UNIVERSITY HEALTHCARE DATABASE SYSTEM

PROJECT REPORT

Group 14

Aarushi Sharma

Aniket Sakharkar

857-869-7912 (Aarushi Sharma)

857-230-5126 (Aniket Sakharkar)

sharma.aaru@northeastern.edu

sakharkar.a@northeastern.edu

Percentage of Effort Contributed by Aarushi: 50%

Percentage of Effort Contributed by Aniket: 50%

Signature of Aarushi: *Aarushi*

Signature of Aniket: *Aniket*

Submission Date: 10th December 2022

EXECUTIVE SUMMARY

When a student walks in, a doctor may not have the time to go through mountains of data. As a result, the information must be queried, filtered, and retrieved using the appropriate technology. Universities across the United States are planning to create a database system to address the problem with the respective university's healthcare database, where students must fill out the student health history form, and the data is stored not only in the university's record but also the hospital based on the zip code that each student enters. As needed, students can seek medical attention and receive the necessary vaccinations based on their previous records. If a student needs the assistance of medical professionals from other facilities, a database system enables information interchange to be in place. With this university healthcare database system, the centre will be able to retrieve easily, update, and report patient data, assisting medical professionals and doctors in providing prompt, accurate diagnoses.

1. The US Healthcare Department manages the database for all the universities in the United States. 2. Officials will need to log in with their respective university login credentials to access their University's Healthcare Database in addition to the details of the nearby healthcare service providers, depending on their location and zip code.

3. Universities can have access to healthcare service providers' details from the University Healthcare Database based on their location and zip code.

4. All the healthcare service providers within the location of the University will have access to those universities' databases to provide facilities when needed.

5. Each student will fill the forms with their respective University's student login credentials, which will then be stored in the database system.

6. With the help of the details provided by the students, vaccination providers can also set up how many vaccinations would be needed per university in the upcoming days/months.

7. A student can fill out multiple forms to report any condition or update in their medical history, which will be updated in the database system against their previous instance. 8. Health care providers who are no longer affiliated with the organization and whom the students may have named on the medical form will not have access to the database, demonstrating the various levels of security that this database system will have.

INTRODUCTION

The need for well-organized, well-maintained, and user-friendly databases in the healthcare industry is greater than ever due to the quick development of technologies utilized in the industry and changes in healthcare operations following COVID-19. The healthcare database system helps us the address the following issues:

• Affordability and timeliness of care

• Barriers to seeking healthcare

• How appropriate or necessary are the services

• Whether the healthcare facilities are adhering to best practices

• Availability of patient information to clinicians and other institutions

• Whether patient information is accessible to physicians and other institutions

• Whether the services are appropriate for the recognized healthcare issues

• The application of innovation in healthcare delivery, including financing medical facilities and other issues A manual system for managing healthcare records cannot support the expansion of the healthcare industry, the rising demand for healthcare services, or the requirements of the healthcare regulatory framework. Large amounts of records can be managed manually, however doing so is costly both administratively and monetarily for the following reasons:

1. Inadequate data and information security

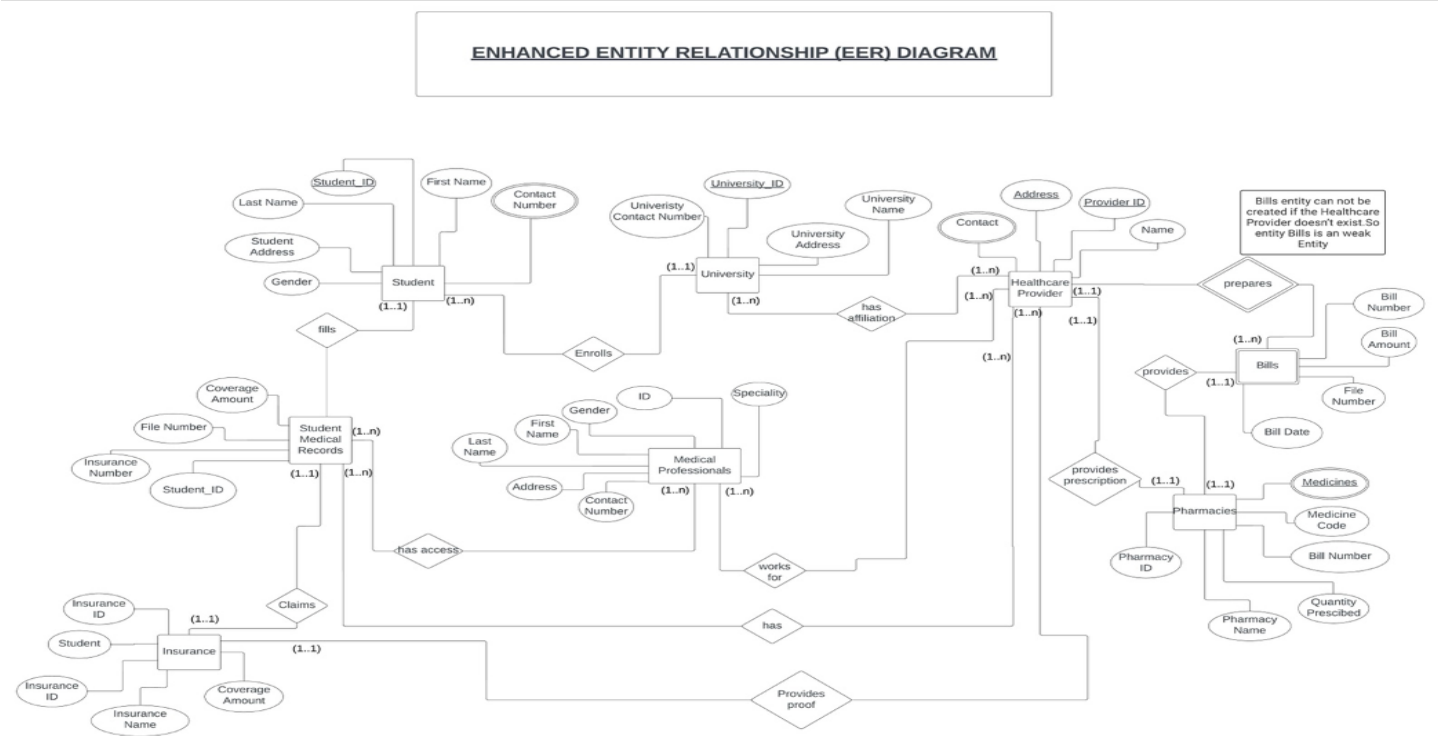
2. Data and information about customers could be subject to illegal change.

3. A manual record management system is expensive to maintain.

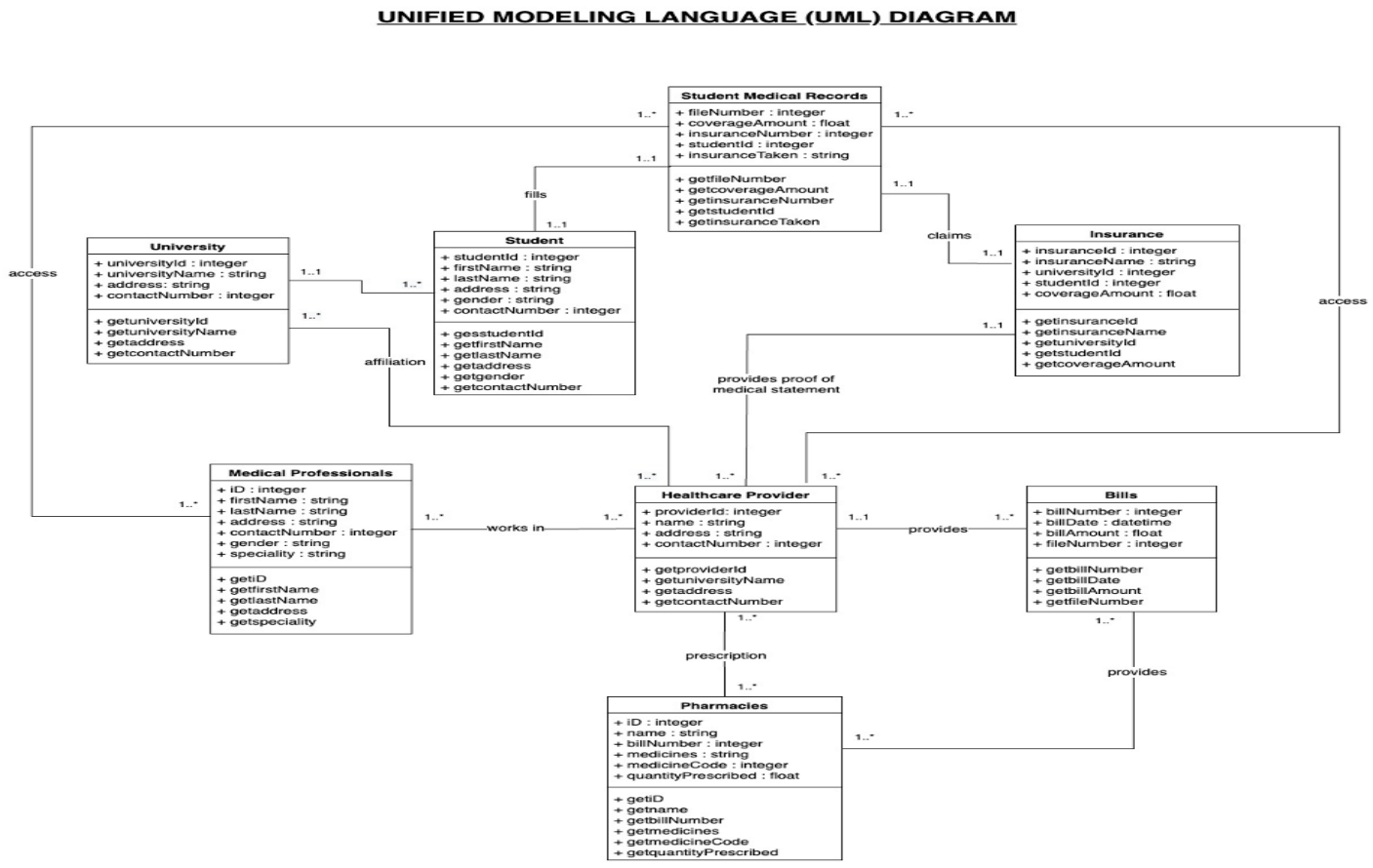
4. Inadequate data access control Our fundamental objective is to promote open science, responsible data exchange with universities, and open research. Its secondary goals are to advance research and public health as a whole and to improve the health of young people in the USA.

