

VIETNAM NATIONAL UNIVERSITY,
HO CHI MINH UNIVERSITY OF TECHNOLOGY,
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



(CO2014) Database Systems

ASSIGNMENT 2

HOSPITAL DATABASE

Advisor: Phan Trọng Nhân
Students: Huỳnh Minh Trí - 1953041
Nguyễn Trúc Phương - 1952402
Nguyễn Võ Hoàng Thi - 1952996
Trần Hưng Cường - 1952606

HO CHI MINH CITY, October 2021



Contents

1	Member list & Workload	2
2	Physical Database Design	2
2.1	Implementing the database	2
2.1.1	Requirements	2
2.1.2	Result	2
2.1.3	Note	15
2.2	Insert data	16
2.2.1	Requirements	16
2.2.2	Result	16
3	Store Procedure / Function / SQL	18
4	Building applications	22
4.1	Create user	22
4.2	Requirement function	22



1 Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Huỳnh Minh Trí	1953041	Implement the database Implement the procedure	25%
2	Nguyễn Trúc Phương	1952402	Implement the database Implement the procedure Report Completion	25%
3	Nguyễn Võ Hoàng Thi	1952996	Build the application Report completion	25%
4	Trần Hưng Cường	1952606	Build the application Implement procedure for application	25%

For this assignment, our team choose **MYSQL** as the DBMS to apply the requirements.

2 Physical Database Design

2.1 Implementing the database

2.1.1 Requirements

You have to implement your database, based on your assigned topic, into the physical database. Giving the full explanation of your choices of data types, data length, and constraints in your database.

2.1.2 Result

In assignment 1, we have already built a schema for hospital database. It includes 16 tables which are implemented into physical database as the following information:

Table: EMPLOYEE	
Attr: EMPcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, UNSIGNED NOT NULL
Explantion	Record the employee's ID in the form of integer in range of 0-65535. It is also the primary key of this table.
Attr: Address	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	DEFAULT NULL
Explanation	Record the Address in the form string with the maximum of 50 characters
Attr: First Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes

Constraints	NOT NULL
Explanation	Record the First Name in the form string with the maximum of 50 characters
Attr: Last Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	NOT NULL
Explanation	Record the Last Name in the form string with the maximum of 50 characters
Attr: Date of Birth	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the birth date in the form of 'yyyy-mm-dd'
Attr: Gender	
Data type	CHAR(1)
Maximum storage	1 Bytes
Explanation	We define Male: 'M', Female: 'F', Unisex: 'U', unknown: NULL
Attr: Start date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the date when the employee has started working at the hospital, in the form of 'yyyy-mm-dd'
Attr: SpecName	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	DEFAULT NULL
Explanation	Record the name of major which each employee specialized in, in the form string with the maximum of 50 characters
Attr: Spec_Degree's year	
Data type	TINYINT(2)
Maximum storage	1 Bytes
Constraints	UNSIGNED NOT NULL
Explanation	The number of year which the employee specialized in a major, it could be small. Range: 0-255
Attr: DEPcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('DEPcode')</i> <i>REFERENCES 'mydb'. 'DEPARTMENT' ('DEPcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Each employee belongs to one department.

Table: DOCTOR	
Attr: DOCcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('DOCcode')</i> <i>REFERENCES 'mydb'. 'EMPLOYEE' ('EMPcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	DOCcode is a identification code of a doctor in the hospital so it is a primary key. DOCTOR is a subclass of EMPLOYEE so it has a foreign key to the table EMPLOYEE

Table: DEPARTMENT	
Attr: DEPcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, UNSIGNED NOT NULL
Explanation	The ID of a particular department, so it must be the primary code.
Attr: Title	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	NOT NULL
Explanation	Allow title of department's written in length with maximum is 50
Attr: MGRcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('MGRcode')</i> <i>REFERENCES 'mydb'. 'DOCTOR' ('DOCcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Record the code of department's manager who is also a doctor, so it has a foreign key to the table DOCTOR.

Table: EMP_PHONENUMBERS	
Attr: EMPcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('EMPcode')</i> <i>REFERENCES 'mydb'. 'EMPLOYEE' ('EMPcode')</i>

	<i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	EMP_PHONENUMBERS is a multivalued attribute of table EMPLOYEE, so it links with the table EMPLOYEE by foreign key EMPcode.
Attr: phone number	
Data type	CHAR(10)
Maximum storage	10 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	The phone number is a multivalued attribute of EMPLOYEE, so it is the primary key simultaneously with EMPcode. We save it as CHAR because the phone number usually begins with digit '0' that can be automatically eliminated.

Table: NURSE	
Attr: NURcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('NURcode')</i> <i>REFERENCES 'mydb'. 'EMPLOYEE' ('EMPcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	NURcode is an identification code of a nurse in the hospital so it is a primary key. NURSE is a subclass of EMPLOYEE so it has a foreign key to the table EMPLOYEE

Table: PATIENT	
Attr: PATcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	LEFT(PATcode, 2) == 'OP' OR LEFT(PATcode, 2) == 'IP', NOT NULL, PRIMARY KEY
Explanation	Patient's code has the format of 'OP' (for outpatient) or 'IP' (for inpatient) and followed by 9 digits
Attr: Address	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	NULL
Explanation	Record the Address in the form string with the maximum of 50 characters
Attr: First Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	NULL

Explanation	Record the First Name in the form string with the maximum of 50 characters
Attr: Last Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	NULL
Explanation	Record the Last Name in the form string with the maximum of 50 characters
Attr: Date of Birth	
Data type	DATE
Maximum storage	3 Bytes
Constraints	NULL
Explanation	Record the birth date in the form of 'yyyy-mm-dd'
Attr: Gender	
Data type	CHAR(1)
Maximum storage	1 Bytes
Constraints	NULL
Explanation	We define Male: 'M', Female: 'F', Unisex: 'U', unknown(default): NULL
Attr: Phone number	
Data type	CHAR(10)
Maximum storage	10 Bytes
Constraints	NULL
Explanation	The phone number, of a patient, usually has 10 numbers, default null. We save it as CHAR because the phone number usually begin with digit '0' that can be automatically eliminated.

