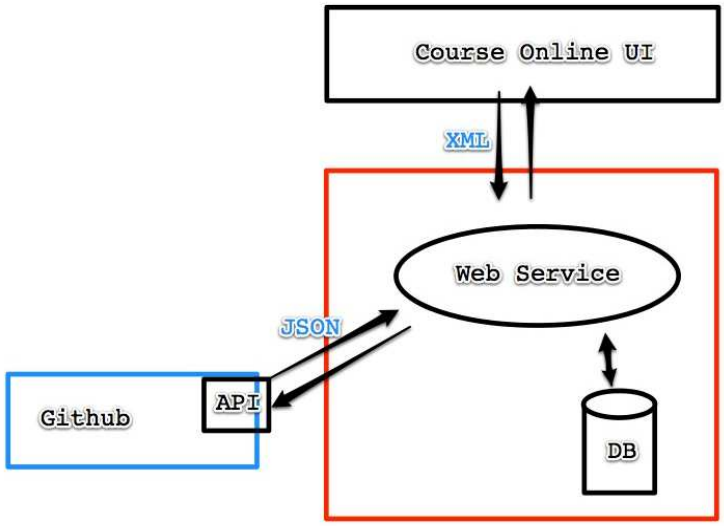
Code management (online course)

Design Document

Group 3: *Bo Li 3319406* *Xiang Xiao 3321515*
Ni Xin 3308139 *Halun Zhang 3354270*

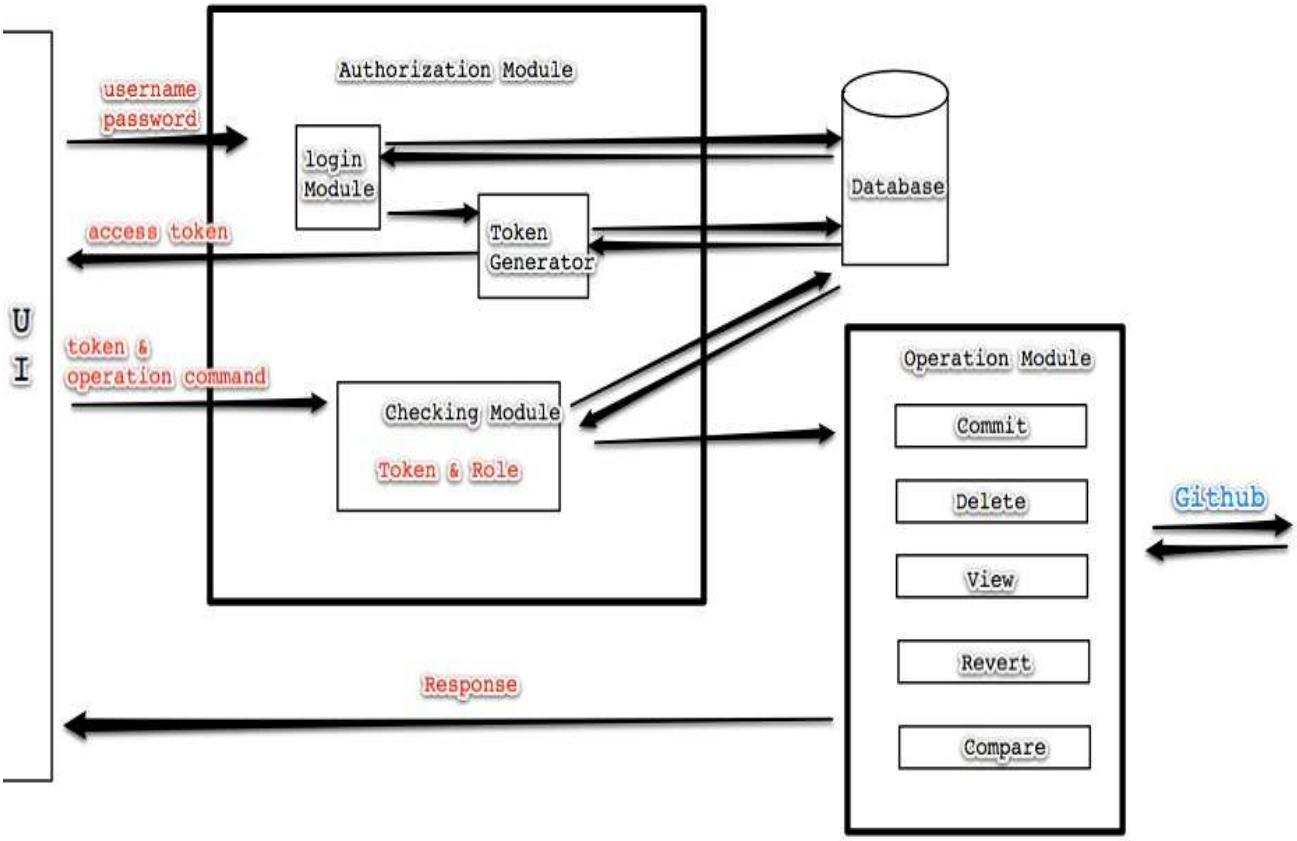
1. Architecture

1) General architecture



The graph shows a general view of the project architecture. The web service contacts with UI model through XML. In addition, it implements the basic functions of online course with the API that offered by Github. JSON is the default communication form between our web service and Github. We use mysql database to store the user information, resource URI, etc.

