The National University of Lesotho

Department of Mathematics and Computer Science

Faculty of Science and technology

CS4430: Distributed Database Systems

Group Name: Datamine

Project - SYSTEM DESIGN

Due: 23 April 2023, 17:00 hrs

Student Number	Surname, initials	Year of study	Major
201902555	Mahlelebe T.I	IV	BSc MACS
201902306	Mokete K.J	IV	BSc MACS
201902292	Semethe B.C	IV	BSc MACS
201903871	Mabothile T.E	IV	BSc CS & Stats
201903584	Nkuka M.A	IV	BSc CS & Stats

1.System Architecture

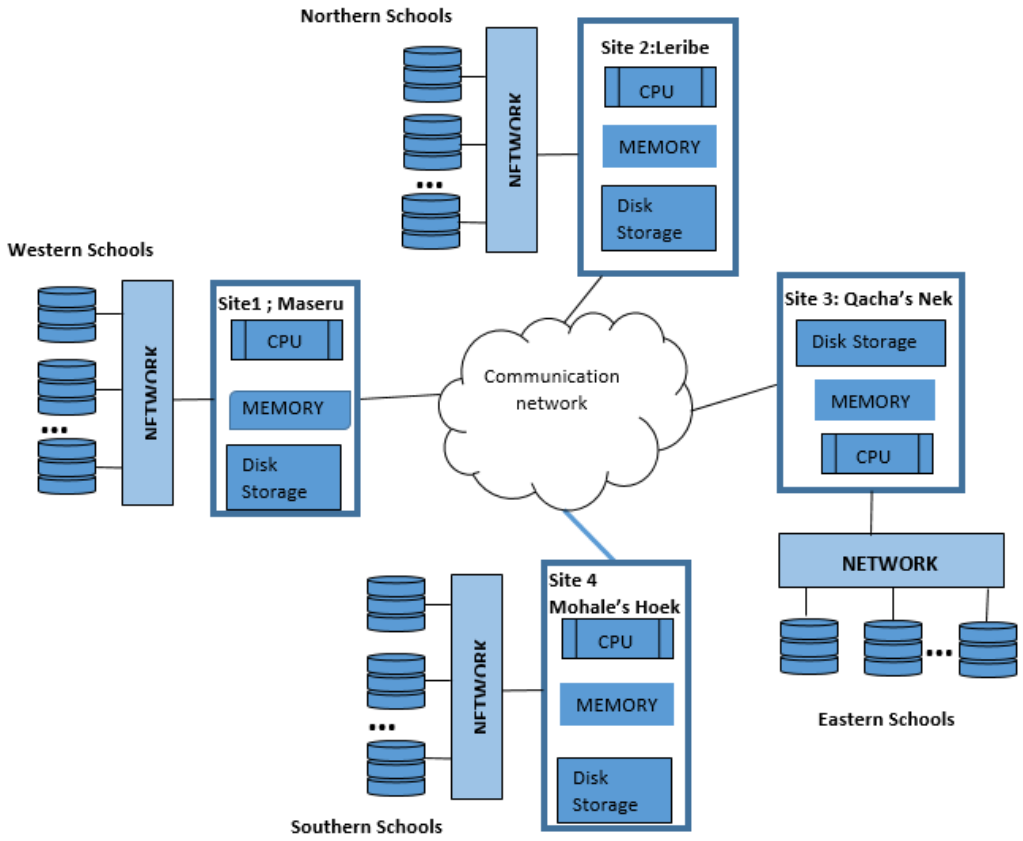


Fig 1

Figure 1 shows the Database Architecture and hardware architecture.

1.1 Database Architecture

Peer-to-Peer and Client/Server hybrid Architecture:

The Database Architecture combines a Peer-to-Peer architecture and a Client-Server Architecture, where four master nodes(which act as peers) are assigned to four regions and each of the four nodes is designated as a master node for schools in its region.

This architecture balances the high availability and fault tolerance of peer-to-peer architecture and the scalability, improved access control and security of client/server architecture.

