The National University of Lesotho

Department of Mathematics and Computer Science

Faculty of Science and technology

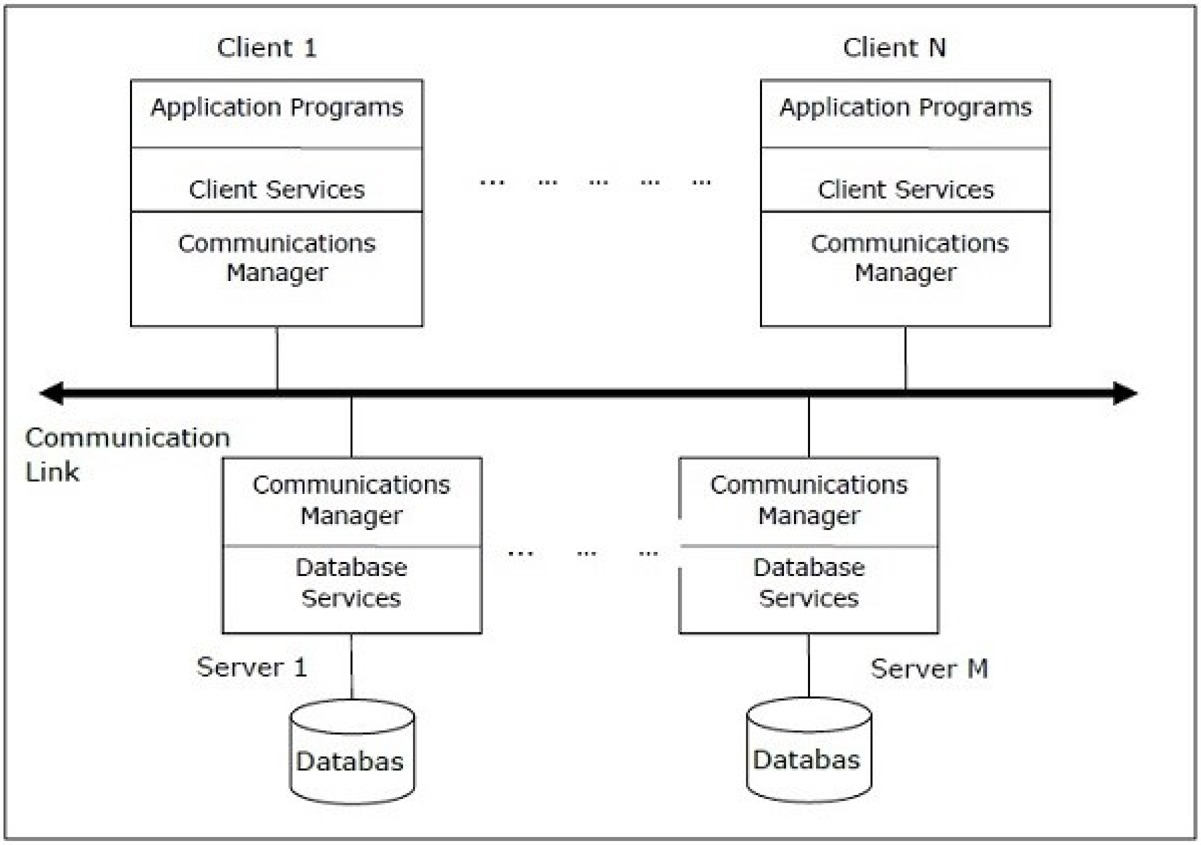
CS4430: Distributed Database Systems

Task: System design

	Student Number	Surname, initials	Major
1.	201903092	Matlali, T.M	B.Eng. Computer Systems and Networks
2.	201804372	Motsoane, T. E	B.Sc. Maths and CS
3.	201904387	Mohajane, S. B	B. Sc. CS
4.	201902164	Putsoane T.	B.Eng. Computer Systems and Networks
5.	201901893	Mosese, R. S	B.Eng. Computer Systems and Networks

1 Detailed Real-World Scenario system Architecture Diagram.

1.1 The implementation design of this system is by use of a two-level architecture where the functionality is divided into servers and clients. The server functions primarily encompass data management, query processing, optimization and transaction management. Client functions include mainly user interface. However, they have some functions like consistency checking and transaction management.



1.2 Hardware architecture.

- **Client Layer:** The client layer would consist of devices used by end-users to access the CTMS software. These devices could include desktops, laptops, tablets, or smartphones, depending on the needs of the users. The hardware requirements for the client layer would depend on the complexity of the software, but generally, modern devices with a fast processor, sufficient memory, and a high-quality display would be recommended.

- **Server Layer:** The server layer would consist of the physical servers that host the CTMS software and manage the database. The server layer can be deployed in different ways, depending on the needs of the organization. For example, a cloud-based solution could be used, where the servers are hosted by a third-party provider and accessed over the internet. Alternatively, an on-premises solution could be used, where the servers are located in-house and managed by the organization's IT department.

In either case, the server layer would require the following components:

- **Server Hardware:** The server hardware should be powerful enough to handle the demands of the CTMS software and the database. This would typically include a fast processor, a large amount of memory, and sufficient storage capacity. Redundant power supplies and network connections should also be included to ensure maximum uptime.

- **Networking Equipment:** The server layer would require networking equipment such as routers, switches, and firewalls to manage traffic between the client layer and the server layer. The network should be designed to ensure fast and reliable communication between the two layers, with adequate security measures in place to protect against unauthorized access.

  Storage Devices: The server layer would require storage devices to manage the CTMS database. Depending on the size of the database, this could include solid-state drives (SSDs), hard disk drives (HDDs), or network-attached storage (NAS) devices.

1.3 Networking architecture (Local Site and Remote).

The following components would be included:

A Local Area Network (LAN) for connecting the study staff computers and the servers at the local site.

A Wide Area Network (WAN) for connecting the local site to remote locations. This basically supports the connections between servers at different sites, distributed geographically.

