

PROBLEM STATEMENT

The problem entails designing and implementing a unified hospital management database system that addresses challenges such as data integration, security, scalability, user-friendliness, real-time access, and interoperability to improve patient care and streamline hospital operations while ensuring regulatory compliance.

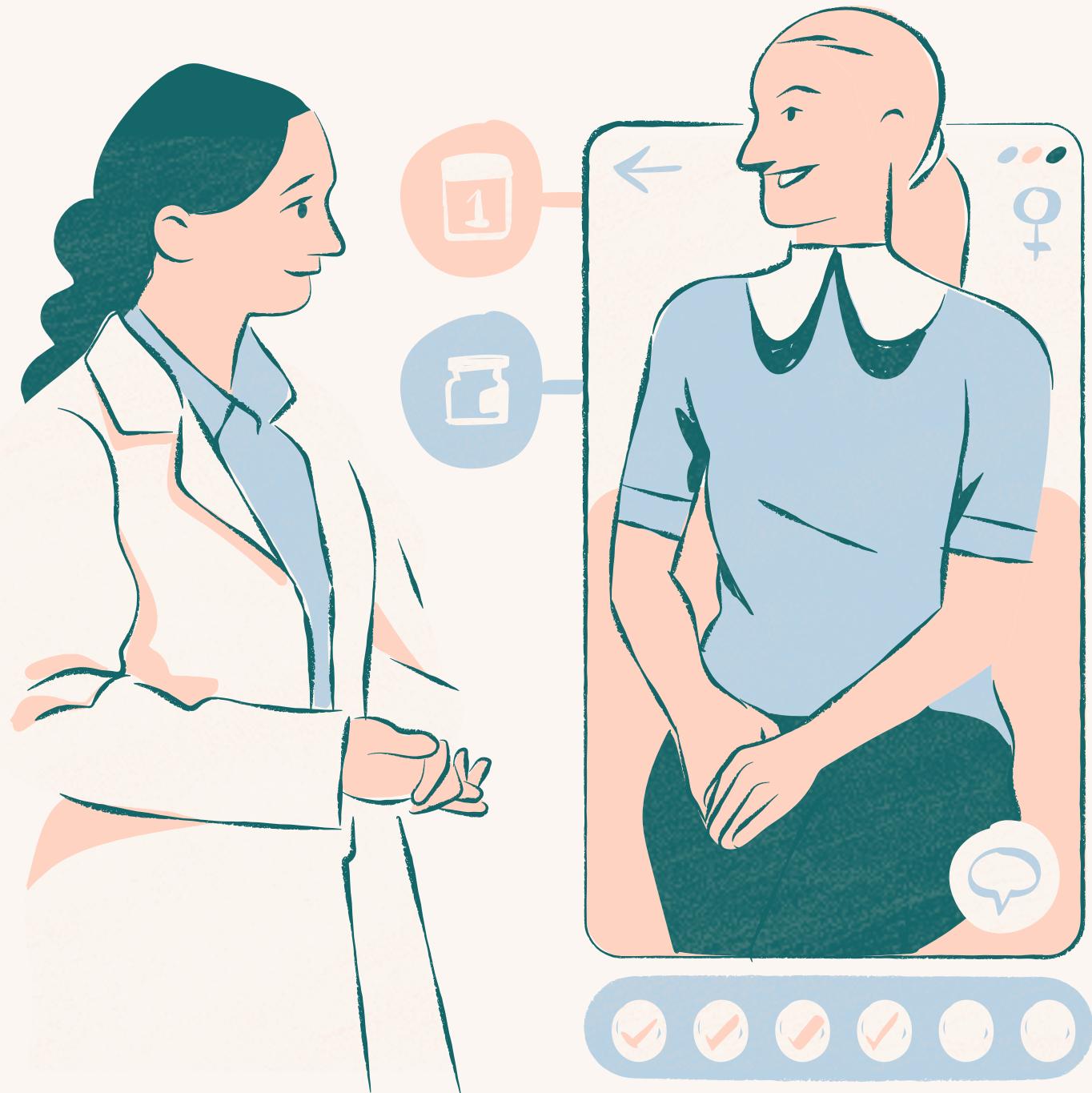
PROBLEM DESCRIPTION

In response to the increasing pressure on hospitals to improve efficiency and improve patient care, the aim of this project is to develop a robust database system for monitoring hospital activities. The current landscape often suffers from outdated and fragmented systems, resulting in inefficiencies and errors. The main challenges addressed were:

- **Data Integration:** Integrate disparate databases for patient records, scheduling, inventory, and billing to facilitate informed decision making.
- **Data Security and Privacy:** Ensure compliance with HIPAA and other legal standards to protect patient privacy.