1.2Hardware Architecture

Shared-Nothing and Shared-Memory hybrid architecture

The system has four sites , and each of the four sites operate independently meaning they do not share anything with each other, each has its own CPU, Memory and Disk storage. Within each region

the schools share memory and disk storage on a site.The shared memory architecture is fast and efficient but becomes a bottleneck as the system grows in scale as all processors would have access to the same memory, therefore the shared-nothing architecture between the sites , is introduced to reduce the bottleneck.This architecture combines the benefits of shared memory and shared nothing architectures,and provides a balance between performance, scalability and fault tolerance.

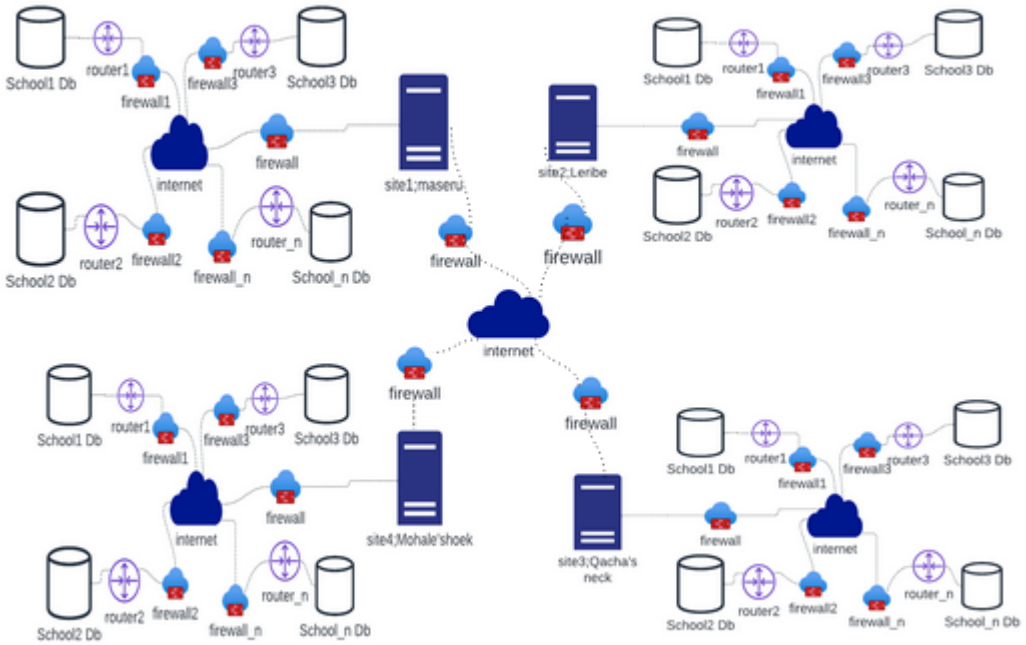


Figure 2 shows the network architecture.

1.3 Network Architecture

Each school in every region will have its own database server connected to school local network(LAN).Each school LAN will have a router connecting school LANs to the internet allowing them to communicate and share information directly in the form of a mesh network
The mesh networks in each region will be connected to the server located in that region.
Finally ,the servers in different regions will as well be connected to each other through the internet.
In the case where a server or a LAN is to be connected to the internet ,firewalls will be installed between such a server or a LAN and the internet to control traffic and enhance security.

2.Conceptual Design

DATA REQUIREMENTS
The following are anticipated to be the most important queries:

User registration Register, retrieve and edit schools, school admins, researchers/policy-makers, educators

School cooperation

- Retrieve information on successful strategies or interventions, implementation and outcomes implemented at different schools

Educational equity

- Percentage of pupils with special needs(e.g. Those with disabilities) grouped by special needs and region
- Percentage of pupils(grouped by gender) that successfully graduate high school
- Correlation changes between pupils' academic success and their socio-economic status per year
- Computes the average subject (e.g mathematics) proficiency for each school type (e.g urban ,rural) ,and groups the results by school type , allowing researchers to compare subject proficiency rates between different types of schools.

Curriculum Development

- Aggregate performance metrics by subject from each school.
- Identify the schools whose performance is poor in the particular subjects.
- Identify common weaknesses among the schools so as to develop targeted interventions for improvement.
- Compare school performance(e.g. pass rate) by resources available at schools.

