

Sprint Planning Meeting 11/04/2010

[18:55] <joran> so i dont see alot of differences in capabilities between python and php

[18:56] <joran> seems like php documentation is a little better

[18:56] <joran> and php structure is closer to c/c++

[18:56] <joran> so it will probably be easier to pick up for people who have c/c++ than python

[18:56] <joran> but thats really the only argument against python that i have

[18:57] <joran> plus i dont know if you need to do special server stuff for it. ... php is easy to setup on a server (if it doesnt already have it) which is a benifit but not a huge one

[18:57] <Stephen\_F> I already know PHP, but I'm down for whatever

[18:57] <joran> yeh thats my case php is real simple and im real good with it

[18:58] <joran> python doesnt look too bad at all from my PoV but im worried about some other people in the class w/ python

[18:59] <joran> but i wouldnt mind doin it in python (I think we should avoid django though... customizing it is a bit of a pain if you arent using blocks that someone else already built... at least how i remember it...)

[18:59] <joran> web2pi or whatever might work

[18:59] <joran> theres a handful of CMS's for python

[19:00] <mike\_s> true. i think about half our group doesn't know php. including me

[19:00] <joran> (I still like codeigniter (but thats more a framework than an cms)... and its php)

[19:01] <joran> i think alot more people are in class who dont know python than that dont know php... but just my 2c

[19:01] <joran> and i think php has a structure much closer to c/c++

[19:01] <joran> which everyone knows

[19:01] == RevanTal [4b57ff00@gateway/web/freenode/ip.75.87.255.0] has joined #uidaho\_cs383

[19:01] <joran> lol cut n past for jake?

[19:01] <mike\_s> you can use a c/++ like style in python

[19:02] <RevanTal> what did i miss?

[19:02] <Stephen\_F> Everything!

[19:02] <Stephen\_F> Nothing

[19:02] <RevanTal> aww...

[19:02] <joran> lol

[19:02] <Stephen\_F> We're just talking PHP vs Python vs ???

[19:02] <joran> like i said im down for it and it seems simple enough to me

[19:02] <joran> mostly those 2 ...

[19:04] <joran> did u look at both code examples jake? since you dont know either?

[19:05] <RevanTal> more or less yeah

[19:05] <sasha\_\_> and the decision is???

[19:05] <joran> one look more manageable than the other to you?

[19:05] <Stephen\_F> \*drumroll\*

[19:05] <mike\_s> did you see the light?

[19:05] <RevanTal> I liked the output of python better, but I suspect that has to do more with the fact that you were doing text output, and the python just looked nice ;)

[19:05] <RevanTal> but other than that, they both looked like things I didn't know, and fairly strait forward to learn

[19:06] <joran> yeh i agree they both are fairly straight forward

[19:07] <mike\_s> so how many people are we missing... max, nathan....

[19:07] <RevanTal> from what you both have said, i'd almost say php because it seems to already be ready to go with the basic features we need, but mike says that python also has this, so i guess i'm pretty much good with either

[19:07] <RevanTal> max isn't really missing...

[19:07] == Sverk [4cb2a75b@gateway/web/freenode/ip.76.178.167.91] has joined #uidaho\_cs383

[19:07] <mike\_s> oh, are you to both on?

