DBMS Project Master Document

Technologies used

Server-end scripting language: GO
GO(golang) is an open source programming language created by Google in 2007. Being statically typed it is quite scalable to large systems. It supports networking and multiprocessing admirably well. Deliberate effort was made to make the syntax and style of the language similar to dynamically typed languages like Python. In recent years it has gained tremendous popularity and remains one of the most used back-end languages.

Additionally, it integrates seamlessly with MySQL drivers; thus emerging as our obvious choice.

HTML

Since a browser interface is infinitely more user-friendly than a command-line one.

JSON

To transmit data objects consisting of attribute-value pairs.

AJAX

To send and receive data from the server asynchronously without interfering with the display and behaviour of current page.

JavaScript

For processing within the HTML document.

CSS:

For describing the presentation of the document(layout of the webpage)

Tables

For each table, write:

1. List of attributes
2. Their types
3. Whether that attribute is the primary key, foreign key, auto increment, not null, etc.

eg. name varchar(32) not null
If you have any suggestions please suggest the edit by changing ‘editing’ in the top right to ‘suggesting’.

Accounts

Stores details of all users

- AccountID int primary key auto increment
- FirstName varchar(32) not null
- LastName varchar(32) not null
- Email varchar(64) not null
- RollNo varchar(32)
- PrivilegeLevel int not null < 0 - student, 1 - teacher, 2 - dean
- ActivationKey char(64) < for account confirmation through email, null indicates already confirmed

Courses

Stores details of courses

- CourseID int primary key auto increment
- CourseName varchar(32) not null
- CourseCode varchar(16) not null
- CourseCreatorID int not null foreign key references AccountID
- CourseStartDate date not null
- CourseEndDate date not null
- MaxGraceDays int not null

Roles

Stores the role of an account for a course. An account may have different roles for different courses.

- AccountID int not null foreign key references AccountID
- CourseID int not null foreign key references CourseID
- Role int not null < 0 - student, 1 - ta, 2 - teacher

Assignments and Announcements

Stores details of assignments and announcements. Combined since they have similar structure, and an announcement is basically an assignment without any submissions required.

- AssignmentID int primary key auto increment
- CourseID int not null foreign key references CourseID
- CreationTime datetime not null
- CreatorID int not null foreign key references AccountID
- DueTime datetime < NULL in case of announcement
- MaxSubmissionTime datetime < NULL in case of announcement or no restriction
- TitleString text not null
- SubmissionType varchar(4) not null < ‘file’ or ‘text’ or ‘none’ in case of announcement
- MaximumMarks int not null = 0 in case of announcement

Grades

Stores grades of students for each assignment. Also stores no. of grace days used.

- SubmittorID int not null foreign key references AccountID
- AssignmentID int not null foreign key references AssignmentID
- GivenGrade int < NULL indicates not graded
- GraceDaysUsed int not null

Files

Stores details of all files, including submissions as well as question files

- FileID int primary key auto increment
- Hash char(64) not null < for quick storage and retrieval of file from filesystem
- FileName varchar(64) not null

Submission Files

Stores details of student submissions for assignments

- FileID int not null foreign key references FileID
- SubmittorID int not null foreign key references AccountsID
- AssignmentID int not null foreign key references AssignmentID
- SubmissionTime datetime not null

Question Files

Stores details of files from the instructor that are part of an assignment specification

- FileID int not null foreign key references FileID
- AssignmentID int not null foreign key references AssignmentID

Comments

Stores user comments

- CommentString text not null
- CommentorID not null foreign key references AccountID
- AssignmentID not null foreign key references AssignmentID
- CommentTime datetime not null

Submission Strings

Stores answer strings

- SubmissionString text not null
- SubmittorID int not null foreign key references AccountID
- AssignmentID int not null foreign key references AssignmentID
- Submission Time datetime not null

Grace Days

Stores no. of grace days remaining for each student for each course.

- CourseID int foreign key not null references CourseID
- StudentID int not null foreign key references AccountID
- GraceDaysRemaining int not null

Supported Operations

For each operation, write operation effect and SQL statement.

Account Operations

Common

- Creation of Account
- Email verification of Account
- Deletion of Account

Dean Only

- Elevating student accounts to dean or instructor level

Instructor Only

- Creation of Course
- Deletion of Course

Course Operations

Common

- Commenting on an Assignment or Announcement

Instructor Only

- Adding other Instructors, TAs, and Students
- Removing other Instructors, TAs, and Students
- Seeing student grades (weighted average of marks in all assignments, out of 100)

Instructor or TA

- Adding Announcements or Assignments
- Modifying Announcements or Assignments
- Removing Announcements or Assignments
- Accessing Student Submissions
- Grading Submissions
- Modifying Grades

Student Only

- Adding a submission
- Modifying a submission (upto last submit date, if not graded)
- Removing a submission (upto last submit date, if not graded)
- Adding Announcements

SQL Statements

Table Creation

- create table account
  (accountid int primary key auto increment,
  firstname varchar(64) not null,
  lastname varchar(64) not null,
  email varchar(64) not null,
  rollno varchar(32),
  privilegelevel int not null,
  activationkey char(64));
- create table course
  (courseid int primary key auto increment,
  coursename varchar(64) not null,
  coursecode varchar(16) not null,
  creatorid int not null foreign key references account(accountid),

startdate date not null,
enddate date not null,
maxgracedays int not null);

- create table role
  (accountid int not null foreign key references account(accountid),
  courseid int not null foreign key references course(courseid),
  role int not null);
- create table assignment
  (assignmentid int primary key auto increment,
  courseid int not null foreign key references course(courseid),
  creatorid int not null foreign key references account(accountid),
  creationtime datetime not null,
  duetime datetime,
  maxsubmittime datetime,
  titlestring text not null,
  submissiontype varchar(4) not null,
  maxmarks int not null);
- create table grade
  (assignmentid int not null foreign key references assignment(assignmentid),
  submittorid int not null foreign key references account(accountid),
  givenmarks int,
  gracedaysused int not null);
- create table file
  (fileid int primary key auto increment,
  hash char(64) not null,
  filename varchar(64) not null);
- create table subfile
  (fileid int not null foreign key references file(fileid),
  submittorid int not null foreign key references account(accountid),
  assignmentid int not null foreign key references assignment(assignmentid),
  submittime datetime not null);
- create table qfile
  (fileid int not null foreign key references file(fileid),
  assignmentid int not null foreign key references assignment(assignmentid));
- create table comment
  (commentstring text not null,
  commentorid int not null foreign key references account(accountid),
  assignmentid int not null foreign key references assignment(assignmentid),
  commenttime datetime not null);
- create table substring
  (answerstring text not null,
  submittorid int not null foreign key references account(accountid),
  assignmentid int not null foreign key references assignment(assignmentid),
  submittime datetime not null);
- create table gracedays
  (courseid int not null foreign key references course(courseid),

```
studentid int not null foreign key references account(accountid),
availabledays int not null);
```

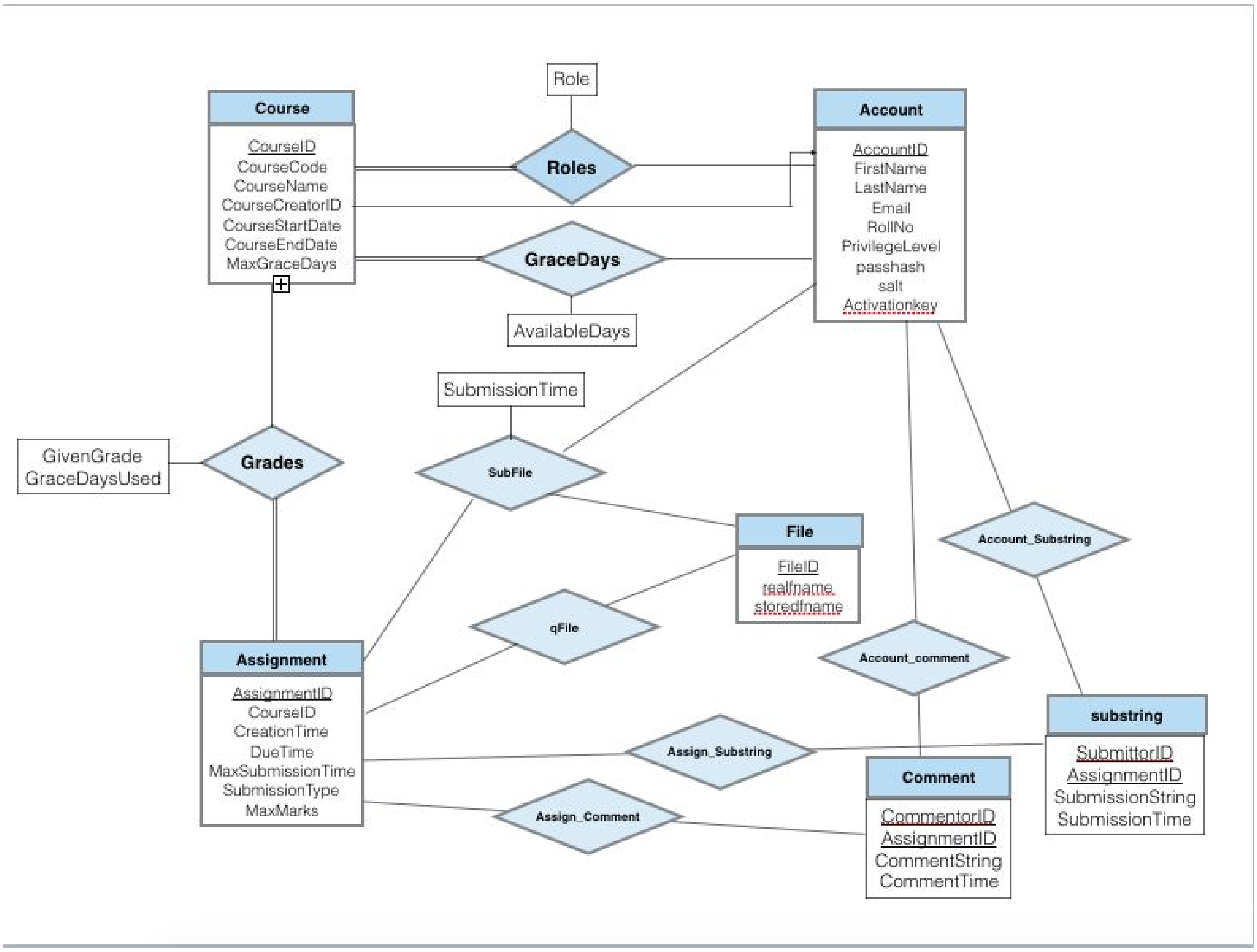
Account Operations

- Creation of Account
- Email verification of Account
- Deletion of Account
- Elevating student accounts to dean or instructor level
- Creation of Course
- Deletion of Course

Course Operations

- Commenting on an Assignment or Announcement
- Editing previous comments
- Deleting comments
- Adding other Instructors, TAs, and Students
- Removing other Instructors, TAs, and Students
- Seeing student grades (weighted average of marks in all assignments, out of 100)
- Adding Announcements or Assignments
- Modifying Announcements or Assignments
- Removing Announcements or Assignments
- Accessing Student Submissions
- Grading Submissions
- Modifying Grades
- Adding a submission
- Modifying a submission (upto last submit date, if not graded)
- Removing a submission (upto last submit date, if not graded)

ER-Diagram


Course
CourseID
CourseCode
CourseName
CourseCreatorID
CourseStartDate
CourseEndDate
MaxGraceDays
Role
Roles
Account
AccountID
FirstName
LastName
Email
RollNo
PrivilegeLevel
passhash
salt
Activationkey
GraceDays
AvailableDays
SubmissionTime
GivenGrade
GraceDaysUsed
Grades
SubFile
File
FileID
realfname
storedfname
Account_Substring
qFile
Account_comment
Assignment
AssignmentID
CourseID
CreationTime
DueTime
MaxSubmissionTime
SubmissionType
MaxMarks
Assign_Substring
substring
SubmittorID
AssignmentID
SubmissionString
SubmissionTime
Assign_Comment
Comment
CommentorID
AssignmentID
CommentString
CommentTime

Web Pages

PRIVILEGES:

- D - Dean
- I - Instructor
- T - TA
- S - Student

Web Page	URL	Contributor	Done
Login (All)	/login/	Keyur	Yes
Registration (All)	/register/	Keyur	Yes
File server (All)	/download/<FileID>	Keyur	Yes
Profile (All)	/profile/	Akshita	Yes
Account promotion (D)	/deanhome/	Sreekar	Yes
Course Creation (I)	/newcourse/	Arjun	Yes
Student Page (ITS)	/students/	Akshita	Yes
TA/Instr Appointing (I)	/chgroles/<CNum>	Akshita	Yes
Course Selection (ITS)	/home/	Akshita	Yes
Course Main Page (ITS)	/course/<CNum>	Akshita	Yes
Assignment creation (IT)	/assnCreate/<CNum>	Sreekar	Yes
Assignment page (ITS)	/assn/<ANum>	Arjun	Yes
Assignment grading (IT)	/grade/<ANum>	Keyur	Yes
Assignment grading (IT)	/grade/<ANum>/<AccNum>	Keyur	Yes