Demographic research

- Aggregate number of students; group by Gender and age
- Aggregate count of students relation with certain passlevel; group by socioeconomic status

Education policy & planning

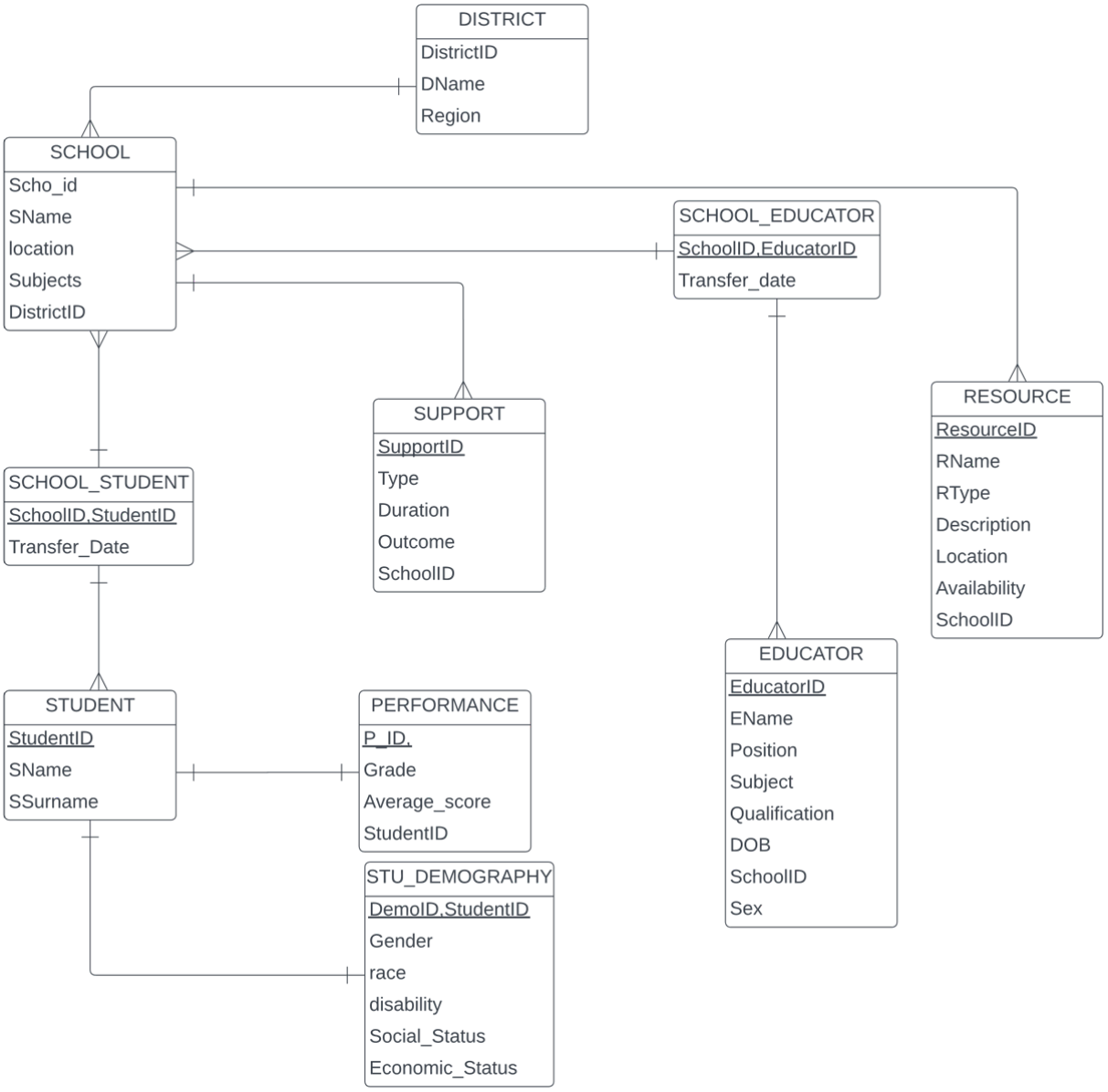
- Query the database for information about graduation rates, standard test scores and teacher qualifications, such as their level of education and teaching experience. This information would help determine how teacher quality contributes to students' performance.