Table: OUTPATIENT	
Attr: OUTcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	LEFT(OUTcode, 2) == 'OP', NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('OUTcode')</i> <i>REFERENCES 'mydb'.'PATIENT' ('PATcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Oupatient's code has the format of 'OP' following by 9 digits is the primary key. OUTPATIENT is a subclass of table PATIENT so it has a foreign key to PATIENT.
Attr: ExaminingDOCcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('ExaminingDOCcode')</i>

	<i>REFERENCES 'mydb'. 'DOCTOR' ('DOCcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Each outpatient is examined by a Doctor so it has a foreign key to table DOCTOR

Table: OUTPATIENT_EXAMINATION	
Attr: OUTcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('OUTcode')</i> <i>REFERENCES 'mydb'. 'OUTPATIENT' ('OUTcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	OUTPATIENT_EXAMINATION is a multivalue attribute of table OUTPATIENT so it has a foreign key OUTcode to the table OUTPATIENT
Attr: Date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION is a multivalue attribute of table OUTPATIENT so it has all the attributes being primary keys. It records the date the outpatient going for examination in form of 'yyyy-mm-dd'
Attr: Fee	
Data type	DECIMAL(10,2)
Maximum storage	4 + 1 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION is a multivalue attribute of table OUTPATIENT so it has all the attributes being primary keys. The fee for the particular examination, in range range from 0.00 to 99999999.99
Attr: Next examination date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION is a multivalue attribute of table OUTPATIENT so it has all the attributes being primary keys. It records the date the outpatient will make a next appointment with doctor, in form of 'yyyy-mm-dd'
Attr: Diagnosis	
Data type	VARCHAR(50)
Maximum storage	50 Bytes
Constraints	NOT NULL, PRIMARY KEY

Explanation	OUTPATIENT_EXAMINATION is a multivalue attribute of table OUTPATIENT so it has all the attributes being primary keys. It is a short description of diseases with max length = 50 characters.
-------------	--

Table: MEDICATION	
Attr: MEDcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
Explanation	Medication has code as the series of number so it is the primary key of medication
Attr: Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Explanation	Name of Medicine, it could be as long as text at maximum 50 characters.
Attr: Price	
Data type	DECIMAL(10,2)
Maximum storage	4+1 Bytes
Explanation	The price of medication can be in decimal, range from 0.0 to 99999999.99
Attr:Expiration Date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	IF Expiration Date > current_date() THEN Expiration Date = '0000-00-00', DEFAULT NULL
Explanation	Mark the out-of-date medication by set it to '0000-00-00', unknown information: NULL

Table: OUTPATIENT_EXAMINATION_MEDICATION	
Attr: OUTcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION_MEDICATION is a multivalue attribute of table OUTPATIENT_EXAMINATION so it has all the attributes being primary keys.
Attr: Date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION_MEDICATION is a multivalue attribute of table OUTPATIENT_EXAMINATION so it has all the attributes being primary keys. It records the date the outpatient going for examination in form of 'yyyy-mm-dd'

Attr: Fee	
Data type	DECIMAL(10,2)
Maximum storage	4 + 1 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION_MEDICATION is a multivalued attribute of table OUTPATIENT_EXAMINATION so it has all the attributes being primary keys. The fee for the particular examination, in range range from 0.00 to 99999999.99
Attr: Next examination date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION_MEDICATION is a multivalued attribute of table OUTPATIENT_EXAMINATION so it has all the attributes being primary keys. It records the date the outpatient will make a next appointment with doctor, in form of 'yyyy-mm-dd'
Attr: Diagnosis	
Data type	VARCHAR(50)
Maximum storage	50 Bytes
Constraints	NOT NULL, PRIMARY KEY
Explanation	OUTPATIENT_EXAMINATION_MEDICATION is a multivalued attribute of table OUTPATIENT_EXAMINATION so it has all the attributes being primary keys. It is a short description of diseases with max length = 50 characters.
Attr: MEDcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	NOT NULL, PRIMARY KEY FOREIGN KEY: <i>FOREIGN KEY ('MEDcode')</i> <i>REFERENCES 'mydb'. 'MEDICATION' ('MEDcode')</i> <i>ON DELETE NO ACTION</i> <i>ON UPDATE NO ACTION</i>
Explanation	It is a medication in use so it has a foreign key MEDcode to the table MEDICATION.

Table: INPATIENT	
Attr: INcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	LEFT(INcode, 2)='IP', PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('INcode')</i> <i>REFERENCES 'mydb'. 'PATIENT' ('PATcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>



Explanation	Inpatient's code has the format of 'IP' following by 9 digits. It is a subclass of table PATIENT, so it has link to it.
Attr: NURcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('NURcode')</i> <i>REFERENCES 'mydb'. 'NURSE' ('NURcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Nurse who is taking care for inpatient has a NURcode which is link to table NURSE
Attr: Date of admission	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the Date of admission in form of 'yyyy-mm-dd', allow null value
Attr: Date of discharge	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the Date of discharge in form of 'yyyy-mm-dd', allow null value
Attr: Fee	
Data type	DECIMAL(10,2)
Maximum storage	4 + 1 Bytes
Constraints	DEFAULT NULL, PRIMARY KEY
Explanation	The fee for the particular inpatient, in range range from 0.00 to 99999999.99
Attr: Sickroom	
Data type	TINYINT
Maximum storage	1 Bytes
Constraints	UNSIGNED DEFAULT NULL
Explanation	The number of sickroom for the particular inpatient, in range range from 0 to 255.
Attr: Diagnosis	
Data type	VARCHAR(50)
Maximum storage	50 Bytes
Constraints	DEFAULT NULL
Explanation	It is a short description of diseases with max length = 50 characters.

Table: IS_TREATED_BY

Attr: DOCcode	
Data type	SMALLINT

Maximum storage	2 Bytes
Constraints	PRIMARY KEY, UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('DOCcode')</i> <i>REFERENCES 'mydb'. 'DOCTOR' ('DOCcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	It is the ID of doctor who treats for the particular patient in a relationship, so it is a foreign key to table DOCTOR.
Attr: INcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	LEFT(INcode, 2) == 'IP', PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('INcode')</i> <i>REFERENCES 'mydb'. 'INPATIENT' ('INcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	It is the ID of patient who is treated in a relationship, so it is a foreign key to table INPATIENT.
Attr: Start date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the start date in the form of 'yyyy-mm-dd'
Attr: End date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Record the end date in the form of 'yyyy-mm-dd'
Attr: Result	
Data type	TINYTEXT
Maximum storage	255 Bytes
Constraints	DEFAULT NULL
Explanation	Short description about each treatment effect

Table: IS_TREATED_BY_MEDICATION	
Attr: DOCcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, UNSIGNED NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('DOCcode')</i> <i>REFERENCES 'mydb'. 'IS_TREATED_BY' ('DOCcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>

Explanation	It is the ID of doctor who treats for the particular patient in a relationship, so it is a foreign key to table DOCTOR. IS_TREATED_BY_MEDICATION is a multivalued attribute so all the attributes from IS_TREATED_BY are primary keys for this table.
Attr: INcode	
Data type	CHAR(11)
Maximum storage	11 Bytes
Constraints	LEFT(INcode, 2) == 'IP', PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('INcode')</i> <i>REFERENCES 'mydb'. 'IS_TREATED_BY' ('INcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	It is the ID of patient who is treated by a particular medicine in a relationship, so it is a foreign key to table IS_TREATED_BY. IS_TREATED_BY_MEDICATION is a multivalued attribute so all the attributes from IS_TREATED_BY are primary keys for this table.
Attr: MEDcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('MEDcode')</i> <i>REFERENCES 'mydb'. 'MEDICATION' ('MEDcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	It is a medication in use so it has a foreign key MEDcode to the table MEDICATION.

