

# **Ninja Typers Web Application**

## **Design Document**

**By**

**Marvin Farrell**

**C00141725**

## Table of Contents

|                              |    |
|------------------------------|----|
| 1. Introduction .....        | 2  |
| 2. Django Files .....        | 2  |
| 2.2. Project Files .....     | 2  |
| 2.3. Application Files ..... | 3  |
| 3. Use Cases .....           | 3  |
| 4. Context Diagram .....     | 4  |
| 5. Classes .....             | 4  |
| 4.1. Teacher .....           | 4  |
| 4.2. Student .....           | 5  |
| 4.3. Lesson .....            | 6  |
| 4.4. Payment .....           | 7  |
| 4.5. Certificate .....       | 7  |
| 4.6. Guidesheet .....        | 8  |
| 4.7. Typing .....            | 8  |
| 5. Software Stack .....      | 9  |
| 6. Database Design .....     | 9  |
| 7. Views .....               | 10 |
| 7.1. TypingApplication ..... | 10 |
| 7.2. Login .....             | 11 |
| 7.3. Register .....          | 12 |
| 7.4. Payment .....           | 13 |
| 8. Design Pattern .....      | 13 |
| 9. Cloud Platform .....      | 14 |
| 10. Appendix .....           | 14 |
| 10.1. Acronyms .....         | 14 |
| 11. References .....         | 14 |

## 1. Introduction

The purpose of this document is to highlight the different design aspect needed to implement this application. This document contains important information on files in the Django project as well as information on the classes that are contained in the project. A number of user interfaces have been included to show what the web pages and application will look like. This document is an overview of the typing application that is being developed for the Ninja Types club<sup>i</sup>.

## 2. Django Files

When a new Django project is created a number of key files are automatically created. Each of these files serves a very important purpose as well as providing a good way of separating different areas of functionality within the project. The same thing happens when an app is created with the project.

