

Databases

Project Progress Report

Hongyi Liu
hl2638

- 1) Name of the Team Members:

Hongyi Liu

- 2) What technologies are you are using?

I'm using Django in Python with MySQL configured as the database. I'm following [Corey Schafer's tutorial on Django](#).

- 3) What's your plan to finish the project in time?

The deadline is May 13. There are still a bit more than 3 weeks to go. In the three weeks I'm going to implement views for airplanes, flights and tickets, views for managing them, and views for advanced queries such as getting top customers. I'm slightly behind schedule but as I've already built the framework the rest should be less bulky and more of implementation details.

- 4) Are you currently facing any problems?

No. I've just managed to solve some problems regarding custom user models, custom authentication, etc.

- 5) How many use cases have you completed so far?

Register and Login. A custom User Models extend Django's AbstractUser class. Three Models, AirlineStaff, BookingAgent and Customer, all have a one-to-one field that is attached to it, in order to achieve low coupling. I also created separate register and login views as well as customized authentication backends, so their unique login procedures can be implemented.

After clicking 'register' or entering the URL, the user will be prompted to choose to register as an Airline Staff, Booking Agent or Customer. The customized register form will generate a User Model and a corresponding Model (AirlineStaff, BookingAgent or Customer) that links to the User Model. Though to a large degree customized, some lower-level authentication, including hashing the password as a SHA-256, is done by built-in modules in Django.

Login is implemented by a custom login form and a custom authentication backend for each user type. The form specifies which credentials to enter for a type of user and the corresponding backend deals with authentication and getting the User object.

/ I have to say the design of users in the prompt is somewhat bad practice. The login credentials are not cohesive and it makes the three types of users very different. I had to re-implement the User Model and find a way to satisfy Django's principle,*

*where the Users have to have the same username field, and at the same time making it possible to log in as an airline staff with username and log in as a customer with email address. This means that what I have done is reinventing the wheel to some degree, and took me longer than I expected. */*

- 6) Did you create the tables in the databases? Do you plan to use any additional attributes?

As is mentioned above, I drew the common features of the users into a User Model and decoupled the user type-specific attributes into separate Models linked to the User Model. I can see the possibility to add more attributes as the development progresses.

- 7) Did you insert data in those tables?

During testing, I inserted a few airlines and registered a few user accounts. These are all I have so far.