Jiaoling Chen

Field Experience Work Log—Girl Develop It Course Website

June 2nd

For a web development project, it is necessary to get familiar with Github since it is a great tool for version control and team collaboration. I am studying a Github tutorial for web designers at lynda.com.(Link: http://www.lynda.com/GitHub-tutorials/GitHub-Web-Designers/162276-2.html?org=unc2.edu)

We plan to start the project with Persona Creation. We will focus on three groups of users: 1. Chapter Leader 2. Teacher 3. Student. In order to understand the users' characteristics and needs, we planned to conduct interviews with these three groups. Thus, I asked Sarah and Sylvia via Slack to gain contacts of teacher Dan Lucas since he is going to teach the upcoming class on June 6th. And we can get contacts of students through Dan's upcoming class attendants.

I researched resources to learn about the general knowledge of the front end and back end websites technology. Two UNC course materials will be used as my tutorial: 1)COMP426 Web Development (Link: https://workflowy.com/s/afIlgGeZGL) 2)INLS672 Web Development (Link: http://ils.unc.edu/courses/2015_summerI/inls672_001/schedule.html).

June 3rd

Today I continued to learn git by watching Github Tutorial video and started to learn the two courses: COM426 and INLS672 by reading their materials online.

June 4th

Thanks Sylvia for emailing all local chapter instructors for asking help on this project. Three instructors (Daniel Lucas, Amy Gori and Autumn Welles) replied back offering to help. I scheduled separate time for Hangout video interviews with each person.

What I did today is to first learn about what Persona is. And this is a good website to start with: http://www.usability.gov/how-to-and-tools/methods/personas.html. From the reading, it mentions that the best practice for Person is to first conduct user research. We are on the right track! Asking questions/making surveys is important for such research. I came up with the following questions to ask the teachers for the Persona after reading related articles:

1. What is your job role in GDI?
2. Have your taught any class before at GDI?

3. (If the answer is Yes)
4. - What was the class about?
5. - What was your responsibility as a teacher?

6. - What materials did you have for the class?
7. - How did you manage these materials? Did you use any system to manage them?
8.
9. We are going to build a new website to store and manage those curriculum:
10. - what will you use this website for?
11. - What tasks do you expect to perform on this website as a teacher?
12. - What's your expectations about this website in its performance and functions?
13.
14. (If the answer is No)
15. We are going to build a new website to store and manage those curriculum, if you will be teaching one class:
16. - What topic do you want to teach?
17. - what will you use this website for?
18. - What tasks do you expect to perform on this website as a teacher?
19. - What's your expectations about this website in its performance and functions?

June 5th

I finished the Github Tutorial. Using git commands via terminal seems a better practice for me over Github local app. And it is more important for a team to learn to make branches out of master for independent work.

Jun 6th

Prepared interview questions for each group.

June 7th

Information Architecture: focuses on organizing, structuring and labeling contents in an effective and sustainable way. The goal is to help users find information and complete tasks. Involve: Organization Scheme(Taxonomy), Labeling System, Navigation Systems, and Search Systems. (Reading Link: http://www.usability.gov/what-and-why/information-architecture.html)

Ethnography techniques: participant observation (Reading Link: http://www.socialinnovator.info/process-social-innovation/prompts-and-inspirations/research-and-mapping/ethnographic-research-techniques)

June 8th

I am learning about Personas more in details now. I followed several readings from the INLS 718 User Interface Design course- Developing the user profile(s) topic section and here are some of my take-aways and notes:

1.

For General Guidance:

Personas: http://www.usability.gov/how-to-and-tools/methods/personas.html

2.
The comic strip mentioned (Link: http://www.smashingmagazine.com/2012/04/23/mental-model-diagrams-cartoon/) is quite helpful. It stresses that we should discover the underlying thoughts and feelings of the users, which can explain their behaviors. More people show trends in behavior. Group these behaviors, emotions and beliefs into blocks. Assign features to each block.

Mental Model: http://rosenfeldmedia.com/category/mental-models/

3.
User Expertise Stagnates at Low Levels: We should accept the fact that Learning is hard work, and users don't want to do it; they don't explore the user interface and don't know about most features. Advice mentioned encouraging user learning is helpful.

