --- | --- | --- |
| **Environment:** | |  |
| **Operation:** | |  |
|  | **When the user uploads the files, replicas of the files are done** |  |
|  | **The replicas are created in all the servers so that when the server is down it** |  |
|  | **doesn’t affect the customer** |  |
| **This is done through the restful API** |  |

**Expected result: Creation of the replicas of the files that are uploaded into the database through the restful API**

**Result:**

**4. References:**

**http://php.net/manual/en/book.mysql.php**

**http://search.cpan.org/~capttofu/DBD-mysql-4.031/lib/DBD/mysql.pm Sommerville, Ian. Software Engineering, 9th ed. Addison-Wesley, 2011**