CONCEPTUAL DATA MODELING

ENHANCED ENTITY RELATIONSHIP (EER) MODEL



UNIFIED MODELING LANGUAGE (UML) CLASS DIAGRAM



MAPPING CONCEPTUAL MODEL TO RELATIONAL (LOGICAL) MODEL

- **PK** (Bold and Underlined) **–** Primary Key
- ***FK*** (Bold and Italics) – Foreign Key

1. Student (**Student_ID,** First Name, Last Name, Gender, ***University_ID***)
   a. Primary Keys: - Student_ID (NULL – not allowed)
   b. Foreign Keys: - University_ID

2. Student_Contact_Number (**Student_ID, Contact Number**)
   Contact Number – multi-valued attribute -- decomposed into another relation
   a. Primary Keys: - Student_ID, Contact Number (NULL – not allowed)
   b. Foreign keys:- Student_ID

3. Student_Address (**Student_ID, Address)**
   Student_Address – multi-valued attribute -- decomposed into another relation
   a. Primary Keys: - Student_ID, Address (NULL – not allowed)
   b. Foreign keys: - Student_ID

4. University (**University_ID**, University Name, University Address, University Contact Number, ***Provider_ID***)
   a. Primary Keys: - University_ID (NULL – not allowed)
   b. Foreign Keys: - Provider_ID

5. Medical Professionals (**MP_ID**, Speciality, Gender, First Name ,Last Name ,Address, Contact Number, ***File_Number, Provider_ID*)**
   a. Primary Keys: - MP_ID (NULL – not allowed)
   b. Foreign Keys: - Provider_ID,File_Number

6. Insurance (**Insurance_ID**, Insurance Name, Coverage Amount)
   a. Primary Keys: - Insurance_ID (NULL – not allowed)

7. Student Medical Records (**File_Number**, Insurance Number, Coverage Amount ***MP_ID, Provider_ID, Student_ID)***
   a. Primary Keys: - File_Number (NULL – not allowed)
   b. Foreign Keys: - MP_ID, Provider_ID, Student_ID

8. Healthcare Provider (**Provider_ID**, Healthcare_Provider_Address, Healthcare_Provider_Name, ***Insurance_ID, File_Number, MP_ID, University_ID***)
   a. Primary Keys: - Provider_ID (NULL – not allowed)
   b. Foreign Keys: - Insurance_ID, File_Number,MP_ID, University_ID