PROBLEM DESCRIPTION



- **User-friendly interfaces:** Designing user-friendly interfaces to increase usability by health professionals.
- **Real-time access and reporting:** Timely delivery of accurate information for clinical decision making and performance management.
- **Interoperability:** Ensure easy data exchange with external systems such as EHRs and billing systems for continuous monitoring.

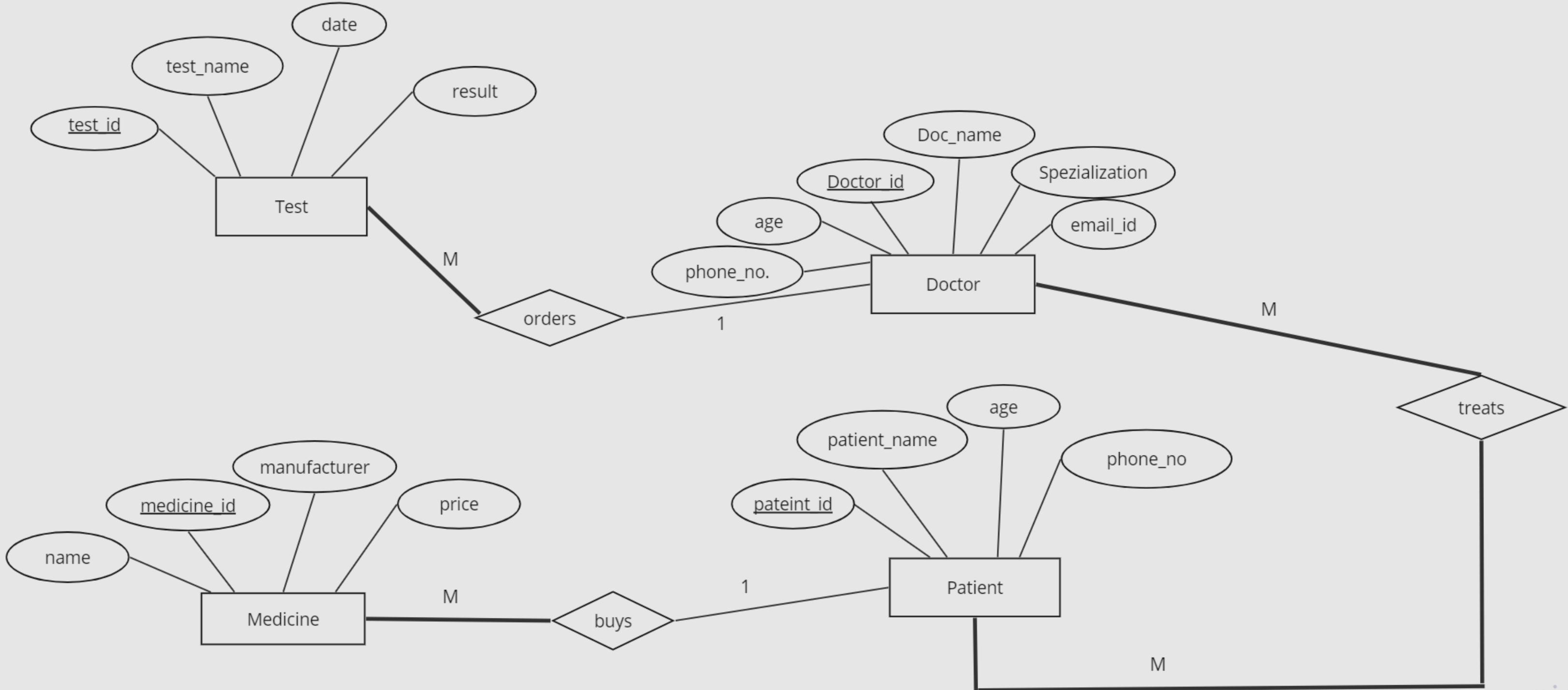
By addressing these challenges, the project aims to provide a comprehensive clinical management database system that improves resource utilization, improves patient care, and improves overall clinical performance.