### 2.2. Project Files

“`django-admin.py startproject myproject`”<sup>ii</sup> is the command used to create the project named `myproject`. Below is a look at what is created after the command is used.

```
myproject/
  manage.py
  myproject/
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

- `myproject/`  
The outer `myproject` folder is the container for the whole project.
- `manage.py`  
This file is a command line utility that enables the user to make use of pre-built functions to interact with the Django project.
- `myproject/`  
The inner `myproject` folder is the actual Python package for the Django project. It is very important for importing files into the project (e.g. `myproject.settings`)
- `__init__.py`  
“This file is an empty file that is in place to let Python know that this directory is a Python package”.<sup>iii</sup>
- `settings.py`  
This file contains all the settings and configurations for this Django project. An example of the settings would be `TEMPLATE_DIRS` () which lets Django know of the location of the templates used in the project. Also `DATABASES` () contains information on the database used in the project. Information needed in this project are what type of database is being used, the database location and passwords.
- `urls.py`

This file is a way of separating all the urls for the project in one central location. The urls.py uses regular expressions and Python functions.

- wsgi.py  
“This is an entry-point for WSGI-compatible web servers to serve the Django project”.<sup>iv</sup>

### 2.3. Application Files

“python manage.py startapp myapp”<sup>v</sup> is the command used to create the application myapp. Below is a look at what is created after the command is used.

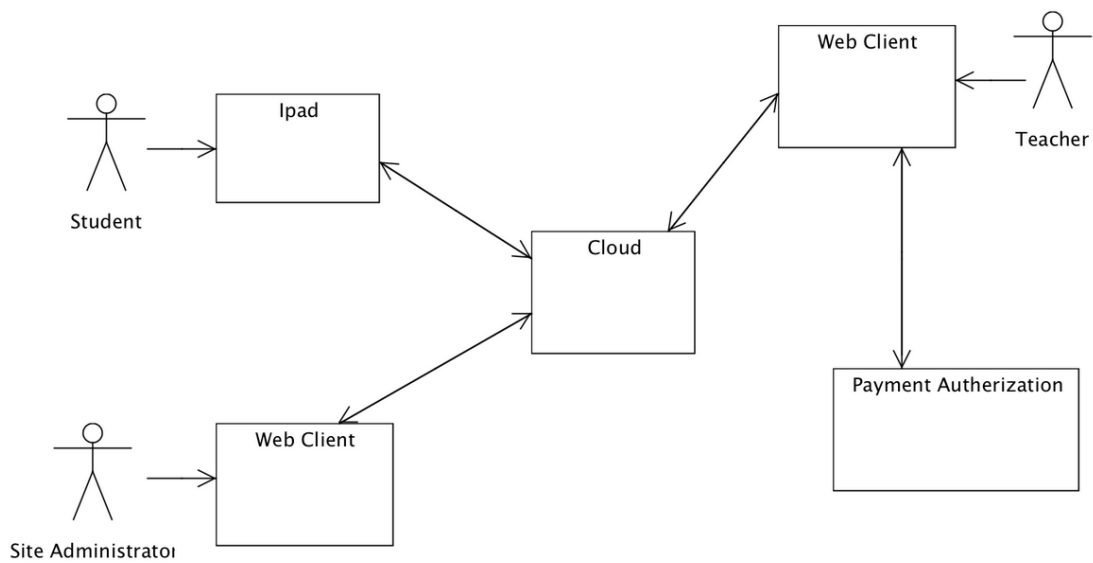
```
myapp/  
  __init__.py  
  admin.py  
  views.py  
  models.py
```

- myapp/  
This folder is used to contain all data and files for your application.
- \_\_init\_\_.py  
This file is an empty file that is in place to let Python know that this directory is a Python package.
- admin.py  
This file is used for administrative uses with the project and to access the administration part of the Django project.
- views.py  
This file is used to store all the views for the Django project. The views are the functions that display the different web pages for the project and also a way to access each one.
- models.py  
This file is used to store all the data that will be used in the project. This is the file that classes for the project will be stored. Each model is essentially a Python class and each model attribute represents a database field for the object that will be instantiated using the Python class.

## 3. Use Cases

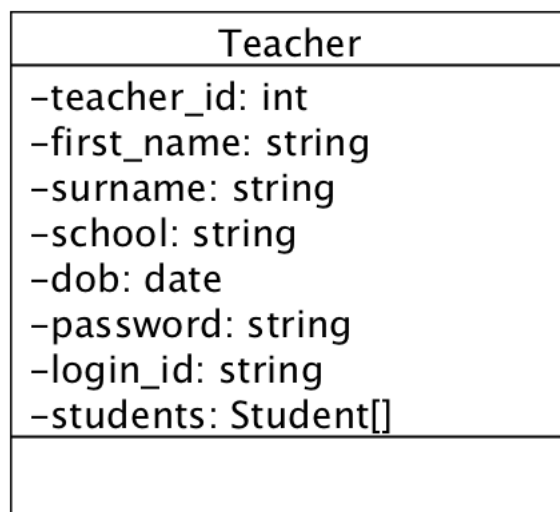
All information regarding use cases, detailed use cases and the use case model for this project can be found in the Functional Specification.

## 4. Context Diagram



## 5. Classes

### 4.1. Teacher



The Teacher class is used to create a new teacher object and add it to the database. The teacher will be responsible for the accounts of students enrolled on their course.

Attributes:

- **teacher\_id:** This attribute is of type integer that is unique to each teacher and is used as the primary key for the teacher object in the database.
- **first\_name:** This attribute is of type string and is used to store the first name of the teacher.

- surname: This attribute is of type string and is used to store the second name of the teacher.
- school: This attribute is of type string and is used to store the name of the school or institution that the application is being used in.
- dob: This attribute is of type date and is used to store the date of birth of the teacher.
- password: This attribute is of type string that must be greater than 6 characters that is used to store the password of the teacher.
