

**Software Engineering Design 2019 Group 2 Meeting Minutes**

Date	2019/10/15
Time	20:00 - 22:30
Location	CSIE building room 544
Facilitator	Chih-Hsuan Yen
Recorded by	Yifan Wu
Objective	Decide on how to breakdown web2py

**Participants**

Name	E-mail	Role	Present or not
Chih-Hsuan Yen	yan12125@gmail.com	Team Leader	Y
Chen-Hung Wu	ac791228@gmail.com	Team Member	Y
Po-Sheng Lin	b890052@gmail.com	Team Member	Y
Cheng-Jhih Shih	cs861324@gmail.com	Team Member	Y
Jui-Che Wu	ss900405twtw@gmail.com	Team Member	Y
Yifan Wu	eurekalilychou@gmail.com	Team Member	Y
Pin-Yen Huang	pinyentraffic@gmail.com	Team Member	Y
Wei-Jie Liang	jack55513608@gmail.com	Team Member	Y

**Meeting Agenda**

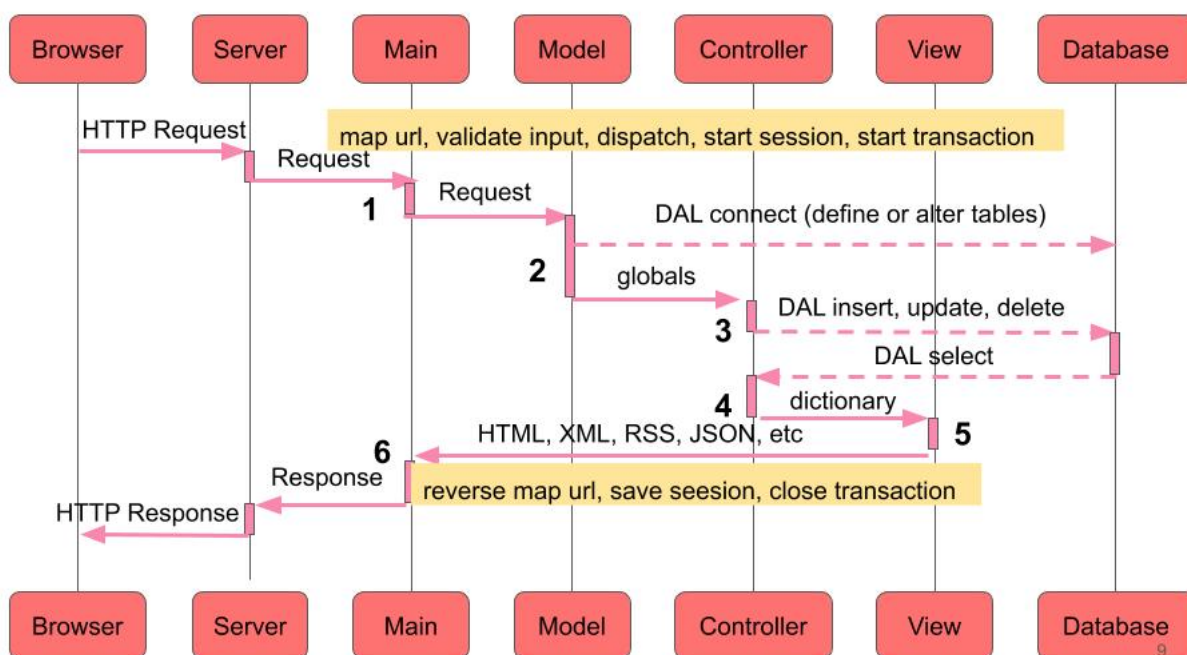
- Requests to responses
- How PyDAL works
- Overall structure
- Discuss the first quiz

- Overall structure

- Study each component of web2py and bring up a plan of breakdown
- The tool pdb can be used to track how an application runs line by line, similar to gdb for C/C++ programs. With that we can build a top-down walkthrough for web2py.
- Files to study
  - \* gluon/\*.py: the core of web framework
  - \* gluon/contrib/\*: additional useful modules, including modules for different databases (pymysql, pyodbc, etc.)
- dependency of files
  - \* Generated with the pydeps tool.
  - \* Can be used to identify files with similar or related functions, and thus helps WBS.

