2017年1月12日

Online Modeling by Drug Record and Drug Store APP

1641467 陈启源 1641472 高　屹

1641473 胡永豪 1641478 潘瑞峰

（姓名无先后顺序，按学号排）

1. Requirement

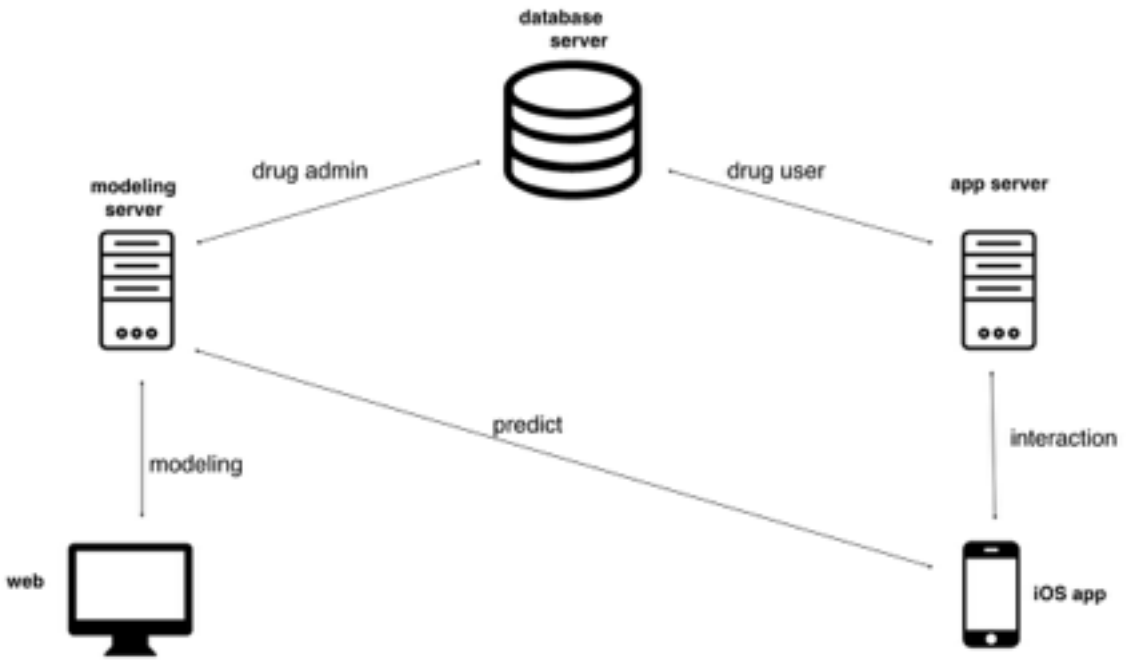
This project is aimed at design and implementation an Online Modeling by Drug Record and Drug Store APP to help people chooses a drug according to their physical status. The project has two parts : Modeling part and Drug Store App.

The Modeling part used to build a model using the drug record data and provide the restful api for Drug Store App to predict drug.

People can register an account and sign in the Drug Store App to predict what drug he or she can buy and buy it.

2. System Architecture

The system architecture as the diagram bellow:

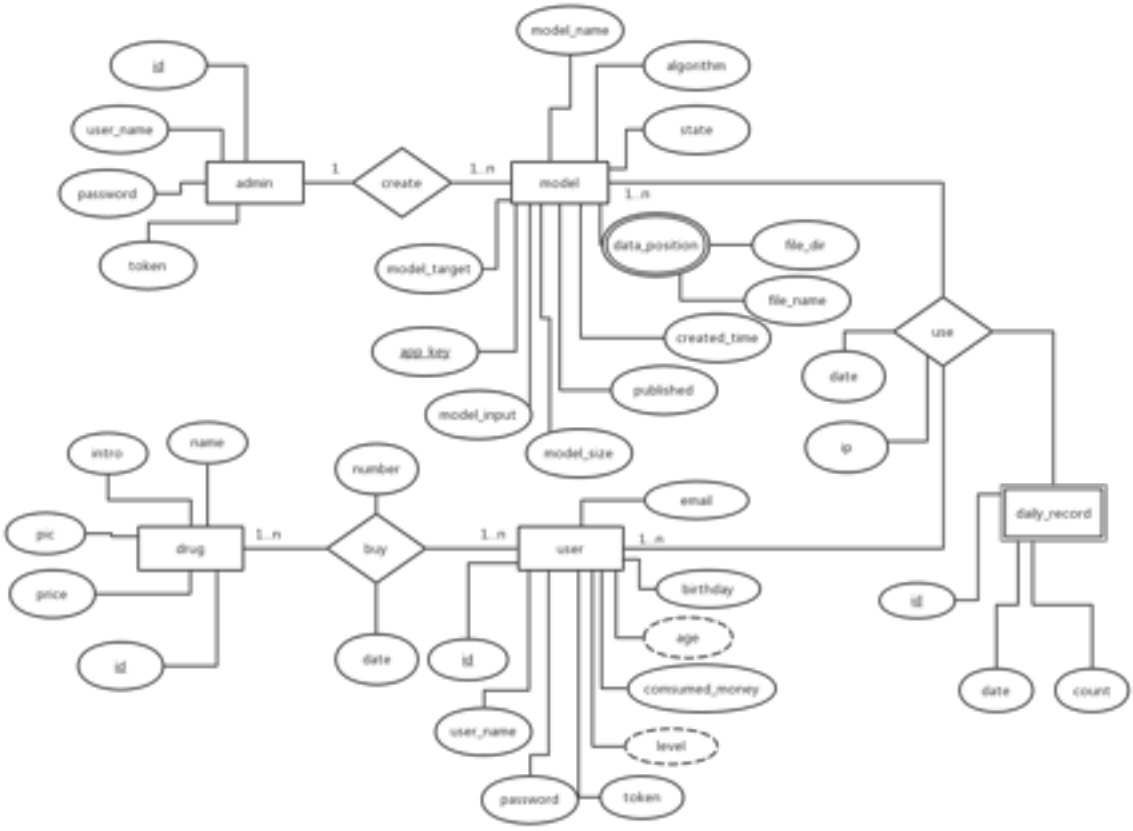


In the diagram we can see, there are two parts in our project, one is a web application, the other is an iOS application. The two part all have its own server and they share one database. The web application is an online modeling application, the admin use the web to create a model using drug record training data and the web can generate a restful api and iOS app call the api using the model to predict drug.

3. Database Design and Implementation

3.1 Database Design

The E-R diagram of our database as bellow:



3.2 Database Implementation

We in all have 7 tables:

We have two users in the system and iadmin table used to storage the information of admin users. Includes user name, password and token.

iadmin table

Attribute Name	Type	Description	Constraint
id	number(6)	The PK of the table,each user has an only id.	PK
user_name	varchar2(16)	The name of the admin user	unique, not null
password	varchar2(32)	The password of the admin user	not null
token	varchar2(256)	The UUID to identify unique user	unique, not null

imodel table used to storage the model information which used to predict the drug and it has many attributes.

imodel table

Attribute Name	Type	Description	Constraint
id	number(6)	The PK of the table, each model has an only id.	PK
model_name	varchar2(32)	The name of the model	not null
algorithm	varchar2(32)	The algorithm required for model training	not null
state	number(1)	The state of the model, default 0 means training, 1 means completed	not null
file_name	varchar2(256)	The file name of the train data	not null
created_time	date	The date of the model was trained	not null
published	number(1)	The model is private or public, 0 means private, 1 means private.	not null
model_size	number(6)	The size of the model, default is 0	
model_input	varchar2(256)	The input of the model	
model_target	varchar2(256)	The target of the model	
app_key	varchar2(256)	Which app used this model	
admin_id	number(6)	Which admin train this model	not null, FK—iadmin(id)

record table used to storage the model record, contains the app who used this model and the ip of the model.