- login\_id: This attribute is of type string and is used for the teacher to login with. It is the username for that teacher.

## 4.2. Student

| Student   |
|---|
| -student_id: string<br>-first_name: string<br>-surname: string<br>-school: string<br>-gender: bool<br>-password: string<br>-login_id: string<br>-teacher_id :string<br>-accuracy: double<br>-words_type: int<br>-wpm: int<br>-grade: string |
|   |

The student class is used to instantiate a student object that has been created by a teacher to use the application. The class is then added to the database.

Attributes:

- student\_id: This attribute is of type int and is used to store primary key which is used to distinguish between different Student objects stored in the database.
- first\_name: This attribute is of type string and is used to store the first name of the student.
- surname: This attribute is of type string and is used to store the surname of the student.
- school: This attribute is of type string and will be the same as the school attribute as the teacher object associated with the student object.
- gender: This attribute is of type bool and is used to get the gender of the student. E.g. true for male and false for female.

- password: This attribute is of type string and is used to store the password of the student for access to the system.
- login\_id: This attribute is of type string and is used to login to the system.
- teacher\_id: This attribute is of type string and is used to form the association between the student and teacher object. This is the foreign key for database use.
- accuracy: This attribute is of type double and is used to store the accuracy of the student while using the typing application.
- words\_typed: This attribute is of type int and is used to store the total amount of words typed while using the application.
- time\_typing: This attribute is of type double and is used to store the total time spent using the application.
- wpm: This attribute is of type int and is used to determine how many words can be typed per minute
- grade: This attribute is of type string and is used to keep track of what grade the student is currently at.

### 4.3. Lesson

| Lesson  |
|---|
| -lesson_id: int<br>-name: string<br>-grade: string<br>-pass_percentage: double<br>-certificate: Certificate<br>-guide_sheet: GuideSheet |
|   |

This Lesson class is used for the lessons that are stored on the database that the students will take when they are using the application.

Attributes:

- lesson\_id: This attribute is of type int and is used as a unique identifier for each lesson.
- name: This attribute is of type string and is used to name the lesson.
- grade: This attribute is of type string and is used to assign the grade to each lesson.
- content: This attribute is of type string and is the body of the lesson. This is what the students will be typing.
- pass\_percentage: This attribute is of type double and is the percentage value that the students accuracy of the current lesson must be above to advance to the next lesson.

#### 4.4. Payment

| Payment   |
|---|
| -payment_id: int<br>-paid_by: string<br>-amount: double<br>-students: string[]<br>-date_paid: date<br>-payment_type: string |
|   |

The payment class is used to information to payments for use of the application. The class will not store any information regarding credit cards but just who made the payment, when it was paid and the amount paid etc.

Attributes:

- payment\_id: This attribute is of type int and is used an unique identifier for the payment.
- paid\_by: This attribute is off type string and is used to denote who made the payment.
- amount\_paid: This attribute is of type double and is used for the total amount paid.