ALL POSSIBLE ASSUMPTIONS

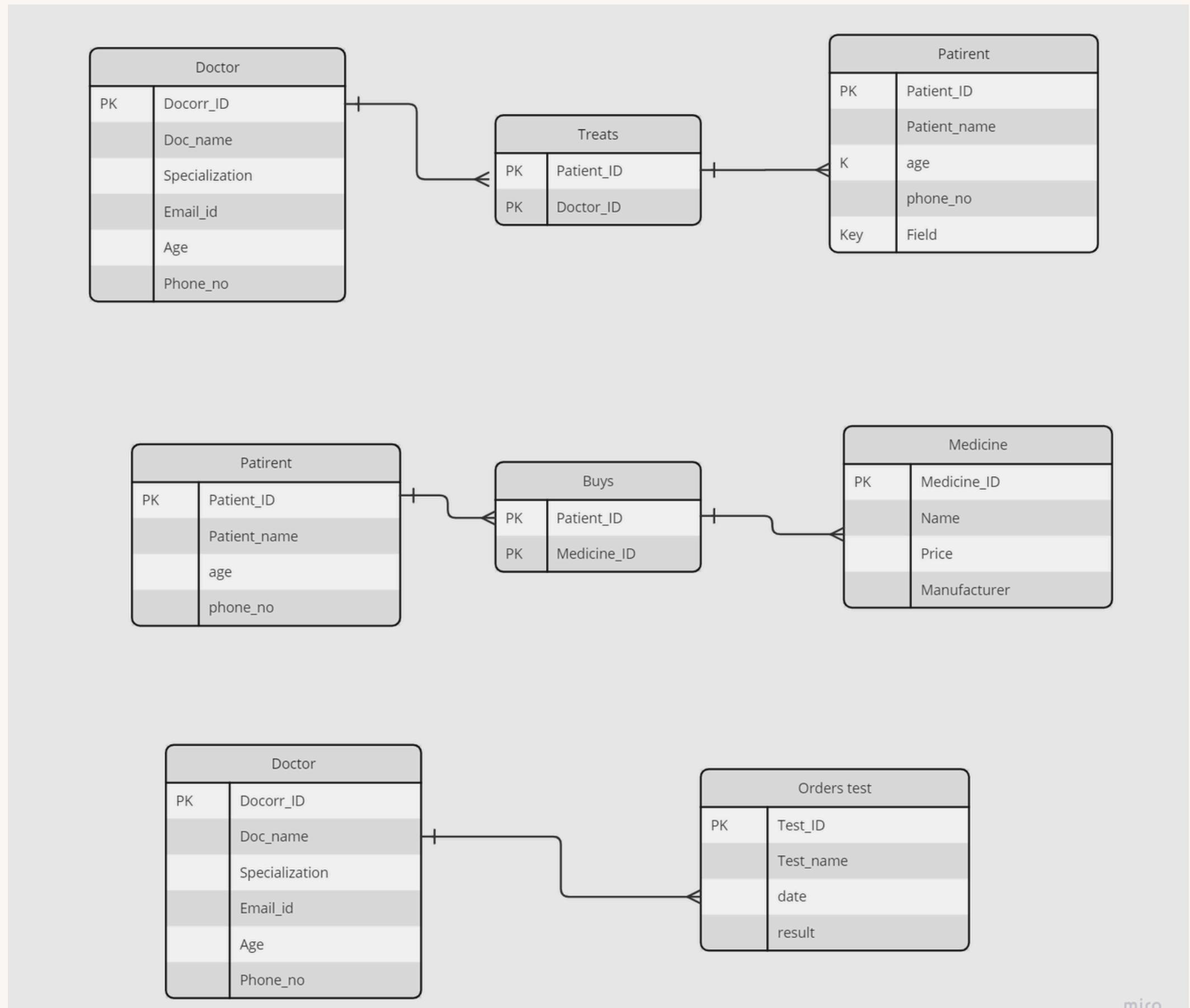
1. There is a strong entity **patient** which refers to the patient who visits our clinic and it has attributes like patient_id primary key and other attributes as patient_name, age, phone_no.
2. There is a strong entity **doctor** which refers to the doctors working in our clinic and it has primary key doctor_id and other attributes like phone_no, email_id, doc_name, specialization and age.
3. There is a strong entity **Test** which refers to the tests which the doctor orders the patient to take with primary key test_id and other attributes like test_name, date and result
4. There is a strong entity **medicine** which refers to the medicine list as per the test and it has primary key attribute medicine_id and other attributes like name, manufacturer and price.



ER DIAGRAM



RELATIONAL MODEL



Normalization

PATIENT TABLE

Patient_id	Patient_name	age	Phone_no.