2.1Global Conceptual Schema

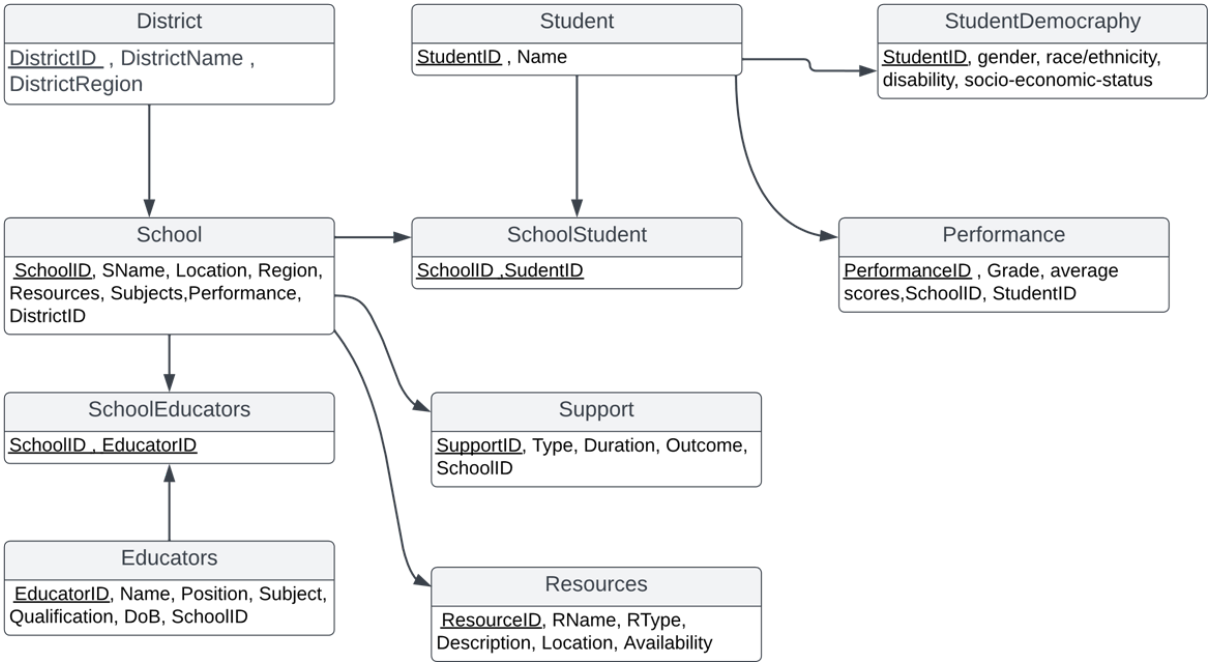
Entities
Student(StudentID , Name)
Educators(EducatorID, Name, Position, Subject, Qualification, DoB, SchoolID)

Resource(ResourceID, RName, RType, Description, Location, Availability)
School(SchoolID, SName, Location, Region, Resources, Subjects,Performance, districtID)
Student Demography(StudentID, gender, race/ethnicity, disability, socio-economic-status)
Support(SupportID, type, duration, outcome, schoolID)
Performance(PerformanceID , Grade, average scores,schoolID, studentID)
District(districtID, districtName, districtRegion)

ER-DIAGRAM



Join Graph



Fragmentation

Applying PHF on the District relation we get:

From , Simple predicates are:

P1 : Region = “North”,

P2: Region = “South”,

P3: Region = “East”,

P4: Region = “west”

Pr = {P1, P2 , P3, P4}

Resulting complete and minimal predicates are:

Pr’ = {P1, P2 , P3 }

Set of implications:

P4 = ¬P1 and ¬P2 and ¬P3

Now generating minterms:

M1 = P1

M2 = P2

M2 = P3

M2 = P4

M’ = {M1, M2 , M3 , M4} = M ; since already simplified

Therefore the resulting fragments are M1 = “North” ,M2 = ”West” , M3 = ”East” and M4 = ”South” .

