You will add some more and edit where is needed.

- **About Doctor Patient Portal**

  Doctor Patient Portal is a system designed to help doctors handle their "paper" work more efficiently and also help patients to book appointments and view their medical progress. This system allows the doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The system manages the appointment data for multiple doctors for various date and times. Each time a user visits a doctor his/her medical entry is stored in the database by doctor. Next time a user logs in he may view his/her entire medical history as and when needed. At the same time a doctor may view patients' medical history even bore the patient visits him. This allows for an automated patient doctor handling system through an online interface. Our system also consists of organ donor module. This module allows for organ donation registration as well as organ search. The module is designed to help urgent organ requirements through easy/instant searches.

- **The development processes and System Analysis**

  The system is broken into modules, namely

  - ❖ **Admin Login:** The system is under supervision of admin who manages the bookings made.
  - ❖ **User login/registration:** Users have to first register themselves to login into the system.
  - ❖ **Medical History:** System allows to update and view patient medical history.
  - ❖ **Doctor Search:** System allows for doctor search through categories, name and location.
  - ❖ **Appointment availability check:** User can click on spaces to view the availability.
  - ❖ **Appointment booking online for date and time:** Users can book appointment for their required date and time.
  - ❖ **Booking cancellation:** User may even cancel their bookings by login into the system any time.
  - ❖ **Email on appointment booking:** When user is successful in appointment confirmation and 'thank you' email regarding the slot booked.
  - ❖ **Feedback:** The system has a feedback form, where user can provide feedback into the system.

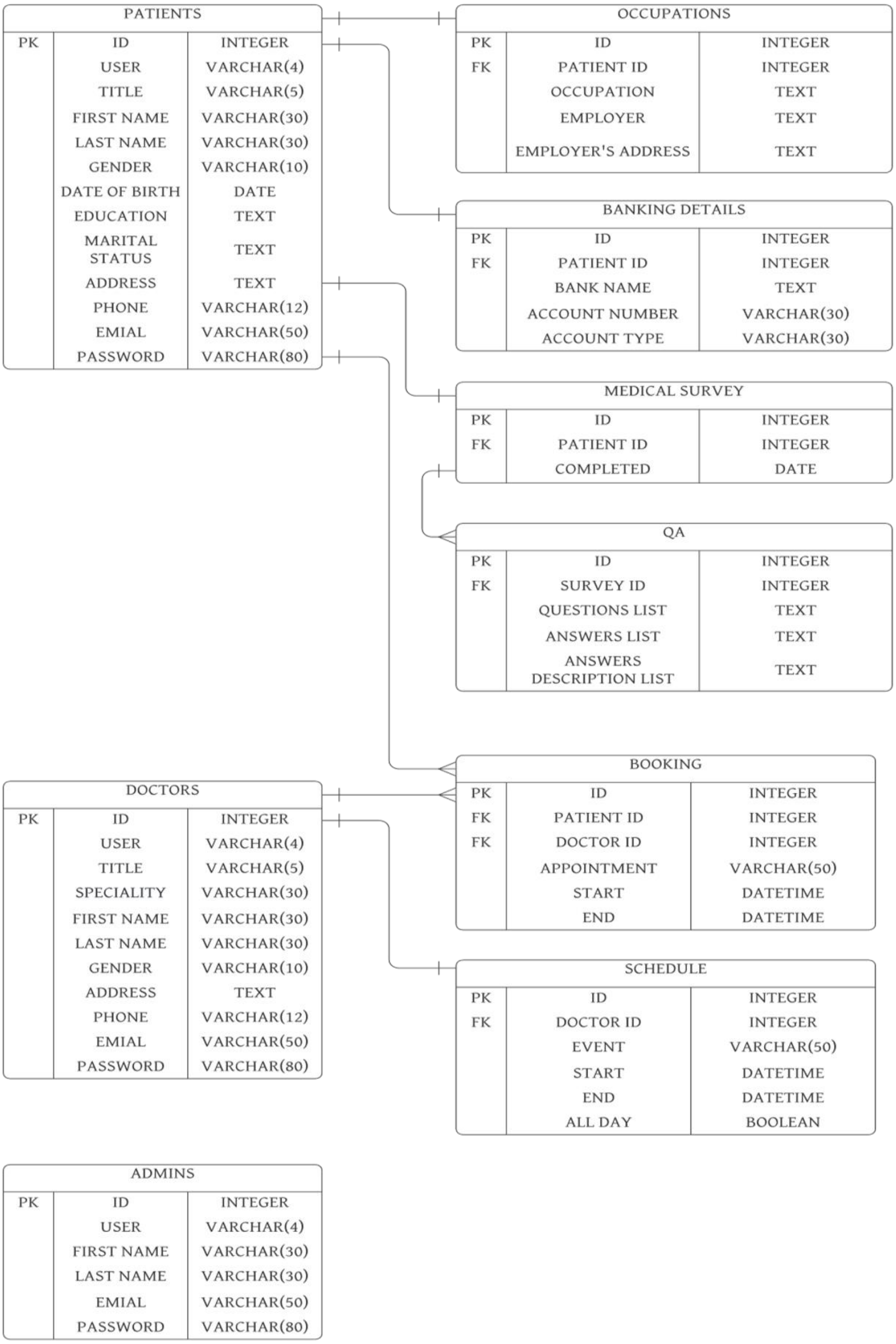
- **System Design**

  The system was designed or made using the following tools

- **Python Programming Language for back-end programming, query handles, database connections, and also provided the Flask framework.**
- **FLASK framework was used as a bridge between the back-end and front-end of the system.**
- **Jinja2 is a modern and designer friendly templating language for Python.**
- **And for the System User Interface JavaScript, HTML and CSS were used to provide the signing up, logging in and all of the systems' forms and interactive features.**