- Request-response flow

- Study the request-to-response workflow to determine the integral parts or modules of web2py
- Divide the interactions between components (loosely) into 6 steps
- gluon/rocket.py: contains the worker server object and worker thread class and subclasses that are responsible for receiving connections and will run an application to process the the connection
- gluon/rewrite.py: parse incoming URLs and format outgoing URLs
- URLs are mapped directly to Python modules and function calls, which return dictionary for rendering



- How PyDAL works

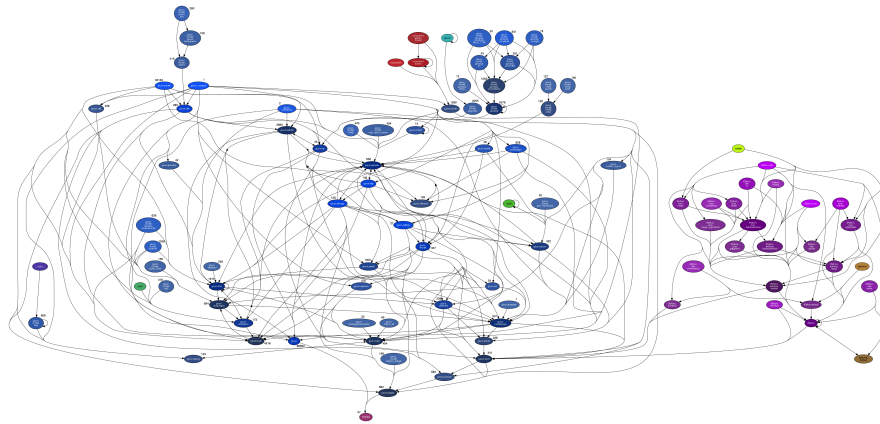
- Check if PyDAL plays an important role in web2py, so that we can determine whether we should read it
- Reasons for NOT NEEDED: PyDAL is a standalone package and has a set of well-defined APIs. It can be treated as a black box and we can understand how web2py works without investigating PyDAL.
- Reasons for NEEDED: if example applications are excluded from web2py, cloc reports only around 60,000 lines of code, so PyDAL should be included.
- Conclusion: include PyDAL for code reading.

- PPT

- Overall structure, request-response flow and pydal

## Issues

- Assign codes to team members
  - Two stages: core modules first, then extensions and addons
  - Split the web framework into multiple parts by functionality HackMD document
  - according to personal preference



- References
  - Github
  - Official website
  - Tutorial
  - Complete reference manual

Action Items					
No.	Action Items	Responsibility	Deadline	Status	Remark
1	Survey Grails	Chih-Hsuan Yen, Yifan Wu	10/1	<b>Closed 10/1</b>	
2	Survey Web2py	Po-Sheng Lin, Chen-Hung Wu	10/1	<b>Closed 10/1</b>	
3	Survey DropWizard	Cheng-Jhih Shih, Pin-Yen Huang	10/1	<b>Closed 10/1</b>	
4	Survey Nutch	Jui-Che Wu	10/1	<b>Closed 10/1</b>	
5	Requests to responses	Billy, Yifan	10/8	<b>Closed 10/8</b>	
6	How PyDAL works	WJ, WuCH, Jui-Che Wu	10/8	<b>Closed 10/8</b>	
7	Overall structure	Chih-Hsuan Yen, Pin-Yen, Cheng-Jhih	10/8	<b>Closed 10/8</b>	
8	Overall structure of PyDAL	WJ, WuCH, Jui-Che Wu	10/15	<b>Closed 10/15</b>	
9	Function of gluon/*.py	Pin-Yen	10/15	<b>Closed 10/15</b>	
10	Function of gluon/contrib/[a-o]*.py	Chih-Hsuan Yen	10/15	<b>Closed 10/15</b>	
11	Function of gluon/contrib/[p-z]*.py	Cheng-Jhih	10/15	<b>Closed 10/15</b>	
12	scheduler.py	WuCH, Yifan	10/29	<b>Ongoing</b>	
13	rocket.py	Po-Sheng Lin Pin-Yen	10/29	<b>Ongoing</b>	
14	rewrite.py	Cheng-Jhih	10/29	<b>Ongoing</b>	
15	widget.py	Jui-Che Wu	10/29	<b>Ongoing</b>	
16	sql.py	Wei-Jie Liang	10/29	<b>Ongoing</b>	
17	html.py	Chih-Hsuan Yen	10/29	<b>Ongoing</b>	