Table: MED_EFFECT	
Attr: MEDcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('MEDcode')</i> <i>REFERENCES 'mydb'. 'MEDICATION' ('MEDcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	It is a medication in use so it has a foreign key MEDcode to the table MEDICATION. MED_EFFECT is a multivalued attribute of MEDICATION so the primary key of MEDICATION is also the primary key for MED_EFFECT
Attr: MED_EFFECT	



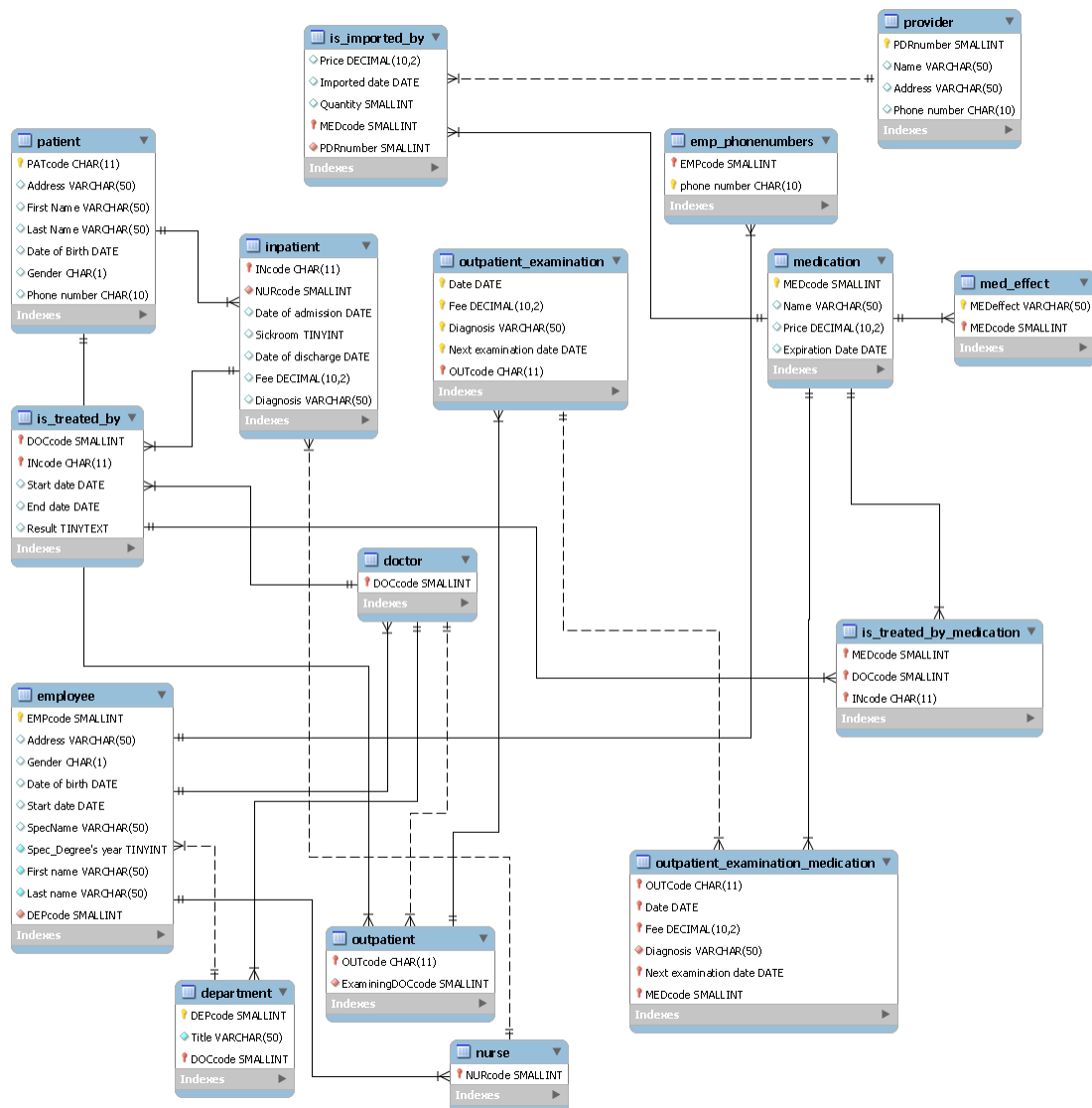
Data type	VARCHAR(50)
Maximum storage	50 Bytes
Constraints	PRIMARY KEY, NOT NULL
Explanation	Effect of medicine, it could be as long as text. The main value for this table as a multivalue attribute of MEDICATION.

Table: PROVIDER	
Attr: PDRnumber	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, NOT NULL, AUTO_INCREMENT
Explanation	Provider has code as the series of number
Attr: Address	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	DEFAULT NULL
Explanation	Record the Address in the form string with the maximum of 50 characters
Attr: Name	
Data type	VARCHAR(50)
Maximum storage	50 (+2) Bytes
Constraints	DEFAULT NULL
Explanation	Record the name in the form string with the maximum of 50 characters
Attr: Phone number	
Data type	CHAR(10)
Maximum storage	10 Bytes
Constraints	DEFAULT NULL
Explanation	The phone number of a provider, usually has 10 numbers.

Table: IS_IMPORTED_BY	
Attr: MEDcode	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	PRIMARY KEY, NOT NULL FOREIGN KEY: <i>FOREIGN KEY ('MEDcode')</i> <i>REFERENCES 'mydb'. 'MEDICATION' ('MEDcode')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Medication has code as the series of number
Attr: PDRnumber	
Data type	SMALLINT
Maximum storage	2 Bytes
	PRIMARY KEY, NOT NULL

	FOREIGN KEY: <i>FOREIGN KEY ('PDRnumber')</i> <i>REFERENCES 'mydb'. 'PROVIDER' ('PDRnumber')</i> <i>ON DELETE CASCADE</i> <i>ON UPDATE CASCADE</i>
Explanation	Provider has code as the series of number
Attr: Price	
Data type	DECIMAL(10,2)
Maximum storage	4+1 Bytes
Constraints	DEFAULT NULL
Explanation	The price of consignment, can be in decimal, range from 0.0 to 99999999.99
Attr: Imported date	
Data type	DATE
Maximum storage	3 Bytes
Constraints	DEFAULT NULL
Explanation	Date when the provider imports the drug into warehouse, if unknown information: NULL
Attr: Quantity	
Data type	SMALLINT
Maximum storage	2 Bytes
Constraints	DEFAULT NULL
Explanation	Number of each medications in one consignment, in range of 0-255, if unknown information: NULL

After that, we will build a physical database in MYSQL due to the information above. The source code is in file [PART1.sql](#) attached in zip. Also, some extra triggers(*insert_inpatient*, *insert_outpatient*) and procedures (*insert_inpatient*, *insert_outpatient* and *insert_emp*) used to checking the constraints while insert and update the patient ('IP' and 'OP') and employee('Doctor' and 'Nurse'). Their source code are in file [extra.sql](#). The schema then performed



as:

2.1.3 Note

There are some notes for task 1A:

- employee ID: 1->8. Where:
 - Doctor: 1 2 3 4
 - Nurse: 5 6 7 8
- Patient:
 - INPATIENT: IP0...1->4
 - OUTPATIENT: OP0...1->4

2.2 Insert data

2.2.1 Requirements

- Insert data for all tables in the database.
- Requirements: The data in the tables must be meaningful, and each table has at least 4 rows.