[19:07] <mike\_s> \*two

[19:07] <RevanTal> there ya go  
[19:07] <Sverk> wouldn't let me use Max... =/  
[19:08] <RevanTal> huh  
[19:08] <joran> #!/usr/bin/env python # -\*- coding: UTF-8 -\*- from cgi import escape  
import sys, os from flup.server.fcgi import WSGIServer def app(environ,  
start\_response): start\_response('200 OK', [('Content-Type', 'text/html')])  
yield '<h1>FastCGI Environment</h1>' yield '<table>' for k, v in  
sorted(environ.items()): yield '<tr><th>%s</th><td>%s</td></tr>' %  
(escape(k), escape(v)) yield '</table>' WSGIServer(a  
[19:08] <mike\_s> ouch... my eyes...  
[19:09] <joran> lol thats theyre sample web python example to see if its installed  
right  
[19:09] <Stephen\_F> ^^  
[19:09] <joran> <http://docs.python.org/howto/webserver.html>  
[19:09] <joran> about halfwaydown  
[19:09] <mike\_s> i thought we were using a framework either way? if we did that page,  
we'd have to worry about lots of solved problems  
[19:10] == NMazu [81651204@gateway/web/freenode/ip.129.101.18.4] has joined  
#uidaho\_cs383  
[19:11] <Stephen\_F> Is that all of us? Do we get to start laying down the law now? :)  
[19:12] <joran> i think so  
[19:12] <mike\_s> i think that is everyone, yes  
[19:12] <mike\_s> so, do we turn in the irc log as minutes?  
[19:12] <sasha\_\_> yep  
[19:12] <mike\_s> okay guys, step one:  
[19:12] <mike\_s> Sprint Review Meeting (16 minutes) Except on the first one, review  
the outcome of the past sprint. 8 minutes per team, in front of class. Demo finished  
deliverables. Graded.  
[19:13] <mike\_s> done? since this IS the first one?  
[19:13] <joran> yeh i think so  
[19:13] <Sverk> yep  
[19:13] <NMazu> Are we going to have to review the outcome of this one in class  
tomorrow?  
[19:13] <NMazu> Or is that like outcome over the week  
[19:13] <NMazu> ?  
[19:13] <RevanTal> i thought the last one was technically the 'first' one and that is  
why there was no review, because we hadn't done anything before that  
[19:13] <joran> end of sprint  
[19:13] <joran> i think its next thurs  
[19:14] <RevanTal> yeah  
[19:14] <NMazu> k, that makes sense  
[19:14] <mike\_s> all who think sprint is over, say aye  
[19:14] <joran> lol  
[19:14] <Stephen\_F> aye guess?  
[19:14] <joran> i doubt it  
[19:15] <mike\_s> step 2 also doesn't apply, since we didn't sprint anywhere yet  
[19:15] <NMazu> What about the other two parts?  
[19:15] <NMazu> Especially the third  
[19:15] <mike\_s> here goes: Sprint Planning (20 minutes) 12 minutes in your team,  
updating requirements and backlog. 4 minutes per team to negotiate with Dr. J on  
items committed to complete for the next sprint.  
[19:16] <NMazu> So are we basically planning for what we want to get done next week?  
[19:16] <mike\_s> i think so... this one we have those five additional steps... like  
figuring out how we meet  
[19:16] <joran> establish what fields are needed for groups and members  
[19:16] <joran> determine implementation methods  
[19:17] <joran> prototype some member/group functions (add/create/delete)  
[19:17] <joran> (me and mike pretty much did that :P code is fun :P)  
[19:17] <mike\_s> okay, according to this business, looks like we should start here:

[19:17] <mike\_s> Figure out in your team, the closest approximation to a daily scrum that you are willing to commit to. Scrums are <= 15 minutes. Approximation means a schedule that may not include every day, or every person every day. Approximation this year means virtual scrums via CVE, Skype or other tools some of the time.

[19:18] <joran> 7pm works for me pretty much any given day

[19:18] <Sverk> same here

[19:18] <joran> m/w/f ?

[19:18] <mike\_s> can we commit to meeting online in some form most days? or do we want to do MWF or something?

[19:18] <NMazu> I think MWF is adequate, but wouldn't be opposed to more often

[19:18] <joran> i vote m/w/f

[19:18] <Sverk> MWF works for me

[19:18] <joran> but if need be we could meet more often

[19:18] <NMazu> We can increase it later if we decide it isn't enough

[19:19] <joran> but we gotta time box those meetings well

[19:19] <mike\_s> so, MWF at 7:00 sharp for 15 minutes until decided otherwise?

[19:19] <joran> i dun want 15 min scrum turning into an hour all the time :P

[19:19] <NMazu> Yeah, sounds good

[19:19] <Sverk> yep

[19:19] <RevanTal> yup

[19:19] <joran> aye aye matey

[19:19] <Sverk> はい, yay it works

[19:20] <joran> rofl

[19:20] <Stephen\_F> Ja

[19:20] <mike\_s> okay. done. the next one is more complex:

[19:20] <mike\_s> Look for steps in your use cases where what exactly the user clicks is not spelled out yet. Add statecharts and collaboration, interaction, or activity diagrams as needed to describe dynamic behavior in your system, per the criterion discussed in class.

