Client Meeting 5 - For retrospective 15:00 - 16:59

Agenda:

Showed Ros the website.

Retrospective

Attendees:

Henry

Augustin

Notes:

R: Numbers and no names because data needs improvement?

R: **date format confusing, order universities by name.**

R: if it was 2013 then it had been approved, does it just disappear?

H: atm the app doesn’t work cause it needs to support many to many. Staff search requiring login, which shows same stuff except open up comments, show who made what edits, a screen with extra information.

R: does the cart work?  
 H: yep, but if you went student application, it’d automatically take what you selected.

R: Checkout button?

H: 2 views for search: student and staff separately, work involved to doing that. No need for changes to database structure. With current constraints better to aim for data entry, then later the application. Student enters it, staff opens the application and can edit it.

H: anything to add?

R: i only care about the outcome! Can’t really give meaningful feedback.

H: we’ll go through the points we raised.

Working website.

Sample data is there.

1-1 mapping, application page was done, now it’s a M-N mapping, the form might get longer and longer.

Meetings have been documented.

We could improve:

better requirements gathering.

better scoping (week 7 tash told us new scope).

H: backend logic on unit equivalence, if it was denied but equivalent, show as equivalent (approved). Student context not so relevant here.

Collaborative idea, as normally client would have more technical knowledge, so in a way those things couldn’t really be avoided, the cost of figuring out things as we go along.

Scope creep - big issue. Value of student application depends on staff page. We’d streamline development. That didn’t work out so well. First pass of search page, staff page, then revamp the search page, staff page etc.

R: what’s the , in terms of going forward after the database, if we want to fix it up or upload a new spreadsheet/tidy it up, would we have the capacity to do this.

H: one way it’d be a good way to work, a dropbox/google drive where you can upload your spreadsheet to. Whenever it boots up it’ll load it from the database (spreadsheet is database). As long as it’s in the right format, upload it, although this means you’d have to restart the app.

R: it’s quite important isn’t it, it’s a living document. If all is working, the student enters something and that goes into the spreadsheet/database.

H: spreadsheet read from disk into memory, all query is read from memory, what it can do is periodically … upload to spreadsheet everyday.

R: what happens if we want to get the other faculties on board with it?

H: upload spreadsheet screen. Upload updated spreadsheet to the website, if it’s rejected, then use old data.

Last sprint coming up, search page with staff view and student view done, and that’s the finished product. Uwa makes their systems produce a spreadsheet for this system.

H: regular progress reports. Distribution of work - don’t know how to fix, cause people don’t have the technical knowledge. Project management - avoiding last minute meetings like this. Know what meetings you have to organise. Deadlines - get something done by some date, complications come up that delay deadlines. To be followed up - people can’t catch-up when we allocated 3 pages to 3 sprint, skills audit more pessimistically done, “just search units” so we had time to learn.

Staff page layout uncertain? Decision states and filters, not sure.

R: if we had something approved, the database would only have if a particular uni, unit , has been approved, we won’t have an opportunity to enter it again. Potentially the comment would only be relevant the first time? Don’t want students seeing comments, but would staff see the comments?

H: atm when someone makes a request, 1 set of units in ex for 1 set of units (can be just 1 unit), there’s a field per set that asks if there’s an equivalence precedence. Should only show unique instances of equivalence.

Maybe we can do a button to download the spreadsheet, that way if this dies, you have a way to see the latest information. Could be done daily as backup. Simple so we don’t need to build a page for that.

H: maximise value without integration. Integration not as important for the value. Data is self contained, maybe you’d be able to auto suggest unit-coordinator? In this context where we won’t fully implement, making sure it’s easy to upload, download, get a backup are valuable things.

H: M-N design, tough problem to solve. Search result by unit names.

Responsiveness?

Problem encountered with Heroku website: search by exchange universities encountered an error, website search is working but table isn’t.

R: she thought we’d be able to think of stuff she didn’t think of. $100 dollar test, realistic outcomes to be expected, good management of expectations, looking at the product so far, i like the look of it, simple, and once we pointed out the functionality, it was obvious to use, unlike uwa counter intuitive websites. Basically, i think you’ve understood what’s required and so far come close to producing something. I’m worried about our data and the accuracy of it, but that’s my problem not yours. Spreadsheets are boring, but a website feels nicer to use, more appealing product. Most importantly to me, the fact is if it works properly and a student can enter things, and we eliminate the double handling and back and forth… makes more sense, if that works and we have the ability to tidy it up (when student enters something that doesn’t make sense). Sometimes information is not entered into the spreadsheet because we get lazy.

H: path1: download spreadsheet in current format, change UBS, upload it. The complication is when we have to merge spreadsheets.

R: we’d only put it on live when all faculties have merged their spreadsheets. Email as backup daily as the safety net, “rush hour application” …

H: special search screen for staff VS application approval, which is more important?

R: application approval.

H: let’s walkthrough the cart, someone adds units to the cart, the first thing is, all electives have to be reapproved, all is per student a decision is made, the things in the database are either core or complementary?

R; if a unit is approved, then the details around core/complementary, there’s no way the database knows that so it’s up to us to check that, it’s useful to the student to know basket weaving isn’t equivalent to x at uwa.

A: a matter of how complex the unit is to be worth 6points not just it’s got an equivalent unit here at uwa.

H: uwa unit displays unit level. Someone does an elective, but they’ve done an equivalent here, so you’d say you can’t, contextual approval, academic level, equivalence. Core / complementary unit must satisfy these. Frontend ignores contextual decisions, only shows if something is approved at an academic level.

R: student sees academic level equivalence and adds to basket. student rejected because they already did a similar unit here at uwa.

R: we speak to students beforehand and work out what they have left to do, so explicitly defining core/complementary/elective is not needed.

A: add a note at the top of the page informing the student that the results are based on equivalence and staff has final say on context based decisions.

H: student’s pov of unit set: name of ex unit, ex code, ex url, uwa name, uwa unit, uwa url, 3 ticks, for equivalence, context, academic level (greater than level 0).

R: integrate with Pheme for user accounts.

R: as long as we can see what we’ve approved/rejected when the students are back from exchange that’s way better than currently fishing through their email.

H: unit coordinator stuff is out of bounds.

R: totally fine, that’s complicated anyways, we will have that list of units from students, that won’t be downloadable wil it?

H: all is backed up on spreadsheet with format similar to what you’ve got.

R: ideally, easy way to plot to email (working links).

H: clipboard icon to copy to email.

A: confirm this is only for staff view?

R: unit coordinator replies by email, then we update the database, any functionality for student to look

H: no that comes with the problem of students knowing student numbers.

R: alright, people might say that’s sensitive data, so probably not, forget about that. But we would have that view, we’d update the database.

H: much easier if at a point in time you say, yep this app is done, send email, but then you’d have to send an email per application. Oooo….

R: tickbox for each unit set, then copy to clipboard and send to students (doesn’t matter if they are told again).

A: could we have a copy/template of informing students of their unit set approval.

A: level 0, copy all, you’ll manually edit, level 1: copy the ticked unit sets, level 2: automated emails, is that correct?

R: yes, but we wouldn’t want that because we’d want it integrated with askuwa at that point.  COULD we have something where we assign a student to a member of staff? Certain staff get x number of students to look after.

H: api methods that write the code for the front end.

R: if i had to choose, easier email format with cut and paste is more important than keeping track of whose staff is responsible for student.