1.4 Software architecture (Platform and Application)

The software architecture for this system would include the following parts:

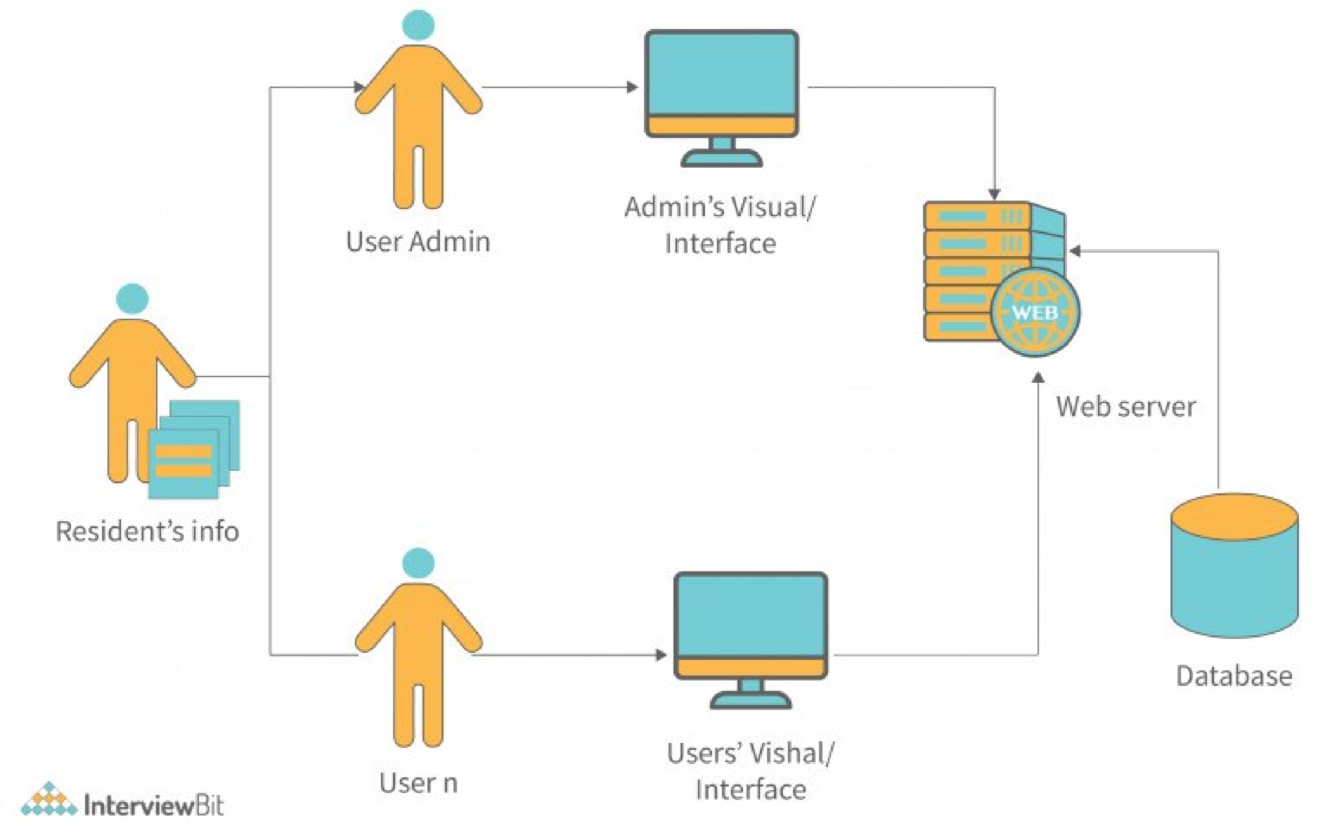
1.4.1.1 A platform that includes the operating system, web server, and database management system (DBMS) for hosting the different modules of the system.

1.4.1.2 Web-based application for the different modules of the system, such as study setup and design, participant management, data management and reporting and analytics.

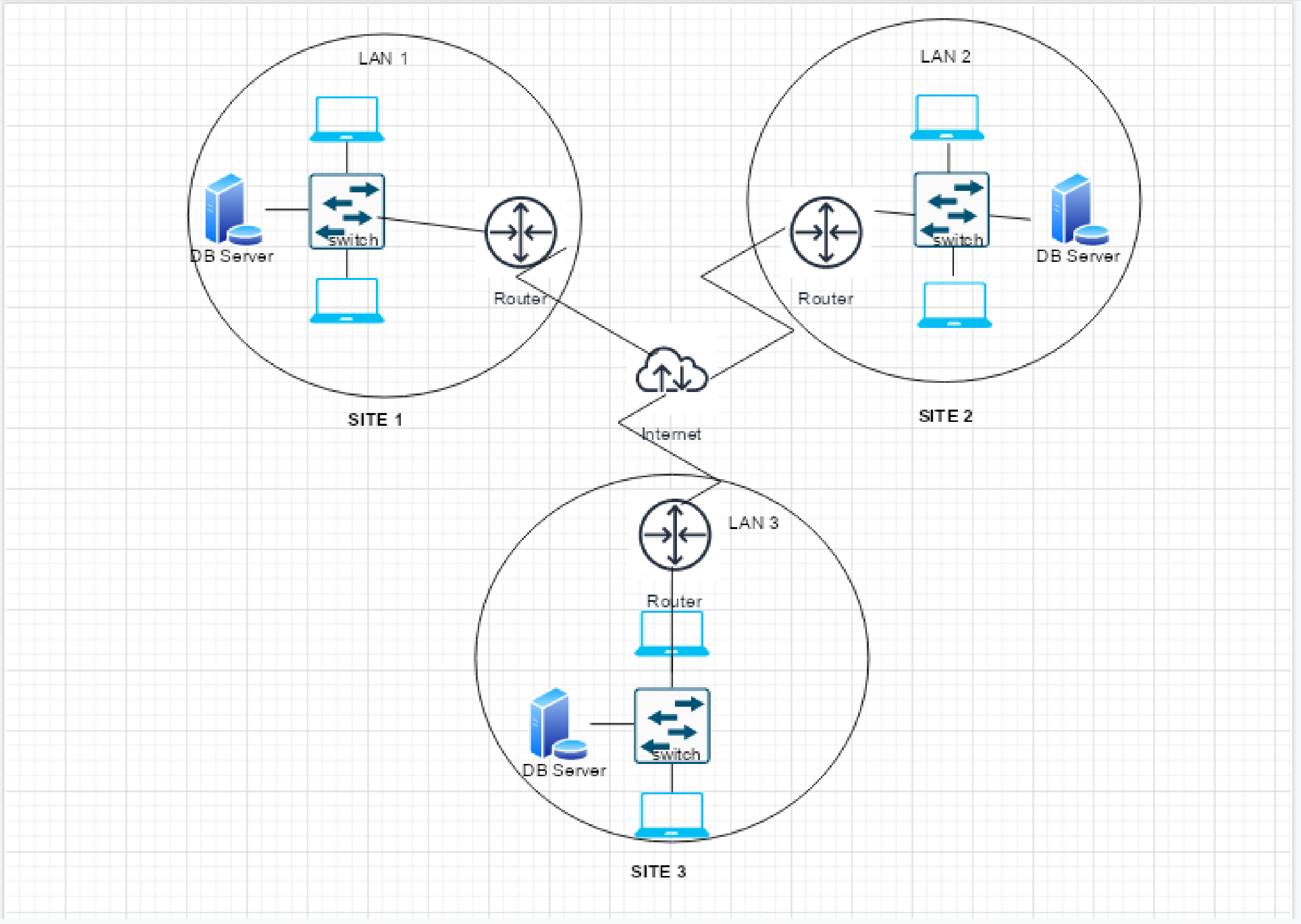
1.4.1.3 Third party software for data visualization and statistical analysis.

1.5 Rationale for the design choices.

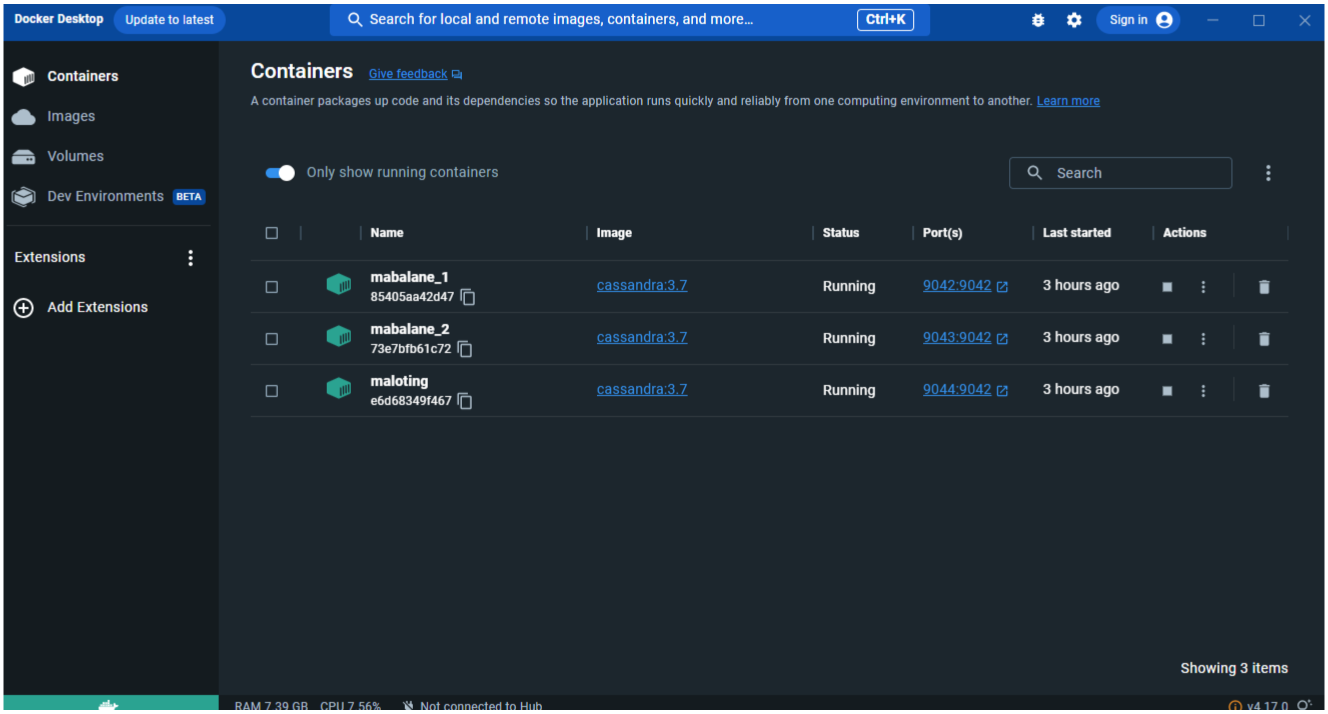
The distributed clinical trials management system is designed to be scalable, secure and user-friendly. The use of multiple database servers ensures that the system can handle large volume of participant data and study-related information. The LAN and WAN provide secure communication between the different components of the system. The web based applications make it easy for the stuff to access the system from any location, and the use of a third party software for data visualization and statistical analysis ensures that the system provides valuable insights into the study data.



2. Detailed Demonstration System Architecture Diagram



The system is distributed by district (location) considering whether or not the country is in the highlands or lowlands of Lesotho. There are 10 fragments. The data however is distributed over 3 nodes, one in the highlands(maloting) and 2 in the lowlands (mabalane_1 and mabalane_2). The reason there are two nodes in the lowlands is for load balancing since there is much population in the lowlands and one in the highlands is due to the fact that there is not much activity expected in the highlands.