record table

Attribute Name	Type	Description	Constraint
id	number(6)	The PK of the table	PK
app_key	varchar2(256)	The app who used this model	not null
idate	date	The date it was used	not null
ip	varchar2(32)	The ip of the model	

daily_records table used to storage how many times the predict has been call.

daily_record table

Attribute Name	Type	Description	Constraint
id	number(6)	The PK of the table	PK
app_key	varchar2(256)	The key of the app	not null
idate	date	The date to call the predict	not null
count	number(6)	How many time the model was call.	

drug table used to storage the information of drugs that we want to sold.

drug table

Attribute Name	Type	Description	Constraint
id	number(6)	The PK of the table, each drug has its only id	PK
name	varchar2(256)	The name of the drug	not null, unique
intro	varchar2(256)	The introduction of the drug	
pic	varchar2(256)	The picture of the drug	
price	number(16,4)	The price of the drug	

iuser table used to storage the user information includes user name passwords and so on.

iuser table

Attribute Name	Type	Description	Constrain
id	number(6)	The PK of the table	PK
user_name	varchar2(16)	The name of the user	unique, not null
password	varchar2(32)	The password of the user	not null

Attribute Name	Type	Description	Constrain
token	varchar2(256)	The UUID to identify unique user	unique, not null
ilevel	number(2)	The level of the user, buy more drug can improve the level. Default 0	not null
consumed_money	number(2)	How much has the user consumed in all	not null
birthday	date	The birthday of the user	
age	number(3)	How old the user is.	
email	varchar2(30)	The email of the user	not null check(email like '_%@_%._%')

buy_record table used to storage the record of the drug that was sold.

buy_record table

Attribute Name	Type	Descriptpion	Constraint
id	number(6)	The PK of the table	PK
idate	date	The date that the drug was sold	not null
user_id	number(6)	The id of the user who buy the drug	not null, FK—iuser(id)
drug_id	number(6)	The id of the drug	not null, FK—drug(id)
amount	number(6)	The number of the drug that was sold	not null

Beside the table, there are also some trigger, procedure, function and so on in our database.

There are two roles *drug admin and drug user* in our database, they have different authority. Drug admin can train an model and manage the database and also grant the authority to an user. The user can only use the app and do some related operation.

There is a view *user_buy_record* for the project, to view what user has bought.

For each table there is an id attribute. The "id" column will be set as the primary key and will be auto-incrementing. For each id we create a sequence and a trigger to implement the id auto-incrementing.

And also there are 3 logical triggers:

(1) *model_records_trigger* used when a user use model. It will insert a record into table record, and this will cause the record in table daily_records to add a record or count in daily_record add 1.

(2) *cal_age* used to calculate you age when you input or update your birthday.

(3) *update_user_level* used when user buy drugs,it will update user's consumed money and change his level.

Also there are two packages for the database:

(1) IDRUG : The package has the functions and procedures about the drug user, it includes one function and three procedures. The function is *login_get_token*, when drug user login the app, it will return a token. The procedures are *add_user, change_user_details* and *bug_drug.*

(2) ADMIN_COMMAND : The package has the functions and procedures about the drug admin. There are two procedures *exportDrugSales* and *exportUserBugRecord* in it, for drug admin to export the sale conditions and the user buy record.

Also there are 15 index in the database.

Type	Number
Roles	2, Drug User and Drug Admin
View	1, User_by_record
Sequence	7, for id auto-incrementing
Trigger	10, 7 for id auto-incrementing, 3 for logical use
Package	2, one for drug user, one for drug admin
Function	1, return user token
Procedure	5, in the two package
Index	15, To make query faster

4. System Implementation

Our project has three parts. Online modeling part, the drug store app and the database. In the online modeling part we use HTML5, CSS, JavaScript and AngularJS to implement the front-end and use SpringMVC and Hibernate to implement the back-end. For drug store app we choose iOS system and use iOS afnetworking, SpringMVC and hibernate to implement.

5. Test

For the test part, we choose a third party testing platform—travis CI. It uses github deposit and when we submit the code, the third platform will compile the code automatically. It can only check the compile error.