- **Database**
    - **Entity Relationship Diagrams**

DOCTOR PATIENT PORTAL ERD



PATIENTS

PK	ID	INTEGER
	USER	VARCHAR(4)
	TITLE	VARCHAR(5)
	FIRST NAME	VARCHAR(30)
	LAST NAME	VARCHAR(30)
	GENDER	VARCHAR(10)
	DATE OF BIRTH	DATE
	EDUCATION	TEXT
	MARITAL STATUS	TEXT
	ADDRESS	TEXT
	PHONE	VARCHAR(12)
	EMIAL	VARCHAR(50)
	PASSWORD	VARCHAR(80)

OCCUPATIONS

PK	ID	INTEGER
FK	PATIENT ID	INTEGER
	OCCUPATION	TEXT
	EMPLOYER	TEXT
	EMPLOYER'S ADDRESS	TEXT

BANKING DETAILS

PK	ID	INTEGER
FK	PATIENT ID	INTEGER
	BANK NAME	TEXT
	ACCOUNT NUMBER	VARCHAR(30)
	ACCOUNT TYPE	VARCHAR(30)

MEDICAL SURVEY

PK	ID	INTEGER
FK	PATIENT ID	INTEGER
	COMPLETED	DATE

QA

PK	ID	INTEGER
FK	SURVEY ID	INTEGER
	QUESTIONS LIST	TEXT
	ANSWERS LIST	TEXT
	ANSWERS DESCRIPTION LIST	TEXT

DOCTORS

PK	ID	INTEGER
	USER	VARCHAR(4)
	TITLE	VARCHAR(5)
	SPECIALITY	VARCHAR(30)
	FIRST NAME	VARCHAR(30)
	LAST NAME	VARCHAR(30)
	GENDER	VARCHAR(10)
	ADDRESS	TEXT
	PHONE	VARCHAR(12)
	EMIAL	VARCHAR(50)
	PASSWORD	VARCHAR(80)

BOOKING

PK	ID	INTEGER
FK	PATIENT ID	INTEGER
FK	DOCTOR ID	INTEGER
	APPOINTMENT	VARCHAR(50)
	START	DATETIME
	END	DATETIME

SCHEDULE

PK	ID	INTEGER
FK	DOCTOR ID	INTEGER
	EVENT	VARCHAR(50)
	START	DATETIME
	END	DATETIME
	ALL DAY	BOOLEAN

ADMINS

PK	ID	INTEGER
	USER	VARCHAR(4)
	FIRST NAME	VARCHAR(30)
	LAST NAME	VARCHAR(30)
	EMIAL	VARCHAR(50)
	PASSWORD	VARCHAR(80)

❖ Database variable names

- **ID →** It stores the entity's id.

- **USER →** It stores the user type. There are three users here doctor, patient and admin.
- **TITLE →** It holds the user's title such as Mr., Mrs., Dr.
- **FIRST NAME →** It stores the user's first name.
- **LAST NAME →** It stores the user's last name or surname.
- **GENDER →** It stores the user's gender. It can have values M, Male or F, Female.
- **DATE OF BIRTH →** It stores the user's date of birth.
- **ADDRESS →** It stores the user's address.
- **PHONE →** It stores the user's phone number.
- **EMAIL →** It stores the user's email.
- **PASSWORD →** It stores the user's password.
- **EVENT/APPOINTMENT →** It stores the name or reason for event of appointment.
- **START →** It stores the date and time of which the event or appointment will end.
- **END →** It stores the date and time of which the event or appointment will end.
- **ALL DAY →** It stores True or False depending on whether the event takes all day or not.
- **COMPLETED →** It stores the date which the survey was taken.
- **OCCUPATION →** The person's accupation.
- **EMPLOYER →** The employer can be person's names or company name(s).
- **EMPLOYER'S ADDRESS →** It stores the employer's address.
- **BANK NAME →** It stores the name of the bank.
- **ACCOUNT NUMBER →** It stores the person's accoount number.
- **ACCOUNT TYPE →** It can have values like, "Savings, Cheque and Credit."
- **QUEATIONS LIST →** It is a list of questions of the survey.
- **ANSWERS LIST →** It is a list of provided answers.
- **ANSWERS DESCRIPTION LIST →** It stores the descriptions of provided answers.
- **DOCTOR ID →** It refferences the '**Doctors**' table's id.
- **PATIENT ID →** It refferences the '**Pateints**' table's id.
- **SURVEY ID →** It refferences the '**Medical Survey**' table's id.

- **System Overview**

  System requirements:

- Python Environment.
- SQLITE.
- WSGI Server for deployment(The development server comes with the Python Module: Flask)
- A web browser to run the system(i.e. Mizilla Firfox, Google Chrome)