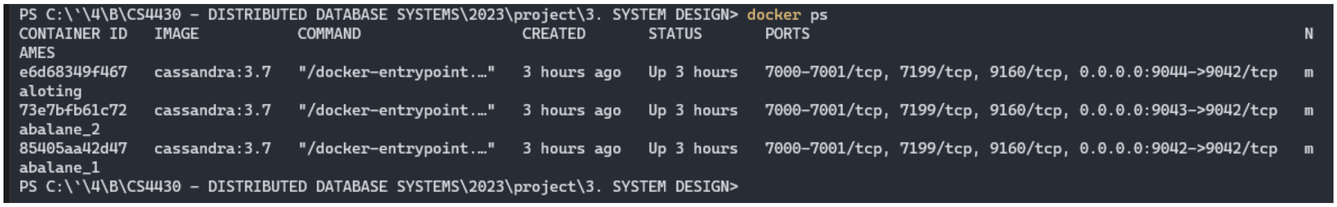


Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

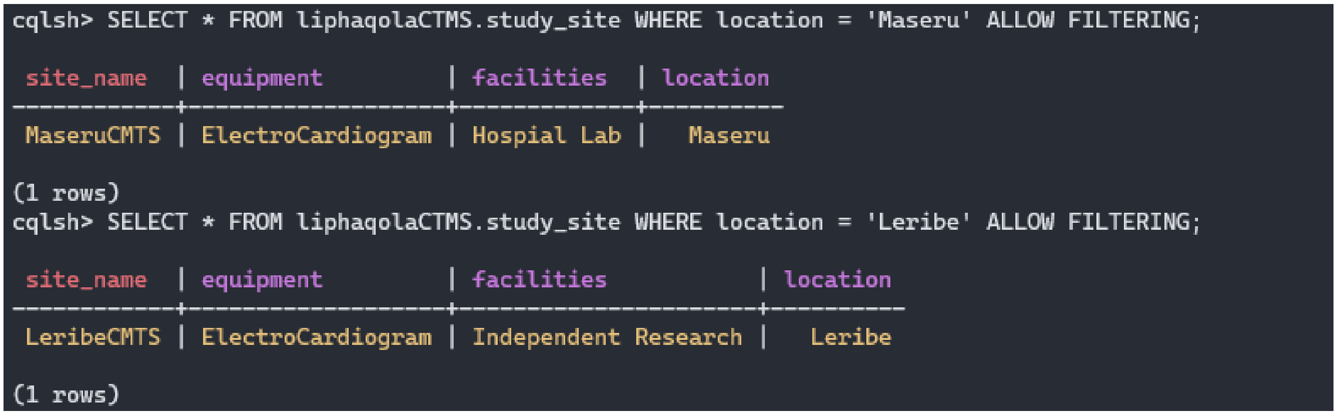
	Name	Image	Status	Port(s)	Last started	Actions
□	mabalane_1 85405aa42d47	cassandra:3.7	Running	9042:9042	3 hours ago	
□	mabalane_2 73e7bfb61c72	cassandra:3.7	Running	9043:9042	3 hours ago	
□	maloting e6d68349f467	cassandra:3.7	Running	9044:9042	3 hours ago	

Showing 3 items



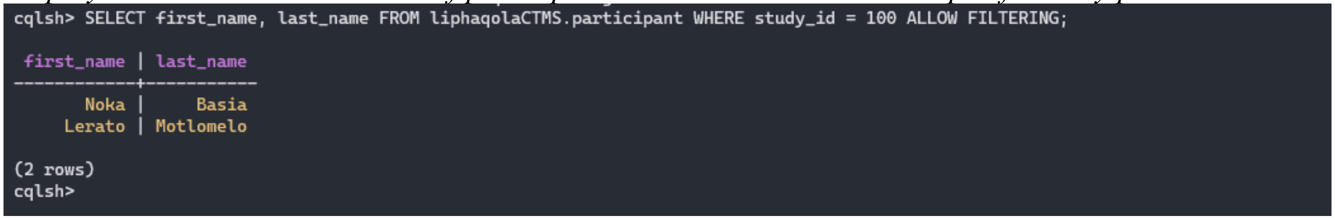
```
PS C:\`\4\B\CS4430 - DISTRIBUTED DATABASE SYSTEMS\2023\project\3. SYSTEM DESIGN> docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED       STATUS        PORTS                                                        N
AMES
e6d68349f467   cassandra:3.7   "/docker-entrypoint.…"   3 hours ago   Up 3 hours    7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9044->9042/tcp   m
aloting
73e7bfb61c72   cassandra:3.7   "/docker-entrypoint.…"   3 hours ago   Up 3 hours    7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9043->9042/tcp   m
abalane_2
85405aa42d47   cassandra:3.7   "/docker-entrypoint.…"   3 hours ago   Up 3 hours    7000-7001/tcp, 7199/tcp, 9160/tcp, 0.0.0.0:9042->9042/tcp   m
abalane_1
PS C:\`\4\B\CS4430 - DISTRIBUTED DATABASE SYSTEMS\2023\project\3. SYSTEM DESIGN>
```

Display all the information about a site in a specific district



```
cqlsh> SELECT * FROM liphaqolaCTMS.study_site WHERE location = 'Maseru' ALLOW FILTERING;

 site_name  | equipment         | facilities  | location
------------+-------------------+-------------+----------
 MaseruCMTS | ElectroCardiogram | Hospial Lab |   Maseru

(1 rows)
cqlsh> SELECT * FROM liphaqolaCTMS.study_site WHERE location = 'Leribe' ALLOW FILTERING;

 site_name  | equipment         | facilities           | location
------------+-------------------+----------------------+----------
 LeribeCMTS | ElectroCardiogram | Independent Research |   Leribe

(1 rows)
```

Display the names and surnames of participants which are involved in a specific study protocol