2.2.2 Result

After building a database in the task 1A. We insert the data for corresponding table in **MYSQL** as:

Table 'mydb':EMPLOYEE'
INSERT INTO mydb.employee VALUE (1, '5 ManCity', 'F', '2001-1-1', '2019-1-1', 'Ophthalmologists', 2, 'A', 'Nguyen', 1);
INSERT INTO mydb.employee VALUE (2, '5 ManUtd', 'F', '2001-1-2', '2019-1-1', 'Ophthalmologists', 3, 'B', 'Nguyen', 2);
INSERT INTO mydb.employee VALUE (3, '5 Liverpool', 'M', '2001-1-1', '2019-1-1', 'Dermatologists', 2, 'C', 'Nguyen', 3);
INSERT INTO mydb.employee VALUE (4, '5 Ansernal', 'F', '2001-1-2', '2019-1-4', 'Dermatologists', 3, 'D', 'Nguyen', 4);
INSERT INTO mydb.employee VALUE (5, '5 ManCity', 'M', '2001-1-1', '2019-1-1', 'Ophthalmologists', 2, 'E', 'Nguyen', 1);
INSERT INTO mydb.employee VALUE (6, '5 ManUtd', 'F', '2001-1-2', '2019-1-1', 'Ophthalmologists', 3, 'F', 'Nguyen', 2);
INSERT INTO mydb.employee VALUE (7, '5 Liverpool', 'M', '2001-1-1', '2019-1-1', 'Dermatologists', 2, 'G', 'Nguyen', 1);
INSERT INTO mydb.employee VALUE (8, '5 Ansernal', 'M', '2001-1-2', '2019-1-4', 'Dermatologists', 3, 'H', 'Nguyen', 4);
Table 'mydb':DEPARTMENT'
INSERT INTO mydb.department VALUE ('1', 'A', '4');
INSERT INTO mydb.department VALUE ('2', 'B', '2');
INSERT INTO mydb.department VALUE ('3', 'C', '1');
INSERT INTO mydb.department VALUE ('4', 'D', '3');
Table 'mydb':EMP_PHONENUMBERS'
INSERT INTO emp_phonenumbers VALUE ('1', '111222333');
INSERT INTO emp_phonenumbers VALUE ('2', '111222334');
INSERT INTO emp_phonenumbers VALUE ('3', '111222335');
INSERT INTO emp_phonenumbers VALUE ('4', '111222336');
INSERT INTO emp_phonenumbers VALUE ('5', '111222337');
INSERT INTO emp_phonenumbers VALUE ('6', '111222338');
INSERT INTO emp_phonenumbers VALUE ('7', '111222339');
INSERT INTO emp_phonenumbers VALUE ('8', '111222340');
INSERT INTO emp_phonenumbers VALUE ('7', '111222341');
INSERT INTO emp_phonenumbers VALUE ('6', '111222342');
INSERT INTO emp_phonenumbers VALUE ('5', '111222343');
INSERT INTO emp_phonenumbers VALUE ('4', '111222344');
Table 'mydb':NURSE'

INSERT INTO nurse VALUE(5); INSERT INTO nurse VALUE(6); INSERT INTO nurse VALUE(7); INSERT INTO nurse VALUE(8);
Table 'mydb'.PATIENT
INSERT INTO patient VALUES ('IP000000001', '1 Mancity', 'A', 'Nguyen', '2000-12-2', 'F', '123456789');
INSERT INTO patient VALUES ('IP000000002', '1 ManUtd', 'B', 'Nguyen', '2000-12-2', 'U', '123456710');
INSERT INTO patient VALUES ('IP000000003', '1 Ansernal', 'C', 'Nguyen', '2000-12-2', 'F', '123456711');
INSERT INTO patient VALUES ('IP000000004', '1 Liverpool', 'D', 'Nguyen', '2000-12-2', 'F', '123456712');
INSERT INTO patient VALUES ('OP000000001', '1 Mancity', 'E', 'Nguyen', '2000-12-2', 'M', '123456713');
INSERT INTO patient VALUES ('OP000000002', '1 ManUtd', 'F', 'Nguyen', '2000-12-2', 'M', '123456714');
INSERT INTO patient VALUES ('OP000000003', '1 Mancity', 'G', 'Nguyen', '2000-12-2', 'M', '123456715');
INSERT INTO patient VALUES ('OP000000004', '1 Ansernal', 'H', 'Nguyen', '2000-12-2', 'M', '123456716');
Table 'mydb'.OUTPATIENT
INSERT INTO outpatient VALUES ('OP000000001', 1); INSERT INTO outpatient VALUES ('OP000000002', 4); INSERT INTO outpatient VALUES ('OP000000003', 1); INSERT INTO outpatient VALUES ('OP000000004', 2);
Table 'mydb'.OUTPATIENT_EXAMINATION
INSERT INTO outpatient_examination VALUES ('2021-1-1', 12.0, 'Diabetes', '2021-2-1', 'OP000000001');
INSERT INTO outpatient_examination VALUES ('2021-1-1', 100.10, 'Chickenpox', '2021-2-1', 'OP000000002');
INSERT INTO outpatient_examination VALUES ('2021-1-1', 90.30, 'Depression', '2021-2-1', 'OP000000003');
INSERT INTO outpatient_examination VALUES ('2021-1-1', 134.0, 'Cyclospora', '2021-2-1', 'OP000000004');
Table 'mydb'.OUTPATIENT_EXAMINATION_MEDICATION
INSERT INTO outpatient_examination_medication VALUES('OP000000001', '2021-10-5', 100,'2021-11-5',1); INSERT INTO outpatient_examination_medication VALUES('OP000000002', '2020-10-5', 50,'2020-11-5',2); INSERT INTO outpatient_examination_medication VALUES('OP000000003', '2019-10-5', 4000,'2019-11-5',4); INSERT INTO outpatient_examination_medication VALUES('OP000000004', '2018-10-5', 12,'2018-11-5',2);
Table 'mydb'.INPATIENT
INSERT INTO inpatient VALUES ('IP000000001', 1,'2021-1-1', 12, '2021-2-1', 100.0, 'Pharyngitis');

```
INSERT INTO inpatient VALUES ('IP000000002', 2, '2021-1-1', 11, '2021-2-1', 100.0, 'Pharyngitis');
INSERT INTO inpatient VALUES ('IP000000003', 1, '2021-1-1', 100, '2021-2-1', 100.0, 'Ulcerative colitis');
INSERT INTO inpatient VALUES ('IP000000004', 4, '2021-1-1', 19, '2021-2-1', 100.0, 'Ulcerative colitis');
```

