Online Preliminary Medical Service Management

Project Team 17: Xuejiao Dong, Chang Ge, Xiaoyang Jin, Yuanhao Li, Jiqing Sun

Database Purpose:

In view of the current global epidemic of the COVID-19, online medical services can reduce the risk of infection to some extent. The purpose of the database is to maintain the data used to generate and support patients getting preliminary medical services online. It will be mainly used by both patients and doctors, and meet the information needs of appointment service, online diagnosis service, medical record management service, patient and doctor information management service, feedback management and other sub services.

Business Problems Addressed:

- Allow doctors and patients to easily complete the initial medical process without contact.
- Allow doctors, patients or administrators to obtain useful information. (e.g., patients can choose the right doctor through DoctorSchedule).
- Provide the necessary information for performance appraisal of the doctor. Allow the service manager to adjust online medical services through performance appraisal.
- Provide a complete medical record from booking to feedback which can help the administrator track the information efficiently.
- Allow doctors to analyze their patients' condition base on all treatment information of them and supply insight to effective treatment.
- Allow patients to order prescripted drugs directly from the platform.

Business Rules:

- Each account should belong to a doctor or a patient but not both.
- Each department may have one or more doctors.
- Each doctor can handle more than one symptom.
- Each patient has only one primary phone number stored.
- Each patient may have zero or more bookings.
- Each doctor may have zero or more doctor schedule.
- Each doctor can only treat one patient at a scheduled time slot.
- Each booking may have only one case associated with.
- Each booking may have only one consultant room associated with.
- Each booking may be associated with at most one scheduled time slot.
- Each case may be associated with at most one feedback.
- Each case may be associated with at most one electronic prescription.
- Each electronic prescription may have one or more drug prescripted.
- Each electronic prescription may have zero or one drug order.

Design Decisions:

Entity Name	Why Entity Included	How Entity is Related to Other Entities
Account	As an online preliminary medical service management platform, it is essential that everyone must have an account. Whenever a user (no matter a doctor or a patient) wants to use this platform, he/she must log into the account. It makes the management of user information easier. In an account, the account type is specified to indicate whether the user is a doctor or a patient. Some necessary attributes like username, birthday and gender are also stored here.	The primary key of this entity is AccountID. The attribute AccountID relates the entity to the Doctor Entity and the Patient Entity. The account should be either a doctor or a patient and there is no such account which has more than one user.
Doctor	As a medical service platform, doctors' information needs to be stored. Doctors should be registered under a department and doctors also have titles to somehow indicate the experience and position of the doctor.	The Doctor Entity is related to Account Entity, Department Entity, DoctorSchedule Entity and Symptom Entity. The primary key of this entity is DoctorID. Due to a many-to-many relationship, this entity is related to the symptom entity through an associative entity. It is common that a doctor can handle multiple symptoms and it makes sense that one symptom can be handled by more than one doctors. The primary key is also used to relate the entity to the DoctorSchedule Entity. One doctor can have no time slot or multiple slots. As for the relationship to Department and Account, there are DepartmentID and DoctorID as foreign keys to do the job.
Symptom	The Symptom entity is a very important entity for both patient and doctor. For patients, when they need medical services, they usually search the right doctor through symptoms. For doctors, different doctors are good at dealing with different symptoms. The classification and storage of symptoms information can help patients quickly find a doctor that suits them.	The Symptom Entity is related to the Doctor Entity through an associative entity (DoctorSymptomSpeciality Entity) due to many-to-many relationships. The primary key of this entity is SymptomID, which is used to relate the entity to the doctor entity. Each symptom can be dealt with more than one doctors, and each doctor can be good at dealing with more than one symptom.

Doctor Symptom Specialty	DoctorSymptomSpecialty is an associative entity, which stores symptoms that doctors can handle, helping to build a relationship between the symptom entity and the doctor entity.	The associative entity (DoctorSymptom Specialty Entity) is related to both the Symptom Entity and the Doctor Entity. It helps to solve the many-to-many problem between the Symptom Entity and the Doctor Entity. There're SymptomID and DoctorID as primary keys and foreign keys to maintain these relationships.
Department	Department is a basic entity in a medical service platform. It records departments' basic information and classify doctors into different groups. In this way, the hospital can manage doctors more efficiently, and patients can also find the right doctor through the department.	The Department Entity is related to the Doctor Entity. The primary key of this entity is DepartmentID, which is used to relate the entity to the Doctor Entity. A department includes one or more doctors, and a doctor must belong to a specific department.
Patient	Patient is also an essential entity in a medical service platform. A patient entity is relatively easy to store since all the common information is stored in the account entity. In the patient entity, the phone number used to contact the patient should be stored.	The Patient Entity is related to the Booking Entity and the Account Entity. It is related to the Account using the foreign key PatientID in it. The primary key of this entity is PatientID, it is used to relate it to the Booking Entity. A patient may have zero to multiple bookings.
Doctor Schedule	When the patient wants to book for an appointment, it is important for both the patient and the doctor to have available time. The doctor schedule can help this system to judge whether the doctor is available or not.	The DoctorSchedule Entity is related to the Doctor Entity and the Booking Entity. It is related to the Doctor Entity using the foreign key DoctorID in it. The primary key of this entity is ScheduleID. A doctor may have zero or multiple schedules.
Booking	Booking is the start of the online preliminary medical service. It allows the patent and the doctor to be able to have the appointment.	The Booking entity is related to the DoctorSchedule entity, the Case entity, the Patient entity and the ConsultationRoom entity. It is related to Case using the primary key BookingID, and related to the DoctorSchedule entity, the Patient entity and the ConsultationRoom entity using foreign keys ScheduleID, PatientID and Consultation RoomID. A patient may have zero or many bookings.