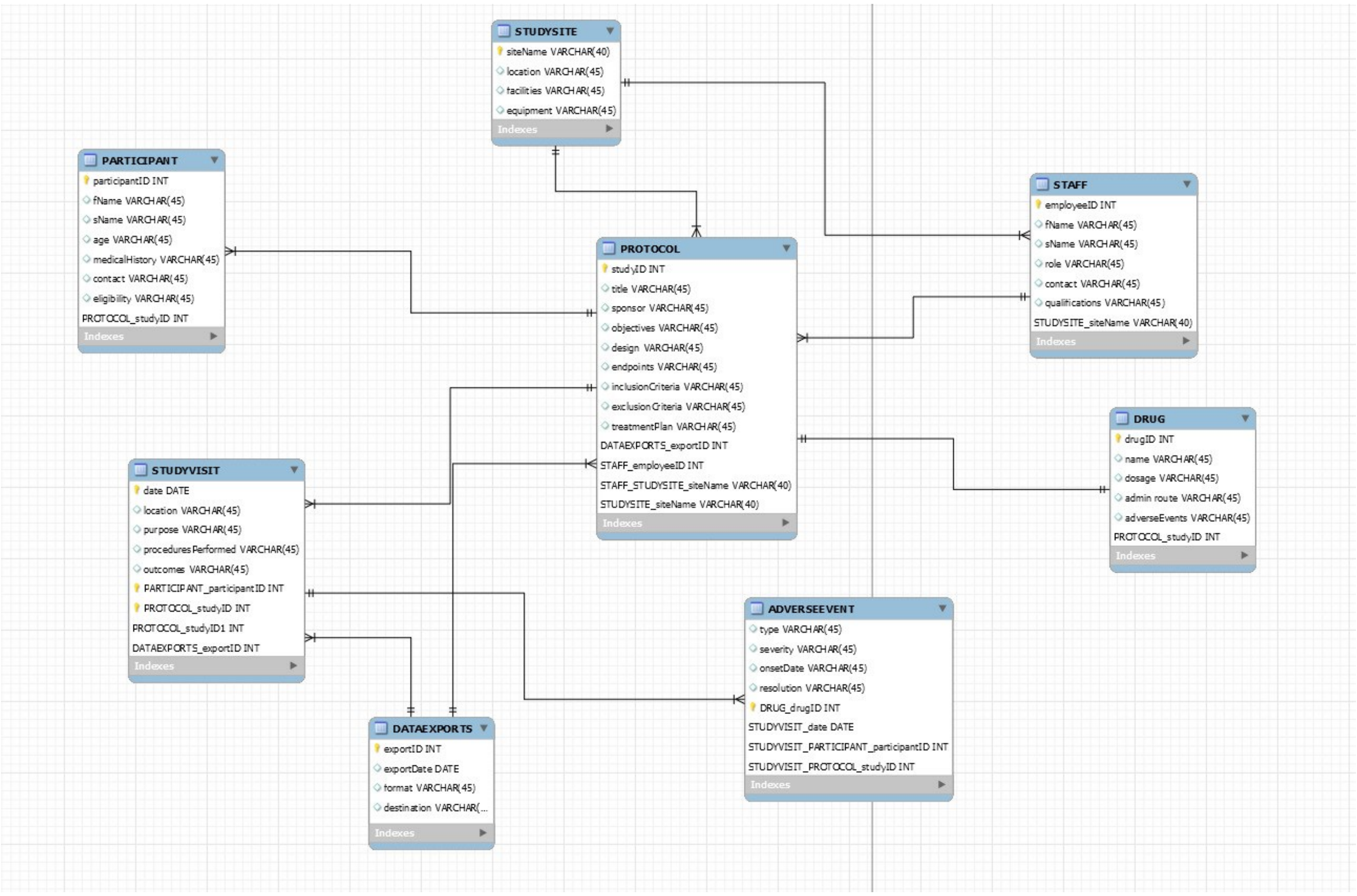
```
cqlsh> SELECT first_name, last_name FROM liphaqolaCTMS.participant WHERE study_id = 100 ALLOW FILTERING;

 first_name | last_name
------------+-----------
       Noka |     Basia
     Lerato | Motlomelo

(2 rows)
cqlsh>
```

QUERIES:

```
CREATE KEYSPACE liphaqolaCTMS WITH REPLICATION - { 'class' : 'NetworkTopologyStrategy', 'datacenter1' : 3 };

CREATE TABLE liphaqolaCTMS.protocol (study_id int, title text, sponsor text, objectives text, PRIMRY KEY (study_id));

CREATE TABLE liphaqolaCTMS.protocol (study_id int, title text, sponsor text, objectives text, PRIMARY KEY (study_id));
cqlsh> CREATE TABLE liphaqolaCTMS.protocol (study_id int, title text, sponsor text, objectives text, PRIMRY KEY (study_id));
cqlsh> CREATE TABLE liphaqolaCTMS.participant (participant_id int, first_name text, last_name text, age int, contact text,
study_id int, PRIMARY KEY (participant_id));
cqlsh> CREATE TABLE liphaqolaCTMS.drug (drug_id int, drug_name text, dosage decimal, study_id int, PRIMARY KEY (drug_id));
cqlsh> INSERT INTO liphaqolaCTMS.study_site (site_name, location, facilities, equipment) VALUES ('MokhotlongCMTS',
'Mokhotlong', 'Clinical Research', 'MRI Scanners');
cqlsh> INSERT INTO liphaqolaCTMS.study_site (site_name, location, facilities, equipment) VALUES ('QachaCMTS', 'Qacha',
'Univeristy Lab', 'X-Ray Scanners');

cqlsh> INSERT INTO liphaqolaCTMS.study_site (site_name, location, facilities, equipment) VALUES ('MaseruCMTS', 'Maseru',
'Hospial Lab', 'ElectroCardiogram');

cqlsh> INSERT INTO liphaqolaCTMS.study_site (site_name, location, facilities, equipment) VALUES ('LeribeCMTS', 'Leribe',
'Independent Research', 'ElectroCardiogram');
cqlsh> INSERT INTO liphaqolaCTMS.protocol (study_id, title, sponsor, objectives) VALUES (100, 'Nutritional Study', 'Minister of
Security', 'Dietrary Intervention');
cqlsh> INSERT INTO liphaqolaCTMS.protocol (study_id, title, sponsor, objectives) VALUES (101, 'Observational', 'Minister of
Security', 'Cross-Sectional Studies');
cqlsh> INSERT INTO liphaqolaCTMS.protocol (study_id, title, sponsor, objectives) VALUES (102, 'Medical Device', 'Minister of
Health', 'Feasibility Study');
cqlsh> INSERT INTO liphaqolaCTMS.protocol (study_id, title, sponsor, objectives) VALUES (103, 'Drug Development', 'Minister of
Health', 'Marketing Surveillance');
cqlsh> SELECT * FROM liphaqolaCTMS.protocol;

cqlsh> INSERT INTO liphaqolaCTMS.participant (participant_id, first_name, last_name, age, contact, study_id) VALUES (1000,
'Lerato', 'Motlomelo', 23, '53216540', 100);
cqlsh> INSERT INTO liphaqolaCTMS.participant (participant_id, first_name, last_name, age, contact, study_id) VALUES (1001,
'Noka', 'Basia', 28, '53219740', 100);
cqlsh> INSERT INTO liphaqolaCTMS.participant (participant_id, first_name, last_name, age, contact, study_id) VALUES (1002,
'Thabo', 'Moshe', 30, '53228440', 101);
cqlsh> INSERT INTO liphaqolaCTMS.participant (participant_id, first_name, last_name, age, contact, study_id) VALUES (1003,
'Rabasotho', 'Mosese', 37, '59221230', 102cqlsh> SELECT * FROM liphaqolaCTMS.participant;

cqlsh> INSERT INTO liphaqolaCTMS.drug (drug_id,drug_name, dosage, study_id) VALUES (9010, 'Pfizer', '2.21 ml', 100);
cqlsh> INSERT INTO liphaqolaCTMS.drug (drug_id,drug_name, dosage, study_id) VALUES (9011, 'Johnson & Johnson', '0.57 ml',
101);
cqlsh> INSERT INTO liphaqolaCTMS.drug (drug_id,drug_name, dosage, study_id) VALUES (9012, 'AbbVie', '1.34 ml', 102);
cqlsh> INSERT INTO liphaqolaCTMS.drug (drug_id,drug_name, dosage, study_id) VALUES (9013, 'Bayer', '3.00 ml', 103);
cqlsh> INSERT INTO liphaqolaCTMS.drug (drug_id,drug_name, dosage, study_id) VALUES (9014, 'Moderna', '1.00 ml', 104);
```

