

Database Implementation/Joint Usage Proposal

USER TABLE

Registration:

1. Look for duplicate entries before inserting a new registration.
2. Generate a salt, verify salt against database to make sure unique
3. apply salt to password, then hash. Store unique salt key in user row.

Login:

1. Accept user input, look for email match (unique key), get salt, apply salt to inputted password, hash inputted password and check for match against password in user row.

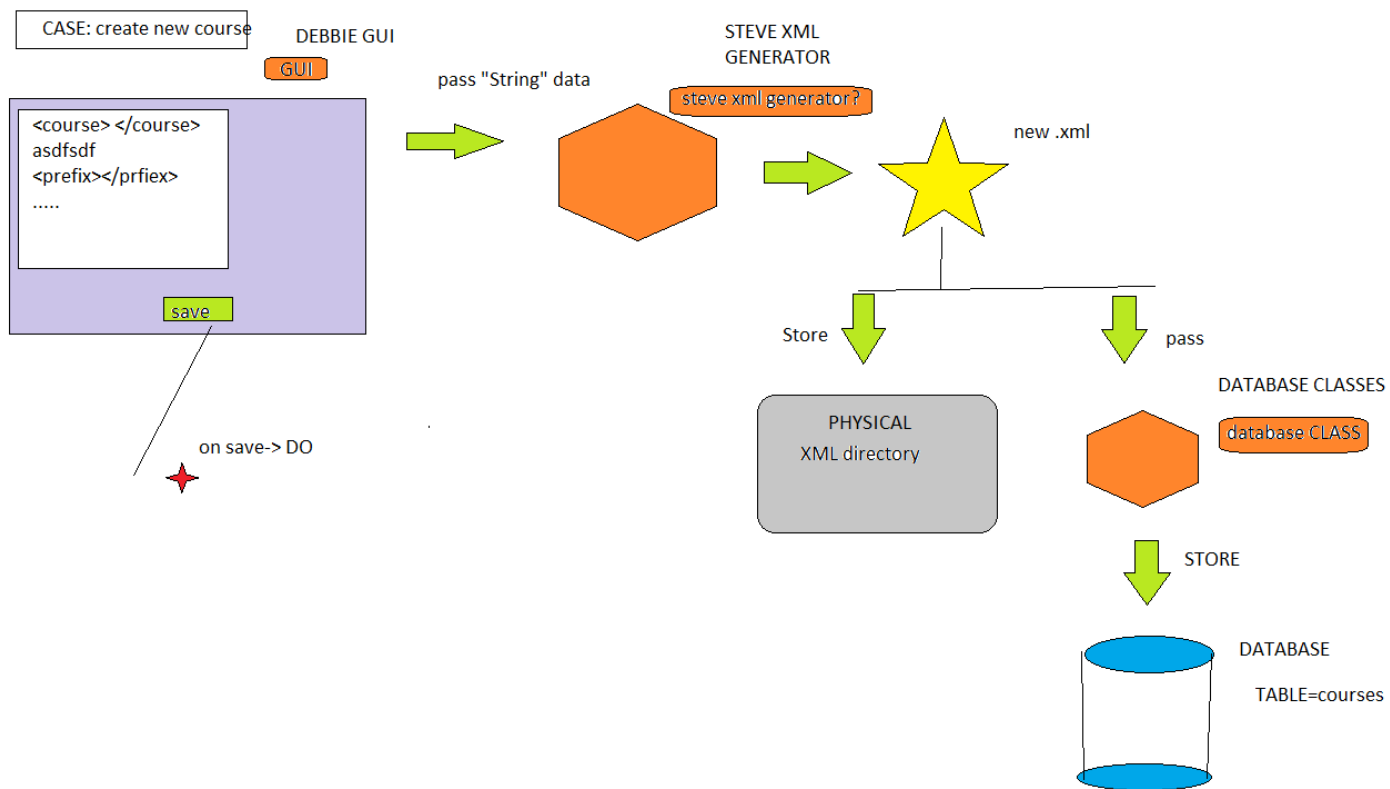
Columns:

- **Courses column**
- how does it store all user classes? Linkage, foreign keys? Array List
- How will it return all the courses a user has if called to do so?
- How does it add/store additional courses? Remove?