- students: This attribute is a collection of student names to associate with the payment.
- date\_paid: This attribute is of type date and is used to store the date of the payment.
- payment\_type: This attribute is of type string and is used to save the method used when the payment was made.

#### 4.5. Certificate

| Certificate  |
|--|
| -certificate_id: int<br>-name: string<br>-grade: string<br>-certificate: img |
|  |

The certificate class is for the certificate that the child using the application is awarded on the completion of a lesson of a certain grade.

Attributes:



- `certificate_id`: This attribute is of type `int` and is used as an identifier for the certificate. This will be the primary key in the database.
- `name`: This attribute is of type `string` and is used to name the certificate.
- `grade`: This attribute is of type `string` and is so that the relationship can be formed between the certificate and the lesson.
- `certificate`: This attribute is of type `img` and the image of actual certificate.

#### 4.6. Guidesheet

| Guidesheet   |
|--|
| - <code>guidesheet_id</code> : <code>int</code><br>- <code>name</code> : <code>string</code><br>- <code>grade</code> : <code>string</code><br>- <code>guidesheet</code> : <code>img</code> |
|  |

The Guidesheet class is for the guide sheet that is associated with each lesson. Each guide sheet contains finger positions and key colours.

Attributes:

- `guidesheet_id`: This attribute is of type `int` and is used as an identifier for each. This will be the primary key in the database.
- `name`: This attribute is of type `string` and is used to name the current guidesheet.
- `grade`: This attribute is of type `string` and is so that the relationship can be formed between the lesson and the guidesheet.
- `guidesheet`: This attribute is of type `img` and the image is just the guidesheet that is associated with that lesson.

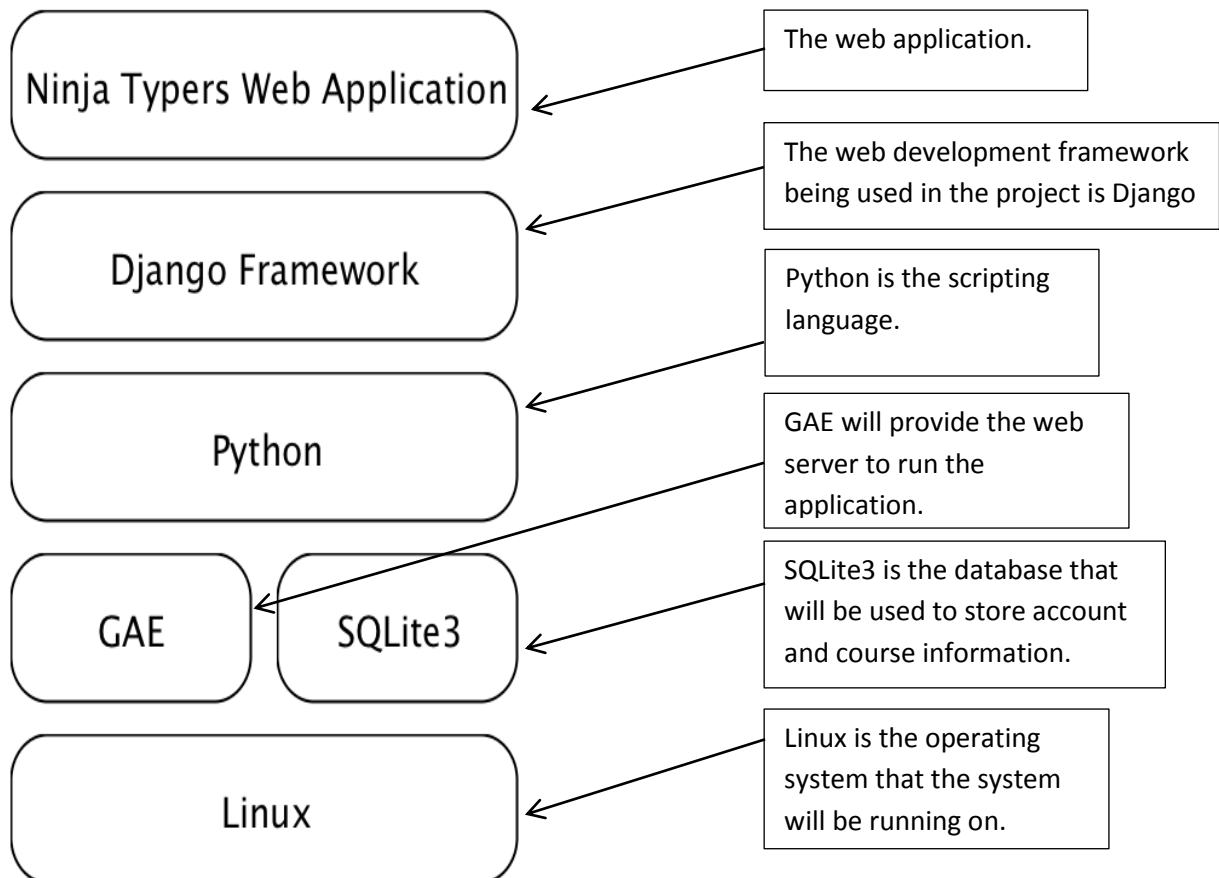
#### 4.7. Typing

The typing class is used to gather statistical information about a typing lesson.

Attributes:

- `accuracy`: This attribute is of type `double` and is used to get the accuracy of the lesson taken.
- `time`: This attribute is of type `double` and is used to get the time that it has taken to complete the current lesson.
- `words_typed`: This attribute is of type `int` and is used to get the amount of words that the student has typed in the lesson.

## 5. Software Stack



## 6. Database Design

Below is the design of the database for the information that it will store. This design may change when implementing the project.

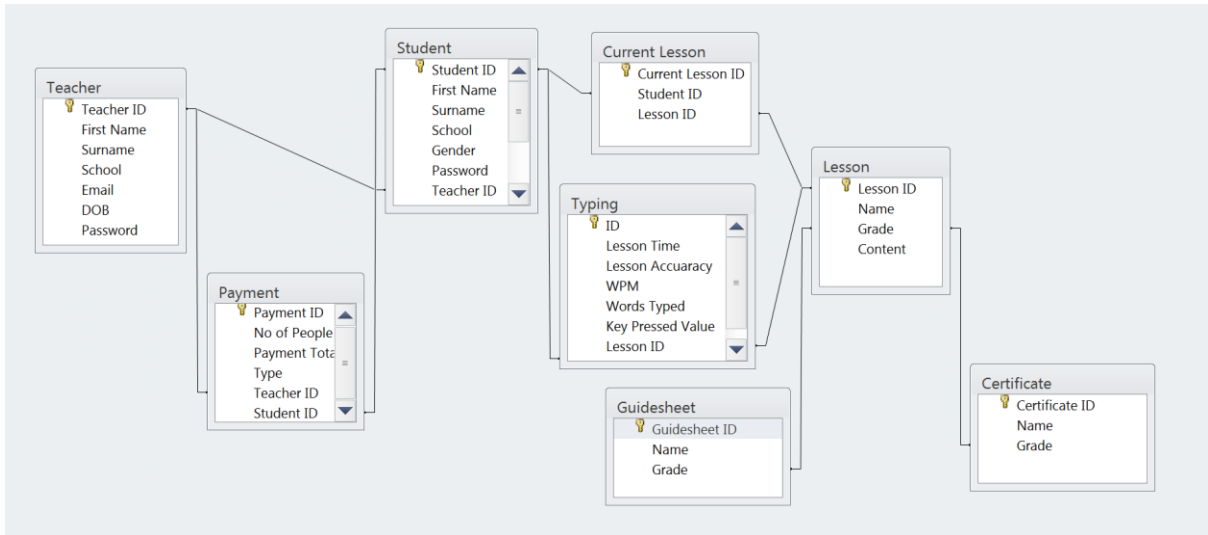


Figure 1 Database

## 7. Views

The views are the different web pages and user interfaces that are contained in the project. All the views and information relating to each view are stored in the views.py file that is located with the application folder of the project.

### 7.1. TypingApplication



## 7.2. Login

|   |   |                                |                                |                                  |                                  |                                   |
|---|---|--------------------------------|--------------------------------|----------------------------------|----------------------------------|-----------------------------------|
| <p>Ninja Typers Banner <small>800 x 160px</small></p>   |   |                                |                                |                                  |                                  |                                   |
| <p><small>420 x 380px</small></p> <p><u>Graphics + Benefits</u><br/>of<br/><u>Children Typing</u></p>   | <p><small>314 x 380px</small></p> <p>Log In <input type="text"/></p> <p>Password <input type="password"/></p> <p><a href="#">New User</a>   <a href="#">Forgot Password</a></p> |                                |                                |                                  |                                  |                                   |