- The patient table has multivalued attribute phone no._ and therefore it is not in 1NF so creating a separate table for it so that it is in 1NF

- This is normalized form since the patient table is in 1NF as it has no multivalued attribute
- The table is also in 2NF as it has no partial dependencies and non-key attributes fully functionally depend on the candidate key Patient_id
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

Patient_id	Patient_name	age

Normalization

PATIENT-PHONENO. TABLE

<u>Patient_id</u>	<u>Phone_no</u>

- This is normalized form since the table is in 1NF as it has no multivalued attribute
- The table is also in 2NF as it has no partial dependencies and non-key attribute fully functionally depends on the candidate key Patient_id
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

Normalization

DOCTOR TABLE

<u>Doctor_id</u>	Doc_name	age	Phone_no.	Email_id	Specialization
------------------	----------	-----	-----------	----------	----------------

- The doctor table has multivalued attribute phone_no and email_id and therefore it is not in 1NF so creating a separate table for it so that it is in 1NF

- This is normalized form since the patient table is in 1NF as it has no multivalued attribute
- The table is also in 2NF as it has no partial dependencies and non-key attributes fully functionally depend on the candidate key Doctor_id
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

<u>Doctor_id</u>	Doc_name	age	Specialization
------------------	----------	-----	----------------

Normalization

DOCTOR-PHONE_NO. TABLE

Doctor_id	Phone_no.

- This is normalized form since the table is in 1NF as it has no multivalued attribute
- The table is also in 2NF as it has no partial dependencies and non-key attribute fully functionally depends on the candidate key Doctor_id
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

Normalization

DOCTOR-EMAIL_ID TABLE

Doctor_id	Email_id

- This is normalized form since the table is in 1NF as it has no multivalued attribute
- The doctor_email-id table is also in 2NF as it has no partial dependencies and non-key attribute fully functionally depends on the candidate key Doctor_id
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

Normalization

MEDICINE TABLE

<u>medicine_id</u>	name	manufacturer	price
--------------------	------	--------------	-------

- This is normalized form since the table is in 1NF as it has no multivalued attribute
- The medicine table is also in 2NF as it has no partial dependencies and non-key attribute fully functionally depends on the candidate key Medicine_id.
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

Normalization

TEST TABLE

<u>test_id</u>	test_name	date	result

- This is normalized form since the table is in 1NF as it has no multivalued attribute
- The test table is also in 2NF as it has no partial dependencies and non-key attribute fully functionally depends on the candidate key test_id.
- It also follows 3NF as it has no transitive dependencies.
- Finally the table is in BCNF as the LHS is a super key.

FINAL TABLES

PATIENT TABLE

Field	Type	Null	Key	Default	Extra
p_id	int	NO	PRI	NULL	
patient_name	varchar(20)	YES		NULL	
age	int	YES		NULL	
phone_number	varchar(20)	YES		NULL	

FINAL TABLES

DOCTOR TABLE

Field	Type	Null	Key	Default	Extra
pno	int	YES		NULL	
age	int	YES		NULL	
d_id	int	NO	PRI	NULL	
d_name	varchar(20)	YES		NULL	
spcl	varchar(20)	YES		NULL	
email_d	varchar(20)	YES		NULL	

FINAL TABLES

TEST TABLE

Field	Type	Null	Key	Default	Extra
test_id	int	NO	PRI	NULL	
test_name	varchar(20)	YES		NULL	
date	date	YES		NULL	
result	varchar(20)	YES		NULL	

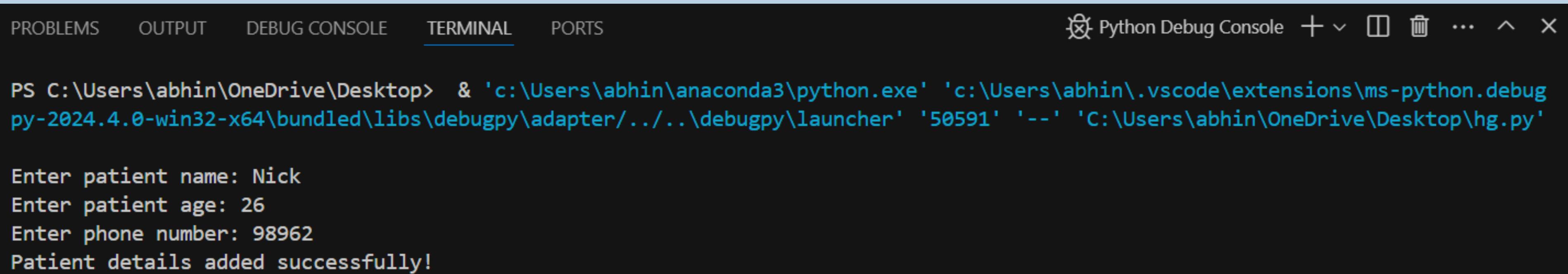
FINAL TABLES

MEDICINE TABLE

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	text	NO		NULL	
price	double	NO		NULL	
manufacturer	text	NO		NULL	