9. Healthcare Provider_Contact (**Provider_ID**, **Healthcare_Provider_Contact**)
   Healthcare_Provider_Contact – multi-valued attribute -- decomposed into another relation
   a. Primary Keys: - Provider_ID, Healthcare_Provider_Contact (NULL – not allowed)
   b. Foreign Keys: - Provider_ID

10. Bills (**Bill_Number**, Bill_Amount, Bill_File_Number, ***Provider_ID)***
    a. Primary Keys: - Bill_Number (NULL – not allowed)
    b. Foreign Keys: - Provider_ID

11. Pharmacies (**Pharmacy_ID**, Pharmacy_Name, Quantity_Prescribed, Medicine_Code, ***Bill_Number)***
    a. Primary Keys: - Pharmacy_ID (NULL – not allowed)
    b. Foreign Keys: - Bill_Number

12. Pharmacies_Medicines (**Pharmacy_ID**, **Medicines**)
    Medicines – multi-valued attribute -- decomposed into another relation
    a. Primary Keys: - Pharmacy_ID, Medicines (NULL – not allowed)
       Foreign Keys: - Pharmacy_ID

IMPLEMENTATION IN MYSQL

The SQL Database has been created on MySQL Workbench using CREATE Statements for each of the 10 tables: Student, University, StudentMedicalRecords, HealthcareProvider, MedicalProfessionals, Insurance, Bills, and Pharmacies.

1. Find the Insurance Name, Associate Healthcare Provider and the Coverage Amount offered.

```
SELECT I.InsuranceName, HP.ProviderID, I.CoverageAmount
FROM Insurance I
JOIN HealthcareProvider HP
ON I.InsuranceID = HP.InsuranceID
ORDER BY CoverageAmount;
```

InsuranceName	ProviderID	CoverageAmount
Howe-Roob	68304	0.00
Greenholt, Waters and Lemke	4	0.00
Rodriguez, Smitham and Auer	33411	0.00
Ortiz Inc	4307497	0.00
Stanton, Bogisich and Harber	64	0.00
Torphy-Kunde	377432	0.00
Mante PLC	4094918	0.26
Hegmann Group	6	0.65
O'Keefe Ltd	6688	1.12
Langosh Inc	56011	1.95
Bednar Group	9999999	2.00
Schulist, O'Kon and Gutmann	875422	2.00
Ratke-Krajcik	26082	2.24
Jenkins Group	1	2.68
Kemmer-Klein	6198410	3.05
Cronin-Witting	95301	4.67
Gerlach Inc	581	5.71

2. Find the StudentID, First Name, Last Name of the Student with File Number '1721255882' with the coverage amount that the Student's Insurance is covering.

```
SELECT S.StudentID, S.FirstName, S.LastName ,TEMP.CoverageAmount, TEMP.FileNumber
FROM Student S
JOIN
(
SELECT SM.StudentID, SM.CoverageAmount, SM.FileNumber
FROM StudentMedicalRecords SM
WHERE EXISTS
(
SELECT * FROM Bills B
WHERE SM.FileNumber = B.FileNumber AND  SM.FileNumber = '1721255882'
)
) TEMP
ON S.StudentID = TEMP.StudentID;
```

StudentID	FirstName	LastName	CoverageAmount	FileNumber
20	Will	Howell	447331.90	1721255882

3. **Find the Insurance Companies whose available Coverage Amount is greater than the Average Coverage Amount offered by all the Insurance Companies in the database.**

```
SELECT InsuranceName, CoverageAmount FROM Insurance
WHERE CoverageAmount > (
SELECT AVG(CoverageAmount) FROM Insurance
);
```

InsuranceName	CoverageAmount
Altenwerth-Bahringer	55512188.31
McLaughlin, Schroeder and Stamm	35482510.58
Hoppe Group	676427282.30
Durgan, Langosh and Anderson	117570728.98
Hessel PLC	767995162.65
Smitham-Kohler	217238457.41
Gusikowski-Kub	41532051.31
Braun, Bradtke and Hills	110907530.61
McClure-Abshire	94754809.08
Beatty-Bartoletti	504441672.36
Gibson, Kessler and Tillman	71621830.51
O'Hara-Volkman	268015125.40
Skiles, Schneider and Bailey	241761079.98