User Expertise Stagnates at Low Levels: http://www.nngroup.com/articles/stagnating-expertise/

4.
Quite a helpful reading about Creating Persona for web development projects

Personas: setting the stage for building usable information sites.pdf

5.
Additional reading mentioned inside is from Alan Cooper: The Inmates are Running the Asylum.pdf (to be read)

June 9th

I did one reading about Essential Use Case Model. (Link: http://ils.unc.edu/courses/2015_spring/inls718_001/materials/Ambler.2000.ws-tip-essentialuse.IBM.pdf)

Currently I am learning about browser-side and web-server side development and technologies following COMP 426 course(Material Link: https://workflowy.com/s/afIlgGeZGL). *Common Gate interface(CGI)* What is the Mechanism? (Link: http://en.wikipedia.org/wiki/Common_Gateway_Interface)

We did an interview yesterday with the instructor, Daniel Lucas, via Hangout Had recorded the interview and summarized an interview note(in the Person folder).

Reading notes and take-aways:
1.
Use interviews to explore users' general attitudes or how they think about a problem; The *critical incident* method is especially useful for such exploratory interviews: Ask users to recall specific instances in which they faced a particularly difficult case or when something worked particularly well.

User Interview: http://www.nngroup.com/articles/interviewing-users/

2.
We did an interview so far and Card Sorting seems to be applicable for information organization according to the reading.

User Experience Research Methods: http://www.nngroup.com/articles/which-ux-research-methods/

June 10th

We conducted a nice interview with Amy and summarized the contents afterwards.

June 11th

Today I attended one course from GDI, the Responsive Web Design. Through Participating a real class enables me to know more about the context of the course website since it will hold all the materials of these courses. We talked to the four to five students, the teacher, one local leader, and one TA there. I found that the needs of these users are quite different from we had expected. And the functionality of the website may be changing a bit, not simply for holding curriculum. The teacher is comfortable with using Github, and the students have no problems of the current materials-sharing approach so far since its not materials-intensive for them. Instead, they want to know more instructors. For the local chapter leader, nice information organization about the teachers and the courses they held as well as TA informations seems to be helpful. (I have made a summarization afterwards)

Some kinds of Pages can be involved:

- Suggesting sequence of the classes to take for beginners
- About career suggestions/ stories(Theres some youtube videos already posted for Women&Tech)
- Whats the other resources for learning?
- Whats the other organizations and websites that targets for younger user groups?(gdi targets for adult women)
- Provide different information/materials sections for different levels of students: beginners, intermediate users

June 12th

We used Qualtrics Survey Software to make a survey for student user group. The users are those who attended the Respond Web Design course. Survey link: https://unc.az1.qualtrics.com/SE/?SID=SV_d1jP2UWkpyF0TAN

We also interviewed one structor, Autumn, via Hangout, and make summarization afterwards.

June 13th

Made one Teacher Persona. Tool: OmniGrapphie

Made one User Case UML for the specification document.

June 14th

Communicate with Emma and Mandie via Google Doc.

First, we finished Instructor Persona. But for the student persona, we only got two feedbacks, still waiting for more answers. Second, we finished the tech specification document. We also paid special attention to the Accessibility requirements of the WCAG 2.0. What's more, I think UML is a good and super clear way to help explain the tech specification document.

One course website to learn from: The San fransisco Chapter Material Website: http://www.teaching-materials.org/

June 15th

Had a group meeting with Emma and Mandie. We made hand-writing wireframe draft and still have the following difficulties:

1. Unknown about the users preference of the page styling. If the courses list should be directly shown on the main page? or make the main page just as simple as it shows the general organization of the whole site?
2. Can not decide the primary user group. The students or the instructors? It can be reconciled in a sense for both users, but also sometimes the design can be a big difference.
3. Have to know some details of the GDI local chapter to better organize information in the website: the list of the instructors? the list of TAs? the list of chapter leader? Is there a course lists (by far we know they just schedule courses one month in advance, and the topic is quite random. It is difficult to summarize all the past course) The talk for the third question is as follows:

   eboettch

   @sarah @sylvia We had a few questions about classes offered - do you have estimates on how many topics you cover in classes/how frequently classes are repeated? And how early do you schedule upcoming classes?