Now on School we perform derived fragmentation

Firstly starting with the SCHOOL Relation:

We perform semi-join with the fragments of the primary source relation and the SCHOOL relation Formula :

SCHOOL_i = SCHOOL ⋉ DISTRICTi, 1 ≤ i ≤ 4, where District_j =σ(Fj)(District), 1 ≤ j≤ 4

SCHOOL_1 = SCHOOL ⋉ DISTRICT_1 ; DISTRICT_1 = σREGION=”WEST”(DISTRICT)

SCHOOL_2 = SCHOOL ⋉ DISTRICT_2; DISTRICT_2 = σREGION=”NORTH”(DISTRICT)

SCHOOL_3 = SCHOOL ⋉ DISTRICT_3; DISTRICT_3 = σREGION=”EAST”(DISTRICT)

SCHOOL_4 = SCHOOL ⋉ DISTRICT_4; DISTRICT_4 = σREGION=”NORTH”(DISTRICT)

Now for the RESOURCE Relation:

RESOURCE_1 = RESOURCE ⋉ SCHOOL_i, 1 ≤ i ≤ 4 , where SCHOOLi =σ(Fj)(SCHOOL), 1 ≤ j≤4

RESOURCE_1 = RESOURCE ⋉ SCHOOL_1

RESOURCE_2 = RESOURCE ⋉ SCHOOL_2

RESOURCE_3 = RESOURCE ⋉ SCHOOL_3

RESOURCE_4 = RESOURCE ⋉ SCHOOL_4

Now for the SUPPORT Relation:

SUPPORT_1 = SUPPORT ⋉ SCHOOL_i, 1 ≤ i ≤ 4 , where SCHOOLi =σ(Fj)(SCHOOL), 1 ≤ j≤4

SUPPORT_1 = SUPPORT ⋉ SCHOOL_1

SUPPORT_2 = SUPPORT ⋉ SCHOOL_2

SUPPORT_3 = SUPPORT ⋉ SCHOOL_3

SUPPORT_4 = SUPPORT ⋉ SCHOOL_4

The STUDENT Relation fragments:

STUDENT_1 = STUDENT ⋉ SCHOOL_i, 1 ≤ i ≤ 4 , where SCHOOLi =σ(Fj)(SCHOOL), 1 ≤ j≤4