4. **Find the Healthcare Providers, its name and the ZipCode it is located in**

```
SELECT R1.ProviderID , R1.Name , R1.Zipcode
FROM HealthcareProvider R1
WHERE R1.ProviderID >= ALL
(SELECT R2.ProviderID
FROM HealthcareProvider R2
WHERE R2.ZipCode = R1.ZipCode);
```

ProviderID	Name	Zipcode
0	Pagac, Barton and Ernser	81601
1	Schuppe PLC	20315
2	Corkery Inc	74772
3	Emmerich PLC	18941
4	Casper-Green	32406
5	Senger Group	33158
6	Hudson-Marvin	10250
7	Gutmann, Jacobson and Zemlak	34115
9	Carter PLC	84647
12	Olson, Durgan and Stoltenberg	33604
14	Gusikowski-Schimmel	12157
31	Casper-Ebert	36127
40	Bergnaum, Bosco and Reichel	91470
41	Maggio PLC	46851

5. **Find the Universities which have more than Average Coverage Amount for Insurance.**

```
SELECT U.universityName, AVG(SMR.corageamount)
FROM student S
JOIN University U
ON S.universityID = U.universityID
JOIN StudentMedicalRecords SMR
ON S.studentID = SMR.studentID
GROUP BY U.universityID
HAVING AVG(SMR.coverageamount) > 50000;
```

universityName	AVG(SMR.coverageamou...
Kertzmann, Kuhic and Torp	15433189.917000
Williamson, Pagac and Carter	12161181.842500
Swaniawski-Kuhic	41596484.728889
Gusikowski LLC	2389864.543636
Dooley-Blanda	2513403.912500
Kertzmann-Bosco	99020951.835714
Goodwin, Pfeffer and Breitenberg	59143037.445000
Windler, Herman and Lind	3319428.388889

IMPLEMENTATION IN NOSQL

- db.Insurance.find().limit(2)

```
> db.Insurance.find().limit(2)
< { _id: ObjectId("6387df3ca901576152ab986a"),
    InsuranceID: 1061087213,
    InsuranceName: 'Skiles, Pacocha and Kilback',
    CoverageAmount: Decimal128("5118.82") }
  { _id: ObjectId("6387df3ca901576152ab986b"),
    InsuranceID: 1148115265,
    InsuranceName: 'Little-Hegmann',
    CoverageAmount: Decimal128("1978312.00") }
```

- db.MedicalProfessionals.findOne({ Speciality : "illo" })

```
>_MONGOSH
> db.MedicalProfessionals.findOne({ Speciality : "illo" })
< { _id: ObjectId("6387df9fa901576152ab98cf"),
    MPID: 284758,
    FirstName: 'Alek',
    LastName: 'Hilll',
    Speciality: 'illo',
    Gender: 'Other',
    Address: '2169 Antonina Point\nNorth Jesse, MO 64728',
    ZipCode: 30465,
    ContactNumber: 3576207775,
    FileNumber: 8188431018,
    ProviderID: 3 }
```

- db.HealthcareProvider.find().skip(4)

```
>_MONGOSH
> db.HealthcareProvider.find().skip(4)
< { _id: ObjectId("6387defca901576152ab981d"),
    ProviderID: 4,
    Name: 'Casper-Green',
    Address: '92880 Eleanore Wall\nPort Henry, MN 61988-4171',
    ZipCode: 32406,
    ContactNumber: 6018677880,
    InsuranceID: 1378781016 }
  { _id: ObjectId("6387defca901576152ab981e"),
    ProviderID: 5,
    Name: 'Senger Group',
    Address: '088 Parker Cliff Apt. 114\nWest Pamela, UT 66736-4085',
    ZipCode: 33158,
    ContactNumber: 5630687746,
    InsuranceID: 4725468334 }
  { _id: ObjectId("6387defca901576152ab981f"),
    ProviderID: 6,
    Name: 'Hudson-Marvin',
    Address: '661 White Mountain Apt. 880\nAbernathymouth, MT 92837',
    ZipCode: 10250,
    ContactNumber: 7171490468,
    InsuranceID: 5901269168 }
  { _id: ObjectId("6387defca901576152ab9820"),
    ProviderID: 7,
```