   sylvia

   We try to publicly announce classes one months in advance. Which means we usually plan them 2 months out. We repeat classes as we have interest/teachers/space

   sarah

   operative word being ;try'

> sylvia
> We do Intro to HTML about 3 times a year; it is the most popular We usually but don't always follow intro HTML with intermediate HTML and intro JS Other classes are about once a year. The past Meetups is the best place to find an actual list. Classes have 1-4 sessions, depending on the topic. The standard topics are 8 contact hours, so we do them in 2 4-hour blocks or 4 2-hour blocks We have a handful of workshops, which are generally 2 contact hours. And @sarah teaches a UX class that is 6 contact hours, 3 sessions.

June 16th

Continue my study of the COM426 course online.

June 17th

Continue my study of the COM426 course online.

June 18th

Finished COM426 course, and continue with the study of INLS672 course.

June 19th

Continue with INLS672 course. The javascript and DOM part is really helpful with my current project. It allows dynamic interactions of a web app.

June 20th

Today we had a meeting with Sarah and Sylvia, the site local chapter leaders, to demo our Persona models and Tech Specification documentation,the website wireframe. It is necessary and more efficient to communicate with the clients about a detailed design plan of the website before implementation. Gladly, they liked the most of our ideas including the layout and the function design. Additional suggestions were offered. For example: the TAs part may not be included within the website design scope this time since it's not of priority right now.

Take-away: When designing websites for nonprofit organizations like GDI, marketing is not taken as priority. The main page design doesn't necessarily have to be marketing-oriented fancy and attractive. Instead, functionality styling prioritizes. Hence, we decide to keep the main page just as simple as it shows the general organization of the whole site.

June 21st

I want to do some practice before building up the django course website. I used a tutorial series, Real Python, which covers basic python programming , Web development in Python, detailed Flask framework, and Django Framework. Since

it is the first time for me to do such complete web app project, i feel it's necessary for me to try a simpler framework first to get familiar with idea of MVC model. Django projects logically use the MVC model too. I tried Flask, which is a light-weighted framework, and easy to build up for beginners.

June 22nd

I went through database programming following the tutorial of Real Python. One take-away is that though there exist convenient database processing packages that can help web developers to interact with database(I read an article that mentioned that some web developers can actually manipulate with database without knowing anything about SQL), it is still necessary to know about basics of SQL. One related blog to tell why: https://gun.io/blog/learn-sql/

June 23rd

I used Flask to build up a basic blog app that can show posts and add posts. This two basic functions will also be included in our website. I summarized the basic work flow of building such a MVC based project.

June 24th

I used Flask to build up a second app- Flasktaskr, which allows adding, updating, deleting tasks. I tried SQLAlchemy for database management this time. Also learned to do User Management, including User Registration, User login/ Authentication to improve the app. These contents will all be included in our website.

June 25th

I made one simple "Hello-World" django project. Read through some documentations of Django and get familiar with its syntax through practices.

June 26th

I attended one PyLadies RDU Meetup Django class: "Django: Beyond the blog". It focused on creating Models, Views, Templates, and URLS. It helps a lot for a more clear understanding of the logic between Models, Views and Templates, and how the parameters are passed within them.

June 27th

Continued with the reading and practice following Real Python

June 28th

Learned more about Javascript and jQuery from the Real Python tutorial.

June 29th

Learning Error Handling, which can make the programming and debugging more efficient.

June 30th

Django ORM Model

We had a group meeting and made some project configuration and database setups. Django projects use a Model-Template-Views architecture (MTV). Such model is logically the same as the Model- View- Controller (MVC) architecture. Model represents your data model/tables. Django ORM(Object Relational Mapping) is used for database organization and management. Templates visually present the data model. Views define the business logic(like the controller in the MVC), which logically link the templates and models. We created four objects for the Model: Course, Instructor, Material, and Topic.

July 1st

Django ORM Model

Today I went through the Model documentation section of Django Documentation, did data manipulation with the database, and refined some details of the models.

Take-away 1: Python classes are usually returned with __str__, but Django models use unicode by default; For python 2.7, we have to customize the models with the following code to populate the real data from the database: "def __unicode__(self): return self.title"

Take-away 2: We need to make migrations for the database every time we make changed to the models.