[19:20] <joran> wait

[19:20] <joran> i thought we were just doin the stuff we told him about last class

[19:20] <joran> or the class b4

[19:20] <joran> whenever

[19:21] <mike\_s> i did too... but the hw5 seems to say otherwise...

[19:22] <mike\_s> so, are we going to try to make some charts?

[19:22] <Sverk> are there any use cases that need it?

[19:23] <NMazu> Do we have a central spot where our "official" use cases and stuff are kept? We all did lots of diagrams, so which ones are we going off of?

[19:23] <sasha\_\_> exactly. Which use cases/class diagrams are we going to use?

[19:23] <sasha\_\_> are we going to merge them?

[19:23] <joran> we have create member/ create group / add member to group / and delete both are the ones we signed up for in class

[19:24] <joran> which are fairly trivial i think... and dont really require statecharts

[19:24] <joran> except maybe add member to group

[19:24] <mike\_s> do those count as use cases? we weren't focusing on users actually doing this things yet

[19:24] == mike\_s [4b57fc0f@gateway/web/freenode/ip.75.87.252.15] has quit [Quit: Page closed]

[19:24] == arkenflame\_ [~arkenflam@75.87.252.15] has joined #uidaho\_cs383

[19:24] <joran> well do users do those things?

[19:25] <joran> so i think they are use cases

[19:25] <joran> but maybe not

[19:25] == arkenflame\_ has changed nick to mike\_s

[19:25] <RevanTal> i would say they are

[19:25] <mike\_s> they are, but not in the form we have them. users aren't going to call functions

[19:26] <RevanTal> true

[19:26] <sasha\_\_> yeah, we can pull use cases out of them, but they are more classes or functions right now.

[19:26] <joran> shuld we goto googdocs to do use cases?  
[19:26] <mike\_s> i got the impression use cases were "click 'add user'"  
[19:26] <joran> yeah thats true  
[19:26] <joran> but thats still a create member use case  
[19:26] <joran> what we prototyped is just functions  
[19:26] <mike\_s> well, i've made about 100 already. we could just copy some of those  
[19:27] <joran> lol that works  
[19:27] <joran> but i think maybe gus has warped way out side your guy's original vision for it....  
[19:27] <mike\_s> "Look for steps in your use cases where what exactly the user clicks is not spelled out yet" -- so how does that jive with what we talked about in class?  
[19:28] <joran> lol ok we got no use cases  
[19:28] <NMazu> Last Thursday class?  
[19:28] <joran> sweet ! :)  
[19:30] <joran> im pretty sure we are supposed to do what we said in class regardless of what the homework says  
[19:30] <mike\_s> makes sense to me  
[19:30] <Sverk> which makes life easier for us :)  
[19:30] <NMazu> Frankly, I'm kind of lost, with little idea what we're supposed to be doing or how to do it.  
[19:30] <joran> but then again i was wrong last time i thought that step 1 would come before step 2  
[19:30] <joran> we merged in class  
[19:31] <joran> and told him we would identify the required members and fields of groups and users  
[19:31] <mike\_s> well, the homework doesn't make any sense, at least that step. unless someone has a practical approach, we should move on  
[19:31] <mike\_s> we could do step 4  
[19:31] <joran> and prototype add/delete/etc  
[19:32] <joran> whats step 4?  
[19:32] <mike\_s> we can specify what in our two prototypes is "public"  
[19:32] <mike\_s> Add a public interface (a set of public methods) for each of the classes in your team class diagram. Methods come from use case steps, events in statecharts, collaboration diagrams, or activity diagrams, or your knowledge of the application domain.  
[19:33] <joran> I think we should maybe clarify use cases for add member , delete member , etc (or at least include the old ones)  
[19:33] <mike\_s> okay. should everyone check and see if they have one of those?  
[19:33] <joran> I think he means the whole system statecharts we turned in last homework  
[19:33] <joran> but not sure  
[19:34] <mike\_s> the class diagrams you mean?  
[19:34] <joran> ummm maybe....  
[19:34] <joran> lol  
[19:34] <NMazu> Yeah, it sounds like it  
[19:34] <joran> so we can do that i guess  
[19:35] <joran> lets just assign blocks  
[19:35] <joran> and we can go over em all next thurs day  
[19:35] <joran> err next scrum  
[19:35] <joran> (friday)  
[19:35] <mike\_s> blocks?  
[19:35] <joran> like we got the 2 diagrams  
[19:35] <joran> we should identify enough to turn it into 1 diagram or just use one or the other i dont care  
[19:36] <RevanTal> we have to have the public interface things done before thursday  
[19:36] <joran> and then just pass out the classes  
[19:36] <joran> tomorrow thursday?  
[19:36] <sasha\_\_> yes, tomorrow thursday  
[19:36] <RevanTal> now that is strange...

