Requirement Specs

User Narratives:

Student—go to website, presented with login credentials

How would they get logins?

* Admin—sets up faculty accounts,
* Faculty—sets up students accounts—enter emails, then send pin to people to register

How should the login be? How would you like the users to be created?

* open registration –do you want someone to have a new account every semester?
  + problem for historical records
    - how are they doing from course to course?
    - no connections between a person from class to class, separation between courses, 120-210 through away records? Is that best?
    - Remove on cascade—want to keep track of how many people over time get questions wrong/right?—What does it mean when you delete a user? Cascade would be bad because it would delete history—Lim want that data to be able to see how people do on a particular question over time

Confirmation code sent out with along with user name and password

* allow them to gain access to another course if they already have an account
* Lock them out of old courses—only what they are currently enrolled in should be available

How would the unique code be administered?

* email, instructor/ admin who hits button to notify people to register for course—could be faculty

Pre add/drop date—basic entry, collect username—plug and chug to get them into course—somehow they have to be entered

More than one course on side, chose which one to start working on—pending assignment—progress/completion status, open up for problems, start coding

CAN THEY SAVE?

* Can we do it? Not the easiest thing. Solving one problem at a time, so that we don’t have to save multiple solutions. Lim wants it at some point, when we get further in the process,
* server breaks—how do we handle it in terms of recovery

Nonfunctional/functional requirements. In a nutshell you can measure functional requirements, but not nonfunctional requirements

* ex)is user interface user friendly, how do you measure that? Is it robust—deal with compilation error, infinite loops—measurable, also is super necessary

Created user page, create user by anyone—gen info, list courses that you are registered in, then you can go into each class and see info for that class

Multiple john Roberts—how does the system know which is which?

* Email user address for user, if you feel there are benefits to Siena email accounts then I (Lim) have no problem. Solves unique problem and administrator can set up classes very easily. Emails would solve both problems—wouldn’t mind

I (Lim) think they should be able to look back at previous problems within that course—not just the pending problems. ,Looking at previous homework solutions to figure out how to do the current homework problem

General information table for grade book

Grade book

* not as formal as grade book in Blackboard—something with the grades shown. Faculty would want this
* They would decide on how the questions would be weighted, JREP shows 8/10

Limit on submissions?

* Disable submission after a certain point-only one.
* Not so much for homework-don’t want them to lose homework
* in general it is a toggle, where the teacher can decide on how many submissions to allow per assignment

Professors:

* Same website-login credentials, provided by admins
* Presented with courses that are teaching—choose a section, create/edit questions individual problems and put them into question set
* should be able to see past question sets

Course coordinator:

* 114 and 010—120/210
* creates a common exam, cohesive across all classes in classes that have a large num of teachers/students
* Make sure that the prof that are teaching the same class are on the right track
* Develop problems and problem sets and administering it to the sections of a course in place of the profs doing it individually

Administrator:

• Set up faculty accounts

• Create courses and assign it to a prof

• Perhaps allowing for a team taught course

• Lock out students

• Faculty would just be able to create assignments edit and grade, everything else would be the administrator

More than one function—helper functions?

* Incorrect to write a main method, cannot change input signature,
* Multiple functions-put through something-added with threads to keep track of shit. (at this point Denis wrote “threads” on Luke’s paper
* How to handle a crash—null pointer: thrown through a standard error, crash program-if the script stops then we should return exit code.

Would a null be the same as index out of bounds—throws grab-able errors. “System. Exist” in Codingbat—try that out. Should keep going; not crash

Exception handling

* Most important is figuring out how to compile the code, then handling exceptions

Absolutes

* grade book—could be live so that when a teacher sets up a problems and a student works on it then something could increment saying they have started.
* Sends students unique code to register; makes the account then the activation code will work.

For Monday—

* User case narratives, wants one for course coordinate—less expectations for it because is just brought it up

Discussion of how awesome we are:

Client Meetings:

Most amount of meetings with the client-an advantage

Increases during the design phase

Webmaster—quick proto-types to go over in future meetings

Design phase—mock ups: draw them story board style, word, paint, layouts-VB draw and drop to get the visual for the final product.

Requirement-Diagrams, uml diagrams, more in-depth