Consultation Room	Consultant room is where the patient and the doctor contact with each other. This room is a virtual room, like a zoom room. As a result, the doctor and the patient can enter this room through the room link at the start time	The ConsultationRoom Entity is related to the Booking Entity. It is related to the Booking Entity using the primary key ConsultationRoomID in it. A booking may only have one consultant room.
Case	Case is a very important document in the entire online medical service. It records the patient's condition and the doctor's advice. At the same time, the doctor will give an electronic prescription for the patient to facilitate the patient's recovery. Each case will have a unique CaseID for recording. Once a patient attends for a consultation, a case will be given by the doctor who serves for the patient.	An electronic prescription is associated with only one case using the primary key prescriptionID, but not every case has an electronic prescription associated with it. This is because the doctor will prescribe a prescription based on the patient's condition. If the condition is mild, there may not be a prescription. One-to-One relationship with the Booking Entity: Once a patient has completed an online treatment, there will be a unique case and it will be related to the Booking Entity directly using the foreign key BookingID. A feedback is associated with only one case using the foreign key CaseID, but not every case has a feedback associated with it. The reasons may be various: sometimes the patient forgets to give feedback, sometimes the patient is unwilling to give.
Feedback	Feedback is the last stage in online medical services. To some extents, feedback can be used to analyze the popularity of a doctor and the degree of satisfaction of patients with medical services. But feedback is not necessary, a complete medical service can be completed without it.	Zero or only one case is associated with a feedback using the primary key CaseID. The reasons may be various: sometimes the patient forgets to give feedback, sometimes the patient is unwilling to give. But once a feedback is generated, it must be linked to a certain case.

Electronic Prescription	Electronic prescription is a result of the doctor's diagnosis. The purpose of creating this ElectronicPrescription is to maintain the ElectronicPrescription information of every patient's Case, including PrescriptionID and Doctor's commends. So that we can query and update the electronic prescription information when we are processing patients' cases.	The ElectronicPrescription Entity's primary key, PrescriptionID, relates it to the DrugPrescripted Entity. One tuple of the ELectronicPrescription Entity is corresponded to many tuples of the DrugPrescripted Entity. All tuples of the ElectronicPrescription Entity are corresponded to at least one tuple of the DrugPrescripted Entity (Mandatory). The ElectronicPrescription Entity's primary key, PrescriptionID, relates it to the DrugOrder Entity. One tuple of the ElectronicPrescription Entity is corresponded to only one tuple of the DrugOrder Entity; Not all tuples of the ElectronicPrescription Entity are corresponded to at least one tuple of the DrugOrder Entity (Optional).
Drug	Drug management is a very important part of online medical service. The purpose of creating this drug entity is to maintain the basic information of drugs, including drugID, drug name, drug company and drug price. So that we can easily query and update the drug infomation when we are processing drug shipments and drug prescriptions.	The Drug Entity's primary key, DrugID, relates it to the DrugPrescripted Entity. One tuple of the Drug Entity is corresponded to many tuples of the DrugPrescripted Entity; Not all tuples of the Drug Entity are corresponded to the DrugPrescripted Entity (Optional).
Drug Prescripted	DrugPrescrited entity is an associative entity between Drug entity and ElectronicPrescription entity to solve the many-to-many problem between them. In detail, DrugPrescripted entity is used to maintain the specific prescription information of each drug in the electronic prescription, including DrugID, PrescriptionID, and DrugQuantity.	The DrugPrescripted Entity's foreign key, DrugID, relates it to the Drug Entity. One tuple of the DrugPrescripted Entity is corresponded to only one tuple of the Drug Entity. All tuples of the DrugPrescripted Entity are corresponded to one tuple of the Drug Entity (Mandatory). The DrugPrescripted Entity's foreign key, PrescriptionID, relates it to the ElectronicPrescription Entity. One tuple of the DrugPrescripted entity is correspond to only one tuple of the ElectronicPrescriptionEntity. All tuples of the DrugPrescripted Entity are corresponded to one tuple of theElectronicPrescription Entity (Mandatory).

DrugOrdered	The purpose of creating this DrugOrder entity is to maintain the information of drug order, including DrugOrderID, PrescriptionID， TotalPrice, OrderData and tracking number. So that we can query and update the drug order information and track the status of drug shipments.	The DrugOrdered Entity's foreign key, PrescriptionID, relates it to the ElectronicPrescripted Entity. One tuple of the DrugOrdered Entity is corresponded to only one tuple of the ElectronicPrescription Entity. All tuples of the DrugOrdered Entity are corresponded to one tuple of the ELectronicPrescriptionEntity (Mandatory).