| <p><small>420 x 380px</small></p> <p><u>Merchandise</u><br/><u>Graphics</u></p>   | <p><small>314 x 380px</small></p> <p><u>Grading</u><br/><u>Structure</u></p>  |                                |                                |                                  |                                  |                                   |
| <table border="1"><tr><td><small>Twitter</small><br/>Icon</td><td><small>Book</small><br/>Icon</td><td><small>Blogger</small><br/>Icon</td><td><small>Email</small><br/>Icon</td><td><small>60 x 60px (Icons)</small></td></tr></table> | <small>Twitter</small><br>Icon  | <small>Book</small><br>Icon    | <small>Blogger</small><br>Icon | <small>Email</small><br>Icon     | <small>60 x 60px (Icons)</small> | <p><small>314 x 181px</small></p> |
| <small>Twitter</small><br>Icon  | <small>Book</small><br>Icon   | <small>Blogger</small><br>Icon | <small>Email</small><br>Icon   | <small>60 x 60px (Icons)</small> |                                  |                                   |
| <p><u>Contact Us</u><br/><u>Phone No</u><br/><u>Email etc.</u></p>  |   |                                |                                |                                  |                                  |                                   |

### 7.3. Register

## Ninja Typers Banner 800 x 160px

**Teacher**

|                      |                      |               |   |
|----------------------|----------------------|---------------|---|
| First Name           | <input type="text"/> | Date of Birth | <input type="text"/>                                  |
| Surname              | <input type="text"/> | Gender        | <input type="checkbox"/> M <input type="checkbox"/> F |
| E-mail               | <input type="text"/> |               |   |
| School/ Organization | <input type="text"/> |               |   |
| Address              | <input type="text"/> |               |   |
|                      | <input type="text"/> |               |   |
|                      | <input type="text"/> |               |   |

**Students**

|           |   |               |                      |        |                            |                            |
|-----------|---|---------------|----------------------|--------|----------------------------|----------------------------|
| Full Name | <input type="text" value="Joe Bloggs 1"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |
| Full Name | <input type="text" value="Joe Bloggs 2"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |
| Full Name | <input type="text" value="Joe Bloggs 3"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |
| Full Name | <input type="text" value="Joe Bloggs 4"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |
| Full Name | <input type="text" value="Joe Bloggs 5"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |
| Full Name | <input type="text" value="Joe Bloggs 6"/> | Date of Birth | <input type="text"/> | Gender | <input type="checkbox"/> M | <input type="checkbox"/> F |

[+ Add another student](#)

## TERMS & CONDITIONS

have read and understand the terms and conditions  NEXT

## 7.4. Payment

# Ninja Typers Banner 800 x 160px

**Students**

|           |   |                  |               |
|-----------|---|------------------|---------------|
| Full Name | <input type="text" value="Joe Bloggs 1"/> | 6 Studentss      | 6 Ninja's     |
| Full Name | <input type="text" value="Joe Bloggs 2"/> |                  | €15 per Ninja |
| Full Name | <input type="text" value="Joe Bloggs 3"/> | Payment required | €90.00        |