Table 'mydb'.IS_TREATED_BY

```
INSERT INTO is_treated_by VALUES(1, 'IP000000002', '2021-1-1', '2021-2-1', 'Conjunctivitis');
INSERT INTO is_treated_by VALUES(4, 'IP000000001', '2021-1-1', '2021-2-1', 'Crabs');
INSERT INTO is_treated_by VALUES(1, 'IP000000003', '2021-1-1', '2021-2-1', 'Cryptosporidiosis');
INSERT INTO is_treated_by VALUES(1, 'IP000000004', '2021-2-1', '2021-3-1', 'Conjunctivitis');
```

Table 'mydb'.IS_TREATED_BY_MEDICATION

```
INSERT INTO is_treated_by_medication VALUES(1,2,'IP000000001');
INSERT INTO is_treated_by_medication VALUES(2,2,'IP000000001');
INSERT INTO is_treated_by_medication VALUES(3,2,'IP000000001');
INSERT INTO is_treated_by_medication VALUES(2,2,'IP000000002');
```

Table 'mydb'.MED_EFFECT

```
INSERT INTO med_effect VALUES('Bladder pain', 1);
INSERT INTO med_effect VALUES('bloody or cloudy urine', 1);
INSERT INTO med_effect VALUES('difficult, burning, or painful urination', 1);
INSERT INTO med_effect VALUES('Swelling of the ankles or feet', 2);
INSERT INTO med_effect VALUES('Difficult or labored breathing', 2);
INSERT INTO med_effect VALUES('Back pain', 3);
INSERT INTO med_effect VALUES('bleeding gums', 3);
INSERT INTO med_effect VALUES('Anxiety', 4);
INSERT INTO med_effect VALUES('Overdose: Cold, clammy skin', 4);
```

Table 'mydb'.PROVIDER

```
INSERT INTO provider VALUES(1,'A','10 ManCity', '123456123');
INSERT INTO provider VALUES(2,'B','10 ManUtd', '123456123');
INSERT INTO provider VALUES(3,'C','10 Liverpool', '123456124');
INSERT INTO provider VALUES(4,'D','10 Real Madrid', '123456125');
```

Table 'mydb'.IS_IMPORTED_BY

```
INSERT INTO is_imported_by VALUES(100,'2021-10-10',200,1,2);
INSERT INTO is_imported_by VALUES(500,'2021-10-10',200,2,1);
INSERT INTO is_imported_by VALUES(400,'2021-10-10',200,3,3);
INSERT INTO is_imported_by VALUES(1000,'2021-10-10',200,4,4);
```

3 Store Procedure / Function / SQL

a. Increase Inpatient Fee to 10% for all the inpatients who are admitted to hospital from 01/09/2020.

```

1  use mydb;
2
3  DELIMITER $$
4  drop procedure if exists increase_fee;
5  CREATE PROCEDURE increase_fee()
6  BEGIN
7      UPDATE INPATIENT
8      SET
9          Fee = Fee*1.1
10     WHERE
11         `Date of admission` > '2020-09-01';
12 END; $$
13 DELIMITER ;
14
15 CALL mydb.increase_fee()

```

RESULT:

	INcode	NURcode	Date of admission	Sickroom	Date of discharge	Fee	Diagnosis
▶	IP000000001	1	2021-01-01	12	2021-02-01	100.00	Pharyngitis
	IP000000002	2	2021-01-01	11	2021-02-01	100.00	Pharyngitis
	IP000000003	1	2021-01-01	100	2021-02-01	100.00	Ulcerative colitis
	IP000000004	4	2021-01-01	19	2021-02-01	100.00	Ulcerative colitis
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

b. Select all the patients (outpatient inpatient) of the doctor named 'Nguyen Van A'

```

1  use mydb;
2
3  DELIMITER $$
4  drop procedure if exists print_patients_of_doctor;
5  CREATE PROCEDURE mydb.print_patients_of_doctor(FirstName varchar(50),
6  LastName varchar(50))
7  BEGIN
8      SELECT P.`Last Name`, P.`First Name`
9      FROM mydb.`PATIENT` AS P
10     INNER JOIN mydb.`OUTPATIENT` AS O ON P.`PATCode` = O.`OUTCode`
11     INNER JOIN mydb.`EMPLOYEE` as E ON O.`ExaminingDOCcode` = E.`EMPcode`
12     WHERE E.`First Name` = FirstName and E.`Last Name` = LastName
13     UNION ALL
14     SELECT P.`Last Name`, P.`First Name`
15     FROM mydb.`PATIENT` AS P
16     INNER JOIN mydb.`is_treated_by` AS T ON P.`PATCode` = T.`INCode`
17     INNER JOIN mydb.`EMPLOYEE` as E ON T.`DOCcode` = E.`EMPcode`
18     WHERE E.`First Name` = FirstName and E.`Last Name` = LastName;
19 END; $$

```

```
19 DELIMITER ;
20
21 CALL mydb.print_patients_of_doctor('A', 'Nguyen')
```

RESULT:

	Last Name	First Name
►	Nguyen	E
	Nguyen	G
	Nguyen	B
	Nguyen	C
	Nguyen	D

c. Write a function to calculate the total medication price a patient has to pay for each treatment or examination

Input: Patient ID

Output: A list of payment of each treatment or examination

```
1 use mydb;
2 DELIMITER $$
3 drop procedure if exists list_payment;
4 CREATE PROCEDURE mydb.list_payment(IN PATcode char(11))
5 BEGIN
6     IF LEFT(PATcode, 2) = 'IP' THEN
7         SELECT A.DOCcode, A. MEDcode, C.`Name`, C.`Expiration Date`, C.price FROM
8         mydb.is_treated_by_medication AS A
9         INNER JOIN mydb.medication AS C
10        ON A.MEDcode=C.MEDcode
11        WHERE A.INcode = PATcode;
12     ELSE -- OP
13         SELECT M.price FROM
14         outpatient_examination_medication AS O
15         INNER JOIN medication AS M
16         USING (MEDcode)
17         WHERE OUTcode = PATcode;
18     END IF;
19 END; $$
20 DELIMITER ;
21
22 CALL mydb.list_payment('IP000000001')
```

Result

	DOCcode	MEDcode	Name	Expiration Date	price
►	2	1	Adderall	2025-05-10	100.00
	2	2	Amlodipine	2025-05-10	70.00
	2	3	Brilinta	2025-05-10	50.00

d. Write a procedure to sort the doctor in increasing number of patients he/she takes care in a period of time

Input: Start date, End date

Output: A list of sorting doctors

```
1  -- INPATIENT, use is_treated_by.`Start date`. Because DOCcode and INcode
   are primary key, then 1 doctor cannot treat 1 patient more than 1 time.
   OUTPATIENT, use .outpatient_examination.`date`
2
3  use mydb;
4
5  DELIMITER $$
6  drop procedure if exists sorting_doctors;
7  CREATE PROCEDURE mydb.sorting_doctors(IN start_date DATE, IN end_date DATE)
8  BEGIN
9      drop table if exists tmp1;
10     CREATE TABLE tmp1
11     SELECT DOCcode, COUNT(INcode) `#patients` FROM is_treated_by
12     WHERE `Start date` > start_date AND `Start date` < end_date
13     GROUP BY Doccode
14     ORDER BY `#patients` DESC;
15
16     drop table if exists tmp2;
17     CREATE TABLE tmp2
18     SELECT ExaminingDOCcode, COUNT(OUTcode) `#patients` FROM
19     outpatient AS A
20     INNER JOIN outpatient_examination as B
21     USING (OUTcode)
22     WHERE `date` > start_date AND `date` < end_date
23     GROUP BY ExaminingDOCcode
24     ORDER BY `#patients` DESC;
25
26     SELECT tmp1.DOCcode, tmp1.`#patients` + tmp2.`#patients` as
27     totalPatients FROM
28     tmp1 INNER JOIN tmp2
29     ON tmp1.DOCcode = tmp2.ExaminingDOCcode;
30 END; $$
31 DELIMITER ;
32
33 CALL mydb.sorting_doctors('2020-1-1','2022-1-1')
```

Result

	DOCcode	totalPatients
►	1	5
	4	2

4 Building applications

Build an application with the following requirements:

- Programming environment: optional (desktop, web, or mobile application).
- Programming language: optional.
- The application connects to the database created in Part 1 and Part 2.
- Display the data on the form and perform the requirements below.
- Students need to prepare data and scripts for demonstration at the reporting session.

4.1 Create user

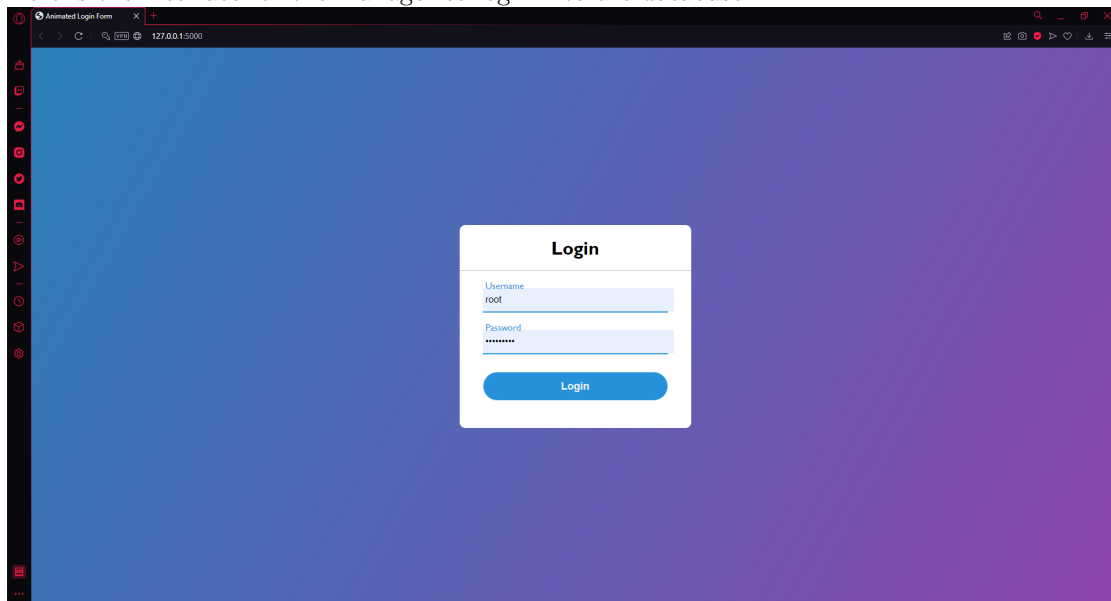
Log in to the database with DBA privileges such as SYS / SYSTEM, create a user named “Manager” and assign all access rights to this user.

We have create a user with the user name root and the password. In the next image in which the login screen is provided, that was the user that we have created in the database.

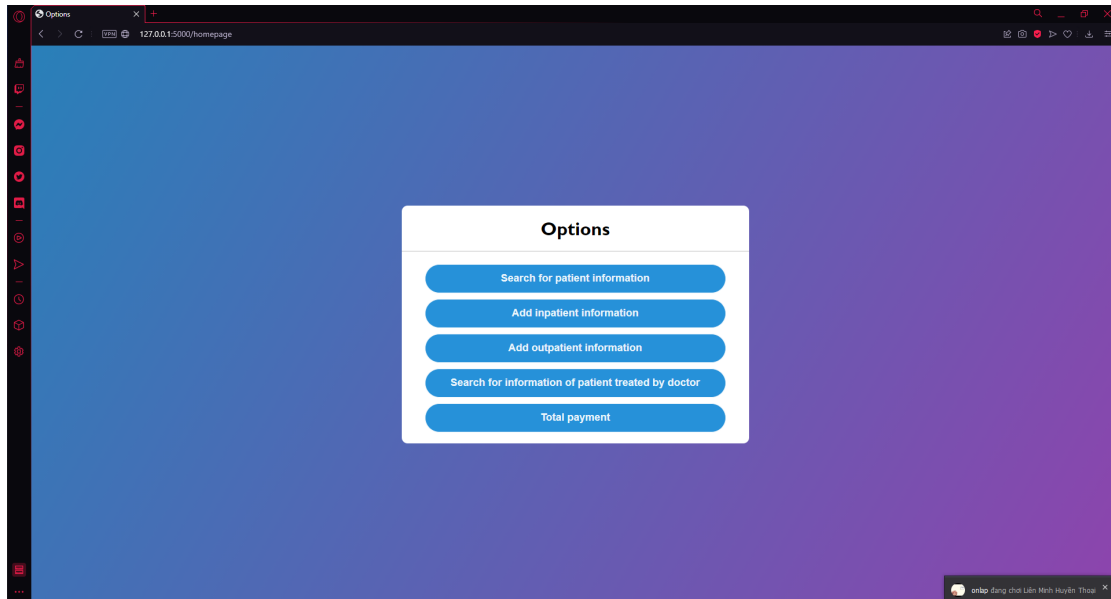
4.2 Requirement function

Log in to the user manager and do the following

Here is the interface for the manager to login into the database.