Take-away 3: Django has nice Fields like ImageField and FileFields that can help restore images and files within a database instead of pure metadata. URLField is also used, which is a subclass of CharField. Pillow and imagekit package can help more advanced image processing for django projects. I plan to use it later.

July 2nd

Django Template

Read through Django Templates Documentation and created HTMLs for the website pages using the Django template language. A template can have a parent or child templates, which ensures page styling conformity to some extent. When templates encounters tags,they use {% tag %}. When templates encounters variables, they use {{ variable }} and when the template engine encounters a variable, it evaluates that variable and replaces it with the result.

Practiced to build up the MTV architecture for the Django course website. Through the practice, the website can now successfully show data from the database through templates.

July 3rd

Had a group meeting with the team mates, Emma and Mandie for more programming. The website can now successfully show related data from the database in the course_list page , course_view page , instructor_list page, instructor_view page. Some details are still need to be refined.

July 6th

Django URL

URL issue: How to put slugs into URLs to make them more readable yet unique.

Solution: Set up URL to take a slug(course name as slug for Course object, instructor name as slug for Instructor object) and an integer(the primary key) as parameters. In the view, use only the integer primary key to access the specific page. The object's "get_absolute_url" object inserts both slug and primary key into the URL and make them fast, readable URLs. I think it is a workaround solution to ensure the uniqueness problem. The reference: https://wellfire.co/blog/fast-and-beautiful-urls-with-django/

The tutorial code:

`urls.py`

```
url(r'^(?P<slug>[-\w\d]+),(?P<id>\d+)/$', view=myviews.article, name='articl
e'),
```

`views.py`

```
def my_view(request, slug, id):
        article = get_object_or_404(pk=id)
        # everything else in your view
```

`models.py`

```
class Article(models.Model):
  @permalink
  def get_absolute_url(self):
        return ('article', (), {
            'slug': self.slug,
            'id': self.id,
        })
```

July 7th
Django's Admin Site

One of the most powerful parts of Django is the automatic admin interface. It reads metadata in your model to provide a powerful and production-ready interface that content producers can immediately use to start adding content to the site.
Admin Oberview: https://docs.djangoproject.com/en/1.8/ref/contrib/admin/
Admin Actions: https://docs.djangoproject.com/en/1.8/ref/contrib/admin/actions/

Basic codes implemented in the website:
1. Local admin-site url pattern in the project/settings: url(r'^admin/', include(admin.site.urls))
2. Register database object into admin site:
for example: for course object, go to course/admin.py:

```
from django.contrib import admin
from course.models import Course
admin.site.register(Course)
```

3. Create the superuser:

```
python manage.py createsuperuser
```

July 8th
Django URL and Templates

Task: Link one page to another within the website. The url for one specific course(course name: responsive-web-design) is like: http://127.0.0.1:8000/course/responsive-web-design-1/. I want to make course names/ instructor names clickable within pages so that one page can lead to another.

Strategy: Use slugify to link course name/ instructor name with '-'in the templates: <a href="/course/{{course.c_name|slugify}}-{{course.pk}}">; accordingly the url is set like this: url(r'^(?P<slug>[-\w]+)-(?P<pk>\d+)/$', views.course_view, name='course_view').
Ref :https://docs.djangoproject.com/en/1.8/ref/utils/

URL issue: when running the server, the meetup page links to url like: http://127.0.0.1:8000/www.meetup.com.

Solution: This is one problem caused in the templates:
{{course.meetup_page}}</td> leads to url like: http://127.0.0.1:8000/http://www.meetup.com
{{course.meetup_page}}</td> leads to url like: http://www.meetup.com/

July 9th
Django Form

Today I created Forms for course and Instructors. Able to create/read/update/delete(CRUD) plain texts/materials on the web page.
Reference Tutorial: http://code.techandstartup.com/django/stuff-app/

Django Form class: In much the same way that a Django model describes the logical structure of an object, its behavior, and the way its parts are represented to us, a Form class describes a form and determines how it works and appears.