[19:39] <mike\_s> well, we told him last week that we'd prototype out basic group/member classes. we kind of did that... twice

[19:39] <joran> yeah

[19:39] <joran> so meh

[19:39] <joran> i think were fine

[19:39] <sasha\_\_> but we have real live code!

[19:40] <joran> it dont matter to me

[19:40] <joran> ok i really gotta run ill post minutes in google groups tonight if you want

[19:40] <joran> and ill open a vote thread for php or python? or have we decided?

[19:40] <mike\_s> so, we put a note that says we couldn't figure out homework in the minutes, add both (or all 3) prototypes, and put it in git?

[19:41] <RevanTal> we could even say that you two were creating a place to start, while the rest of us were getting comfortable with the two different langauges, so we could chose either

[19:47] <joran> next scrum we'll do the identify methods of class diagram thing

[19:47] <joran> and we'll have picked an implementation one way or another

[19:47] <joran> Nathan make sure to chime in php v python convo... in fact everyone should vote on the thread

[19:47] <sasha\_\_> python vote

[19:48] <mike\_s> we should shoot for consensus, not vote

[19:48] <RevanTal> yeah

[19:48] <joran> i agree

[19:48] <sasha\_\_> python consensus

[19:48] <NMazu> I have no opinion, having worked zero with either

[19:48] <joran> lol

[19:48] <NMazu> I'll go with whatever the group decides

[19:48] <joran> did you look at the code samples?

[19:48] <NMazu> And try to pick it up quick

[19:48] <NMazu> yes

[19:48] <NMazu> I've looked at those

[19:48] <joran> they both look like no problem?

[19:48] <NMazu> Manageable, yes

[19:48] <joran> 2 ppl know python 2 know php

[19:48] <Stephen\_F> I'll go with whatever as well, I'm easy like Sunday morning.

[19:49] <joran> but i think me and steven are down to learn python too

[19:49] <joran> so we can go for it

[19:49] <joran> alright python it is ?

[19:49] <mike\_s> i'm down

[19:49] <Sverk> i've fiddled with php before a little and never used python but it didn't look that bad so i'm fina with either

[19:49] <sasha\_\_> YAY!

[19:49] <joran> lol

[19:49] <NMazu> Any naysayers to Python?

[19:49] <Sverk> fine\*

[19:50] <joran> on the condition you do some db shit in a simpple example

[19:50] <sasha\_\_> maybeh.

[19:50] <mike\_s> me? i can try. i'm not so good at the sql

[19:50] <joran> and be carefull when we pick a framework to avoid cms's as thats prolly confusing

[19:51] <mike\_s> ...wouldn't it be nice to have one? never mind, go tutor!

[19:51] <joran> "Insert Into some\_table (user\_id,name) VALUES (NULL,'bob')"

[19:51] <mike\_s> i could just look at your code for the sql

[19:51] <Sverk> lol as its been a few mins again since you last said had to go

[19:51] <joran> maybe if you can find one that allows heavy customization and easily implemented new modules

Here is a basic example in python with output at the bottom:

[http://pastebin.com/embed\\_iframe.php?i=TSe0uFPB](http://pastebin.com/embed_iframe.php?i=TSe0uFPB)

(I am adding a prototype that more closely mirrors mikes python example - joran)

<http://codepad.org/b0aXgLuf> <- this is practically the same as the python  
(also codepad allows you to modify the code and re-execute (output is the code actually  
executing on codepad))

//I used links to codepad so there would be syntax highlights -Joran

<http://codepad.org/pyTpRVB7> <- unit Testing my DB Class (disregard errors, just see  
code)

<http://codepad.org/614HFyKD> <- Gus Specific Data Management Class

<http://codepad.org/7G1Tzi5k> <- Database helper class (not specific to gus)