- db.University.aggregate([{"$group" : {_id:"$UniversityID", count:{$sum:1}}}])

```
>_MONGOSH
> db.University.aggregate([{"$group" : {_id:"$UniversityName", count:{$sum:1}}}])
< { _id: 'Dooley-Blanda', count: 1 }
  { _id: 'Swaniawski-Kuhic', count: 1 }
  { _id: 'Goodwin, Pfeffer and Breitenberg', count: 1 }
  { _id: 'Williamson, Pagac and Carter', count: 1 }
  { _id: 'Windler, Herman and Lind', count: 1 }
  { _id: 'O\'Reilly, Blanda and Bailey', count: 1 }
  { _id: 'Gusikowski LLC', count: 1 }
  { _id: 'Kertzmann, Kuhic and Torp', count: 1 }
  { _id: 'Schultz LLC', count: 1 }
  { _id: 'Corkery, Stracke and Ruecker', count: 1 }
  { _id: 'Kertzmann-Bosco', count: 1 }
```

- db.Insurance.aggregate({ $sort : { CoverageAmount : 1 }})

```
>_MONGOSH
> db.Insurance.aggregate({ $sort : { CoverageAmount : 1 }})
< { _id: ObjectId("6387df3ca901576152ab986f"),
    InsuranceID: 1378781016,
    InsuranceName: 'Greenholt, Waters and Lemke',
    CoverageAmount: Decimal128("0.00") }
  { _id: ObjectId("6387df3ca901576152ab9875"),
    InsuranceID: 1751735176,
    InsuranceName: 'Ortiz Inc',
    CoverageAmount: Decimal128("0.00") }
  { _id: ObjectId("6387df3ca901576152ab9893"),
    InsuranceID: 5094051483,
    InsuranceName: 'Stanton, Bogisich and Harber',
    CoverageAmount: Decimal128("0.00") }
  { _id: ObjectId("6387df3ca901576152ab98a0"),
    InsuranceID: 6351368730,
    InsuranceName: 'Rodriguez, Smitham and Auer',
    CoverageAmount: Decimal128("0.00") }
  { _id: ObjectId("6387df3ca901576152ab98a2"),
    InsuranceID: 6620353538,
    InsuranceName: 'Bernier, Strosin and Osinski',
    CoverageAmount: Decimal128("0.00") }
  { _id: ObjectId("6387df3ca901576152ab98a6"),
    InsuranceID: 6730984949,
    InsuranceName: 'Hudson, Prohaska and Schinner',
    CoverageAmount: Decimal128("0.00") }
```

PYTHON APPLICATION

```
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
```

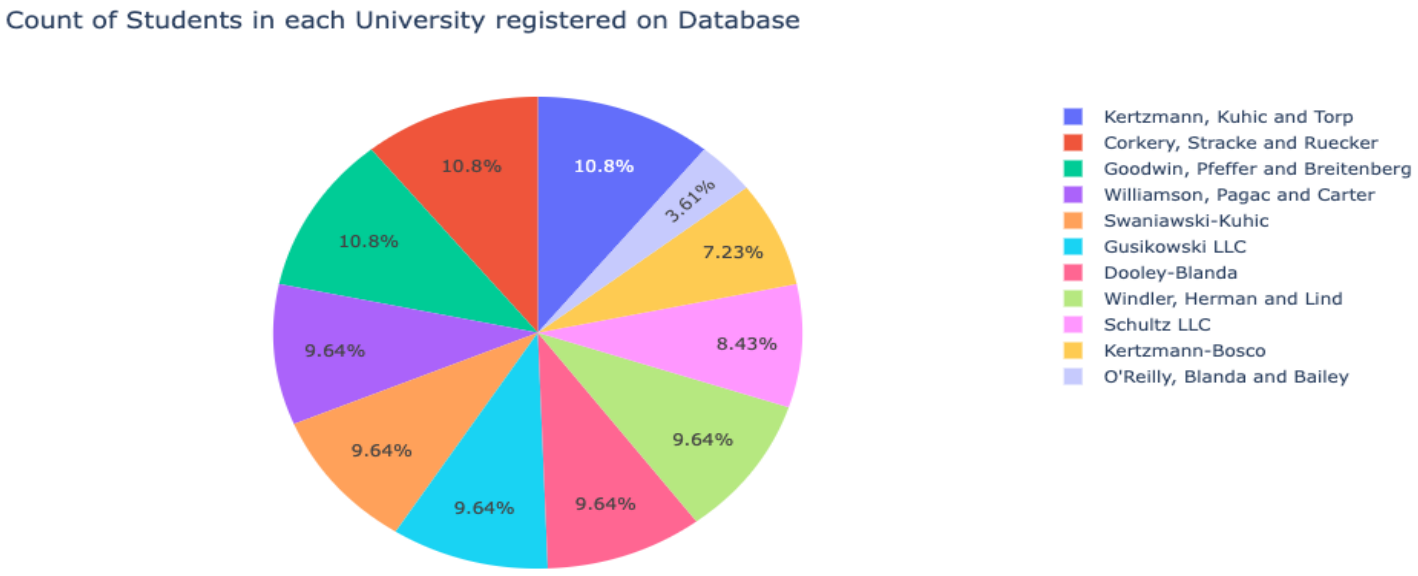
Connecting to the database created on MySQL Workbench

```
mydb = mysql.connector.connect(host = "localhost", user = "root",
passwd = "root", database = "university_healthcare_db")
mycursor = mydb.cursor(buffered = True)
```

Exploratory Data Analysis using Visualizations and executing the SQL Query through Python Application

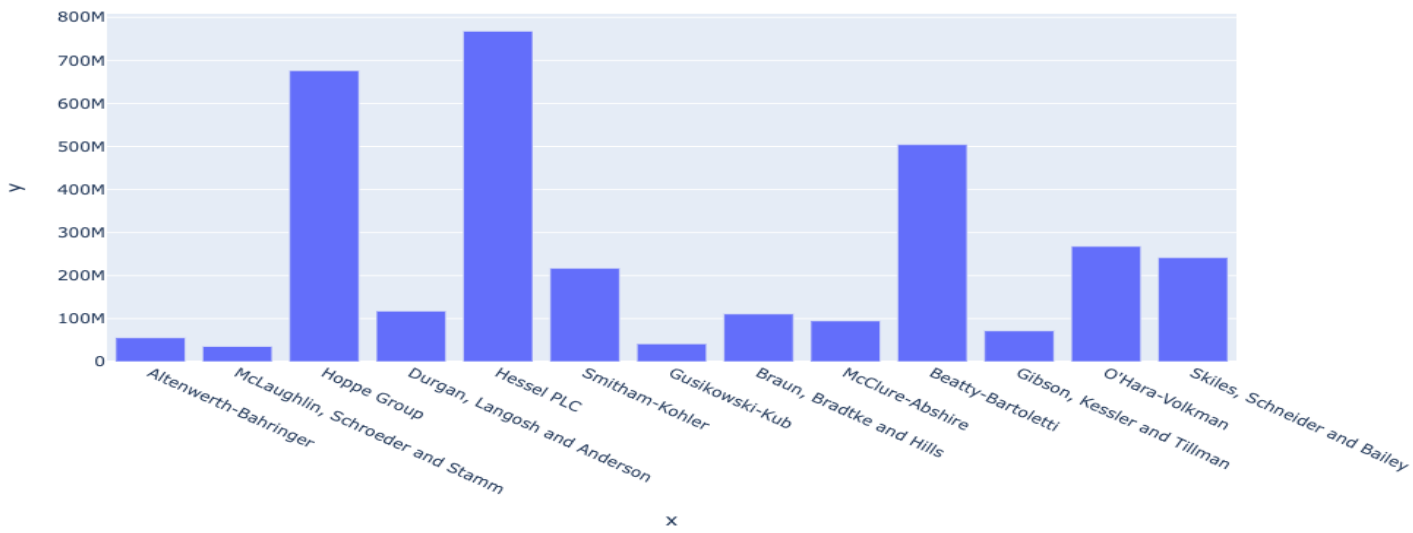
```
mycursor.execute(
    '''
    select count(S.studentID),
    U.universityName from student S join University U on
    S.universityID = U.universityID group by U.universityID
    ''')
result = mycursor.fetchall
count_id = []
uniname = []
for i in mycursor:
    count_id.append(i[0])
    uniname.append(i[1])
fig = px.pie(mycursor, values = count_id, names = uniname, title =
'Count of Students in each University registered on Database')
fig.show()
```

The below graph is a representation of the number of Students in each University which has been registered on the Database. When you hover over the area, we can see the exact number of students and the corresponding University's name.