| Full Name | <input type="text" value="Joe Bloggs 4"/> | Reimbursement    |               |
| Full Name | <input type="text" value="Joe Bloggs 5"/> | to School        | €9.00         |
| Full Name | <input type="text" value="Joe Bloggs 6"/> |                  |               |

I confirm above details are correct

**Organization bank details for refund**

|             |                      |              |                      |
|-------------|----------------------|--------------|----------------------|
| Account No  | <input type="text"/> | Swift Code   | <input type="text"/> |
| Sort Code   | <input type="text"/> | Account Name | <input type="text"/> |
| IBAN Number | <input type="text"/> |              | <input type="text"/> |
|             | <input type="text"/> |              |                      |

**Payment method**

|  |   |
|--|---|
| <input type="text" value="PayPal"/>          | When payment is received you will receive an e-mail confirming receipt of payment and your login details. |
| <input type="text" value="VISA Debit Card"/> |   |
| <input type="text" value="Fund transfer"/>   |   |

## 8. Design Pattern

Any project developed in Django adapts the MVC design pattern. Standard MVC pattern dictate that the Model contains all data that is being used in the application and will be stored on the database, the View is how information is displayed to the user and that the Controller controls how the information from the model is shown in the view. Although Django does adopt this pattern it is slightly different that the standard MVC pattern in that a Template in Django is how information the user. The Model is what data is to be stored in the database similar to the standard MVC pattern. The View in Django describes what information is to be displayed to the user. The framework itself is

said to be the Controller. Even though Django uses the MVC design pattern, this version of that pattern can also be MTV design pattern.<sup>vi</sup>

## 9. Cloud Platform

GAE<sup>vii</sup> is the cloud platform that will be chosen for the project. App Engine executes your Python application code using a pre-loaded Python interpreter in a safe "sandboxed" environment. Your app receives web requests, performs work, and sends responses by interacting with this environment.<sup>viii</sup> GAE also supports the use of large frameworks like Django and has extensive Python libraries. GAE app engine also supports JQuery which is the library which will be used for the JavaScript in the project.

## 10. Appendix

### 10.1. Acronyms

- MVC – Model View Controller
- MTV – Model Template View
- GAE – Google App Engine

## 11. References

---

<sup>i</sup> Lennon, K. (2012). *Ninja Typers*. Retrieved January 02, 2014, from Ninja Typers: <http://www.ninjatypers.com/>

<sup>ii</sup> *Tutorial*. (n.d.). Retrieved January 14, 2014, from Django: <https://docs.djangoproject.com/en/dev/intro/tutorial01/>

<sup>iii</sup> *Tutorial*. (n.d.). Retrieved January 14, 2014, from Django: <https://docs.djangoproject.com/en/dev/intro/tutorial01/>

<sup>iv</sup> *Tutorial*. (n.d.). Retrieved January 14, 2014, from Django: <https://docs.djangoproject.com/en/dev/intro/tutorial01/>

<sup>v</sup> *Tutorial*. (n.d.). Retrieved January 14, 2014, from Django: <https://docs.djangoproject.com/en/dev/intro/tutorial01/>

<sup>vi</sup> *FAQ*. (n.d.). Retrieved January 14, 2014, from Django: <https://docs.djangoproject.com/en/dev/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>

---

<sup>vii</sup> *Python Runtime Enviroment*. (2014, January 14). Retrieved January 14, 2014, from Google Developers: <https://developers.google.com/appengine/docs/python/>

<sup>viii</sup> *Python Runtime Enviroment*. (2014, January 14). Retrieved January 14, 2014, from Google Developers: <https://developers.google.com/appengine/docs/python/>