File Sharing System for CrsMgr – Final Project

Group ID: poc353_1

August 15th 2016

Demostenis Kioussis
Computer Science Student
ID: 2768634

Tian Nan Liu
Computer Science Student
ID: 26580017

Shayne Mclevy
Computer Science Student
ID: 27452160

Jordan Orsini
Computer Science Student
ID: 26471196

Hussain Qabbani
Computer Science Student
ID: 25609879

Table of Content

Login Information:

URL: https://poc353_1.encs.concordia.ca
Username: poc353_1
Password: 353CoMp1

Design Decisions:

We began our project from scratch by designing our E-R diagram. We have referenced some of the special design techniques mentioned in our tutorials and incorporated those principles with our own ideas to generate our E-R diagram.

We started off with the creation of the Users table, which contains all users and information for each user. We needed to differentiate the roles of each user and we created three tables that inherit the attributes of a user (Admins, TAs, Teachers, and Students). We differentiate users by their user_type attribute, which is represented by different integers (1 for admins, 2 for teachers, 3 for teaching assistants, and 4 for students). Only Student type users use the attributes group_id and grade because they are uniquely given to students only, which is implemented on our platform. The whole purpose of this design is to ensure that user roles are well defined and avoid redundancy. All attributes in each table are shown in the E-R diagram.

In order to make sure that students can form into groups to work on their assignments and projects, we made a Groups table. For each group, we have a group_id that uniquely identifies each group and a leader_id that is unique for each group. This is because each group can only have one leader, which is a student, and each student can only enroll in one group. In the Users table which displays the information of all Users, we have a column for their group_id since we use this attribute as a foreign key. As a result, a student will not have a group_id if the group is deleted and a student cannot enroll in multiple groups at the same time due to the implementation of uniqueness in our database structure. We made sure by having a one-to-many relation from the Students table to Groups table. We designed our database so that only users of type 4, which are students, can enroll into groups.

On our actual course platform, only teachers can create new assignments. As new assignments are created, they become accessible to students and each assignment is then stored into the Assignments table. More specifically, this table stores the id of the assignment and the type of the assignment (1 for assignments and 2 for projects).

The file table is used to link assignments and submitted versions for a group. It contains the attributes assignment_id, group_id and file_id. We use this table so that we do not need to repeat group_id and assignment_id for every file submitted. This means that a File can have multiple versions, but the only repeated attribute is the file id. In other words, each group can have many Files and each file can only associate to one specific assignment. This design allows our platform to save memory and improve efficiency. We designed the database so that only admin users can archive files, which are unique from each group. Each file can only be archived by one admin and an admin can archive many files. We use the File_versions table to hold all the files submitted. There are many file versions for one specific type of file. We designed the File_versions table as a weak-entity as the table does not contain a primary key. The version number attribute, together with the primary key, file_id, from Files table, should be unique as there can be many versions for one specific type of assignment.

Rollbacks, Recovers, Deletes and Downloads all share the same attributes : TYPE_id, version_number, file_id, TYPE_date ; where TYPE is replaced by its table name, each entry is a primary key. The reason we made 4 separate tables rather than one table with a “type” attribute is because it's faster to search a smaller, more specific table for a given entry than searching through a large table for an entry. We essentially compartmentalize our data. TYPE_id is used to store the user who performed the action, version number the version of a file, file_id the file we perform the action on, and TYPE_date is the date the action was performed. Each entry is a primary key because they must all be used to identify an entry. For example, if a user deletes a file, recovers that file then deletes it again, the Deletes table will contain two entries with identical delete_id, file_id, version_number, however the date will be the discriminating attribute. Likewise, to know which file we are performing the action on we need to know the file_id and the file_version. Because it’s possible for two users to delete the same file, we must also keep the delete_id as a primary key. The situation is the same for the other 3 tables.

Design Decisions:

System details is a special table in our design as it has no relation to other tables, but it is used to keep track of the maximum data a group can use, and the start and end date of the course. Because and admin can only perform archiving after the class has ended, we use this table as the reference date. Similarly, if a user tries to submit a file which would result in the group using all of their allotted date, this table is where we get the data cap information. Because it doesn't make sense to have more than a single row in this table, we introduce the singleton column. This field cannot be null, and it must be unique and only accepts one enumerated value, the character 'a'. Because of the constraints on this column, we guarantee that there can only be one row.

Limitations of our platform:

- Only usable for one course, it does not work for multiple courses since we did not implement it this way.
- Since we only have one course, we will also need to make a new course table which will take on group ids and ensure that groups that are suppose to be in course A cannot contain members from course B and vice-versa.
- Although we have a grade attribute for students, we do not have a specific grades for each assignments and projects. Our platform also does not have grade distributions for each graded work or required statistics.
- Teachers cannot upload assignment instructions on our platform for students
- Teaching assistants cannot upload assignment feedback

Apps Supported:

- Mozilla Firefox, Internet Explorer, Chrome

Descriptions of the Database:

GROUPS table:

In this table, we have group_id as primary key and we want to differentiate each group with this group_id by implementing an auto increment to ensure uniqueness. Each group will have a leader_id, who is only restricted to students with restrictions implemented in our back-end code. The leader id, which is a foreign key, will reference to the user_id from the Users table. Only students can join groups and each student can only join one group. Each group can only have one leader_id, which indicates that each group can only have one group leader, who must be a student. Lastly, we have data_taken to store the size of files uploaded in each group. The purpose of this table is to allow students to join groups, so that each group of students may have their own depository and uploaded files.

USERS table:

In this table, we have user_id as our primary key which identifies every student with a unique integer. We ensure uniqueness by implementing an auto increment for our primary key. We also have attributes such as username and password so that users may access to their files, upload, download, rollback, and join group later on. In addition, we have their first_name and last_name as attributes for their names. Each user is distinguished by the user_type attribute: 1 for admin, 2 for teacher, 3 for teaching assistant, and 4 for student. Each student will eventually join a group during their course enrollment period and will have a group_id once they successfully join the group. The foreign key, which is the group_id, will reference to the group_id attribute from the Groups table. Each student's last login will be stored into the lastlogin feature and a grade will be given to each student at the end of the course. The purpose of this table is to create and store information regarding each user and provides more specific utility to different types of users, such as only allowing students to join groups.

System_details table:

In this table, we have no primary key. We have start_date, end_date to track the duration of the course. We have max_data as a maximum cap for the total file size allowed to be uploaded by students. We only allow one row in this table and we do this by assigning a unique key to the singleton attribute. The purpose of this table is to automatically trigger the archive function implemented in our back-end, which saves all files uploaded by all groups into a zip folder.

Assignments table:

In this table, we store the different types of graded work such as assignment and project into the assignment_type attribute. We have the assignment_id as primary key, which has the auto increment function implemented to ensure uniqueness. The purpose of this table is to differentiate the different types of work that need to be completed in the enrolled course. There can be many assignments of the same type, but they will all have different assignment_ids.

Descriptions of the Database:

FILES table:

We use this table to store all relevant information about assignments and projects in the course. We differentiate each file with a primary key called file_id, meaning Assignment 1 and Assignment 2 will have different file_id with the implementation of auto increment. The is_archived attribute is used as a Boolean column, whereas one means the file has been archived and zero means the file has not been archived. The group_id is a foreign key that references to the unique group_id from the Groups table. The assignment_id is also a foreign key, but references to the unique assignment_id from the Assignments table. In other words, this table contains the uploaded files, known as the current version, of each assignment from each group.

File_versions table:

Since there are many files and possibly many versions for the same type of file, we need this table to differentiate between the current and all other versions of the same assignment. This is extremely useful so that teachers/TAs may download the correct version from numerous submissions by the students. We have version_number and file_id as the primary key to uniquely identify each file with an auto increment. For instance, if student A has uploaded 10 files for Assignment #1, the current version would be the most recent uploaded file. The is_current attribute checks whether the individual file is the current version of the assignment or not and returns 1 if it is true and 0 if it is false. The data attribute is used to contain the content of the file. We also have numerous attributes that provide more information regarding each file which is shown more in detail on our E-R diagram. Upload_id is an attribute given that it is a foreign key referencing to the user_id, the user who uploaded the file. The file_id is a foreign key that references to the file_id from Files table and makes the primary key of the table with the version_number attribute. We assume that students will upload many versions of the same assignment in the system and we will need a way to differentiate which of those versions is the one that the student wants to upload. The purpose of this table is to store all versions uploaded into the system and allows students to access those files. This table is also linked to other relations which will be shown below:

Downloads table:

In this table, the download_id is a foreign key that references to the user_id who downloaded the file. The version_number is a foreign key that references to the version_number of the file from the Files_versions table. The file_id is a foreign key that references to the file_id from the Files table. The download_date is an attribute that contains the specific date time of each download. Together, these attributes form a single primary key for this table. The purpose of this table is to know exactly who downloaded what file with specific details on what file version and version number. Every user in the system has the ability to download files. However, we only allow teachers and teaching assistants to download the current version of the assignment and allowing students to download any version that are uploaded into the system (not deleted ones).

Rollbacks table:

In this table, the rollback_id is a foreign key that references to the user_id who performed a rollback on a file. The version_number is a foreign key that references to the version_number of the file from the Files_versions table. The file_id is a foreign key that references to the file_id from the Files table. The rollback_date is an attribute that contains the specific date time of each rollback. Together, these attributes form a single primary key for this table. We made this table because we assume students may upload many versions of the same assignment work and they may revert back to older versions before their final submission. The purpose of this table is to know who performed a rollback and on which file with specific details on the file version and version number. Since only students can submit graded work, we only allow students to roll back on their files. Only group leaders can use rollback and each leader can rollback to any versions that are currently on the system.

Descriptions of the Database:

Recovers table:

In this table, the recover_id is a foreign key that references to the user_id who performed a rollback on a file. The version_number is a foreign key that references to the version_number of the file from the Files_versions table. The file_id is a foreign key that references to the file_id from the Files table. The recover_date is an attribute that contains the specific date time of the file recovery. Together, these attributes form a single primary key for this table. We made this table because we assume students may recover files that were deleted from the system. Only students can recover files and each student may recover as many files as they uploaded.

Deletes table:

In this table, the delete_id is a foreign key that references to the user_id who performed a rollback on a file. The version_number is a foreign key that references to the version_number of the file from the Files_versions table. The file_id is a foreign key that references to the file_id from the Files table. The delete_date is an attribute that contains the specific date time of the file deletion. Together, these attributes form a single primary key for this table. We made this table because we assume students may delete files from the system for various reasons. Only students can delete files and each student may delete as many files as they uploaded.

Database Assumptions:

1) Only admin can create, modify and remove Users. Each user is defined by a unique user_id

- We do not want anyone to add new users to the group aside from the admin. This is obvious

2). Admin already exists in DB

- We need to already have an existing admin within the system so we can add new users (including other system admins)

3). Only admin can modify System_details (change start/end date, data cap)

- We do not want students, teachers or ta to be able to change the data caps, or the course dates.

4). Only one row of System_detals

- We are only looking at one class. A class only has one start and end date. We do not want to be able to add more entries in this table because it would not make sense. This table is isolated from the other relations because it is not related to the other tables in any way. An admin can archive files only if the current date > class end date. A student can only upload a file if the sum of all the file sizes are < than limit

5). Only admin can archive all files

- Admin is responsible for dealing with the system. It doesn't make sense for a teacher, ta or student to be capable of archiving

6). Archiving files can only be done after the end of the class

- We should not be able to archive while class is in progress because then new files would not be able to be uploaded, as the system is closed

7). Only students can join groups

- Teachers, Ta's and admins have no reason to be in a group

8). A student can become the leader of only 1 group, identified by leader_id within Groups and must be a member of that group

- We do not want a student to become the leader of more group, because that would mean they are a member of two groups
- A student cannot lead a group in which they are not a member

9). A student can only join one group, identified by group_id within User

- Students cannot be graded multiple times in a class and we want to avoid redundancy in our database. We cannot have student A appearing in multiple groups because teachers will not be able to grade them.

10). Only students can upload/delete files. Deleted files are recoverable for by any student 24 hours.

- We forbid students, admins, and T.As from deleting files because only the student himself should have the right to alter his uploaded files. Due to potential misclicks by the students, we allow students to recover their deleted files within 24 hours of their deletion.

Database Assumptions:

11). Only group leaders can change roll back current version to a different version

- We the group leader to decide which file is to be marked

12). Teacher can create groups, place students in groups, delete groups and choose group leaders

- Since teachers are the only users teaching the course, they have the administrative rights and they are the users who decide the graded works in the course. We forbid admins or teaching assistants since they simply do not teach the course.

13). Only teacher can create new assignments / project, identified by unique assignment_id

- Teachers are the users who teach the class, they are the ones who assign graded works to their students. They should have the right to create assignments and projects. Admins and T.As cannot have the same right as teachers because they do not teach the course. Each assignment is identified by a unique id because there can be many assignments. With unique identifiers, there will not be any redundancy for the Assignments table.

14). Students can upload many different file_versions for a given assignment or project. Only one is the current version, identified by is_current within File_versions. Each file has a unique id, and file_version's key is version_number and file_id (from Files)

- To avoid redundancy, there can only be one current version of the assignment file uploaded by the student since the teacher and the T.A can only download the current version of the assignment file. For each file, we need to have a column to indicate whether the file is the current version of the assignment or not. This will allow the database to column data to tables such as Deletes, Rollbacks, Recovers, and Downloads.

15). Teachers and ta's can only download the current version of an upload, students can download any version.

- Teachers or TA's only need to look at what is to be marked (the current version)
- A student can download any version because it's possible they want to review an older version

Relational Model in 3NF:

Groups(group_id, data_taken, leader_id)
Users(user_id, firstname, last, username, password, user_type, grade, group_id, lastlogin)
System_details(start_date, end_date, max_data, singleton)
Assignments(assignment_id, assignment_type)
Files(file_id, is_archived, group_id, assignment_id)
File_versions(version_number, upload_id, file_id, is_current, data, checksum, upload_ip, size, name, is_deleted, sizechange)
Recovers(recover_id, version_number, file_id, recover_date)
Rollbacks(rollback_id, version_number, file_id, rollback_date)
Deletes(delete_id, version_number, file_id, delete_date)
Downloads(download_id, version_number, file_id, download_date)

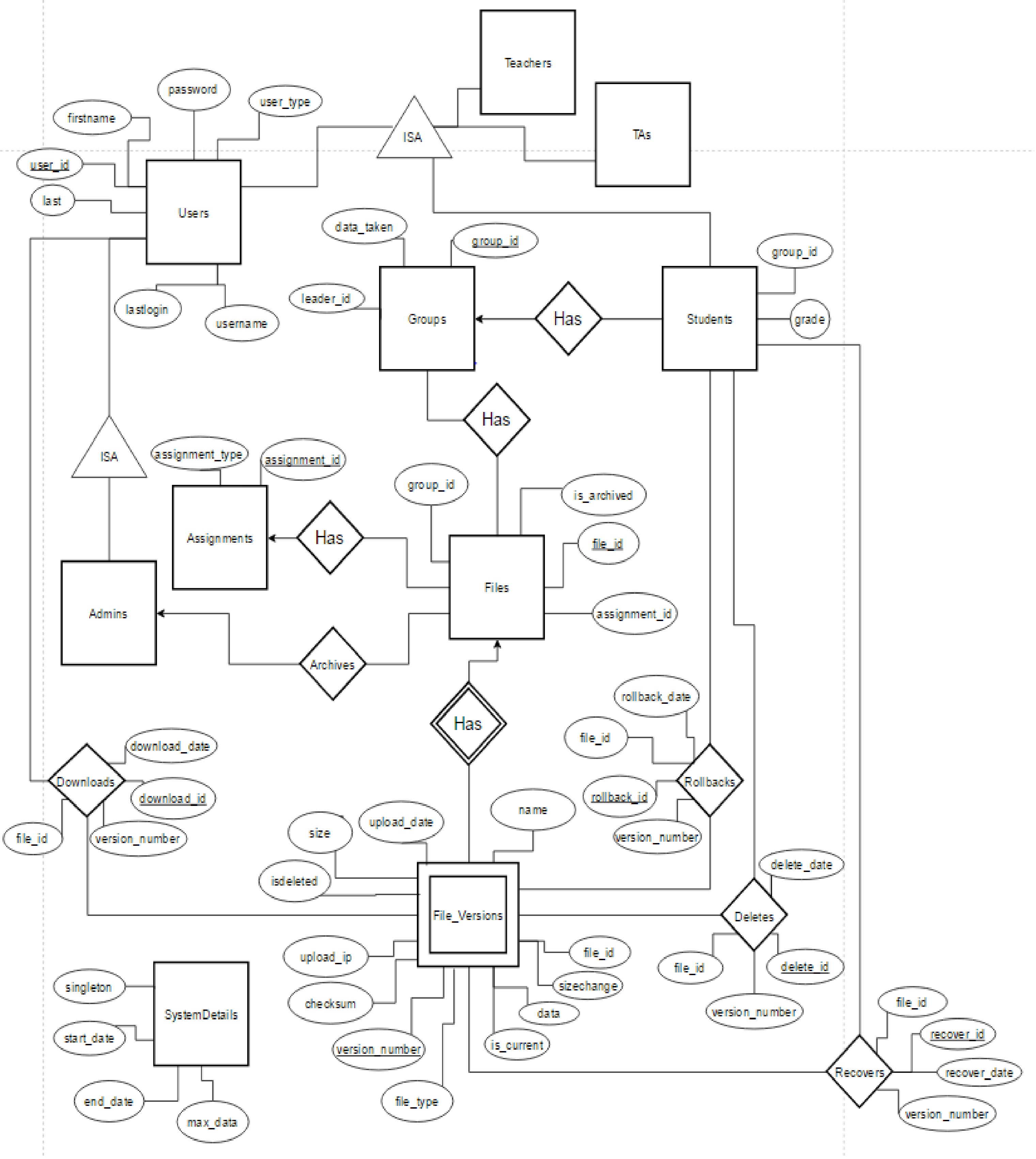
Functional Dependencies:

Groups(group_id → data_taken, group_d → leader_id)
Users(user_id → firstname, user_id → last, user_id → username, user_id → password, user_id → user_type, user_id → grade, user_id → group_id, user_id → lastlogin)
Assignments(assignment_id → assignment_type)
Files(file_id → is_archived, file_id → group_id, file_id → assignment_id)
File_versions(version_number upload_id file_id → is_current, version_number upload_id file_id → data, version_number upload_id file_id → checksum, version_number upload_id file_id → upload_ip, version_number upload_id file_id → upload_ip, version_number upload_id file_id → size, version_number upload_id file_id → name, version_number upload_id file_id → is_deleted, version_number upload_id file_id → sizechange)
Recovers(recover_id → version_number, recover_id → file_id, recover_id → recover_date)
Rollbacks(rollback_id → version_number, rollback_id → file_id, rollback_id → rollback_date)
Deletes(deletes_id → version_number, deletes_id → file_id, delete_id → delete_date)
Downloads(download_id → version_number, download_id → file_id, download_id → download_date)

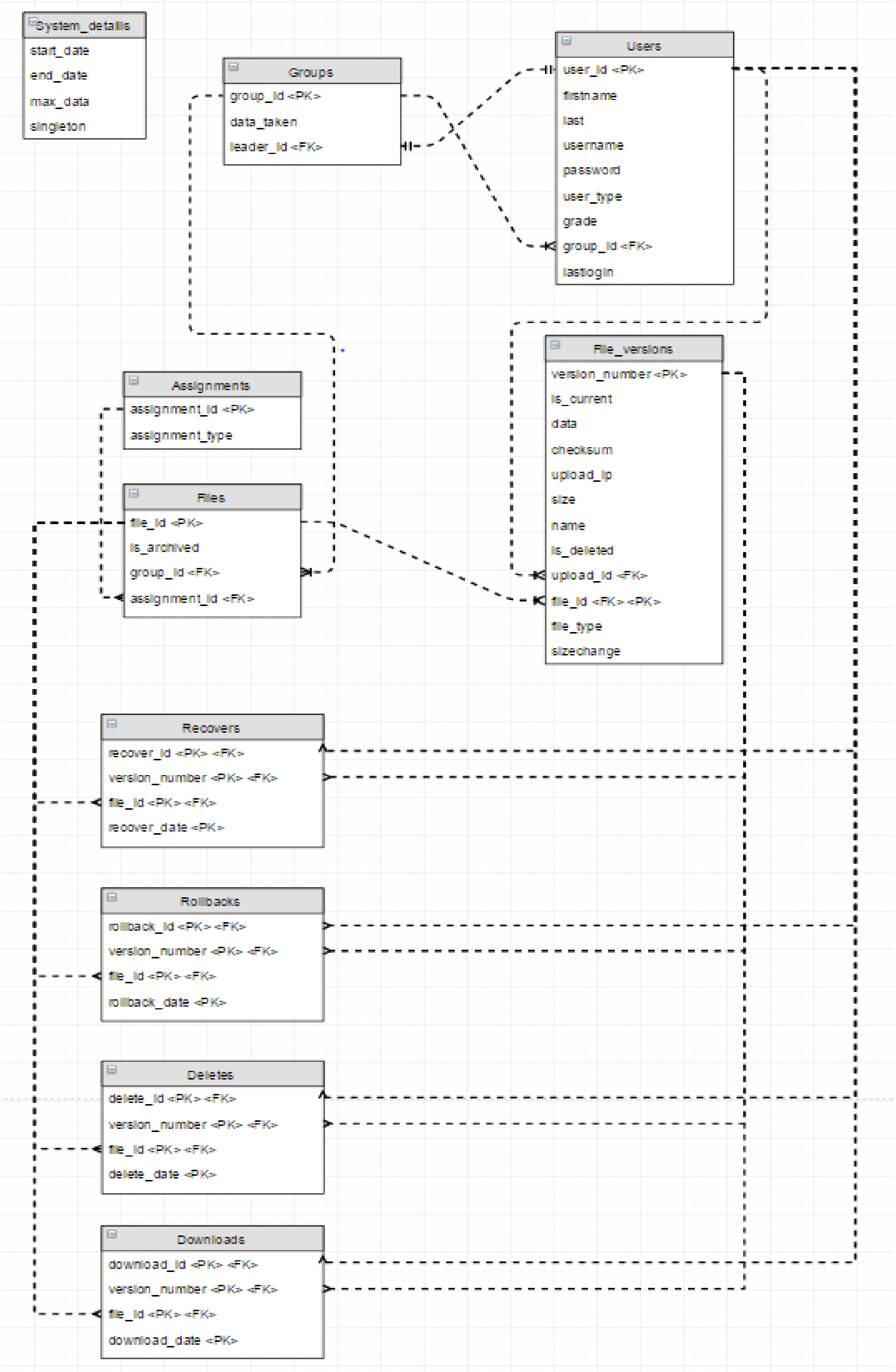
Check for 3NF:

In each table, every non-primary key attributes depend on the primary key of their table. Therefore, there are no transitive dependencies and every relation is non-trivial. For our database, the primary key in each table is also the super key. This has been verified for each functional dependencies in each table, which means that each table respects the conditions in the third normal form.

E-R Diagram:

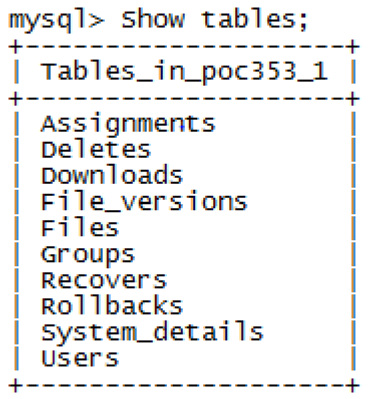

Teachers
TAs
password
user_type
firstname
ISA
user_id
last
Users
data_taken
group_id
group_id
leader_id
Groups
Has
Students
grade
lastlogin
username
Has
ISA
assignment_type
assignment_id
group_id
is_archived
Assignments
Has
file_id
Files
Admins
assignment_id
Archives
rollback_date
Has
file_id
download_date
Downloads
download_id
rollback_id
Rollbacks
name
file_id
version_number
version_number
size
upload_date
isdeleted
delete_date
File_Versions
Deletes
file_id
upload_ip
file_id
delete_id
sizechange
singleton
checksum
data
version_number
file_id
SystemDetails
version_number
is_current
recover_id
start_date
Recovers
recover_date
end_date
file_type
version_number
max_data

Relational Tables:



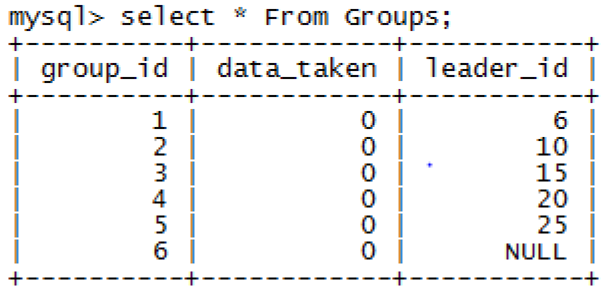
Queries:

1. Show all of your tables



```
mysql> show tables;
+---------------------+
| Tables_in_poc353_1 |
+---------------------+
| Assignments         |
| Deletes             |
| Downloads           |
| File_versions       |
| Files               |
| Groups              |
| Recovers            |
| Rollbacks           |
| System_details      |
| Users               |
+---------------------+
```

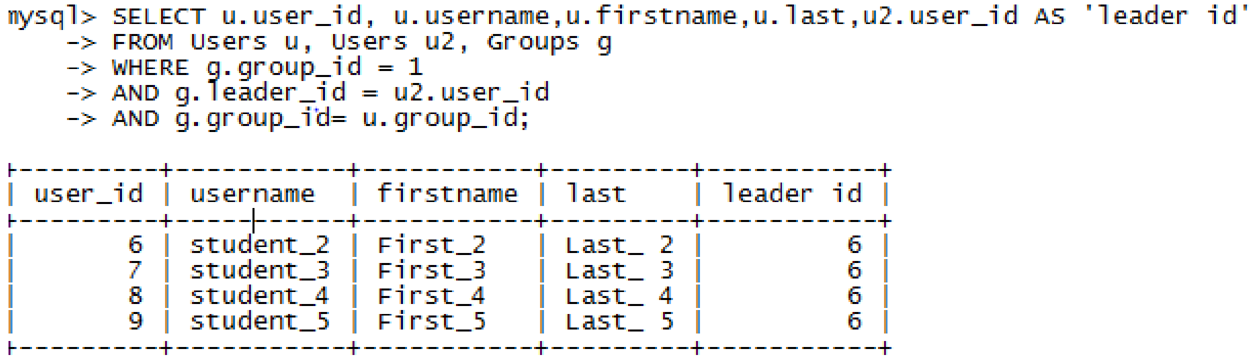
2. List of all groups in the FSS.



```
mysql> select * From Groups;
+----------+------------+-----------+
| group_id | data_taken | leader_id |
+----------+------------+-----------+
|        1 |          0 |         6 |
|        2 |          0 |        10 |
|        3 |          0 |        15 |
|        4 |          0 |        20 |
|        5 |          0 |        25 |
|        6 |          0 |      NULL |
+----------+------------+-----------+
```

3. Provide details for a given group (group members and group leader).

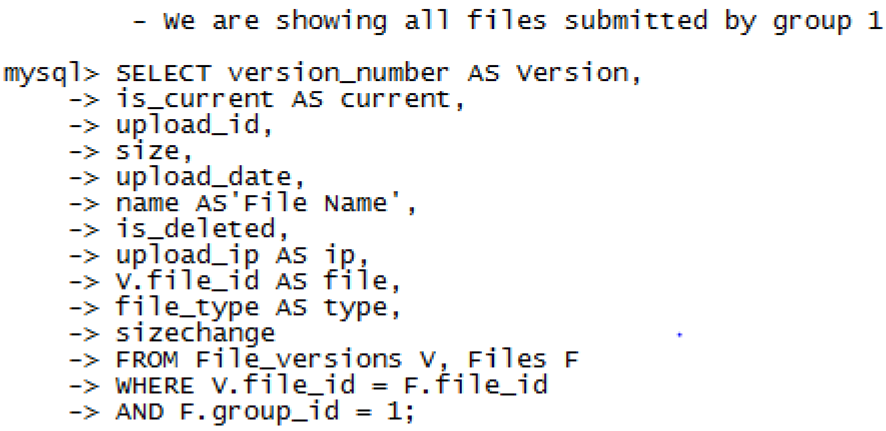
```
        - we are showing details of group 1
```



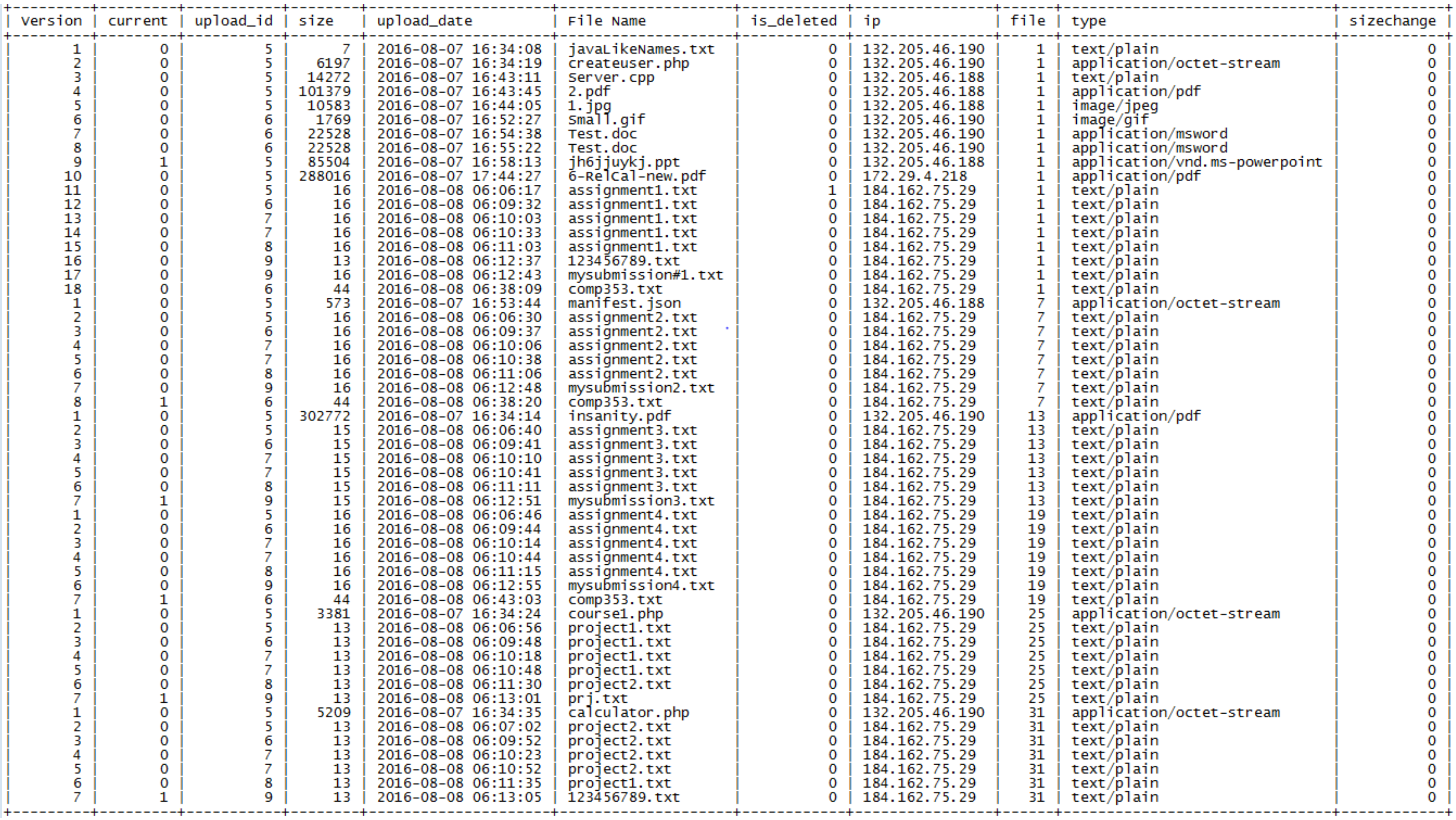
```
mysql> SELECT u.user_id, u.username,u.firstname,u.last,u2.user_id AS 'leader id'
    -> FROM Users u, Users u2, Groups g
    -> WHERE g.group_id = 1
    -> AND g.leader_id = u2.user_id
    -> AND g.group_id= u.group_id;

+---------+-----------+-----------+---------+-----------+
| user_id | username  | firstname | last    | leader id |
+---------+-----------+-----------+---------+-----------+
|       6 | student_2 | First_2   | Last_ 2 |         6 |
|       7 | student_3 | First_3   | Last_ 3 |         6 |
|       8 | student_4 | First_4   | Last_ 4 |         6 |
|       9 | student_5 | First_5   | Last_ 5 |         6 |
+---------+-----------+-----------+---------+-----------+
```

Queries:

4. Provide details of all shared files for a given group (i.e. each project or assignment and the history for each file for each project or assignment).



