# Design pattern

Since we choose to use the Django as the python framework to build the final project. The basic design pattern behind the Django is MTV (Model-Template-View) which is similar to MVC.

To break it down, here’s a general overview of the MTV Concept and how they apply in the Django:

**Model:**

Just like the Model explanation in the MVC pattern, this also takes the same position as the interface or relationship between the data and contains everything related to data access and validation.

In Django, there will be a file called model which defined all the model.py which define all the structure of database.

**Template:** This relates to the View in the MVC pattern as it is the presentation layer that handles the presentation logic in the framework and basically controls what should be displayed and how it should be displayed to the user..

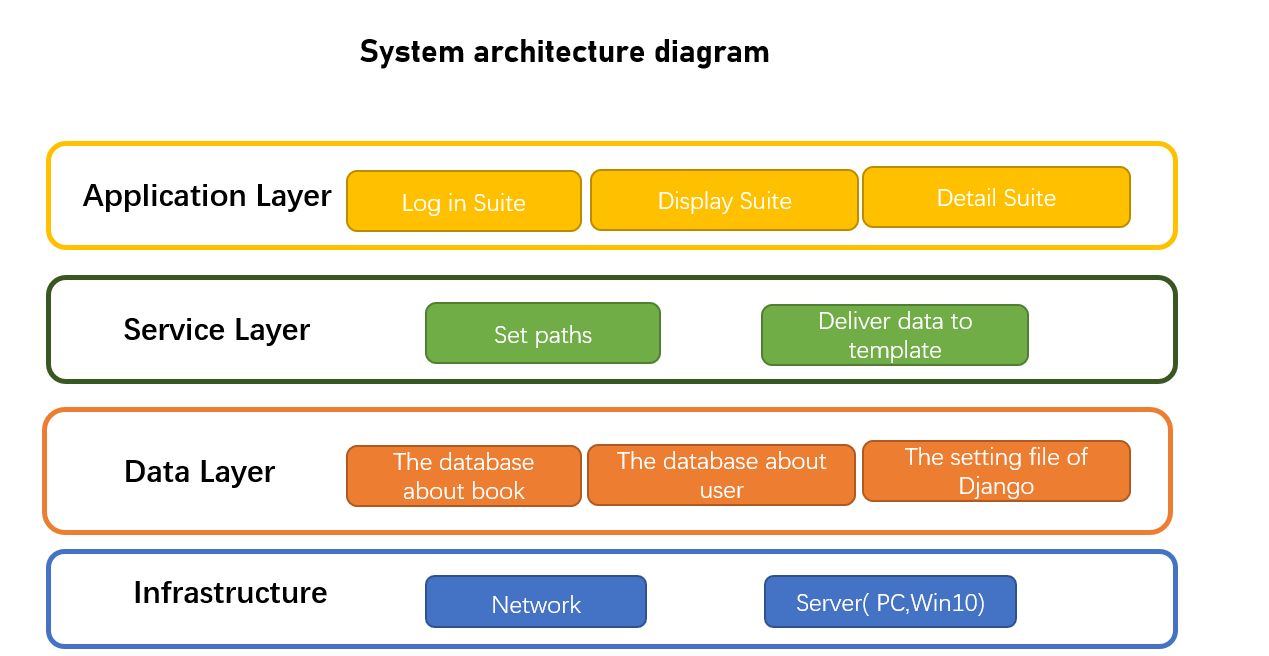
In Django, there will be a folder which is the symbol of template, You can write those template by using Django template language. Which will generated the html dynamically

**View:** This part relates to the Controller in the MVC pattern and handles all the business logic that throws down back to the respective templates. It serves as the bridge between the model and the template.

In Django, there are views.py and path.py which define how the data from the database can represent in the html tags.