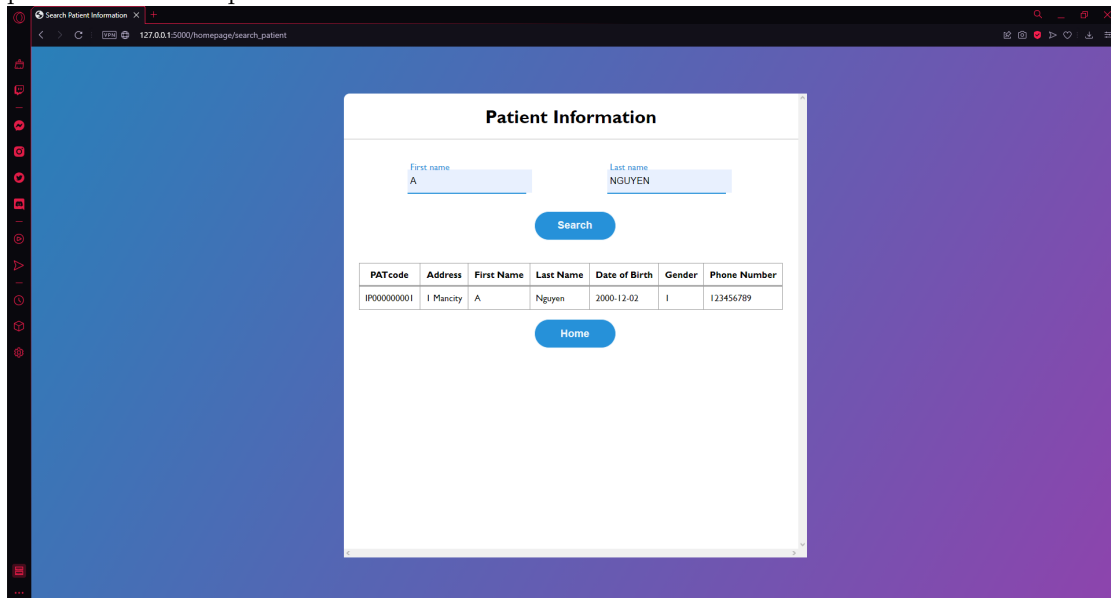


Base on the requirements for Part 3, the manager is able to perform below functions:



1. Search patient information: Search results include the name, phone number and information about the treatment and visit of the patient.

For searching the information of the patient, we provide an interface with the filter by the provided name for patient:

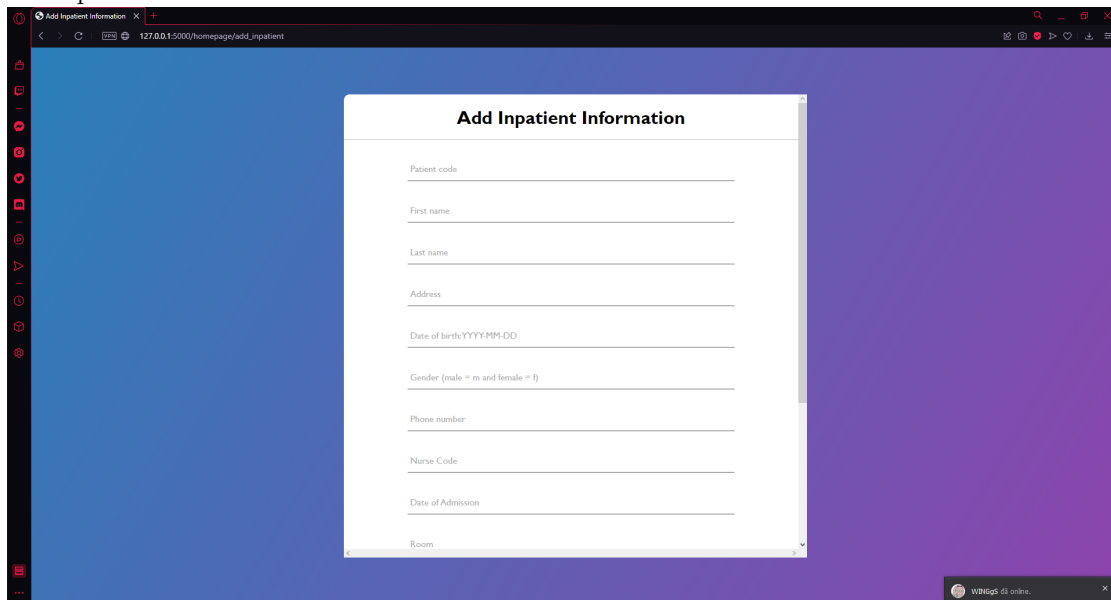


In this option, we provide some information of the patient such as: ID, address, name, birthday, gender and phone number.

2. Add information for a new patient.

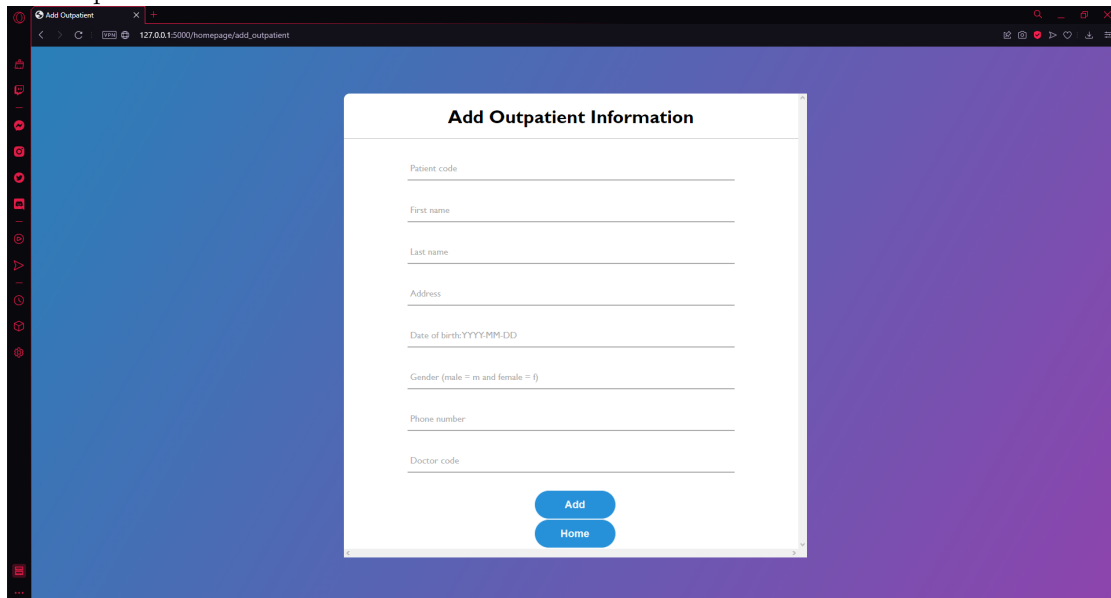
In this option, we split the new patient into 2 types: inpatient and outpatient. Because the information of the inpatient and outpatient is different, the provider have to fill enough information of the new patient to make sure the critical information is not missed.