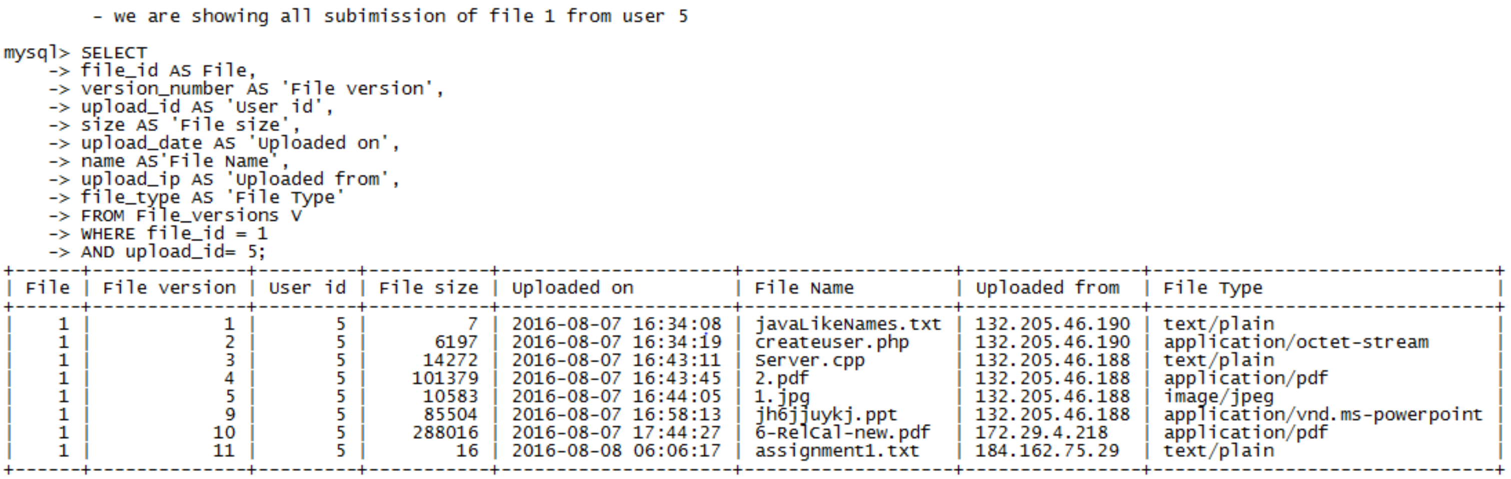
```
        - We are showing all files submitted by group 1

mysql> SELECT version_number AS Version,
    -> is_current AS current,
    -> upload_id,
    -> size,
    -> upload_date,
    -> name AS'File Name',
    -> is_deleted,
    -> upload_ip AS ip,
    -> V.file_id AS file,
    -> file_type AS type,
    -> sizechange
    -> FROM File_versions V, Files F
    -> WHERE V.file_id = F.file_id
    -> AND F.group_id = 1;
```



```
+---------+---------+-----------+--------+---------------------+--------------------+------------+----------------+------+-------------------------------+------------+
| Version | current | upload_id | size   | upload_date         | File Name          | is_deleted | ip             | file | type                          | sizechange |
+---------+---------+-----------+--------+---------------------+--------------------+------------+----------------+------+-------------------------------+------------+
|       1 |       0 |         5 |      7 | 2016-08-07 16:34:08 | javaLikeNames.txt  |          0 | 132.205.46.190 |    1 | text/plain                    |          0 |
|       2 |       0 |         5 |   6197 | 2016-08-07 16:34:19 | createuser.php     |          0 | 132.205.46.190 |    1 | application/octet-stream      |          0 |
|       3 |       0 |         5 |  14272 | 2016-08-07 16:43:11 | Server.cpp         |          0 | 132.205.46.188 |    1 | text/plain                    |          0 |
|       4 |       0 |         5 | 101379 | 2016-08-07 16:43:45 | 2.pdf              |          0 | 132.205.46.188 |    1 | application/pdf               |          0 |
|       5 |       0 |         5 |  10583 | 2016-08-07 16:44:05 | 1.jpg              |          0 | 132.205.46.188 |    1 | image/jpeg                    |          0 |
|       6 |       0 |         6 |   1769 | 2016-08-07 16:52:27 | Small.gif          |          0 | 132.205.46.190 |    1 | image/gif                     |          0 |
|       7 |       0 |         6 |  22528 | 2016-08-07 16:54:38 | Test.doc           |          0 | 132.205.46.190 |    1 | application/msword            |          0 |
|       8 |       0 |         6 |  22528 | 2016-08-07 16:55:22 | Test.doc           |          0 | 132.205.46.190 |    1 | application/msword            |          0 |
|       9 |       1 |         5 |  85504 | 2016-08-07 16:58:13 | jh6jjuykj.ppt      |          0 | 132.205.46.188 |    1 | application/vnd.ms-powerpoint |          0 |
|      10 |       0 |         5 | 288016 | 2016-08-07 17:44:27 | 6-RelCal-new.pdf   |          0 | 172.29.4.218   |    1 | application/pdf               |          0 |
|      11 |       0 |         5 |     16 | 2016-08-08 06:06:17 | assignment1.txt    |          1 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      12 |       0 |         6 |     16 | 2016-08-08 06:09:32 | assignment1.txt    |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      13 |       0 |         7 |     16 | 2016-08-08 06:10:03 | assignment1.txt    |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      14 |       0 |         7 |     16 | 2016-08-08 06:10:33 | assignment1.txt    |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      15 |       0 |         8 |     16 | 2016-08-08 06:11:03 | assignment1.txt    |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      16 |       0 |         9 |     13 | 2016-08-08 06:12:37 | 123456789.txt      |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      17 |       0 |         9 |     16 | 2016-08-08 06:12:43 | mysubmission#1.txt |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|      18 |       0 |         6 |     44 | 2016-08-08 06:38:09 | comp353.txt        |          0 | 184.162.75.29  |    1 | text/plain                    |          0 |
|       1 |       0 |         5 |    573 | 2016-08-07 16:53:44 | manifest.json      |          0 | 132.205.46.188 |    7 | application/octet-stream      |          0 |
|       2 |       0 |         5 |     16 | 2016-08-08 06:06:30 | assignment2.txt    |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       3 |       0 |         6 |     16 | 2016-08-08 06:09:37 | assignment2.txt    |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       4 |       0 |         7 |     16 | 2016-08-08 06:10:06 | assignment2.txt    |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       5 |       0 |         7 |     16 | 2016-08-08 06:10:38 | assignment2.txt    |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       6 |       0 |         8 |     16 | 2016-08-08 06:11:06 | assignment2.txt    |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       7 |       0 |         9 |     16 | 2016-08-08 06:12:48 | mysubmission2.txt  |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       8 |       1 |         6 |     44 | 2016-08-08 06:38:20 | comp353.txt        |          0 | 184.162.75.29  |    7 | text/plain                    |          0 |
|       1 |       0 |         5 | 302772 | 2016-08-07 16:34:14 | insanity.pdf       |          0 | 132.205.46.190 |   13 | application/pdf               |          0 |
|       2 |       0 |         5 |     15 | 2016-08-08 06:06:40 | assignment3.txt    |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       3 |       0 |         6 |     15 | 2016-08-08 06:09:41 | assignment3.txt    |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       4 |       0 |         7 |     15 | 2016-08-08 06:10:10 | assignment3.txt    |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       5 |       0 |         7 |     15 | 2016-08-08 06:10:41 | assignment3.txt    |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       6 |       0 |         8 |     15 | 2016-08-08 06:11:11 | assignment3.txt    |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       7 |       1 |         9 |     15 | 2016-08-08 06:12:51 | mysubmission3.txt  |          0 | 184.162.75.29  |   13 | text/plain                    |          0 |
|       1 |       0 |         5 |     16 | 2016-08-08 06:06:46 | assignment4.txt    |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       2 |       0 |         6 |     16 | 2016-08-08 06:09:44 | assignment4.txt    |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       3 |       0 |         7 |     16 | 2016-08-08 06:10:14 | assignment4.txt    |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       4 |       0 |         7 |     16 | 2016-08-08 06:10:44 | assignment4.txt    |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       5 |       0 |         8 |     16 | 2016-08-08 06:11:15 | assignment4.txt    |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       6 |       0 |         9 |     16 | 2016-08-08 06:12:55 | mysubmission4.txt  |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       7 |       1 |         6 |     44 | 2016-08-08 06:43:03 | comp353.txt        |          0 | 184.162.75.29  |   19 | text/plain                    |          0 |
|       1 |       0 |         5 |   3381 | 2016-08-07 16:34:24 | course1.php        |          0 | 132.205.46.190 |   25 | application/octet-stream      |          0 |
|       2 |       0 |         5 |     13 | 2016-08-08 06:06:56 | project1.txt       |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       3 |       0 |         6 |     13 | 2016-08-08 06:09:48 | project1.txt       |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       4 |       0 |         7 |     13 | 2016-08-08 06:10:18 | project1.txt       |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       5 |       0 |         7 |     13 | 2016-08-08 06:10:48 | project1.txt       |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       6 |       0 |         8 |     13 | 2016-08-08 06:11:30 | project2.txt       |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       7 |       1 |         9 |     13 | 2016-08-08 06:13:01 | prj.txt            |          0 | 184.162.75.29  |   25 | text/plain                    |          0 |
|       1 |       0 |         5 |   5209 | 2016-08-07 16:34:35 | calculator.php     |          0 | 132.205.46.190 |   31 | application/octet-stream      |          0 |
|       2 |       0 |         5 |     13 | 2016-08-08 06:07:02 | project2.txt       |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
|       3 |       0 |         6 |     13 | 2016-08-08 06:09:52 | project2.txt       |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
|       4 |       0 |         7 |     13 | 2016-08-08 06:10:23 | project2.txt       |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
|       5 |       0 |         7 |     13 | 2016-08-08 06:10:52 | project2.txt       |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
|       6 |       0 |         8 |     13 | 2016-08-08 06:11:35 | project1.txt       |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
|       7 |       1 |         9 |     13 | 2016-08-08 06:13:05 | 123456789.txt      |          0 | 184.162.75.29  |   31 | text/plain                    |          0 |
+---------+---------+-----------+--------+---------------------+--------------------+------------+----------------+------+-------------------------------+------------+
```

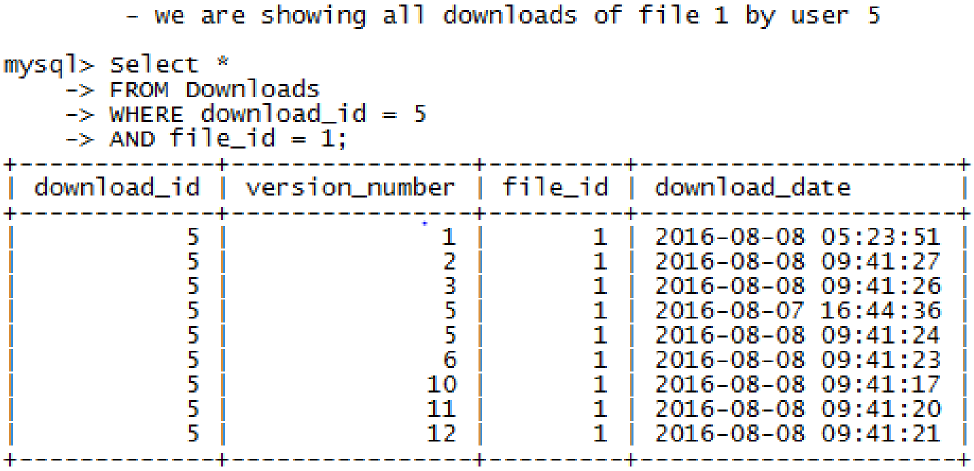
Queries:

5. Upload a given file by a given member.



```
        - we are showing all subimission of file 1 from user 5

mysql> SELECT
    -> file_id AS File,
    -> version_number AS 'File version',
    -> upload_id AS 'User id',
    -> size AS 'File size',
    -> upload_date AS 'Uploaded on',
    -> name AS'File Name',
    -> upload_ip AS 'Uploaded from',
    -> file_type AS 'File Type'
    -> FROM File_versions V
    -> WHERE file_id = 1
    -> AND upload_id= 5;
+------+--------------+---------+-----------+---------------------+-------------------+----------------+-------------------------------+
| File | File version | User id | File size | Uploaded on         | File Name         | Uploaded from  | File Type                     |
+------+--------------+---------+-----------+---------------------+-------------------+----------------+-------------------------------+
|    1 |            1 |       5 |         7 | 2016-08-07 16:34:08 | javaLikeNames.txt | 132.205.46.190 | text/plain                    |
|    1 |            2 |       5 |      6197 | 2016-08-07 16:34:19 | createuser.php    | 132.205.46.190 | application/octet-stream      |
|    1 |            3 |       5 |     14272 | 2016-08-07 16:43:11 | server.cpp        | 132.205.46.188 | text/plain                    |
|    1 |            4 |       5 |    101379 | 2016-08-07 16:43:45 | 2.pdf             | 132.205.46.188 | application/pdf               |
|    1 |            5 |       5 |     10583 | 2016-08-07 16:44:05 | 1.jpg             | 132.205.46.188 | image/jpeg                    |
|    1 |            9 |       5 |     85504 | 2016-08-07 16:58:13 | jh6jjuykj.ppt     | 132.205.46.188 | application/vnd.ms-powerpoint |
|    1 |           10 |       5 |    288016 | 2016-08-07 17:44:27 | 6-RelCal-new.pdf  | 172.29.4.218   | application/pdf               |
|    1 |           11 |       5 |        16 | 2016-08-08 06:06:17 | assignment1.txt   | 184.162.75.29  | text/plain                    |
+------+--------------+---------+-----------+---------------------+-------------------+----------------+-------------------------------+
```

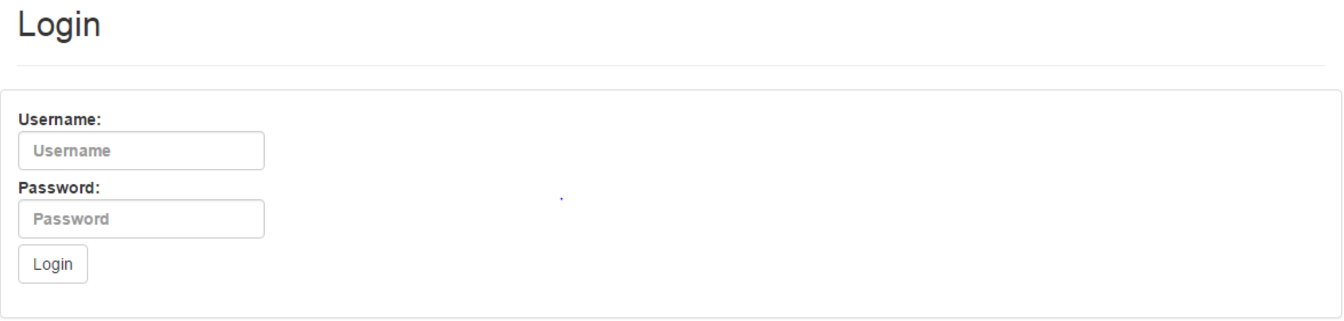
6. Download a given file by a given member



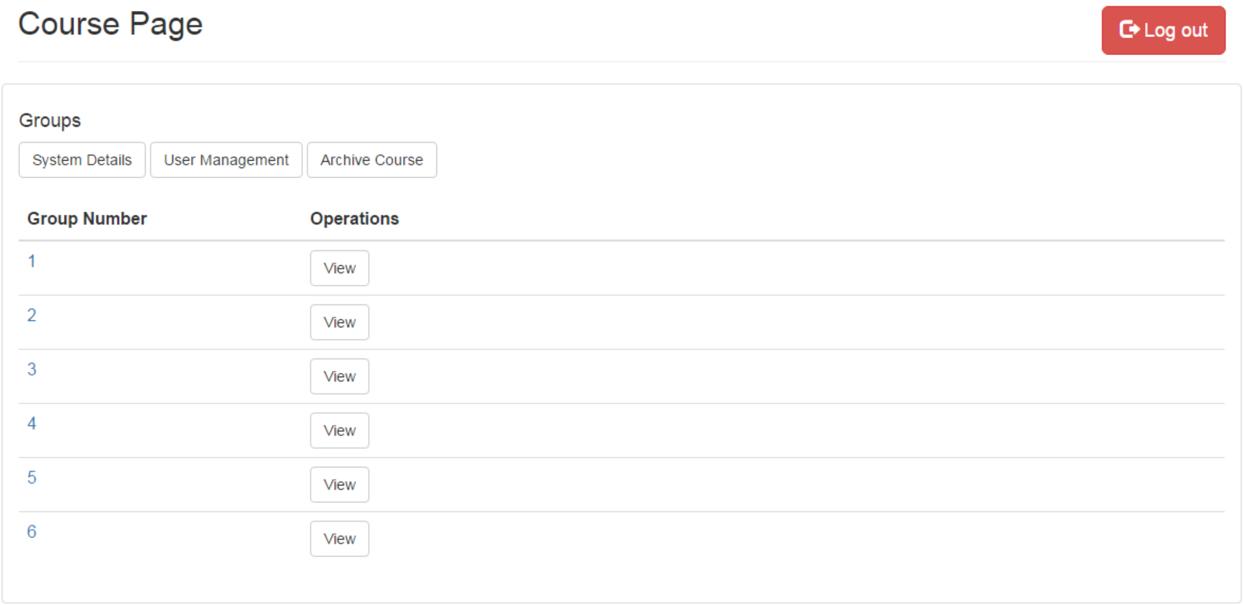
```
        - we are showing all downloads of file 1 by user 5

mysql> select *
    -> FROM Downloads
    -> WHERE download_id = 5
    -> AND file_id = 1;
+-------------+----------------+---------+---------------------+
| download_id | version_number | file_id | download_date       |
+-------------+----------------+---------+---------------------+
|           5 |              1 |       1 | 2016-08-08 05:23:51 |
|           5 |              2 |       1 | 2016-08-08 09:41:27 |
|           5 |              3 |       1 | 2016-08-08 09:41:26 |
|           5 |              5 |       1 | 2016-08-07 16:44:36 |
|           5 |              5 |       1 | 2016-08-08 09:41:24 |
|           5 |              6 |       1 | 2016-08-08 09:41:23 |
|           5 |             10 |       1 | 2016-08-08 09:41:17 |
|           5 |             11 |       1 | 2016-08-08 09:41:20 |
|           5 |             12 |       1 | 2016-08-08 09:41:21 |
+-------------+----------------+---------+---------------------+
```

User Interface - Admin

- User login page: login.php
- User login as admin

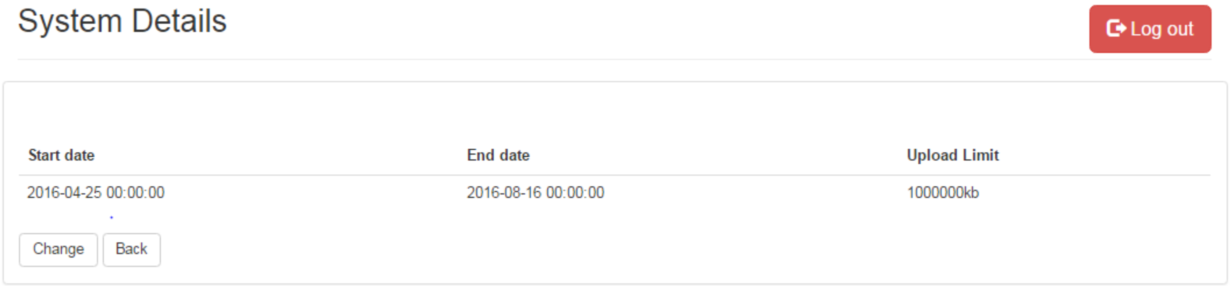


- Course webpage, logged in as an administrator
- Accessible through course.php
- Admins can view submitted assignments by each group with the view button and can archive the entire course with the Archive Course button



Group Number	Operations
1	View
2	View
3	View
4	View
5	View
6	View

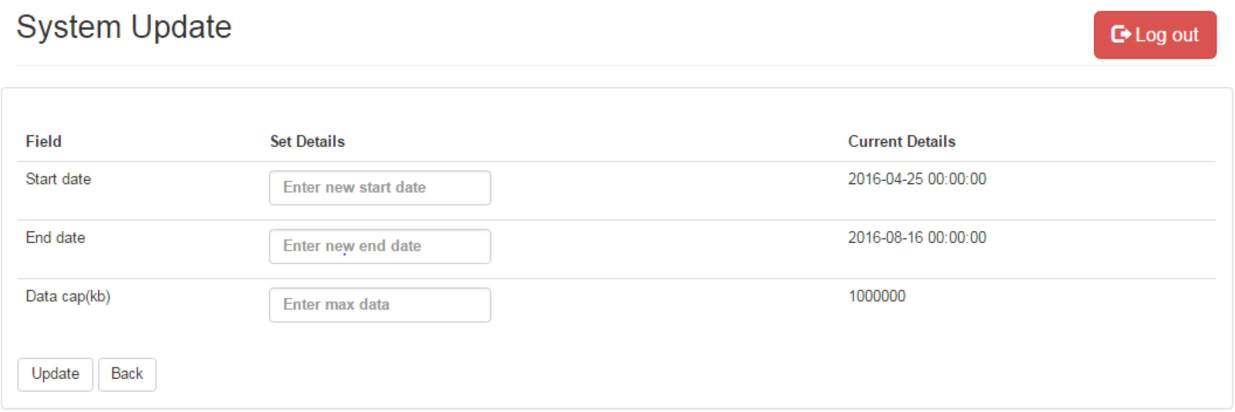
- System details, given the duration of the course and upload limit of file size for the whole class
- Accessible from systemdetails.php



Start date	End date	Upload Limit
2016-04-25 00:00:00	2016-08-16 00:00:00	1000000kb

User Interface - Admin

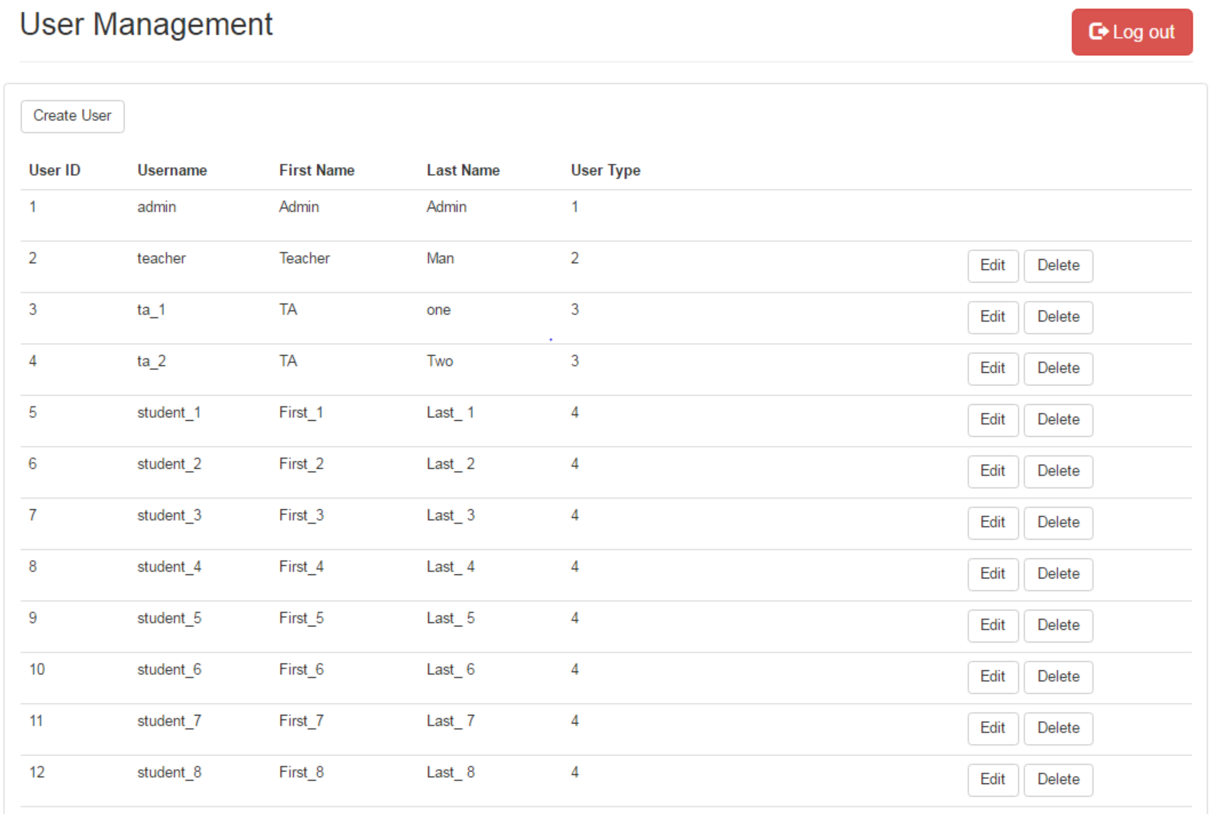
- Change the duration of the course and the upload limit of file size
- Accessible from systemupdate.php



System Update

Field	Set Details	Current Details
Start date	Enter new start date	2016-04-25 00:00:00
End date	Enter new end date	2016-08-16 00:00:00
Data cap(kb)	Enter max data	1000000

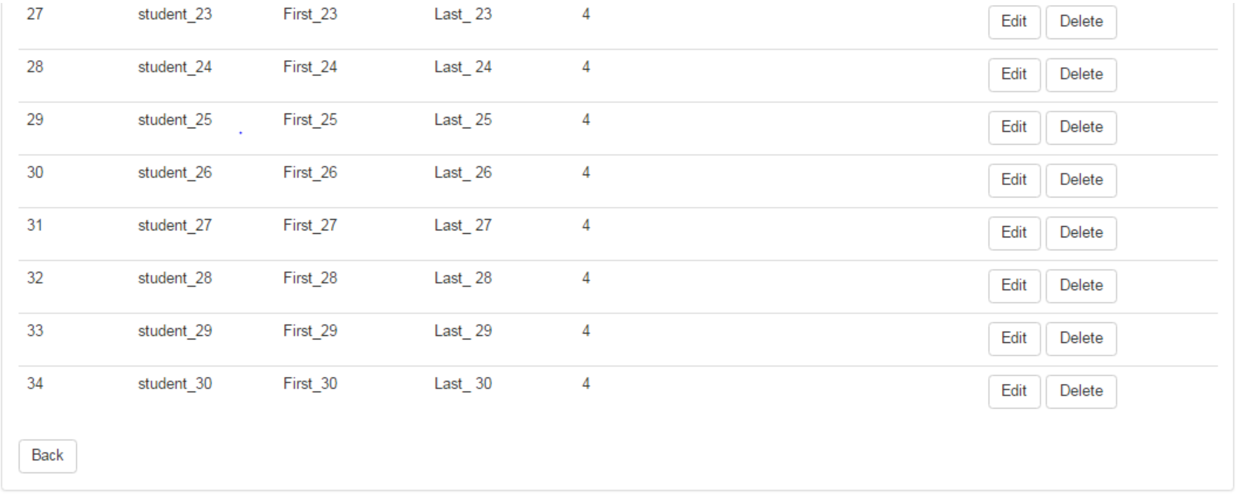
- Contain user information of all users enrolled in the course
- Only the admin can see this page, he can create, edit, and delete users
- Accessible through Usermanagement.php



User Management

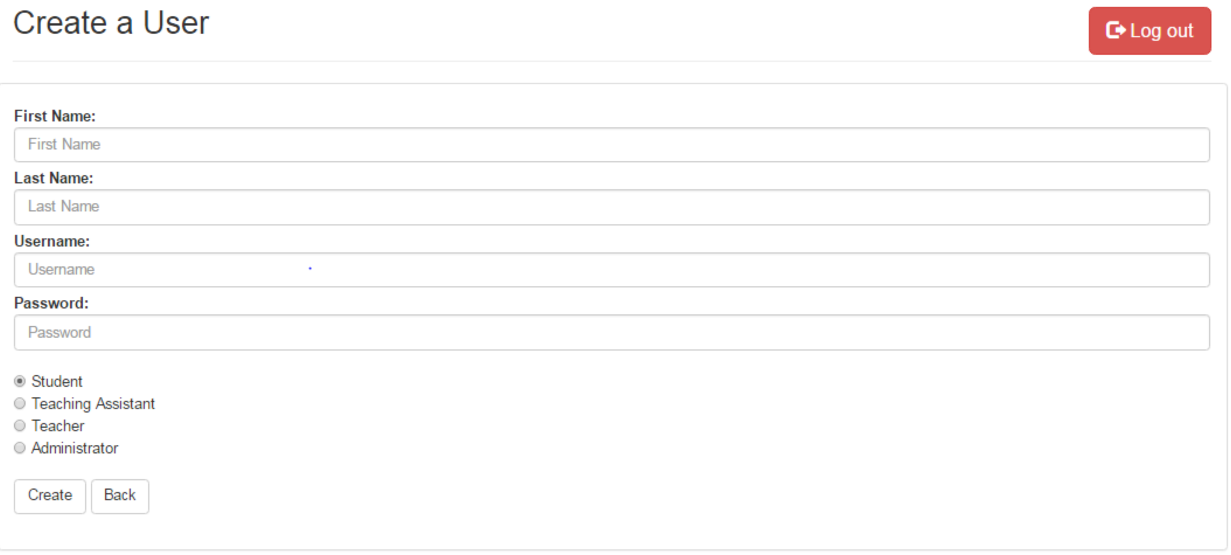
User ID	Username	First Name	Last Name	User Type	
1	admin	Admin	Admin	1	
2	teacher	Teacher	Man	2	Edit Delete
3	ta_1	TA	one	3	Edit Delete
4	ta_2	TA	Two	3	Edit Delete
5	student_1	First_1	Last_ 1	4	Edit Delete
6	student_2	First_2	Last_ 2	4	Edit Delete
7	student_3	First_3	Last_ 3	4	Edit Delete
8	student_4	First_4	Last_ 4	4	Edit Delete
9	student_5	First_5	Last_ 5	4	Edit Delete
10	student_6	First_6	Last_ 6	4	Edit Delete
11	student_7	First_7	Last_ 7	4	Edit Delete
12	student_8	First_8	Last_ 8	4	Edit Delete

User Interface - Admin



27	student_23	First_23	Last_ 23	4	Edit Delete
28	student_24	First_24	Last_ 24	4	Edit Delete
29	student_25	First_25	Last_ 25	4	Edit Delete
30	student_26	First_26	Last_ 26	4	Edit Delete
31	student_27	First_27	Last_ 27	4	Edit Delete
32	student_28	First_28	Last_ 28	4	Edit Delete
33	student_29	First_29	Last_ 29	4	Edit Delete
34	student_30	First_30	Last_ 30	4	Edit Delete

- Admins can create new users by filling the textboxes, a window popup will show on the right corner of the browser when an existing username already exists in the database to prevent duplicate usernames
- Accessible through createuser.php



Create a User

First Name:
First Name

Last Name:
Last Name

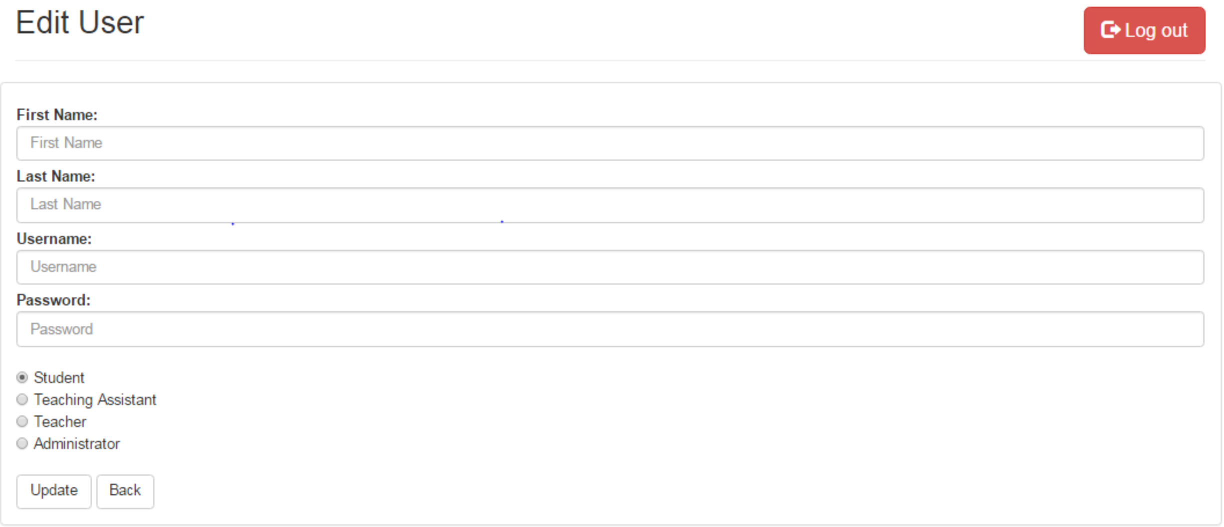
Username:
Username

Password:
Password

● Student
○ Teaching Assistant
○ Teacher
○ Administrator

User Interface - Admin

- Admins can edit existing users by filling the textboxes, a window popup will show on the right corner of the browser when an existing username already exists in the database to prevent duplicate usernames
- Accessible through edituser.php



Edit User

First Name:
First Name

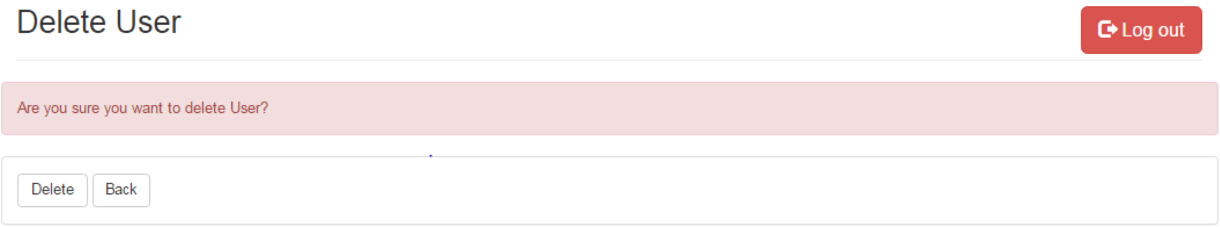
Last Name:
Last Name

Username:
Username

Password:
Password

● Student
○ Teaching Assistant
○ Teacher
○ Administrator

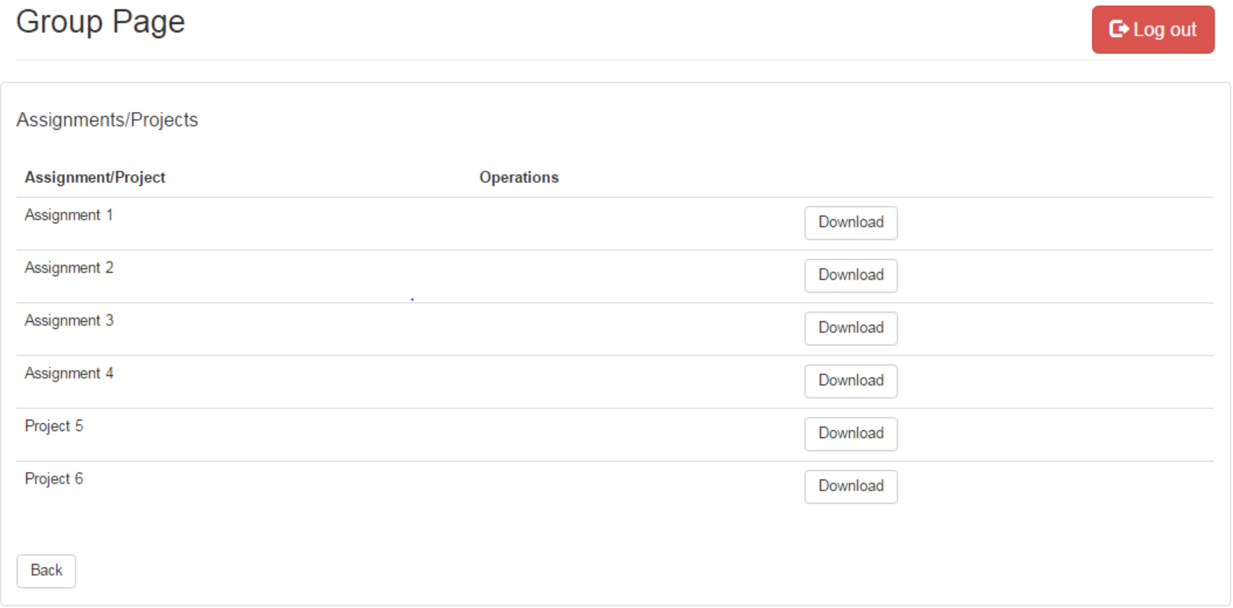
- Admins can delete existing users
- Accessible through deleteuser.php



Delete User

Are you sure you want to delete User?

- Admins can download current versions of each uploaded assignment by each group



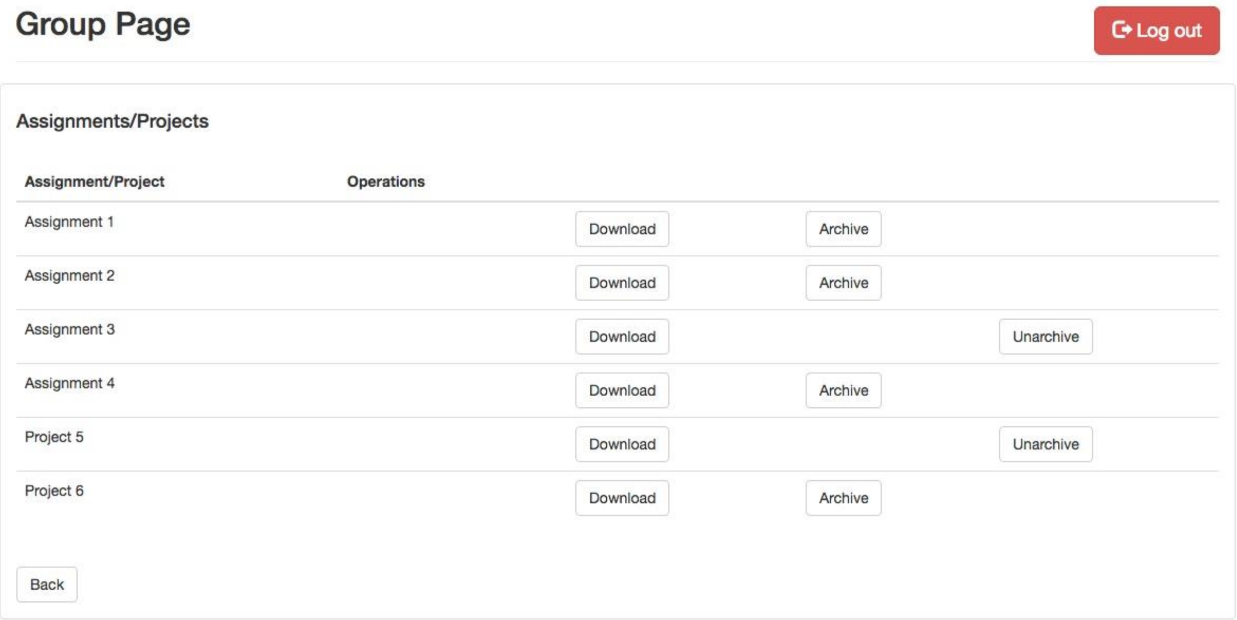
Group Page

Assignments/Projects

Assignment/Project	Operations
Assignment 1	Download
Assignment 2	Download
Assignment 3	Download
Assignment 4	Download
Project 5	Download
Project 6	Download

User Interface - Admin

- Once the course is over the admin can archive a file or unarchive a file that has already been archived
- Archiving a file will remove access to the file to the teacher and TAs (students have no access to anything before the course starts and after it ends)



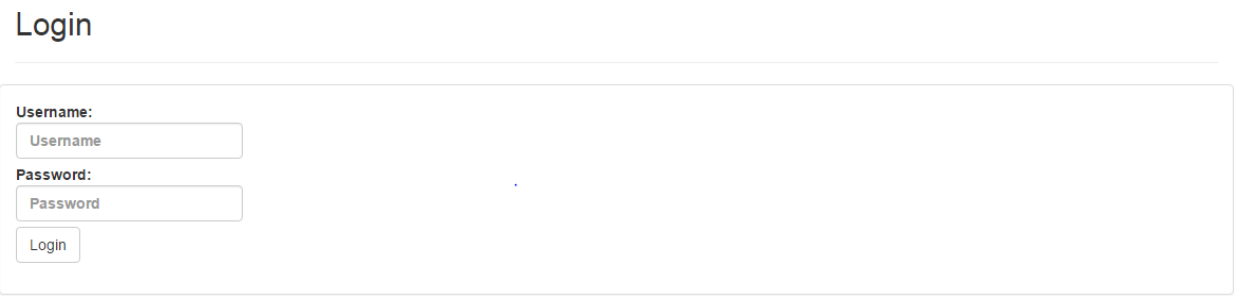
Group Page

Assignments/Projects

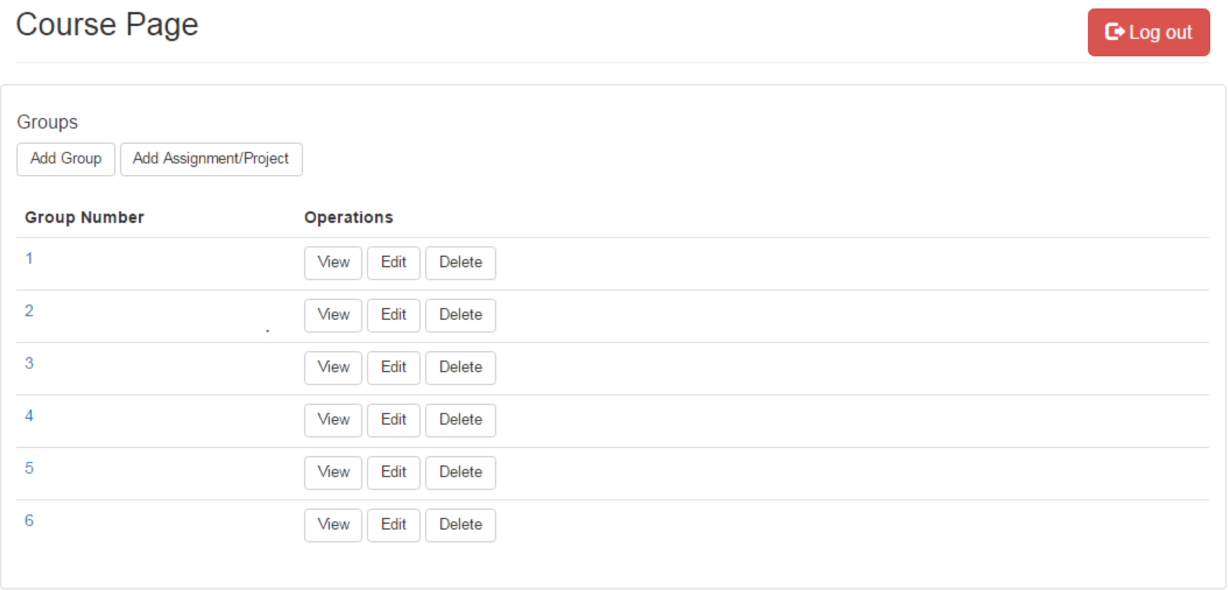
Assignment/Project	Operations		
Assignment 1	Download	Archive	
Assignment 2	Download	Archive	
Assignment 3	Download		Unarchive
Assignment 4	Download	Archive	
Project 5	Download		Unarchive
Project 6	Download	Archive	

User Interface - Teacher

- User login page: login.php
- User login as teacher

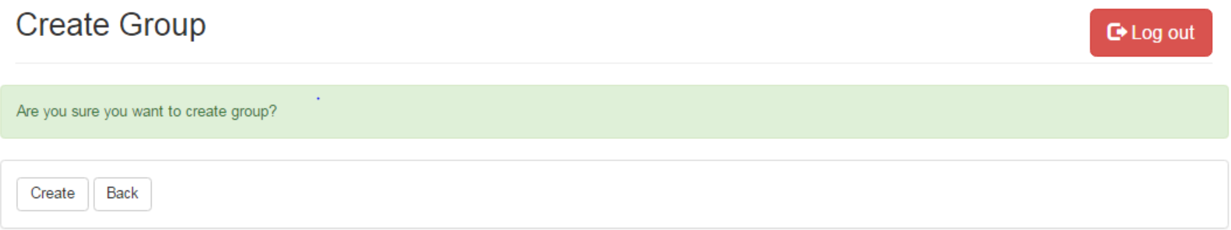


- Course webpage, logged in as a teacher
- Accessible through course.php
- Teachers can view submitted assignments and create new ones for each group with the Add Assignment/Project button and create new groups with the Add Group button
- Teachers can also view the interactions and actions committed for each group
- Teachers can also edit group members and delete group



Group Number	Operations
1	View Edit Delete
2	View Edit Delete
3	View Edit Delete
4	View Edit Delete
5	View Edit Delete
6	View Edit Delete

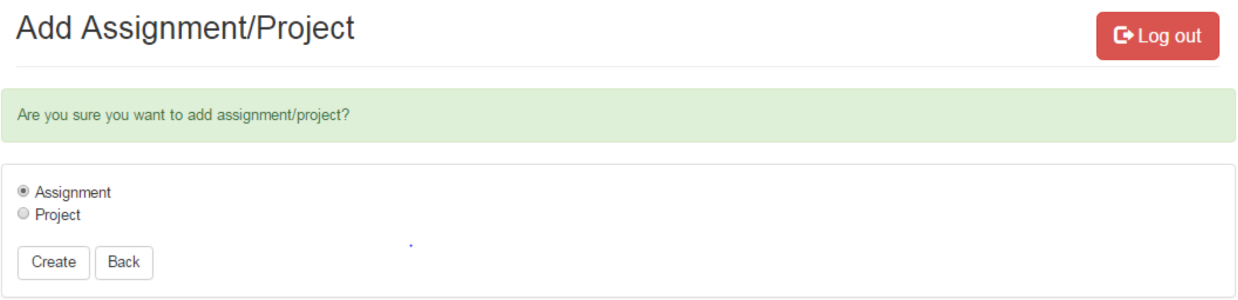
- Teachers can create new groups, Accessible through groupadd.php



Are you sure you want to create group?

User Interface - Teacher

- Teachers can add new assignments which will be added for each group, from assignmentadd.php

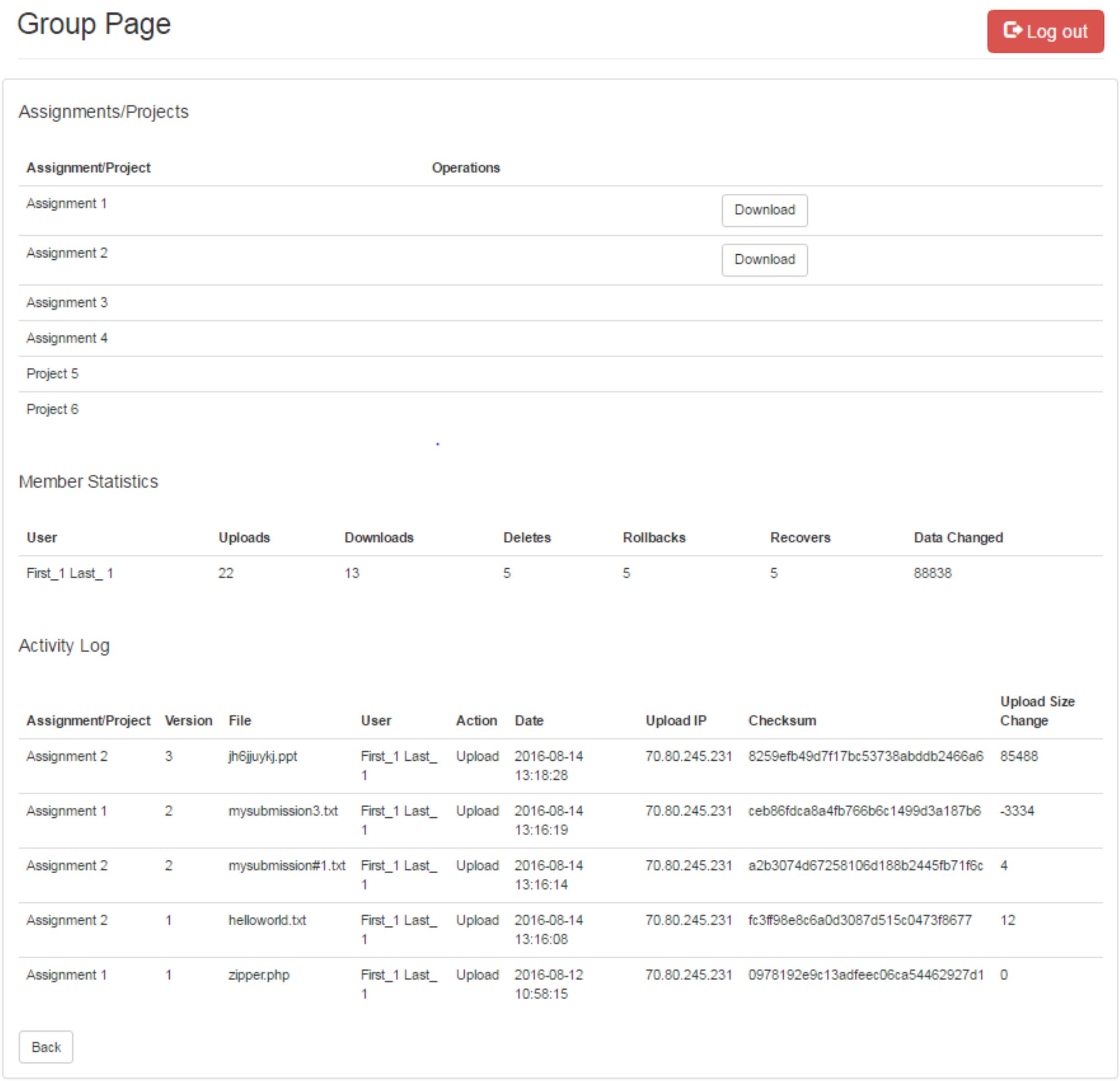


Add Assignment/Project

Are you sure you want to add assignment/project?

● Assignment
○ Project

- Teachers can view interactions made within a group by its group members and the current version of the submitted assignment, along with statistics of each uploaded file
- Accessible through group.php



Group Page

Assignments/Projects

Assignment/Project	Operations
Assignment 1	Download
Assignment 2	Download
Assignment 3	
Assignment 4	
Project 5	
Project 6	

Member Statistics

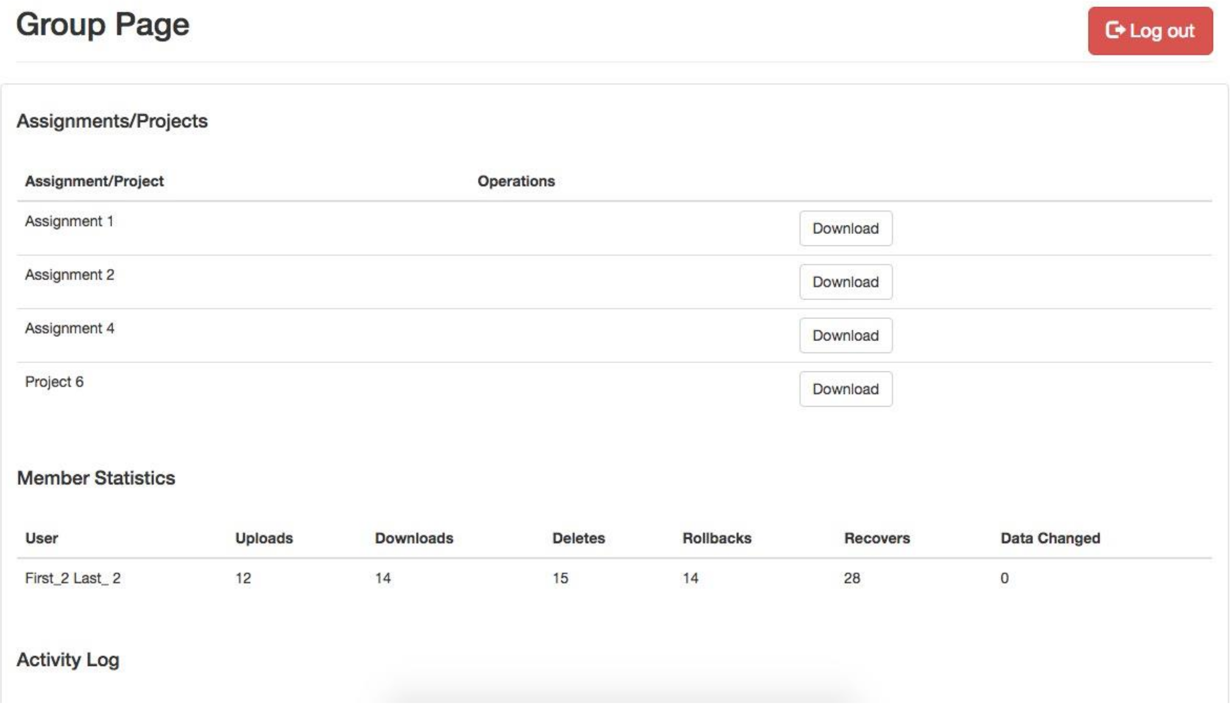
User	Uploads	Downloads	Deletes	Rollbacks	Recovers	Data Changed
First_1 Last_ 1	22	13	5	5	5	88838

Activity Log

Assignment/Project	Version	File	User	Action	Date	Upload IP	Checksum	Upload Size Change
Assignment 2	3	jh6jjuykj.ppt	First_1 Last_ 1	Upload	2016-08-14 13:18:28	70.80.245.231	8259efb49d7f17bc53738abddb2466a6	85488
Assignment 1	2	mysubmission3.txt	First_1 Last_ 1	Upload	2016-08-14 13:16:19	70.80.245.231	ceb86fdca8a4fb766b6c1499d3a187b6	-3334
Assignment 2	2	mysubmission#1.txt	First_1 Last_ 1	Upload	2016-08-14 13:16:14	70.80.245.231	a2b3074d67258106d188b2445fb71f6c	4
Assignment 2	1	helloworld.txt	First_1 Last_ 1	Upload	2016-08-14 13:16:08	70.80.245.231	fc3ff98e8c6a0d3087d515c0473f8677	12
Assignment 1	1	zipper.php	First_1 Last_ 1	Upload	2016-08-12 10:58:15	70.80.245.231	0978192e9c13adfeec06ca54462927d1	0

User Interface - Teacher

- Once the course is over the teacher is still able to access all groups and assignments that have not been archived (in this case Assignment 3 and Project 5 have been archived by the admin after the course has ended)



Group Page

Assignments/Projects

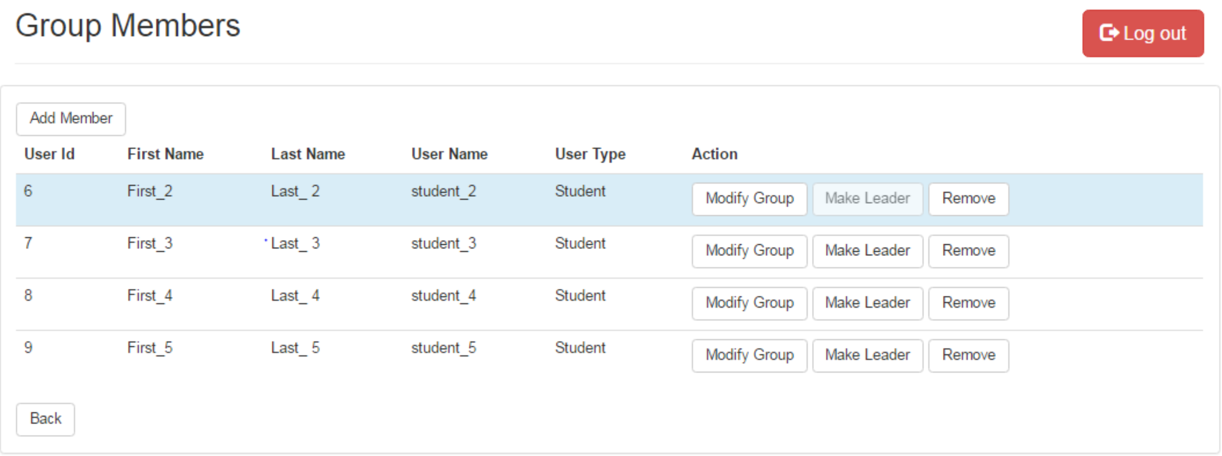
Assignment/Project	Operations
Assignment 1	Download
Assignment 2	Download
Assignment 4	Download
Project 6	Download

Member Statistics

User	Uploads	Downloads	Deletes	Rollbacks	Recovers	Data Changed
First_2 Last_ 2	12	14	15	14	28	0

Activity Log

- Teachers can view group members within a group, modify group leader, remove members, and add new members.
- Accessible through groupedit.php

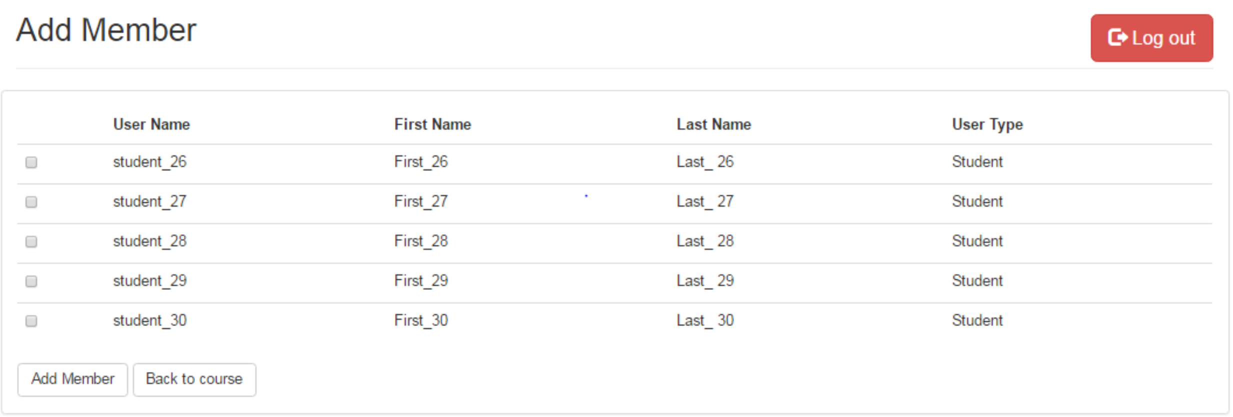


Group Members

User Id	First Name	Last Name	User Name	User Type	Action
6	First_2	Last_ 2	student_2	Student	Modify Group Make Leader Remove
7	First_3	Last_ 3	student_3	Student	Modify Group Make Leader Remove
8	First_4	Last_ 4	student_4	Student	Modify Group Make Leader Remove
9	First_5	Last_ 5	student_5	Student	Modify Group Make Leader Remove

User Interface - Teacher

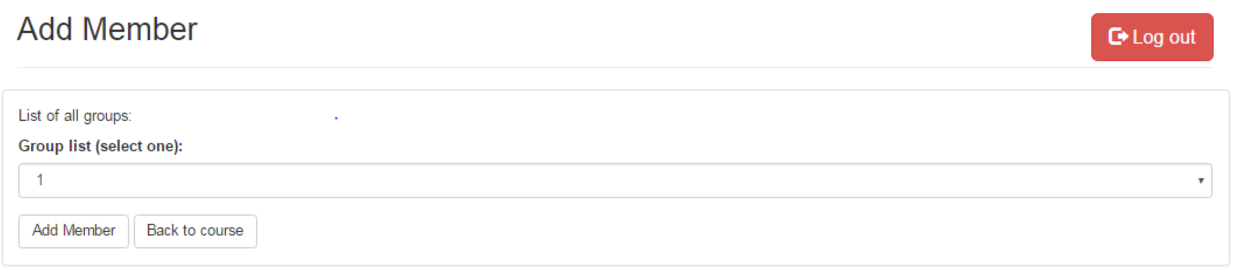
- Teachers can add students enrolled in the course who have not yet joined a group
- Accessible through adduser.php



Add Member

	User Name	First Name	Last Name	User Type
□	student_26	First_26	Last_ 26	Student
□	student_27	First_27	Last_ 27	Student
□	student_28	First_28	Last_ 28	Student
□	student_29	First_29	Last_ 29	Student
□	student_30	First_30	Last_ 30	Student

- Teachers can transfer students in group A to other groups
- Accessible through modifygroup.php

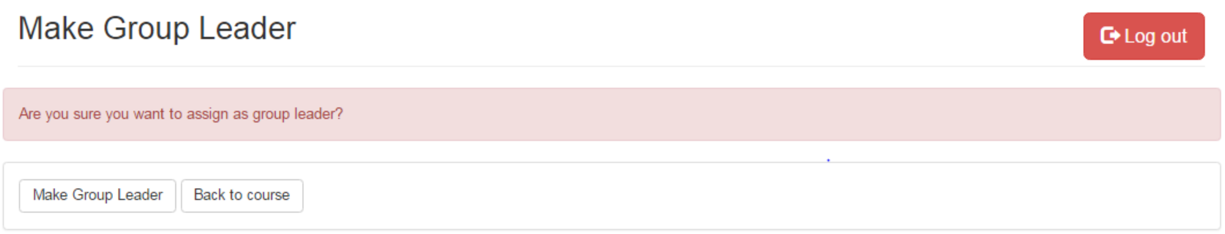


Add Member

List of all groups:

Group list (select one): 1

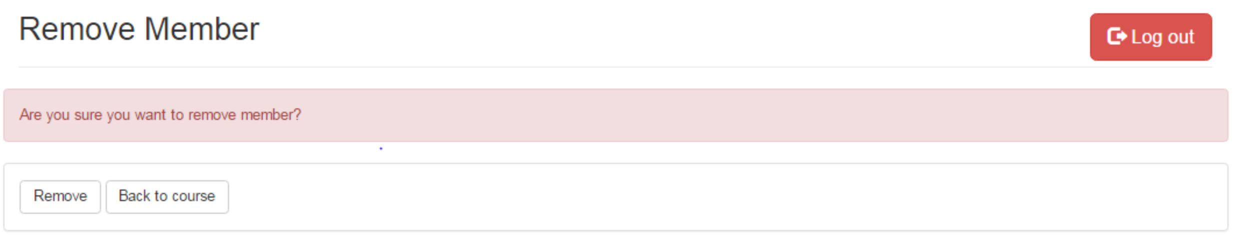
- Teachers can assign a new leader of a group or assign another non-leader member as the new leader
- Accessible through addleader.php



Make Group Leader

Are you sure you want to assign as group leader?

- Teachers can remove members from a group
- Accessible through memberdelete.php

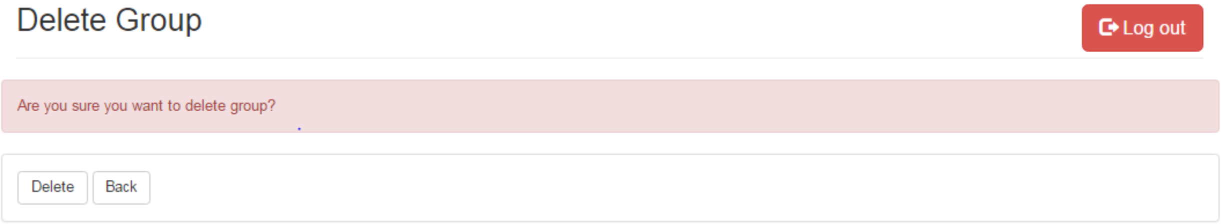


Remove Member

Are you sure you want to remove member?

User Interface - Teacher

- Teachers can delete any group from the course
- Accessible through groupdelete.php

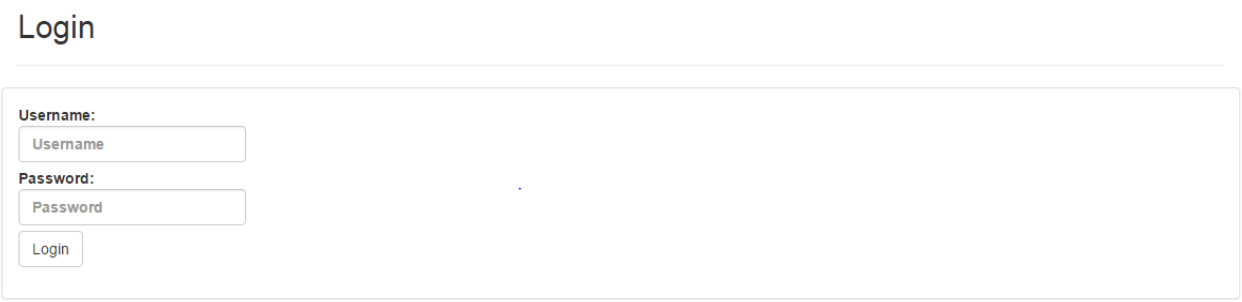


Delete Group

Are you sure you want to delete group?

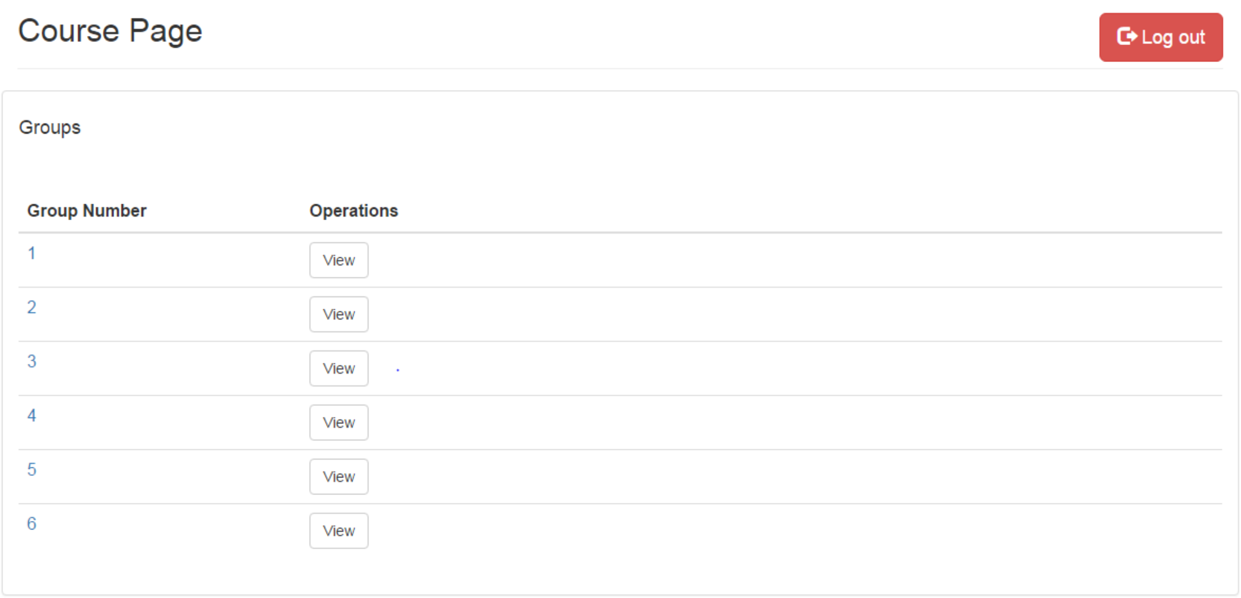
User Interface – Teaching Asisstant

- User login page, login.php
- User login as ta_1, who is a teaching assistant



Login

- TAs can only view the assignments uploaded by students in each group through the view button
- Accessible through course.php



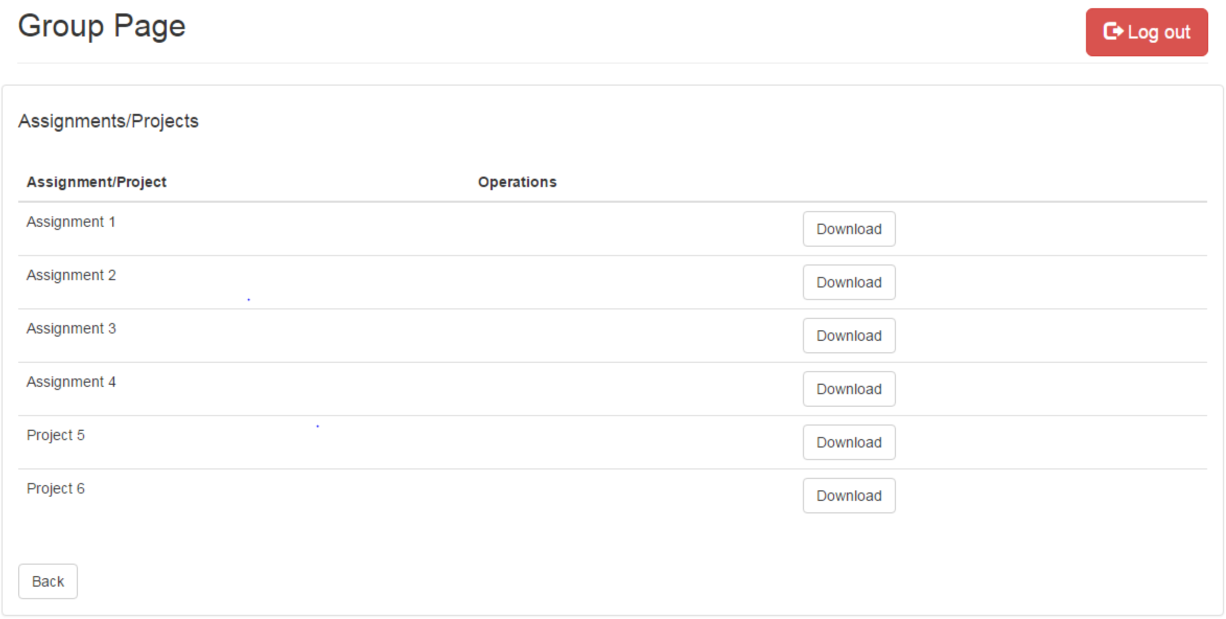
Course Page

Groups

Group Number	Operations
1	View
2	View
3	View
4	View
5	View
6	View

User Interface – Teaching Asisstant

- TAs can download assignments uploaded by each group by clicking download

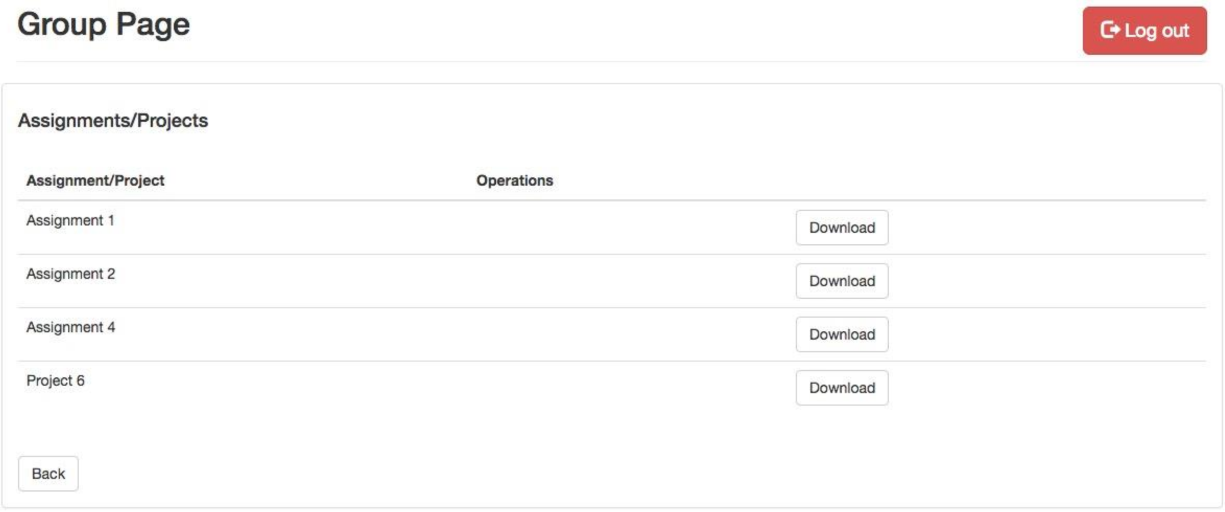


Group Page

Assignments/Projects

Assignment/Project	Operations
Assignment 1	Download
Assignment 2	Download
Assignment 3	Download
Assignment 4	Download
Project 5	Download
Project 6	Download

- Once the course has ended the TAs are still able to access all information unless of course the admin has archived an assignment or project. (in this case the system admin has archived Assignment 3 and Project 5 for this particular group so the TA is unable to access them)



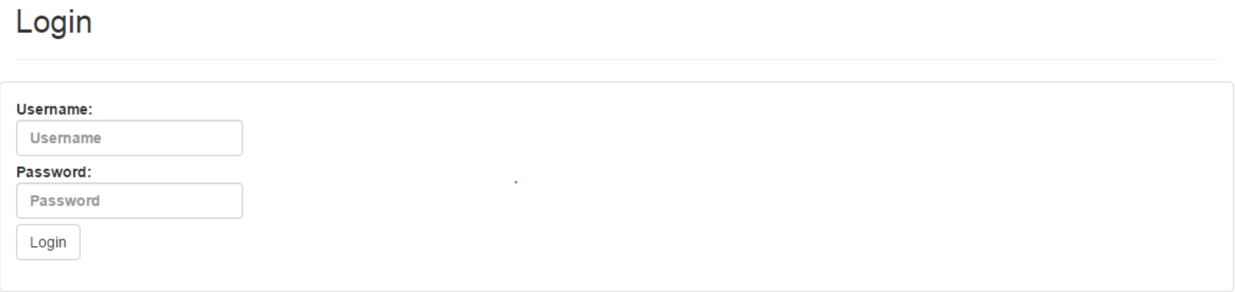
Group Page

Assignments/Projects

Assignment/Project	Operations
Assignment 1	Download
Assignment 2	Download
Assignment 4	Download
Project 6	Download

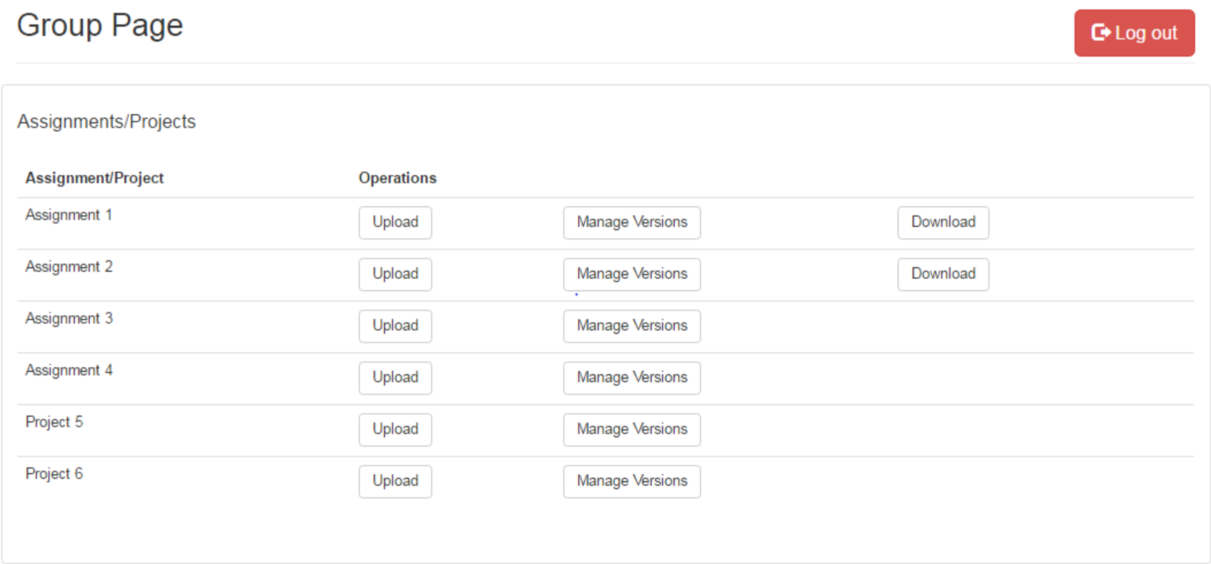
User Interface – Student (Group Leader)

- User login page: login.php
- User login as student_2, who is a leader of a group



Login

- Students can view upload their assignments, choose the version they want to submit, and download the corresponding version that they want to set as the current version
- Accessible through group.php

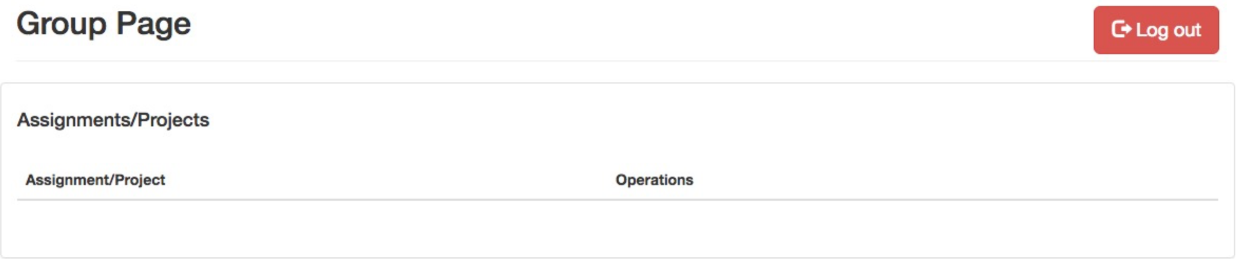


Group Page

Assignments/Projects

Assignment/Project	Operations		
Assignment 1	Upload	Manage Versions	Download
Assignment 2	Upload	Manage Versions	Download
Assignment 3	Upload	Manage Versions	
Assignment 4	Upload	Manage Versions	
Project 5	Upload	Manage Versions	
Project 6	Upload	Manage Versions	

- Once the course is over or before the course has begun the student isn't able to access any information within the database



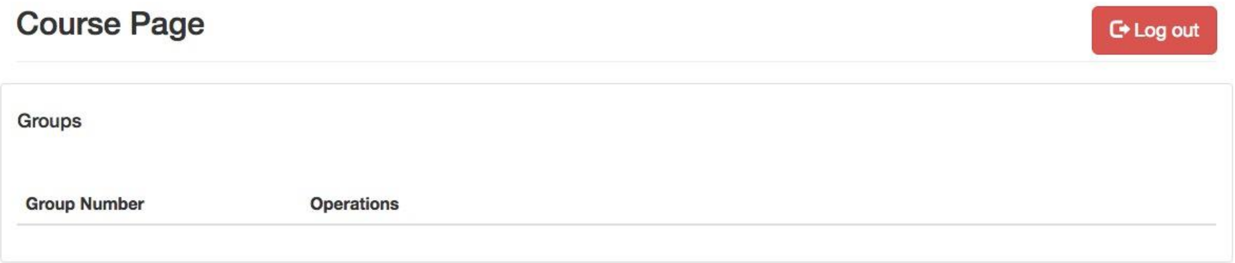
Group Page

Assignments/Projects

Assignment/Project	Operations

User Interface – Student (Group Leader)

- After the course has ended this page is only accessible if a student who has not joined a group tries to log in

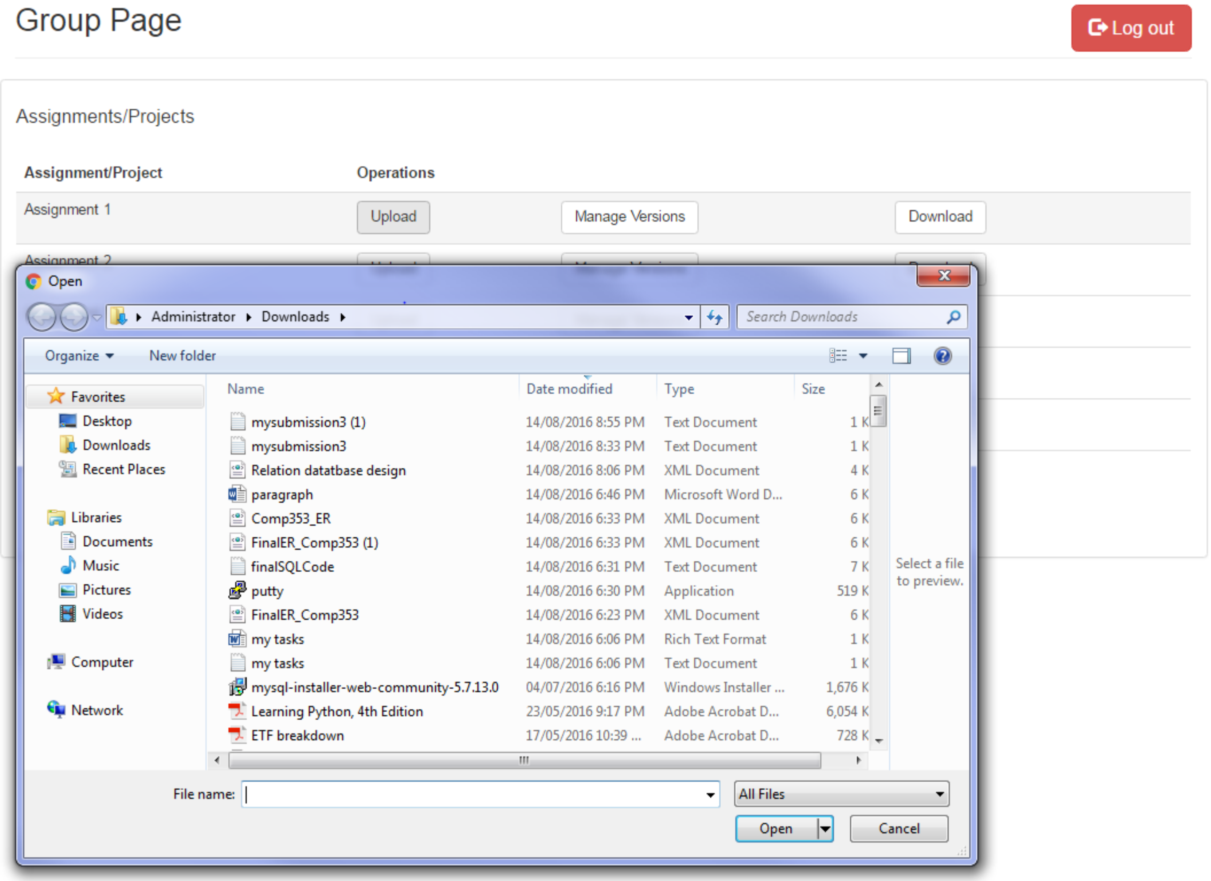


Course Page

Groups

Group Number	Operations

- Student can upload any file from any folder
- Accessible by clicking the Upload button on group.php



Group Page

Assignments/Projects

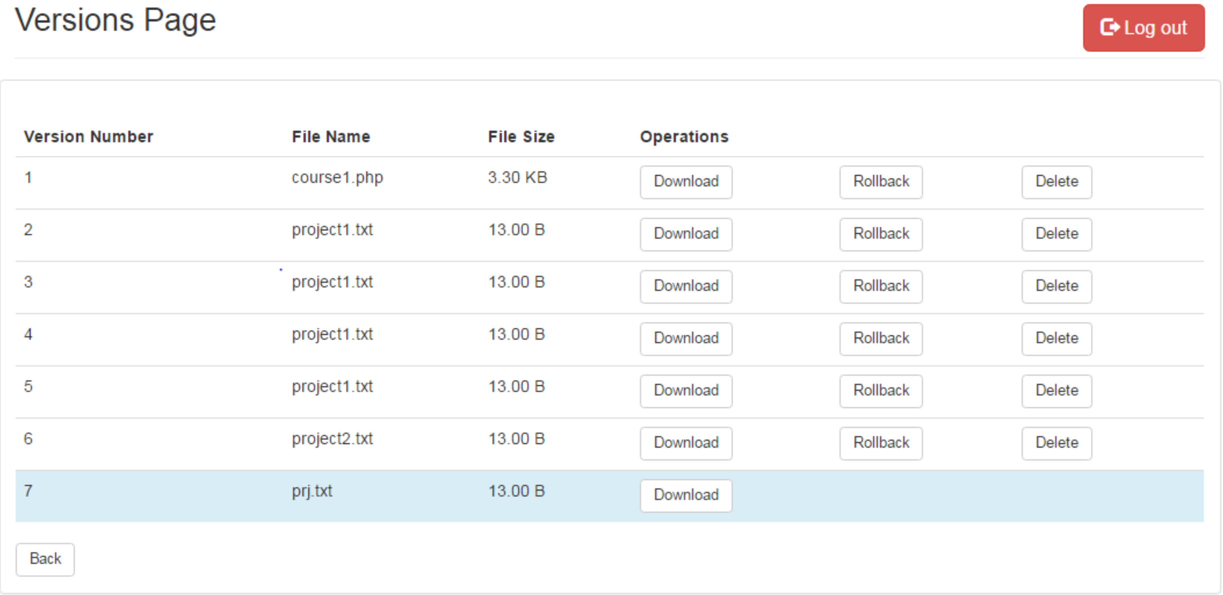
Assignment/Project	Operations		
Assignment 1	Upload	Manage Versions	Download
Assignment 2			

Name	Date modified	Type	Size
mysubmission3 (1)	14/08/2016 8:55 PM	Text Document	1 K
mysubmission3	14/08/2016 8:33 PM	Text Document	1 K
Relation datatbase design	14/08/2016 8:06 PM	XML Document	4 K
paragraph	14/08/2016 6:46 PM	Microsoft Word D...	6 K
Comp353_ER	14/08/2016 6:33 PM	XML Document	6 K
FinalER_Comp353 (1)	14/08/2016 6:33 PM	XML Document	6 K
finalSQLCode	14/08/2016 6:31 PM	Text Document	7 K
putty	14/08/2016 6:30 PM	Application	519 K
FinalER_Comp353	14/08/2016 6:23 PM	XML Document	6 K
my tasks	14/08/2016 6:06 PM	Rich Text Format	1 K
my tasks	14/08/2016 6:06 PM	Text Document	1 K
mysql-installer-web-community-5.7.13.0	04/07/2016 6:16 PM	Windows Installer ...	1,676 K
Learning Python, 4th Edition	23/05/2016 9:17 PM	Adobe Acrobat D...	6,054 K
ETF breakdown	17/05/2016 10:39 ...	Adobe Acrobat D...	728 K

File name:

User Interface – Student (Group Leader)

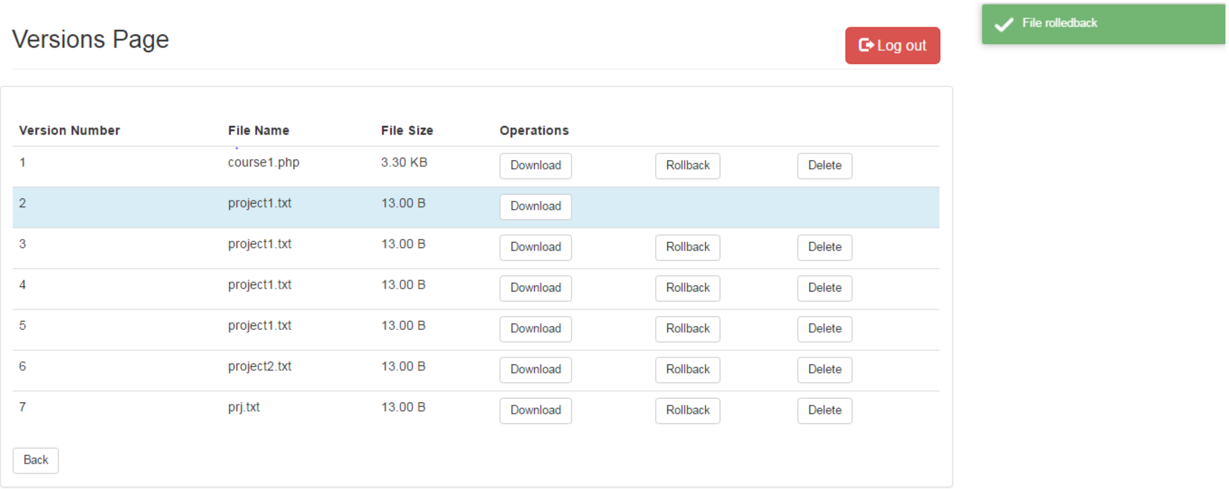
- Student can download any files that were uploaded into the platform for any assignments, rollback between different versions (only if group leader), and delete any file (recoverable within 24 hours)
- Accessible through manageversions.php



Versions Page

Version Number	File Name	File Size	Operations		
1	course1.php	3.30 KB	Download	Rollback	Delete
2	project1.txt	13.00 B	Download	Rollback	Delete
3	project1.txt	13.00 B	Download	Rollback	Delete
4	project1.txt	13.00 B	Download	Rollback	Delete
5	project1.txt	13.00 B	Download	Rollback	Delete
6	project2.txt	13.00 B	Download	Rollback	Delete
7	prj.txt	13.00 B	Download		

- Student (only the group leader) can change the current version of the file to any other version by clicking on rollback
- A deleted file cannot be rolled back to and made the current version
- A current version file cannot be set to be deleted
- Accessible through manageversions.php by clicking rollback



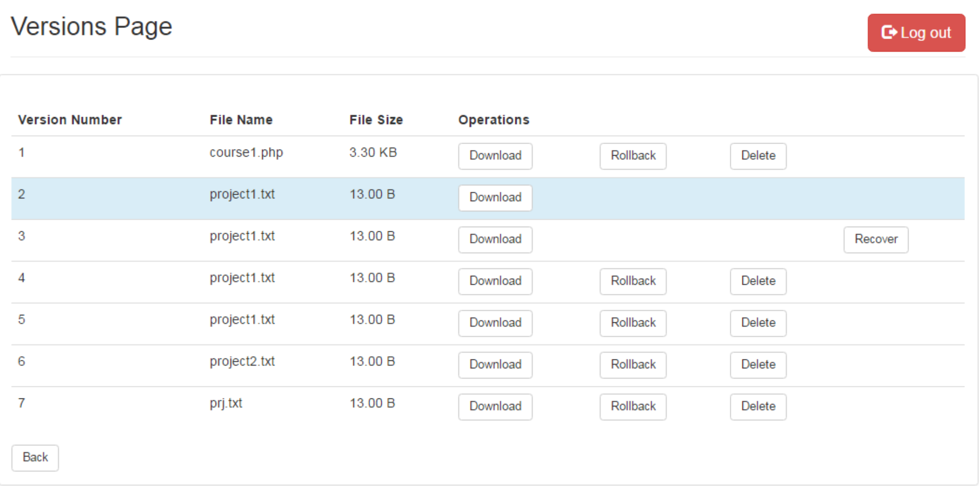
Versions Page

File rolledback

Version Number	File Name	File Size	Operations		
1	course1.php	3.30 KB	Download	Rollback	Delete
2	project1.txt	13.00 B	Download		
3	project1.txt	13.00 B	Download	Rollback	Delete
4	project1.txt	13.00 B	Download	Rollback	Delete
5	project1.txt	13.00 B	Download	Rollback	Delete
6	project2.txt	13.00 B	Download	Rollback	Delete
7	prj.txt	13.00 B	Download	Rollback	Delete

User Interface – Student (Group Leader)

- Student can delete any file versions by clicking delete
- Accessible through manageversions.php by clicking delete

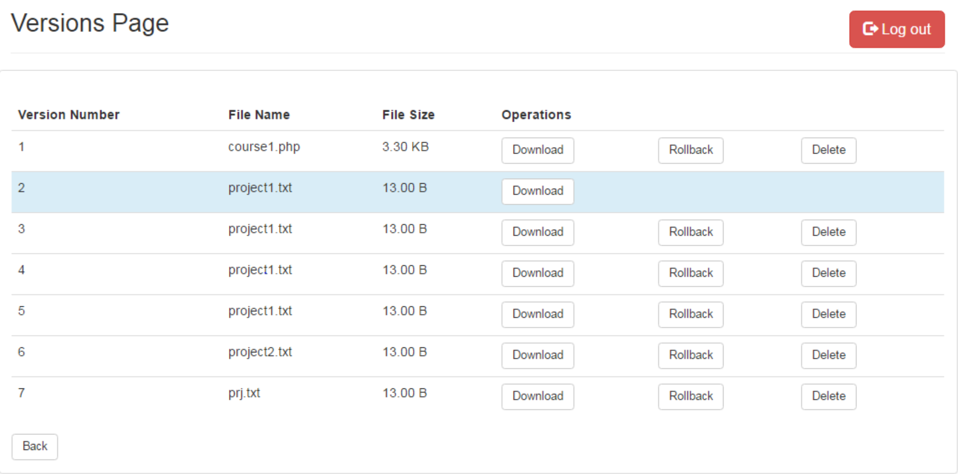


Versions Page

Version Number	File Name	File Size	Operations			
1	course1.php	3.30 KB	Download	Rollback	Delete	
2	project1.txt	13.00 B	Download			
3	project1.txt	13.00 B	Download			Recover
4	project1.txt	13.00 B	Download	Rollback	Delete	
5	project1.txt	13.00 B	Download	Rollback	Delete	
6	project2.txt	13.00 B	Download	Rollback	Delete	
7	prj.txt	13.00 B	Download	Rollback	Delete	

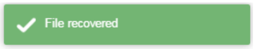
✓ File deleted

- Student can recover any deleted file within 24 hours by clicking recover, which will show up whenever a student deletes a file
- Accessible through manageversions.php by clicking recover



Versions Page

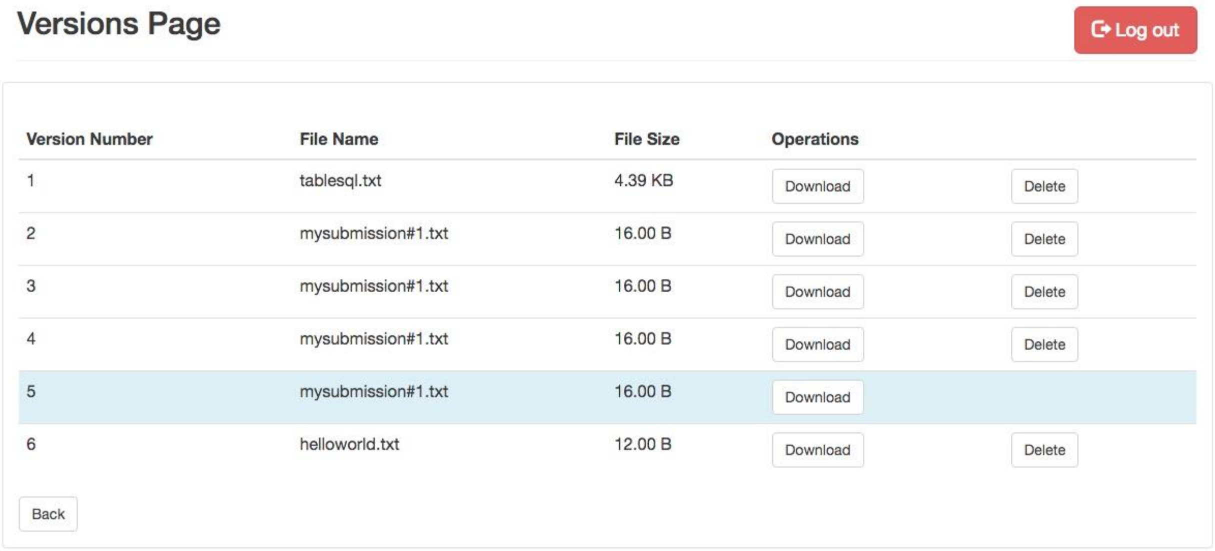
Version Number	File Name	File Size	Operations		
1	course1.php	3.30 KB	Download	Rollback	Delete
2	project1.txt	13.00 B	Download		
3	project1.txt	13.00 B	Download	Rollback	Delete
4	project1.txt	13.00 B	Download	Rollback	Delete
5	project1.txt	13.00 B	Download	Rollback	Delete
6	project2.txt	13.00 B	Download	Rollback	Delete
7	prj.txt	13.00 B	Download	Rollback	Delete



✓ File recovered

User Interface – Student (Non-Group Leader)

- When logged in as a student that is a group member but not a group leader. On the Versions page, the student is not presented with the option to rollback.
- Any group member can still download any version, set any version to be deleted within 24 hours or recover a file set to be deleted within 24 hours.



Version Number	File Name	File Size	Operations	
1	tablesql.txt	4.39 KB	Download	Delete
2	mysubmission#1.txt	16.00 B	Download	Delete
3	mysubmission#1.txt	16.00 B	Download	Delete
4	mysubmission#1.txt	16.00 B	Download	Delete
5	mysubmission#1.txt	16.00 B	Download	
6	helloworld.txt	12.00 B	Download	Delete

Contributions

Demostenis Kioussis:

With Tian, they redesigned the initial E-R diagram using Microsoft Visio, and created the SQL queries used for the warmup project. After receiving feedback on the warmup project, Demostenis helped redesign the E-R diagram with the rest of the team, and modified the tables to remove a recurring error. Using others code as a template, he made systemdetails.php and systemupdate.php. These pages allowed the admin to and/or edit the class start date, end date as well as change the data limit imposed on groups. While working on these pages he also added a constaint to the System_details table, forcing it to contain at most 1 row. He created a script with java to create all the users within our database. Along with Tian, he was responsible for the final report. He was also the group leader.

Tian Nan Liu:

He has started off by working on the E-R diagram that was initially designed by Jordan and Shayne with Demostenis. Both of them worked on redesigning the E-R diagram on Microsoft Visio and completed the warm-up project requirements including the design of the SQL codes, database assumptions. After the warm-up project submission, Tian Nan contributed to the redesign of the E-R diagram based on the feedback given by the T.A along with Demostenis, Jordan, and Shayne. With the help of Shayne, Jordan and based on some of Hussain's codes, he has worked on the usermanagement.php, createuser.php, edituser.php, deleteuser.php pages which allows the admin to create new users and store those information into our database. He ensured that no duplicate users with the same username were allowed. He has also implemented the user editing and delete functions and ensured that no duplicate usernames were allowed after an admin edits an existing user's information. With the help of Shayne, he has managed to work on assignmentadd.php which adds assignments into the system and creates a new assignment and store it into our database. This is a functionality restricted to teachers only. He has also contributed to the manageversion.php page's layout, but the back-end work were done by Shayne. He is also responsible for the final report with Demostenis, which includes updating our E-R diagram and our database design, database assumptions with rational reasons, design decisions, and other guidelines required for the final report.

Hussain Qabbani:

He designed the course webpage by using Bootstrap CSS and JS and he has made great contributions for the group.php file, which links to features such as creating users for the group, creating groups, editing groups. More specifically, the edit group php file allows teachers to assign a leader, delete group, modify group members, and remove members from a group which also deletes members from the database. Hussain added a checkbox to add students in a group and allows the teacher to edit the student's information. He has also worked on the logout feature of our project. Moreover, he helped in populating the database with the rest of the group. As well as he Implemented some functions in archive feature that's zipper.php.

Contributions

Shayne Mclevy:

For the overall system, Shayne ensured that user permissions were on every page (so a student could not access the admin pages, etc.), protected against SQL injection, and ensured the session was active throughout every page in the system. In the database he created the SQL trigger for file deletion. He implemented the index.php page which redirects a user to the correct landing page depending on if they are logged in or their user type. On login.php he created the redirect and updated the User's last login time to trigger the file deletion in the database. On course.php he implemented students being able to join groups if they were not part of one. In the group.php page I implemented the UI, the ability for files to be uploaded and downloaded (includes download.php), as well as the member statistics and activity log tables for the teacher view. In upload, I validated invalid file names and that the size of the file would not exceed the groups maximum file data allowed. For the manageversions.php page, I implemented the download, delete, rollback, and recover actions. For both of these pages I added toasts to notify the user when an action has been completed. Also, I created the initial versions of the ER diagram, and drew many mockups for the web pages.

Db and page modifications for File Size Change feature:
Added sizechange column to the File_versions table.
Added total data added to Member Statistics on group.php
Added upload size change to Activity Log on group.php

Jordan Orsini:

His contributions towards the database project included helping to create the database ER diagram with the rest of the group. He converted the ER diagram into a relational database model in order to facilitate creating the tables in MySQL. He converted the relational database model into table SQL and created these tables in MySQL. He helped to ensure visual continuity throughout our entire application by applying Hussain's created Bootstrap CSS classes to the entirety of our application. Finally, he implemented the first attempt at the archiving feature. This feature would only be accessible by the system admin after the end of the course and would allow the admin to archive assignment or project files from individual groups. Relevant code for the archiving feature can be found in group.php. He implemented checks that ensured that before the course began or after the course finished students would have no access to any information in the database. The checks can be found in group.php as well as course.php to ensure that even students that were not in a group before the course ended would have no access. He also implemented checks in group.php that would ensure that at the end of the course the teacher and TAs were still able to access all files unless the assignment or project was archived by the admin in which case only the admin would have the capability of viewing archived files or un-archiving a file. Furthermore, he helped by filling our database with tuples along with the rest of the group. In manageversion.php he implemented a restriction that ensures that a file set as the current version cannot be set to be deleted, furthermore the restriction also ensures that a file already set to be deleted in 24 hours cannot be rolled back to and be made the current version. Made contributions to the report by providing specs and screenshots and presented the group's application when it came time to demo.