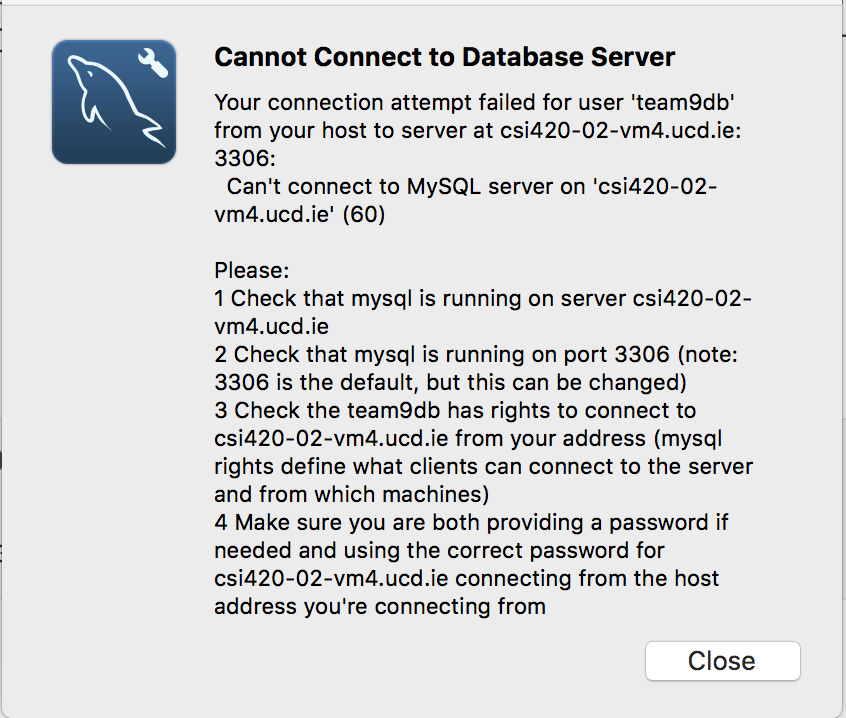
**Set up VM**

Set up virtual machine:

installed **anaconda**, set up a new condo environment called Team9 with Anaconda, used **Sudo** Command. In **Linux**, **sudo**(super user do) allows a system administrator to give certain users or groups of users the ability to run some or all commands as root while logging all commands and arguments.

**Mysql**, when I was connecting mysql to the Workbench, there was a problem with the connection

****

and then I installed iptable, provided by the Linux firewall, after that I can connect to the Workbench.

**Django**

**Topic:Django**

Plekhanova, J. (2009). Evaluating web development frameworks: Django, Ruby on Rails and CakePHP. *Institute for Business and Information Technology*.4-16

Django is a full-stack Python web framework that encourages rapid development and clean, pragmatic design”

**Support for JavaScript libraries**

Django, on the other hand, leaves to the developer the choice of a JavaScript library. “Django includes only a JSON module, leaving JavaScript code and the choice of a JavaScript library to the developer” [2]

**Usability of templates (views)**

Django includes a template language. Templates allow the respective frameworks to separate content from presentation. This paradigm and implementation of the (model-view-controller) MVC architecture is integral to each framework. Templates define the overall layout of the pages while individual page templates define the specific content of a certain page. The key difference between the frameworks is in the way they embed dynamic content.

One of the core reasons for Django’s inclusion of a template language is to separate application and business logic from presentation. There are several advantages to this philosophy, most of which stem from the resulting separation of designer and programmer. Consider:

1. Designers can’t break the application. Since they never touch the application code, there is no worry that a mistake by the designer is going to bring your app to a crashing halt
2. Designers have complete control to do literally anything they want with the content — and they can do it without having to bother the programmer at all.
3. Programmers aren’t futzing with an applica- tion’s design and front-end code. In short, this means no more cleaning up after your pro- grammer, who doesn’t seem to know that the <b> and <i> tags are deprecated.
4. Programmers can add to and change the appli- cation code without affecting the display” [6].

Further, the Django template language is not limited to HTML and makes it possible to create files in CSS, JavaScript, XML and other formats. If more advanced functionality is required (i.e. to print data in a 5 column table), the function is written in a view and pulled by a template. Therefore, Django template language stays clean and simple.

Django’s approach thus offers more advantages for large projects, in which designers can easily understand the language and work independently on layout from programmers.

**Content management**

Django has automatically generated administrative pages, which can save time when developing an application. A developer has to add a few lines of code to get a well functioning back end. Exhibits 2 and 3 show the Django and Rails default administrative interfaces.

In Django, a developer passes attributes and pa- rameters to the Django model classes and their fields. The generated pages have a range of options neces- sary for editing, to create, retrieve, update, and delete(CRUD) items in the model. Records can be also searched, filtered, and sorted. There is an option to look at the history page that lists all changes made to an object via the Django admin, with the timestamp and username of the person who made changes. In the environment where products have to go on the market fast, it means that once the administrative panel is up and running, a user can start adding the content through the back-end, while the developer can further work on setting up the views.

Automatically generated admin pages are help- ful and save time, however they support only simple relationships. For advanced functionality, developers can create a custom administrative panel, overriding the default one. Therefore, this feature is more useful for smaller projects and websites than for large web applications.

**Data management and migration**

Django has ability to create a database schema from the model definition in a model.py file. Currently, Django does not have a tool for database migration; however, Django developers are now work- ing on this problem trying to find the best solution. The advantage of Django is easy integration with legacy databases.

Django ORM’s ease of use, similar syntax to the rest of Django, and ability to handle 90% of SQL queries, makes getting started with Django easy. However, Django’s ORM does not handle edge cases and complex queries well, and one has to write SQL by hand to make it work. SQLAlchemy handles many of these edge cases already, and it has been the Python standard for Object Relational Mapping. SQLAlchemy is the default ORM for most web based and non-web based projects. Whenever the need arises for an ORM, SQLAlchemy is used. Although SQLAlchemy can be used in place of Django’s ORM, it is not the default choice, and still requires some hacks to get integrated. In the early stages of Django development, SQLAlchemy was not an option, so a homebrewed ORM was necessary. Today Django developers are working on a better bridge between SQLAlchemy and Django. Python-based web frameworks, such as Turbogears and Pylons, already support similar behavior. In addition, there is a new breed of databases on the horizon, a non-relational database, including CouchDB, Google App Engine, BigTable, Cassandra, and many more. Being able to plug and play between multiple database mappers, both relational and non-relational alike is a big plus for any web framework.

**Testability**

This measure defines the availability of debugging and testing tools as well as how easy and intuitive it is to solve development problems in each framework. Given that a significant aspect of web development is time spent on testing and debugging, it is important to assess how well a framework supports troubleshooting. This is especially important in the multi-platform, diverse, instantly global realm of web sites.

Django enable debugging in the development environment. When a web application encounters a problem, an error page is displayed. The page usually has a location of the error, the value of variables, a stack trace, and so on.

Inside the code and templates a developer may dump the values of variables. Django provides methods for converting variables to strings and including them in the HTTP response buffer. A log file is another tool that is used by developers and all three frameworks have an ability of creating one to trace through problems.

**Community and maturity**

“The Django website is very well organized and it is easy to navigate and get answers.”

**Conclusion**

The main advantages of Django are automatically generated administrative interface and simple templates which can be used by non-programming web designers.

**Topic: Nginx**

**Topic: Bootstrap**