For inpatient:



The screenshot shows a web browser window with the URL `127.0.0.1:5000/homepage/add_inpatient`. The page has a blue and purple gradient background. A white form titled "Add Inpatient Information" is centered. The form contains the following fields: Patient code, First name, Last name, Address, Date of birth (YYYY-MM-DD), Gender (male = m and female = f), Phone number, Nurse Code, Date of Admission, and Room. A "vibegs id online" status indicator is visible in the bottom right corner of the browser window.

For outpatient:

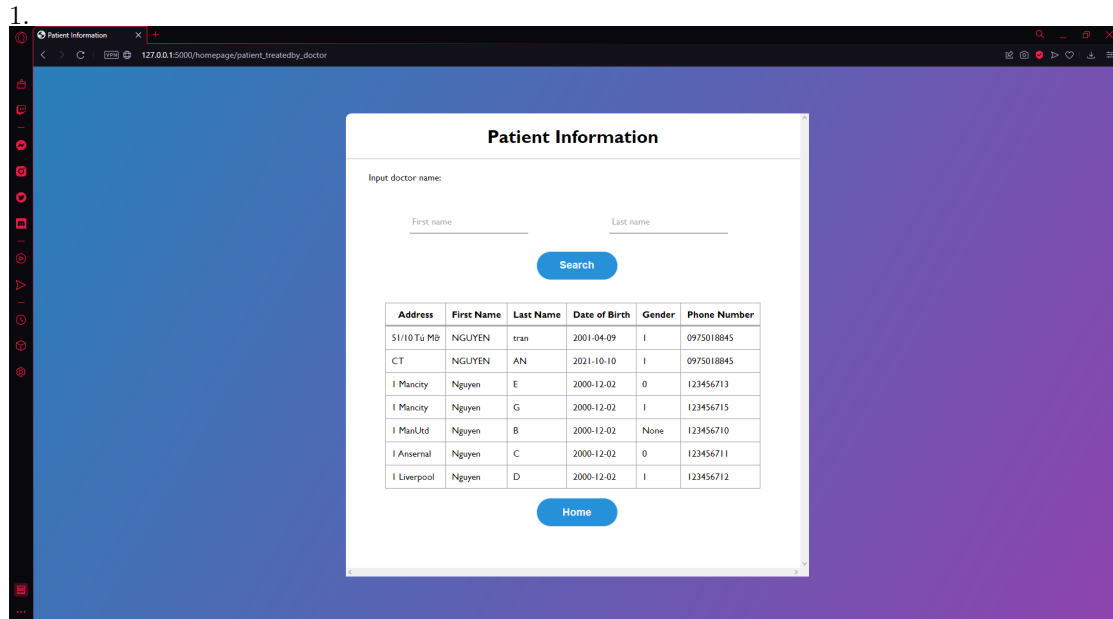


The screenshot shows a web browser window with the URL `127.0.0.1:5000/homepage/add_outpatient`. The page has a blue and purple gradient background. A white form titled "Add Outpatient Information" is centered. The form contains the following fields: Patient code, First name, Last name, Address, Date of birth (YYYY-MM-DD), Gender (male = m and female = f), Phone number, and Doctor code. At the bottom of the form are two buttons: "Add" and "Home".

3. List details of all patients which are treated by a doctor.

By providing the name of the doctor, the system will filter all the inpatient who is treated by the doctor. The information of the inpatient is also the same as the information provided in option

1.



Patient Information

Input doctor name:

First name Last name

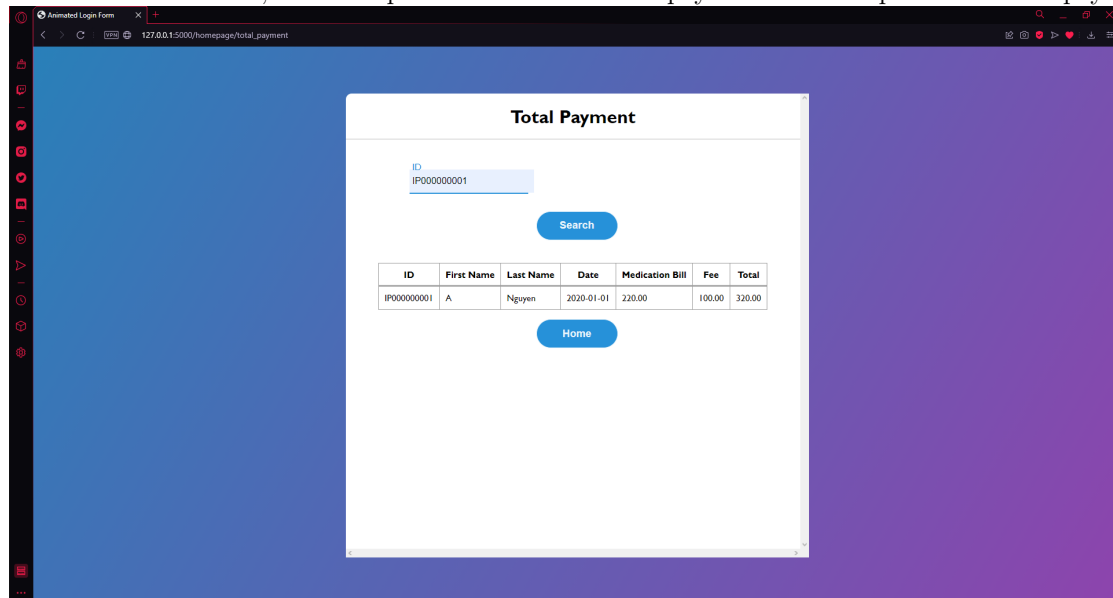
[Search](#)

Address	First Name	Last Name	Date of Birth	Gender	Phone Number
51/10 Tg MB	NGUYEN	tran	2001-04-09	I	0975018845
CT	NGUYEN	AN	2021-10-10	I	0975018845
I Mancity	Nguyen	E	2000-12-02	0	123456713
I Mancity	Nguyen	G	2000-12-02	I	123456715
I ManLtd	Nguyen	B	2000-12-02	None	123456710
I Arsenal	Nguyen	C	2000-12-02	0	123456711
I Liverpool	Nguyen	D	2000-12-02	I	123456712

[Home](#)

4. Make a report that provides full information about the payment for each treatment or examination of a patient.

In this function, we will provide the information about the medication, also the fee for treatment or examination. Then, we sum up these to calculate the payment that the patient have to pay.



Total Payment

ID

[Search](#)

ID	First Name	Last Name	Date	Medication Bill	Fee	Total
IP000000001	A	Nguyen	2020-01-01	220.00	100.00	320.00

[Home](#)