**Ricardo & Liuai: January 28 10h15 - 11h50**

(see more dates below)

3.1: Each requirement / Use case

3.2: Domain model

3.3: Constraints

Questions that came up:

Does system allow us to enroll?

Exceptions to allow student to enroll? (ie: advisor gives permission?)

Notes:

Use case can be used by more than one actor

CRUD: Create, Read (ex: search), Update, Delete

**Next week:**

All use case drafts

Friday 5th show team what we made and get feedback

3.1

Actors - Users:

1. Students
2. Registration Advisors
3. Teachers
4. Admin (tech)

Misuse case (has different formatting):

1. Hacker

Steps:

1. List of actors
2. Fill use case info
3. Diagram for actors

Student:

Login

logout

Update personal info

review my course history

view/search courses

view audit (shows taken courses and missing courses for degree completion)

view current schedule

choose preferences

view course description

enroll(?) - no

save schedule - yes

view saved schedule

generate schedule(s)

finalize schedule (?)

print schedule

Faculty Admin:

Login

Logout

Update personal info

create course (enter requisites etc)

View course

Update course

Delete course

(for create, view, update, delete - > course + requisites)

Professor:

Login

Logout

Update personal info

Update course grades

~~Modify course description (?)~~

Admin:

Login

Logout

Update personal info

Create users

View user

Update

Delete users

(for create, view, update, delete -> account + permissions)

~~Grant permits~~

~~Update user permits~~

Use case:

UC1 - Login

Risk assessment: 5/5

Importance: 5/5

Actor(s): Student, teachers, admin, advisors

Description: Validate user credentials and allow user specific privileges to the system

Goals: Access to the system

Pre-condition(\*this is initial state): User is not logged in

Post-condition: Success: User is logged in. Fail: No access granted. Incorrect credentials

UC2 - Logout

Risk assessment: 3/5 (?)

Importance: 2/5

Actor(s): Student, teachers, admin, advisors

Description: Terminate the session of the user interacting with the system, no longer provide the user his or her personal data

Goals: Disconnection from the system

Pre-condition(\*this is initial state): User is logged in

Post-condition: Success: User is logged out. Fail: User is still logged in.

3.2

Domain model:

More users in domain model

3.3

Possibilities for constraints & nonfunctional requirements:

Performance / response time

Portability

Security, privacy

Reliability / availability: # of 9’s in percentage decimal 99.999% online. how critical system will be online

Scalability: # of student accounts. now 100 later 10000

Language

Useability

Accessibility (impaired users)

Compliance (check if there’s any requirements from ministre du education de QC etc)

Interoperability: if you want system to communicate with another system.

**Anna & Ricardo & Liuai: February 3, 18h - 20h30**

Did the diagrams (upload photos)

Links:

Student 1 has \* schedules

Student \* interacts with 1 scheduler

Student 1 has 1 preferences

Student \* in \* course

Student 1 has 1 academic record

Preferences 1 interacts with \* Schedules

Schedule 1 has \* Courses

Scheduler 1 has \* Schedules

System Admin / Faculty interacts with \* courses

Prof \* interact \* Section (lecture)

Scoping:

Just building student use cases

Ask team if it’s “wishlist” or “enroll”? (eg: do we keep finalize schedule).

\*\* We are using star uml app to make the diagrams

**TODO:**

Anna : Domain Model, Use case, Use case diagram (Professor)

Liuai: Use case, Use case diagram (Student)

Ricardo: Use case diagram (System Admin, Faculty Admin)