STUDENT_1 = STUDENT ⋉ SCHOOL_1

STUDENT_2 = STUDENT ⋉ SCHOOL_2

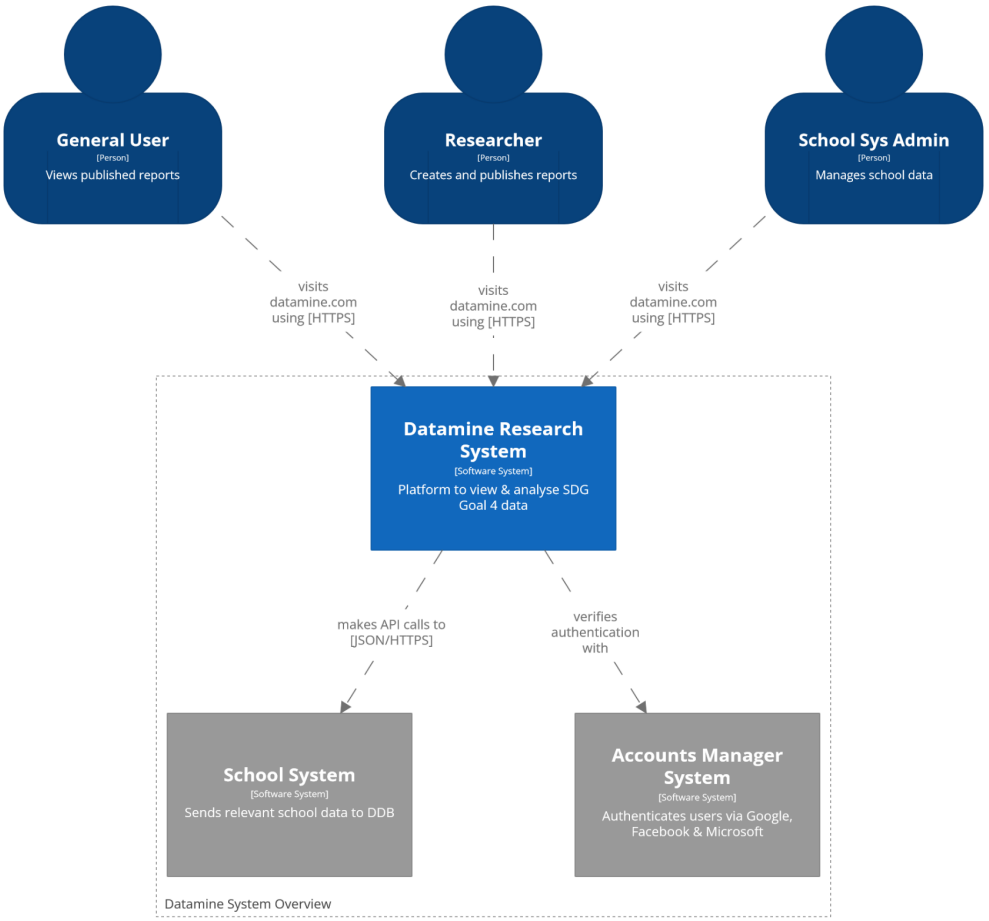
STUDENT_3 =STUDENT⋉ SCHOOL_3

STUDENT_4 = STUDENT ⋉ SCHOOL_4

SOFTWARE ARCHITECTURE USING C4 TEMPLATE

Level I - Software Context Diagram

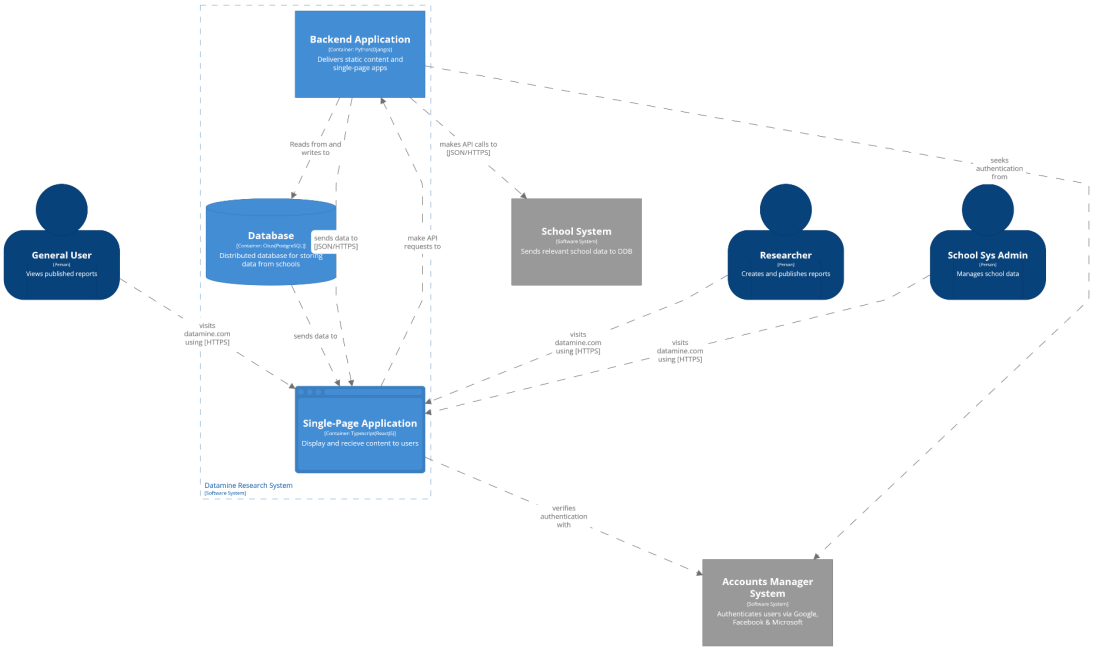
This level provides a broad overview of the Datamine Research System, focusing on the core system, which is the Datamine Software System, as well as the actors and external systems involved. To simplify user authentication, the Accounts Manager system will serve as a third-party application, enabling users to log in using their Microsoft, Google, Facebook, or other accounts without requiring new credentials. Additionally, although only a small percentage of schools maintain databases, the School System represents such systems and can be leveraged to obtain pertinent data for the distributed database.



[System Landscape]
Sunday, 23 April 2023 at 15:53 South Africa Standard Time

Level II - System Container Diagrams

Zooming in into the system software, we have 3 containers: single-page frontend application, distributed database and the backend application. These will be implemented using ReactJS, Citus PostgreSQL & Django respectively. The backend includes an API application written in Django REST framework to manage data requests from and to the single-page application.