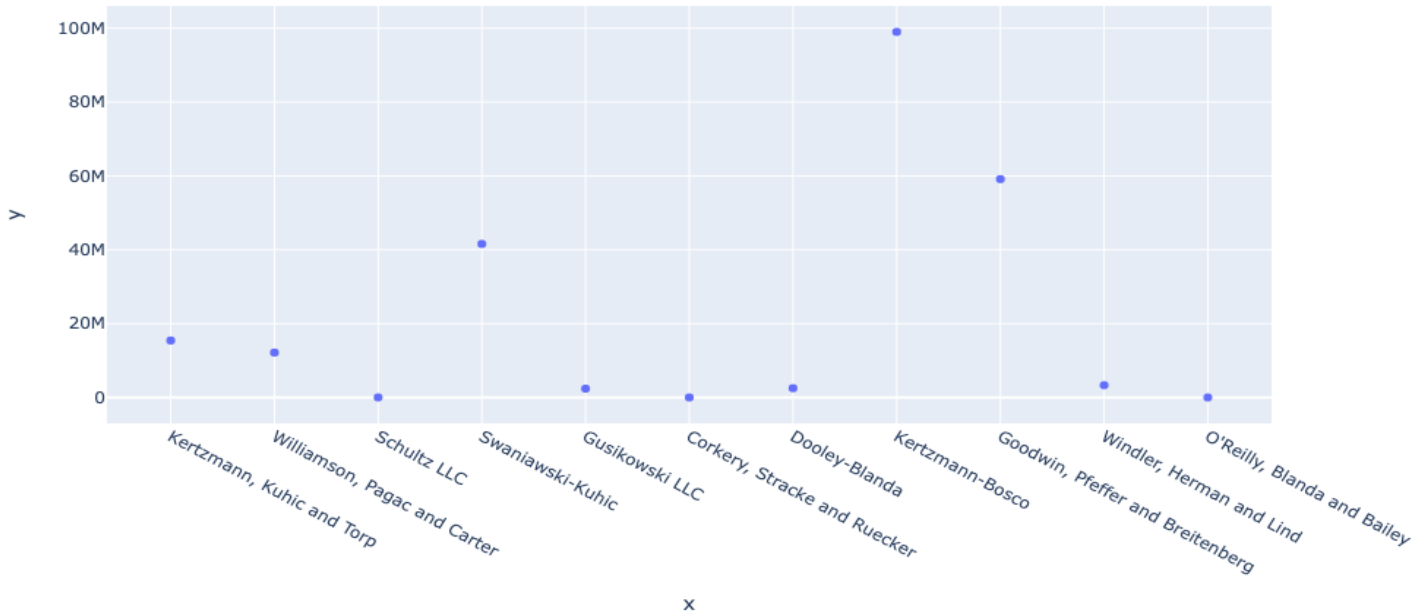


```
mycursor.execute(
    '''
        select InsuranceName, CoverageAmount from Insurance
        where coverageamount > (
        select avg(coverageamount) from insurance;
    ''')
result = mycursor.fetchall
covamt = []
insname = []
for i in mycursor:
    insname.append(i[0])
    covamt.append(i[1])
fig = px.bar(mycursor, x = insname, y = covamt)
fig.show()
```

The below bar graph shows the Insurance Companies which have more than average Coverage Amount to offer to universities as well as Students. We can see the exact amount in Millions if we hover over the bars.



The below scatter plot shows the average Amount of Coverage a university has to offer to its students.



SUMMARY AND RECOMMENDATION

A web-based application called University Healthcare Database is ready for usage and implementation across the United States, enabling users to access and track medical data about multiple universities, healthcare providers, insurance, student medical records, and other billing information. Students fill out the medical form, and their information, along with additional personal medical history information, is registered on the platform.

We started by evaluating all potential entities and their properties to develop the EER and UML diagrams based on this business challenge. Then, we gave each relation a cardinality based on the business problem. To add additional relations and bring the model up to the highest standard form which lowers the data redundancy, we constructed a relational model using the EER model and its cardinalities. We built dummy data for each entity in our database using the fill DB website. To show how it works, we created some SQL queries. We used Python's Jupiter notebook to connect to the database and perform some basic visualizations because the dataset was linked to the main server. We populated the MongoDB database by transferring the constructed tables, and then we built some queries to retrieve required data from the tables.

Universities, doctors, insurance providers, and healthcare providers can all use the database and retrieve the records that will help them manage their inventory of vaccines, upcoming appointments, timely check-ups, etc. To access the already-available data and generate predictions, it is also feasible to add a number of customization choices.