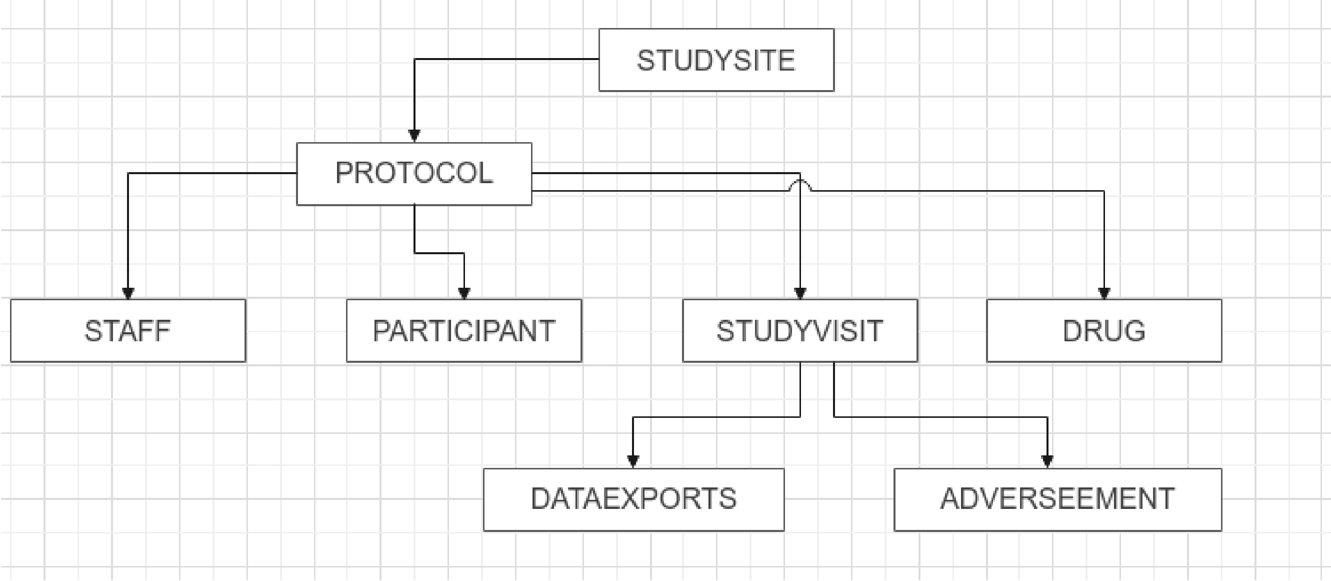
3. Detailed Distributed Database Design

- ERD for the GCS:



Data partitioning:

- Join graph for the above ERD:



From our join graph we can see that the STUDYSITE entity does not have any incoming arrow, hence we can apply Primary Horizontal fragmentation to it as follows:

Assume we fragment it according to the 10 study sites, each for each district (location) in Lesotho.

Predicates (P_i):

P_1: BID = “MSU”
P_2: BID = “MKH”
P_3: BID = “MFT”
P_4: BID = “BB”
P_5: BID = “TT”
P_6: BID = “BR”
P_7: BID = “LR”
P_8: BID = “MSH”
P_9: BID = “QUT”
P_{10}: BID = “QN”

Implications (I_i):

I_1: $P_1 \rightarrow \neg P_2 \Lambda \neg P_3 \Lambda \neg P_4 \Lambda \neg P_5 \Lambda \neg P_6 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_2: $P_2 \rightarrow \neg P_1 \Lambda \neg P_3 \Lambda \neg P_4 \Lambda \neg P_5 \Lambda \neg P_6 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_3: $P_3 \rightarrow \neg P_2 \Lambda \neg P_1 \Lambda \neg P_4 \Lambda \neg P_5 \Lambda \neg P_6 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_4: $P_4 \rightarrow \neg P \Lambda \neg P_3 \Lambda \neg P_1 \Lambda \neg P_5 \Lambda \neg P_6 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_5: $P_5 \rightarrow \neg P_2 \Lambda \neg P_3 \Lambda \neg P_4 \Lambda \neg P_1 \Lambda \neg P_6 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_6: $P_6 \rightarrow \neg P_2 \Lambda \neg P_3 \Lambda \neg P_4 \Lambda \neg P_5 \Lambda \neg P_1 \Lambda \neg P_7 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$
I_7: $P_7 \rightarrow \neg P_2 \Lambda \neg P_3 \Lambda \neg P_4 \Lambda \neg P_5 \Lambda \neg P_6 \Lambda \neg P_1 \Lambda \neg P_8 \Lambda \neg P_9 \Lambda \neg P_{10}$

I_8: $P_8 \rightarrow \neg P_2 \wedge \neg P_3 \wedge \neg P_4 \wedge \neg P_5 \wedge \neg P_6 \wedge \neg P_7 \wedge \neg P_1 \wedge \neg P_9 \wedge \neg P_{10}$
I_9: $P_9 \rightarrow \neg P_2 \wedge \neg P_3 \wedge \neg P_4 \wedge \neg P_5 \wedge \neg P_6 \wedge \neg P_7 \wedge \neg P_8 \wedge \neg P_1 \wedge \neg P_{10}$
I_{10}: $P_{10} \rightarrow \neg P_2 \wedge \neg P_3 \wedge \neg P_4 \wedge \neg P_5 \wedge \neg P_6 \wedge \neg P_7 \wedge \neg P_8 \wedge \neg P_9 \wedge \neg P_1$

	Rule 1 P_i	P_r’	P_r	F	Irrelevant in P_r’?	P_iAction	Done?
1	YES, P_1	{P_1}	{P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 ,P_9 ,P_{10} }	{f_1}	NO	-	NO
2	YES, P_2	{P_1 ,P_2}	{ P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 , P_9 ,P_{10} }	{f_1 ,f_2}	NO	-	NO
3	YES, P_3	{P_1 ,P_2 ,P_3 }	{P_4 ,P_5 ,P_6 ,P_7 ,P_8 ,P_9 ,P_{10} ,}	{f_1 ,f_2 ,f_3 }	NO	-	NO
4	YES, P_4	{P_1 ,P_2 ,P_3 ,P_4}	{P_5 ,P_6 ,P_7 ,P_8 ,P_9 ,P_{10} ,}	{f_1 ,f_2 ,f_3 ,f_4 }	NO	-	NO
5	YES, P_5	{P_1 ,P_2 ,P_3 ,P_4 , P_5}	P_6 ,P_7 ,P_8 ,P_9 ,P_{10} }	{f_1 ,f_2 ,f_3 ,f_4 , f_5}	NO	-	NO
6	YES, P_6	{P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6}	{P_7 ,P_8 ,P_9 ,P_{10} }	{f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6}	NO	-	NO
7	YES, P_7	{P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7}	{P_8 ,P_9 ,P_{10}}	{f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6}	NO	-	NO
8	YES, P_8	{P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 }	{P_9 ,P_{10} }	{f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6 ,f_7 ,f_8 }	NO	-	NO
9	YES, P_9	{P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 ,P_9 }	{P_{10} }	{f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6 ,f_7 ,f_8 ,f_9 }	NO	-	NO
10	YES, P_{10}	{P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 ,P_9 ,P_{10} }	{}	{f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6 ,f_7 ,f_8 ,f_9 ,f_{10} }	YES, P_{10}	P_r,-P_{10}, F-f_{10}	YES

Therefore, Pr’ = {P_1 ,P_2 ,P_3 ,P_4 ,P_5 ,P_6 ,P_7 ,P_8 ,P_9 }

Minterms (M):

m_1: (siteName = “MSU”)
m_2: (siteName = “MKH”)
m_3: (siteName = “MFT”)
m_4: (siteName = “BB”)
m_5: (siteName = “TT”)
m_6: (siteName = “BR”)

m_7: (siteName = “LR”)
m_8: (siteName = “MSH”)
m_9: (siteName = “QUT”)
m_{10}: (siteName = “QN”)

Primarily, the relation STUDYSITE is fragmented horizontally using M into the following fragments:

$F_{STUDYSITE}$={$STUDYSITE_1$,$STUDYSITE_2$,$STUDYSITE_3$,$STUDYSITE_4$,$STUDYSITE_5$,$STUDYSITE_6$,$STUDYSITE_7$,$STUDYSITE_8$,$STUDYSITE_9$,$STUDYSITE_{10}$}

Data allocation including replication scheme:

Since our STUDYSITE relation has been partitioned into 10 fragments, and our goal is to allocate and replicate these fragments to 3 different nodes (sites). We assumed a Round Robin approach where each fragment or set of fragments occupies a certain node for a certain period of time and then switches places with other fragments allocated on the other sites, the allocation and replication scheme would work as follows:

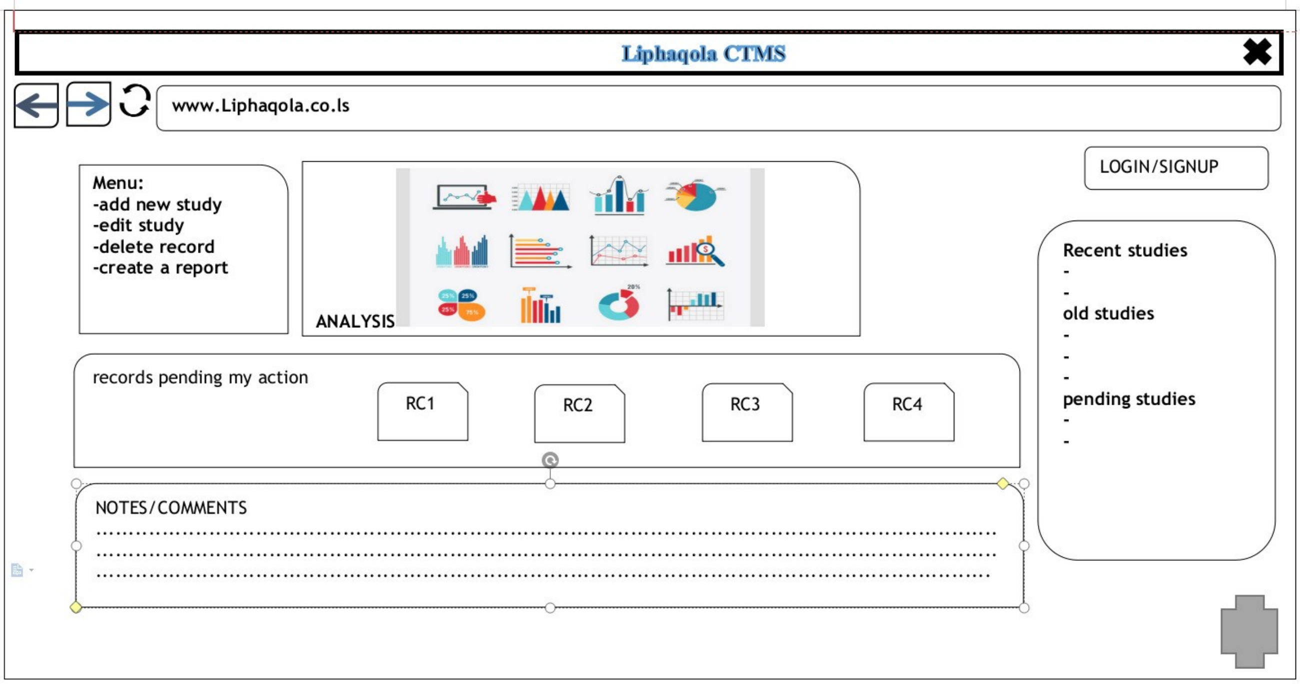
Firstly, Divide the 10 fragments into 3 disjoint sets, with 3 fragments allocated to each of the first two sites, and 4 fragments allocated to the third site.
The second part is where we assign each set of fragments to each site in a Round Robin fashion. For example, the first set could be allocated to Site A, the second to Site B, and the third to Site C. We are to set a time period during which each fragment or set of fragments will reside on a specific site before being moved to another site. The length of the time period can be determined based on the factors such as the size of the fragments, the network bandwidth between the sites, and the expected rate of updates to the fragments. Once the time period for fragment or set has elapsed, move it to the next site in the Round-Robin sequence. Say, the first set of fragments were assigned to Site A, and the time period has elapsed, move it to Site B, and move those of Site B to Site C, and those of Site C to Site A. That is repeat the process for all sets until they have all cycled through each site

Once the allocation and movement is complete, we create two replica copies of each fragment, for a total of 3 copies, to ensure fault tolerance and availability of data in case of site failures. The first replica copy is assigned to the site where the fragment is primarily allocated. The remaining copies are assigned to different sites, randomly.

Rationale for Design Choices:

Using a round-robin for allocation can help ensure that the data evenly distributed across the 3 sites, while also allowing for periodic movement on of the data help balance the load and prevent any one site from becoming overloaded. However, it may require more frequent data movement and replication, which could impact the performance and network bandwidth usage.

4. UI/UX Design



5. Detailed Test Plan

Introduction:
The purpose of this test plan is to ensure that the system architecture diagram for the distributed clinical trials management system is properly designed and implemented. The test plan will include the testing of hardware architecture, networking architecture, software architecture, and rationale for design choices.

Scope:
The scope of this test plan is to ensure that the system architecture diagram meets the following requirements:

1. The system architecture should be scalable and able to handle a large volume of participant data and study-related information.
2. The system should be secure and provide secure communication between the different components of the system.
3. The system should be user-friendly and easy to access from any location.
4. The system should provide valuable insights into the study data through data visualization and statistical analysis.

Test Approach:
The following test approach will be used to ensure that the system architecture diagram meets the requirements:

1. Testing of hardware architecture.
2. Testing of networking architecture (Local Site and Remote).
3. Testing of software architecture (Platform and Application).

4. Testing of rationale for design choices.

Test Environment:
The test environment will include the following:

1. Client devices (desktops, laptops, tablets, or smartphones).
2. Physical servers that host the CTMS software and manage the database.
3. Networking equipment such as routers, switches, and firewalls.
4. Operating system, web server, and database management system (DBMS) for hosting the different modules of the system.
5. Web-based application for the different modules of the system.
6. Third-party software for data visualization and statistical analysis.

Testing Process:

1. Testing of hardware architecture:
   a. Verify that the client layer hardware meets the recommended specifications (fast processor, sufficient memory, and high-quality display).
   b. Verify that the server layer hardware is powerful enough to handle the demands of the CTMS software and the database.
   c. Verify that the server hardware includes redundant power supplies and network connections for maximum uptime.

2. Testing of networking architecture (Local Site and Remote):
   a. Verify that the LAN is properly designed to connect the study staff computers and the servers at the local site.
   b. Verify that the WAN is properly designed to connect the local site to remote locations.
   c. Verify that the network is properly secured with firewalls and other security measures to protect against unauthorized access.

3. Testing of software architecture (Platform and Application):
   a. Verify that the operating system, web server, and DBMS are properly configured to host the different modules of the system.
   b. Verify that the web-based application for the different modules of the system is user-friendly and easy to access from any location.
   c. Verify that the third-party software for data visualization and statistical analysis is properly integrated with the system and provides valuable insights into the study data.

4. Testing of rationale for design choices:
   a. Verify that the system architecture is scalable and able to handle a large volume of participant data and study-related information.
   b. Verify that the system is secure and provides secure communication between the different components of the system.
   c. Verify that the system is user-friendly and easy to access from any location.
   d. Verify that the system provides valuable insights into the study data through data visualization and statistical analysis.

Deliverables:
The following deliverables will be produced as part of this test plan:

1. Test cases for each of the testing processes.
2. Test results for each of the testing processes.
3. Defect reports for any issues found during testing.
4. Final report summarizing the test results and providing recommendations for any necessary improvements to the system architecture.

Conclusion:

This test plan will ensure that the system architecture diagram for the distributed clinical trials management.