IMPLEMENTATION

Patient details being added to the table Patient
after getting input from the user



The screenshot shows a VS Code interface with a dark theme. At the top, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. To the right of the tabs is a Python Debug Console tab with icons for creating a new session (+), saving (down arrow), deleting (trash), and closing (X). The main area displays a command-line session:

```
PS C:\Users\abhin\OneDrive\Desktop> & 'c:\Users\abhin\anaconda3\python.exe' 'c:\Users\abhin\.vscode\extensions\ms-python.debugpy-2024.4.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '50591' '--' 'C:\Users\abhin\OneDrive\Desktop\hg.py'
```

Below the command line, the terminal outputs the following text:

```
Enter patient name: Nick
Enter patient age: 26
Enter phone number: 98962
Patient details added successfully!
```

IMPLEMENTATION

List of doctors being printed to be chosen by the user

List of Doctors:

1. Dr.Amrutha - Cardiologist
2. Dr.Prabhakaran - Psychologist
3. Dr.Charu - Gynecologist
4. Dr.Sneha - Radiologist
5. Dr.Varsha - Dentist

Enter the ID of the doctor you want to choose: 4

Doctor 'Dr.Sneha' chosen successfully!

IMPLEMENTATION

Prints the available tests and lets the user choose the id of the desired test to be taken

```
Available Tests:
```

- 1. Cancer Test
- 2. Polio Test
- 3. Pregnancy Test
- 4. COVID-19 Test
- 5. STD Test

```
Enter the ID of the test you want to choose: █
```

IMPLEMENTATION

The program lets the doctor input the result of the test and the program throws the medicine with regard to the test id produced by the user and if the result is negative it just prints no medicine prescribed by the doctor.

```
Enter the test result (Positive/Negative): Positive
Test record added successfully!
Doctor prescribes the following medications:
COVID-19 Antiviral
COVID-19 Antibiotic
```

IMPLEMENTATION

All the patient details are added to the patient table and the test details are added to the test table.

Table after running the program

mysql> select * from patient;			
p_id	patient_name	age	phone_number
1	Reka	30	1234567890
2	Ashirvad	25	2345678901
3	Kamal	35	3456789012
4	Sreeya	20	4567890123
5	Kaavya	21	5678901234
13	sundari	65	123
14	Sahil	20	98989
15	Jiya	23	1283
16	Hrishit	19	5543
17	Navya	20	3221
18	Mritika	32	2332
19	Kritin	12	5432
23	iamclown	12	123
24	Joginde	84	482
25	Varun	123	212
26	Killua	12	123
27	shiva	54	34
28	Hari	23	543
29	gowri	98	334
30	Lucy	44	32332
34	Ashaz	21	122122
35	Nick	26	98962

22 rows in set (0.00 sec)

mysql> select * from test;			
test_id	test_name	date	result
1	Cancer Test	2022-01-01	Positive
2	Polio Test	2022-02-15	Negative
3	Pregnancy Test	2022-03-20	Positive
4	COVID-19 Test	2022-04-10	Negative
5	STD Test	2022-05-05	Positive
6	Polio Test	2024-04-23	Negative
7	Pregnancy Test	2024-04-23	Positive
8	Cancer Test	2024-04-23	Positive
9	Cancer Test	2024-04-23	Positive
10	Polio Test	2024-04-25	Positive
11	COVID-19 Test	2024-04-25	Negative
12	Cancer Test	2024-04-25	Negative
13	COVID-19 Test	2024-04-25	Positive
14	COVID-19 Test	2024-04-25	Positive
15	Pregnancy Test	2024-04-25	Positive
16	Pregnancy Test	2024-04-25	Positive
17	COVID-19 Test	2024-04-25	Positive
18	Polio Test	2024-04-25	Positive
19	Pregnancy Test	2024-04-25	Positive
20	Cancer Test	2024-04-25	Positive
21	STD Test	2024-04-25	Positive
22	STD Test	2024-04-25	Positive
23	Polio Test	2024-04-25	Positive
24	Polio Test	2024-04-25	Positive
25	Polio Test	2024-04-25	Positive
26	COVID-19 Test	2024-04-25	Positive

26 rows in set (0.00 sec)

CONCLUSION

In conclusion, this hospital management system addresses challenges in healthcare by integrating databases, ensuring security, and scalability and usability. It promises to transform business, delivering timely information for meaningful decisions, and prompt applications. Communication allows for seamless data exchange, increasing oversight and continuous collaboration. This program, in line with industry standards, demonstrates a commitment to excellence, improving and improving the overall efficiency and effectiveness of clinical operations for the benefit of patients and healthcare professionals