

CPSC 304 Milestone 3

Milestone #: 3

Date: 2/11/2022

Group Number: 81

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Abdullah Eshaq	32999518	u0j3b	eshaq@student.ubc.ca
Oussama Saoudi	69249035	p9l4q	bsaoudi@student.ubc.ca
Oliver Oxford	81360497	l3v1o	oliox@student.ubc.ca

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 the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

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

Tasks:

Construct GUI

- Create User login page
 - Create buttons for entering login and new account info
 - Create input fields
 - Link buttons to appropriate query interface
- Create Repository page
 - Create repo name input field
 - Create window showing repositories
 - Create window listing files in selected repository
 - Create sub-window to show the extensions list
 - Create fields for file path, name, and contents and button to submit.
- Create organization page
 - Selection field and button for organization name
 - Buttons to add organization to the user
 - Window for displaying mega contributors
 - Text to show the average contributor commits
- Create tab functionality for User Login, Repositories, and organizations

Construct Query Interface, and associated SQL commands

- Must construct a set of functions to perform insertions, selections, updates, and deletes on the database.
- `select_user(username, password)` -> selects user with username and password in database
- `create_user(username, password)` -> Inserts new user into the user relation
- `get_repositories(repo_name, organization_name)` -> selects list of repository objects matching the given name
 - JOIN
- `get_repository_objects(repository_name)` -> selects list of repository objects from database
 - AGGREGATION WITH HAVING
- `view_contents(file_path, repository_name)` -> selects file contents of appropriate file from the database.
- `add_file(file_path, file_name, contents)` -> inserts file into repository_objects relation
- `delete_file(file_path)` -> removes filepath from dataset,
 - DELETE
- `add_commit(line_number, new_line_contents: int[], file_contents, file_path)` -> Inserts commit_change to database, updates file contents
- `get_organization(organization_name)` -> selects organizations matching the user's input
 - SELECTION
- `get_all_organizations()` -> selects all organizations
- `set_organization(username, organization_name)` -> updates user's organization to a new one.

University of British Columbia, Vancouver

Department of Computer Science

- UPDATE
- get_extensions_list(repository_name)-> Returns string list of the extensions of all files in the repository
 - PROJECTION
-
- get_num_user_commits(username)-> returns the number of commits made by this user
 - AGGREGATION WITH GROUP BY
- get_avg_contributer_commit(organization) -> returns the average number of commits made by contributors of a repository
 - NESTED AGGREGATION WITH GROUP BY
- find_mega_contributers(organization_name) -> find users who have commit to all repositories of an organization
 - DIVISION

Application Core

- Create Internal Repository class
- Create Internal Organization class
- Create internal Repository Object class
- Create Commit class
- Create LineChange class
- Create PageState class which tracks current tab, and other state

Task Assignment and Duedates:

- **Due November 14th, Eshaq:** Complete application core classes
- **Due November 22nd Eshaq:** Complete organization functions/queries and integrate with front end)
- **Due November 14th Oussama:** Construct interface with function stubs, complete user-related functions/queries,
- **Due November 22nd Oussama:** Complete repository-related functions and queries.
- **Due November 14th Oliver:** Complete skeleton of gui. Create the required windows and tabs. Construct buttons for the user page. Complete buttons and input fields in the User page
- **Due November 22nd Oliver:** Create Buttons and fields for user input in repository and organization tabs.
- **Due November 28th Oussama, Eshaq, Oliver:** Integration of components.

Sample UI:

