University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Part 1

Date: September 26, 2018

Group Members:

Name	Student Number	CS Userid	Tutorial Section	Email Address
Damasia Maria Correch	28649151	o2e1b	T1C	damacorrech@gmail.com
Naitong Chen	21539151	q7l0b	T1G	chennaitongubc@hotmail.com
Chloe You	11654150	i0e1b	T1D	chloeyxy@outlook.com
Christina Cheung	37405157	d8f1b	T1D	ccheung256@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

PROJECT DELIVERABLE DOCUMENTATION

Features that stand out:

- We spent a lot of effort in the UI with lots of pre-calculated options to keep the users from entering any long and specific values (project name)
- We have different roles implemented with different access to the DB.

To launch application:

Follow readme.txt in the scripts folder and populate the database in sqlplus.

Overview:

In our deliverable, we used Java to implement the user interface, Oracle for the database and JDBC to connect the relational database to our Java application.
Overall changes we have made in our queries (some are repeats from Project Formal Specification):

In terms of SQL, we split the queries into five groups according to the role in the lab: researcher, PI, lab manager, collaborator and database administrator. Here we will walk you through each role and the corresponding deliverables in SQL one at a time.

Application Design:

We used Java Swing to build the application with 7 main panels (one for each type of role, one for project profile and one for lab profile). The panels are toggled using a card layout. Any context variable (lab id, project name, or user id) is stored in the QueryBuilder class, as well as most of our queries. Other queries are stored in DBAPanelConstants and some insert/update queries in the corresponding panel class. We used rs2xml.jar as a dependency to show result sets. Jar is included in our lib folder in the project.

Outline of deliverables in SQL:

Researcher:

- We will take researcher's id when they login. This id is also stored. We display the researcher's name and the total weekly hours allocated. Researchers can view all information about projects, lab members (with the ability to filter by type) and all labs but does not have authority to update tables or to view information about grants/funding.

PI:

- We will take PI's id when they login. This id is also stored. All queries from researchers are extended to PI. On top of what researchers can do, the extra queries PI have involves updating project category as well as creating new records of collaborators and projects.

Lab manager:

- We will take lab manager's id when they login. This id is also stored.
- All queries from researchers are extended to lab manager as well. Lab manager is only able to see grants without updating it.
- Lab manager can view all open grants (and filter to view different statuses)
- Lab manager will be handling all instances involving study participant bookings. This includes looking at all subjects in the participates table, searching for available participants to book given desired booking details, and inserting into Takes_Booking,

Makes_Booking and Participates. Realistically, mainly psychology labs take study participants. While populating our tables, we only had booking records for psychology labs, but lab managers have the ability to add participants to any study.

⟶ The above three entities are considered as lab members. When they login and type in their Lab ID, we will show the lab they are in, the projects in their lab, their lab members, etc.

⟶ For every lab member in the research institute, they will be able to see a table of all labs (including their own lab), all lab members from other labs. When they click on a lab, they will enter lab profile panel. This is where we display all lab specific info (pi, researchers, projects, etc.) - more details in the UI section

⟶ Clicking on a project will take user to the project info panel, where they can view all project specific info (project name; category; material type and price; PI supervising project; researchers working on project; collaborators in project) - more details in UI section

Collaborators:

- We will take collaborator’s id when they login. This id is also stored. They will only be able to see project information that the collaborator is assigned to. When clicking on a project in the project table, they get taken to the project info panel described above. They also cannot view information about grants.

Database admin:

- DB admin doesn’t need a login id. They can view all grants, add new record of grants open or approved. When a fund gets approved, DB admin can delete the record of the approved grant from the Applies_OpenGrants and add the record to Fund_ApprovedGrant (which are done in a collection of actions described in the changes section). Lastly, DB admin can delete collaborators, update lab member’s name and education. For example, if someone changes their name or graduates.
- DB Admin also has the same ability to view project and lab info panels.

UI:

Lab info panel:

- See the lab’s and the PI’s name AND all the projects in that lab with a count of the number of projects in the lab. Users are not allowed to enter the lab profile panel from here. They’d need to go back and enter from the list of all projects.

Project info panel:

- All users can see grant info except for researchers and collaborators
- See all researchers working on the project AND a count of how many researchers there are
- See all collaborators assigned to project
- See category, material type and material price of the project
- See all grants allocated to a project, a total of all grants amount AND how much funding is left after they have bought the materials for the project
- See all participants booked for a project AND the count of how many experimental and control group participants there are
- See a list indicating how many of each type of researcher are present in the project

Changes from formal specification deliverables:

DB Admin:

- Instead of automatically removing the approved grants and inserting them in the table in terms of UI functions, we decoupled the operation in the UI into 3 parts: the user can first update the status of the specific grants in the Applies_OpenGrant_Date table to *approved* (formerly a LabManager ability, now only DB Admin can do this), then select the option in the drop down menu to remove all approved grants from Open grants, and then add the grants to the Fund_ApprovedGrant table.
- In our Project Formal Specifications, we said that the DB admin can update lab member's name, ID and education. We realized that it would be unwise to tamper with the ID of a tuple, so we decided that DB admin shouldn't be able to update ID. So now DB admin can only update name and education.
- The operation of *LabMember Set ID* is no longer valid; since we believe logically it does not make much sense to change the ID of an existing lab member.
- Delete from approved grant is removed since once a grant is added, it cannot be removed from the record for the for sound financial management practices in the daily operations of the lab.

Lab Member:

- We deleted email address from LabMember's attributes for simplicity. Now LabMember's attributes are (not including the specific attributes for roles): id, labId, name, education, phoneNumber.

Lab Manager:

- We decided to remove Lab Manager's ability to update bookings and subject info - we decided that bookings shouldn't be updated, as this could lead to booking conflicts, and conditions of participants shouldn't be changed after the fact.
- Grant update is now solely handled by the DBA, as the primary keys in the original design complicates this operation for Lab Manager.

Fund_AppliedGrant & Fund_ApprovedGrant

- We changed the table name of Fund_AppliedGrant into Fund_ApprovedGrant, since this entity will only deal with grants that are approved.

PI:

- Decided not to let PI update the project name, since it is also a FK for other tables, for example the grant tables.