COURSE TABLE

- Column names?
- Linkage to User Table (courses columns)
- How do we populate courses table? Do we need to? Automate? Or manual?

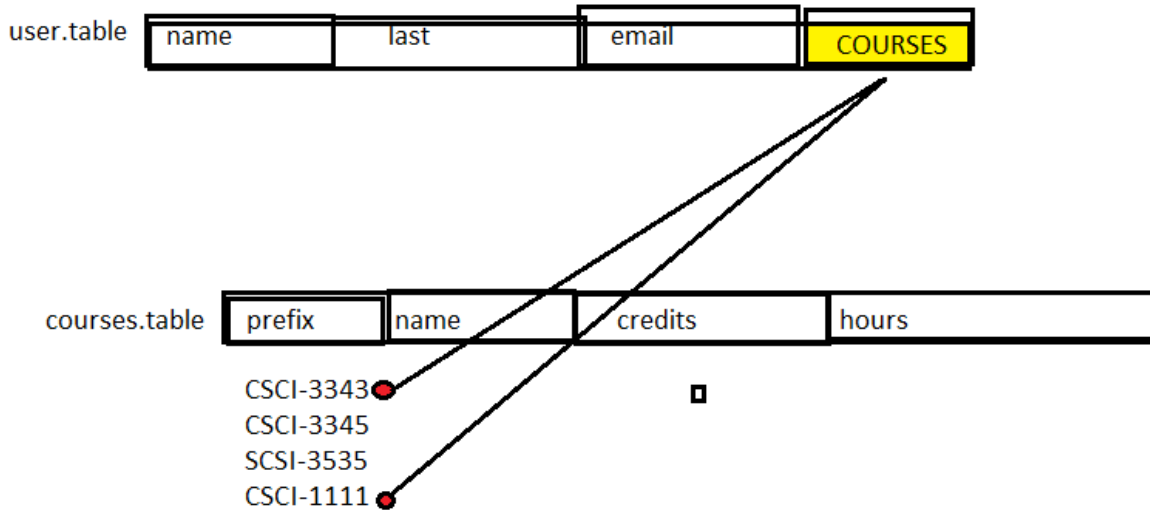
USER CASE: CREATE NEW COURSE



FEASABLE???????

USER COURSE TAB COLUMN

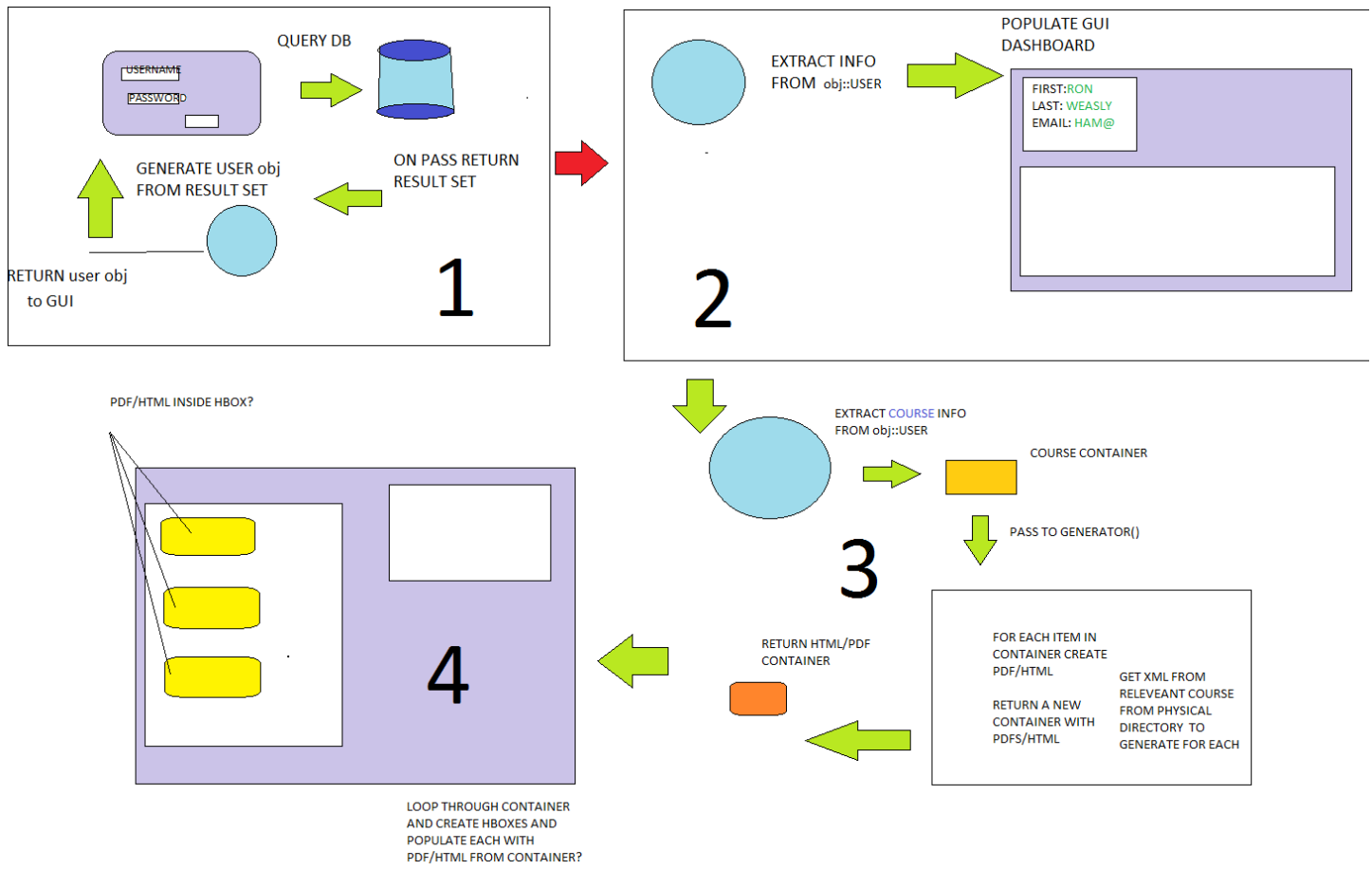
1. Will store courses student has
2. how will will store inside column?
 - ArrayList e.g <coursexxx, coursexxx>
 - table linkage. Link user table with courses table. One to many relationship



USER LOGIN

1. Credentials entered
2. Result set returned
3. match entered email with email in table
4. from that row get unique salt key
5. apply salt to inputted password the hash. Compare against password in that row
6. on match create a obj of class user. Pass user obj to GUI.
7. Extract data from obj. Where data is username..email...COURSES.
8. GUI::how do I populate user schedule? Pass COURSES to PDF/HTML gen
9. GENERATORS::Loop through COURSES, look up XML for each, get XML info, generate PDF/HTML, store in container, return to GUI. Feasible??
10. GUI:: Loop through returned CONTAINER, populate a HBOX with a PDF/HTML for every item in CONTAINER. Place all those HBOXES in master SCORLLPANE to display current classes. Maybe use this functionality to get GPU widget to work? Like in the homework where the meter moved according to what classes you added? Extra field needed for GPA in xml?? DO we have to SAVE GPA relative to each class user has? If so store in DATABASE in extra COLUMN in COURSES TABLE?