# Architecture

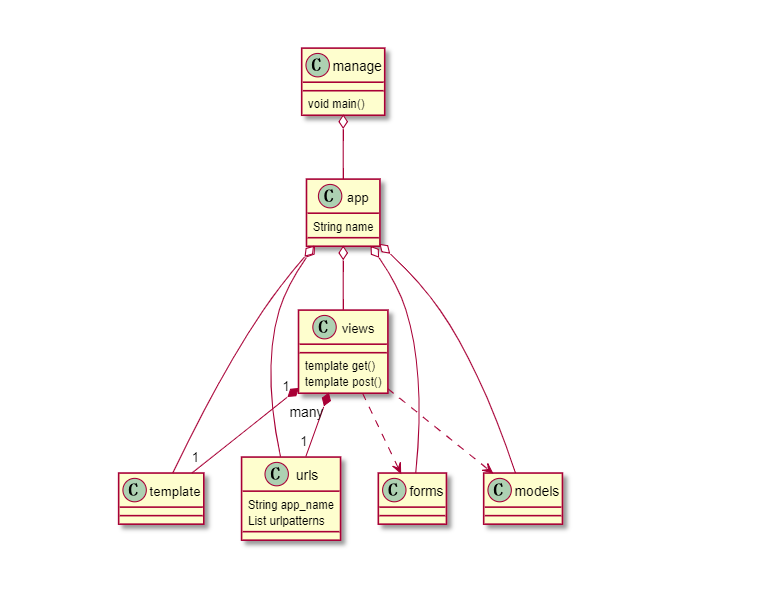
We divide the system into four layers. The basic layer is the infrastructure which contains the outside environment that the program can run. Our application require the server to run the computer. My development environment is Win10. And the running of the program doesn’t need the network. For the data layer, the data types are mainly the database and setting files. The data base is can be divided into the database about the book and the database about the user including their orders. The

 client run on the same

# Data model

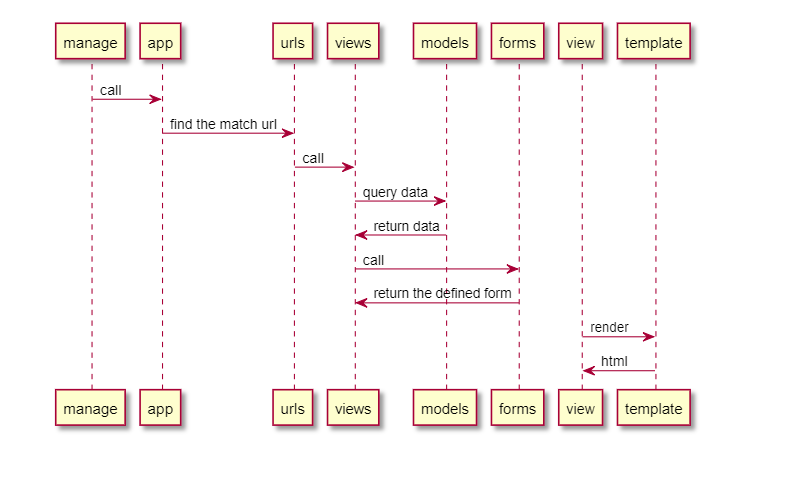
# Class model

There are so many classes in the whole project. So, we talk about them from high-level. All the project will have one manage.py which is the entrance of whole project. And one project could have many apps, but in our project, we just have one, so we can one apps. For every app, we have many views, templates, forms, urls, models. But we will explain it from abstract level. One urls defines the path of several views. So their relationship is one to many. And one template correspond to one url. It just works in my own project. Not all the Django projects has this relationship. It just works fine on my own project. What’s more views depends on the forms and the models to provide data.



# Sequential diagram

When we run the command “python manage.py runserver” . It will call the entrance of whole project whose name is manage. And manage will call the app.py from all the components of the project. When customer type an address in the address bar. They intend to find the match url defined in the urls. And they will call the corresponding views. Since views depends on the forms and models, it will require data and form .Finally it will render the template to generate the html that display in the front-end



# Deployment diagram