**Interview #1**

Interviewee: Carlos Mckinney

Position: Senior Lecturer for Information System related course work.

Time & Location: March 13th 6:45pm

Interviewer: Jas Raj Singh

Functional Questions & Responses (Non-technical)

* What information should we gather from the user before making them recommendations?
  + Interest (interested in), What existing classes have you taken, what did you enjoy in classes, childhood things you wanted to do when you “grow up”
* Should the user have to verify they are a college/UMBC student?
  + No, leaves it open to anyone who is interested in seeing what they want to do (high school students)
* Should the application tell user if classes overlap or provide multiple templates of possible schedules?
  + Order in which you would want to take classes, prerequisites, classes that are only offered in spring/fall, classes that fill up fast/ take it sooner/more urgent for your major.
* Should the application be available in multiple languages (Spanish, Chinese)?
  + Yes would help, but adds more work.
* Should users be allowed to create pathways with multiple majors/minors?
  + Different alternatives are good, don’t want to pitch and hold someone into one “result”,
* Should the system give feedback on the likelihood of success when a user enters certain parameters? (Success = Chance of graduating)
  + “Harder” is still relative, create “math intensive majors”/ “writing intensive majors, language component” (certain classes add this (aka if you have to take calc 3 then it checks the box that says it is a “math intensive” major”))
* Should the system offer full recommended pathways (2/4 year paths)?
  + “Have part where you tell us already how many credits you have” / “having a premade sheet for classes for certain majors” - what can you do with these major/minors

Suggest a forum board to help people communicate with one another

Non-Functional Questions & Responses (Technical)

* Should the system collect cache data for users to be able to open up software again and not have to re-enter information?
  + Sure. maybe include a basic login
* How should we host? AWS/Ocean/Physical server
  + AWS is the best
* Should we collect emails and send out a verification?
  + IF you have a login then emails yes
* Should there be separate mobile development (swift, android)?
  + No
* Mobile web friendly?
  + Just make the site mobile friendly
* Offer a downloadable PDF of course/college pathway outline to device?
  + Sure especially if they aren’t saving it
* How long should the system be up?
  + 24/7

**Interview #2**

Interviewee: Oscar Urquiza

Position: Residential Assistant for Hillside Apartments

Time & Location: Hillside Apartments 5:00 pm March 27th

Interviewer: Jas Raj Singh

Functional Questions & Responses (Non-technical)

* What information should we gather from the user before making them recommendations?
  + Parameters, age, interest, hobbies, parents careers/interest.
* Should the user have to verify they are a college/UMBC student?
  + Only if they want to receiver further advice/guidance from someone on campus.
* Should the application tell user if classes overlap or provide multiple templates of possible schedules?
  + It can, if those settings are checked, ideally the easiest thing to do is just create a bunch of mock templates and course outlines.
* Should the application be available in multiple languages (Spanish, Chinese)?
  + If capable and easy to do, then yes. Spanish ideally.
* Should users be allowed to create pathways with multiple majors/minors?
  + Users should be able to create a pathway with up to 1 major and 1 minor. Allowing them to create more unique pathways compared to just a normal major pathway.
* Should the system give feedback on the likelihood of success when a user enters certain parameters? (Success = Chance of graduating)
  + The system should create a scale and give an output of how many rigorous classes that user may need to take to complete that pathway. A rigorous course can be identified as one that has a high rate of drop/w/failure to complete.
* Should the system offer recommended pathways (2/4 year paths)?
  + It can create default pathways that are common majors and minors grouped. Like comp sci and a math minor, or a biology and psych minor, or Engineering and entrepreneurship minor.

Non-Functional Questions & Responses (Technical)

* Should the system collect cache data for users to be able to open up software again and not have to re-enter information?
  + Doesn’t matter, sure.
* How should we host? AWS/Ocean/Physical server
  + AWS
* Should we collect emails and send out a verification?
  + Only if they want to email their schedule or share it.
* Should there be separate mobile development (swift, android)?
  + No
* Mobile web friendly?
  + Yes
* Offer a downloadable PDF of course/college pathway outline to device?
  + Yes especially if the software doesn’t have a login or any database to save users info.
* How long should the system be up?
  + 24/7

**Summary**

* Keep the software simple, easy to use, and mobile friendly. A non-complex system allows high success rate in which the application can be used. Offer a way to quick save or download the pathway you have created so the user can share or store the template they created. Application should have the least amount of barriers to use; no email verification, no student requirement. This allows people who are younger to anticipate and get an idea of what they maybe want to study in college. Software can provide some bias on coursework suggesting that a class is more “difficult” and requires more “work” to get an A or passing grade in the class.