Use case



obj::USER

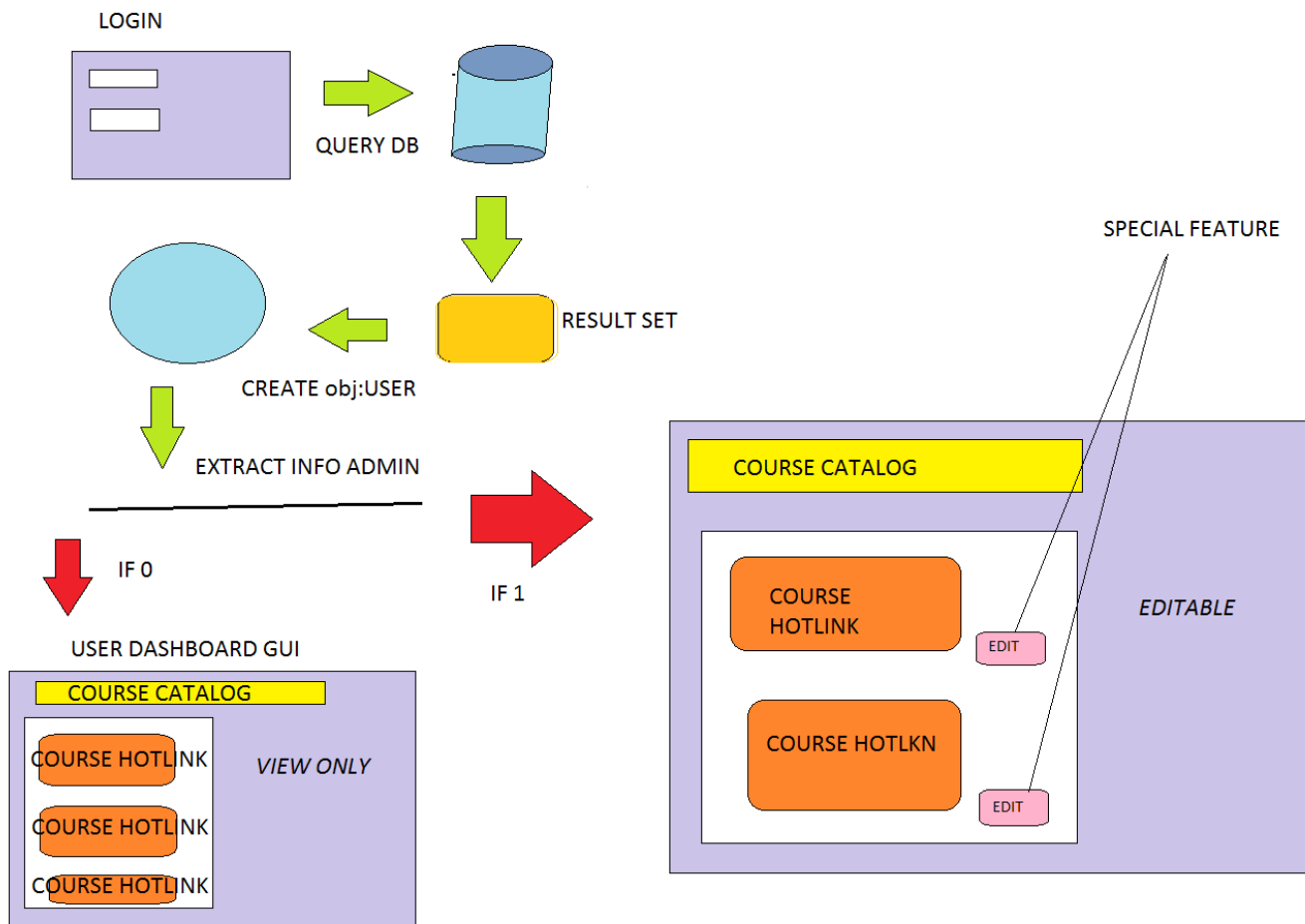
FIRST
dio
LAST NAME
brando
EMAIL
@email
ADMIN:
0 OR 1
COURSES:
<csci-2222,csci-3434,csci-3432>
SALT KEY:
SDF353DFGS
PASSWORD:
h332371010ff1

ADMIN RIGHTS AND COURSE EDITING

USE CASE:

1. User wants to edit a course.
2. How select that particular course? How decide who can edit cours?
<admin rights>

Proposal 1



PROPOSAL 2

- 1 UNIVERSAL GUI
- On edit button click QUERY DB for USER admin rights.
- If 0, message "do not have rights", if 1 allow editing.

ISSUES :

- Who qualifies for admin. Faculty?
- How do they qualify. How distinguish admin from reg user
- email for verification?
- 24 hour waiting period (local machine, unfeasible)
- is a FACULTY email different from student? <-this distinguish factor?