2) KEY MODULE ARCHITECTURE

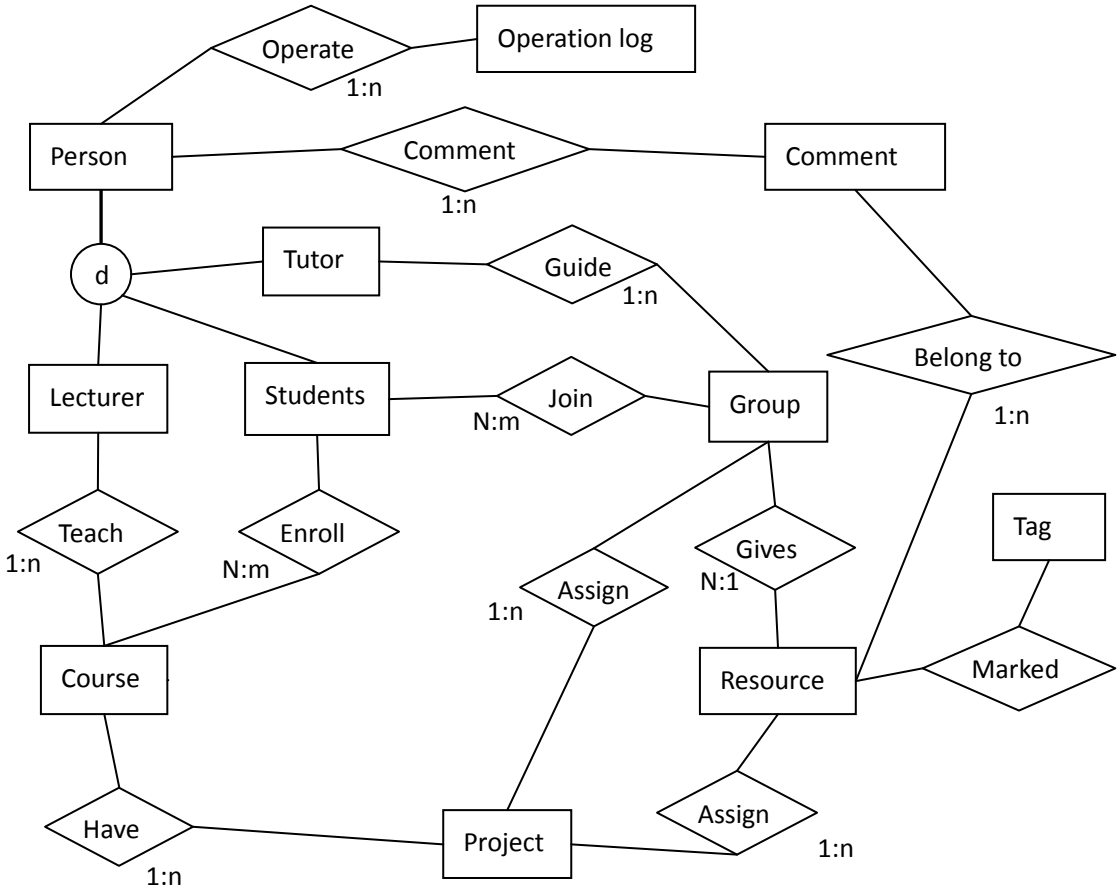


We divide our architecture into two parts. One is the login part. In this part, once the application receives the username and password, the login module checks the login information. If it is correct, then the token generator generates a token, and gives back to the UI. And the token should be written to the local database. This can help application to keep a record of the user's authority.

The second part is the main part of the application. After the user has logged in our application, UI sends the request with the username and access token. After the token and role have been checked, requests are sent to operation module to access the Github server.

2. Data model

1) ER DIAGRAM



2) DATABASE SCHEMA:

Person

Person_id	name	role	password	token

User role (student, lecturer and tutor) is kept in a single person table.

Course

Course_id	Course_name	Lecturer_id

Enrollment_course

Student_id	Course_id

This table maintains the enrollment relation between students and courses.

Group

Group_id	Group_name	Course_id	Tutor_id

Project

Project_id	Course_id

Project table records the relation of project and the course it belongs to.

Group_enrollment

Project_id	Student_id

This table maintains the enrollment relation between groups of students and their projects.

Operation_log

Operation_id	Operation_time	Operation_type	Operater_id	Resource_id

Operation table records any operations (functions that been called) during application running.

Comment

Comment_id	Comment_time	Comment_Content	Commenter_id	Resource_id

Comment table maintains the comment id, time, content and the user id of who gives the comment.

Resource

Resource_id	Group_Id	Project_id

Resource_id keeps track of URI which leads to code. etc in Github and the group and project this resource has been assigned, and the comments given to it.

Tag

Tag_id	Tag_content	Resource_Id

Tag_id keeps track of the tag, and every tag belongs to one resource.