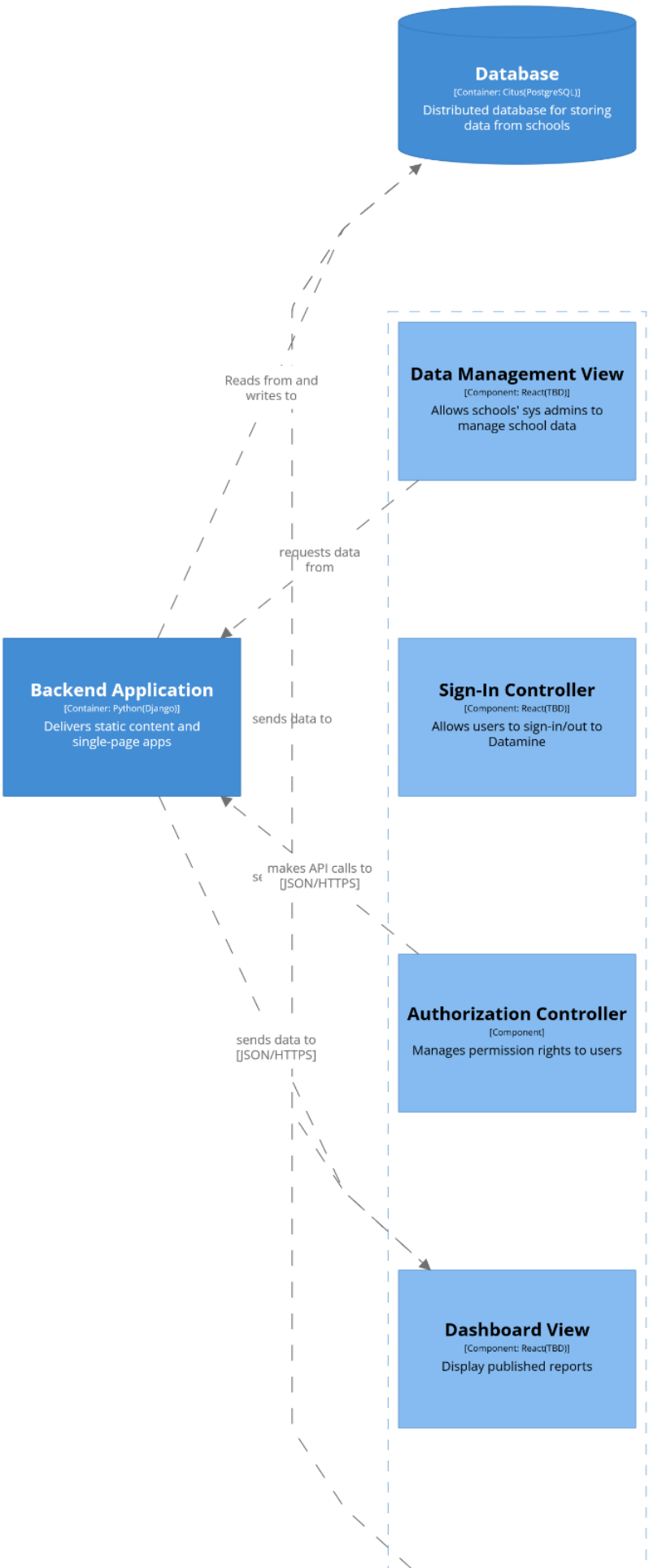


[Container] Datamine Research System

Sunday, 23 April 2023 at 15:55 South Africa Standard Time

Level III - Software Component Diagrams

Focusing on the software system, the Datamine Research System comprises three distinct containers: a single-page frontend application, a distributed database, and a backend application. The ReactJS framework will be utilized for developing the single-page frontend application, while Citus PostgreSQL and Django will be used for the distributed database and backend application, respectively. To manage the communication of data requests between the single-page application and backend, a Django REST framework API application will be incorporated into the system's backend.


Database
[Container: Citus(PostgreSQL)]
Distributed database for storing data from schools
Reads from and writes to
Data Management View
[Component: React(TBD)]
Allows schools' sys admins to manage school data
requests data from
Backend Application
[Container: Python(Django)]
Delivers static content and single-page apps
sends data to
Sign-In Controller
[Component: React(TBD)]
Allows users to sign-in/out to Datamine
makes API calls to
[JSON/HTTPS]
Authorization Controller
[Component]
Manages permission rights to users
sends data to
[JSON/HTTPS]
Dashboard View
[Component: React(TBD)]
Display published reports