In a similar way that a model class's fields map to database fields, a form class's fields map to HTML form <input> elements. (A ModelForm maps a model class's fields to HTML form <input> elements via a Form; this is what the Django admin is based upon.)

A form's fields are themselves classes; they manage form data and perform validation when a form is submitted.(Django Documentation: https://docs.djangoproject.com/en/1.8/topics/forms/)

July 10th
Django Creating/Reading/Updating/Deleting function realization

Today I updated two fields for the Instructor and Course object: the ImageField and FileField, and used Form Model in Django, in order to realize uploading/editing/deleting/creating function. Files including images and slides will be stored in the Media folder, once uploaded, they will have their own url address for direct view.

Had a meeting via Google Doc: discussed about the website styling, and have one discussion about Javascript or Django ModelForm to realize CRUD functions.

July 11th
The importance of setting up virtual environment

Since I forgot to activate the virtual environment during the development, all the packages are installed in my global Python installation. For development, you have to install related packages when using another laptop. When we work on a new project, we need to install a virtual environment, and ACTIVATE it, so that all the packages needed for the project will be installed separately within our virtual environment folder. In this way, anyone can have access to these packages on a different laptop.

How to solve it if we forgot to use the virtual environment:
Solution One:

I generated the requirements.txt(“ pip freeze > requirements.txt”), which shows all the packages in my global Python. You can use “pip install -r requirements.txt” to install the packages. It is quite important to set up virtual environment and activate them, so that all the packages specific for the project can be separated into one folder.
Solution Two(The right one):

1. create the environment: virtualenv - - system site packages address Or using the Preference Panel
2. activate the virtual environment
3. install the things use “pip install - -ignore-installed”/ “pip install -I”
4. That way pip will install what you've requested locally even though a system-wide version exists. Your python interpreter will look first in the virtualenv's package directory, so those packages should shadow the global ones. (Stack overflow reference: http://stackoverflow.com/questions/12079607/make-virtualenv-inherit-specific-packages-from-your-global-site-packages)

General steps of Virtual Environment SetUp and Activate:
You need to activate the virtual environment, after activation, any package that you install using pip will be placed in the venv folder, isolated from the global Python installation.
$ virtualenv venv
$ source venv/bin/activate
$ deactivate

July 12th
Django Authentication
Try1:

Tutorial: Django’s Out-of-the-Box Login: http://code.techandstartup.com/django/login/. Django's out-of-the-box Admin uses the Auth module included with Django. The relevant files (views.py, models.py, forms.py) are in the Python or Virtualenv folder at Lib/site-packages/django/contrib/auth/. All the work is done for you. You just need to dispatch the URLs and create the templates.
Try2:

Following Dango Django Tutorial: http://www.tangowithdjango.com/book17/chapters/login.html#setting-up-authentication. Compared to Try1, this version needs more implementation but more customized flexibility. And I decided to user this version since it makes the website scalable for later development.

There is one Problem that I can't solve right now about the Logged in status. Ideally, once the user logged in, the login tab on the navigation bar will change into the username of the user. To realize it, I added the "user.is_authenticated" to the templates to check if the user is logged in. It works for the lower-structure urls like instructor_view and course_view, however it fails to update the logged in status in the upper urls like course_list, instructor_list or getting started.

July 13th
Django Static and Media folders

Django distinguishes between "static" and "media" files. The former are static assets included with your app or project. The latter are files uploaded by users using one of the file storage backends. Django includes a contrib app, django.contrib.staticfiles for managing static files and, importantly, generating the URLs to them. (REF: http://www.effectivedjango.com/tutorial/static.html)

Combine CSS into the project :
How to insert CSS files:

1.
project/settings.py:
```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
   os.path.join(BASE_DIR,'static'),
]
```
2.
project/urls.py:
```
urlpatterns += static(settings.STATIC_URL, document_root = settings.STATICFILES_DIRS)
```
3.
base.html:
```
{% load staticfiles %}
 <link href="{% static 'css/styling.css' %}" rel="stylesheet" media="screen">
```

July 14th
Chosen jQuery

Chosen is a jQuery plugin that makes long, unwieldy select boxes much more user-friendly. I am unfamiliar with jQuery, so need to learn how to put Chosen into Django.
REF: https://harvesthq.github.io/chosen/