RIT | Golisano College of Computing and Information Sciences
School of Information

ISTE-330 Database Connectivity and Access
Group Project

1. Your task is to create the Data Layer for the system described below. You may write it in Java, PHP, or C#. EVERYONE on the team must be fully capable of writing code with the chosen language. You will each be responsible for being able to explain and work with the code you submit.
2. You may choose to create a front end in order to test your code, but there is no need to hand it in. I will create my own front end and run your code against that. If it works, great. If not, there will be substantial point deductions.
3. You will be graded on:
    a. Functionality—does your code conform to the API
    b. Design—is your code designed for maintenance and robustness
    c. Readability—is your code commented, are variable and method names appropriate, etc.?
    d. Documentation—JavaDocs or equivalent, generated and available
4. The system you will work on is meant to accept submissions to an academic conference. The Papers table is the heart of the system. Each Paper may have multiple subjects and multiple authors. Each paper also has a submitting author and a Type (e.g., paper, poster, panel); each author (or user) has an affiliation (place of employment).
5. The Data Layer should include the usual CRUD methods as well as other special methods that are described on the next page. An ER Diagram is also included with this document and a MySQL dump of the database is in MyCourses. (Note: This is an actual system that Prof. Bogaard and I wrote a few years ago.)
6. Additional credit will be given for the inclusion of additional, useful methods, but only if all the required functions are working and are well-written.
7. The project is due 4/19 at midnight. Peer evals will also be collected and will impact your individual grade for the project.

Data Layer API

CRUD operations for all entities. Include both row-abstraction and table-abstraction DAOs for all entities. Also include appropriate accessors and mutators for all attributes in each class. In addition:

Paper Object

getPaper(int paperId) returns all info (except filename) for specified paper

```
setPaper(int paperId,
        String submissionTitle,
        String submissionAbstract,
        int submissionType,
        String filename,
        String[] subjects,
        String[] coauthors firstnames
        String[] coauthors lastnames
         ) if no paperId (or paperId is zero), creates new entry, OTW updates all specified info.
```

User Object

getPapers(int userId) returns all papers for specified user

getUser() return info, except password, for instantiated user

```
setUser(String lastName
        String firstName,
        String email,
        String pswd,
        String affiliation
        ) if no userId, creates new entry. OTW, updates existing info
```

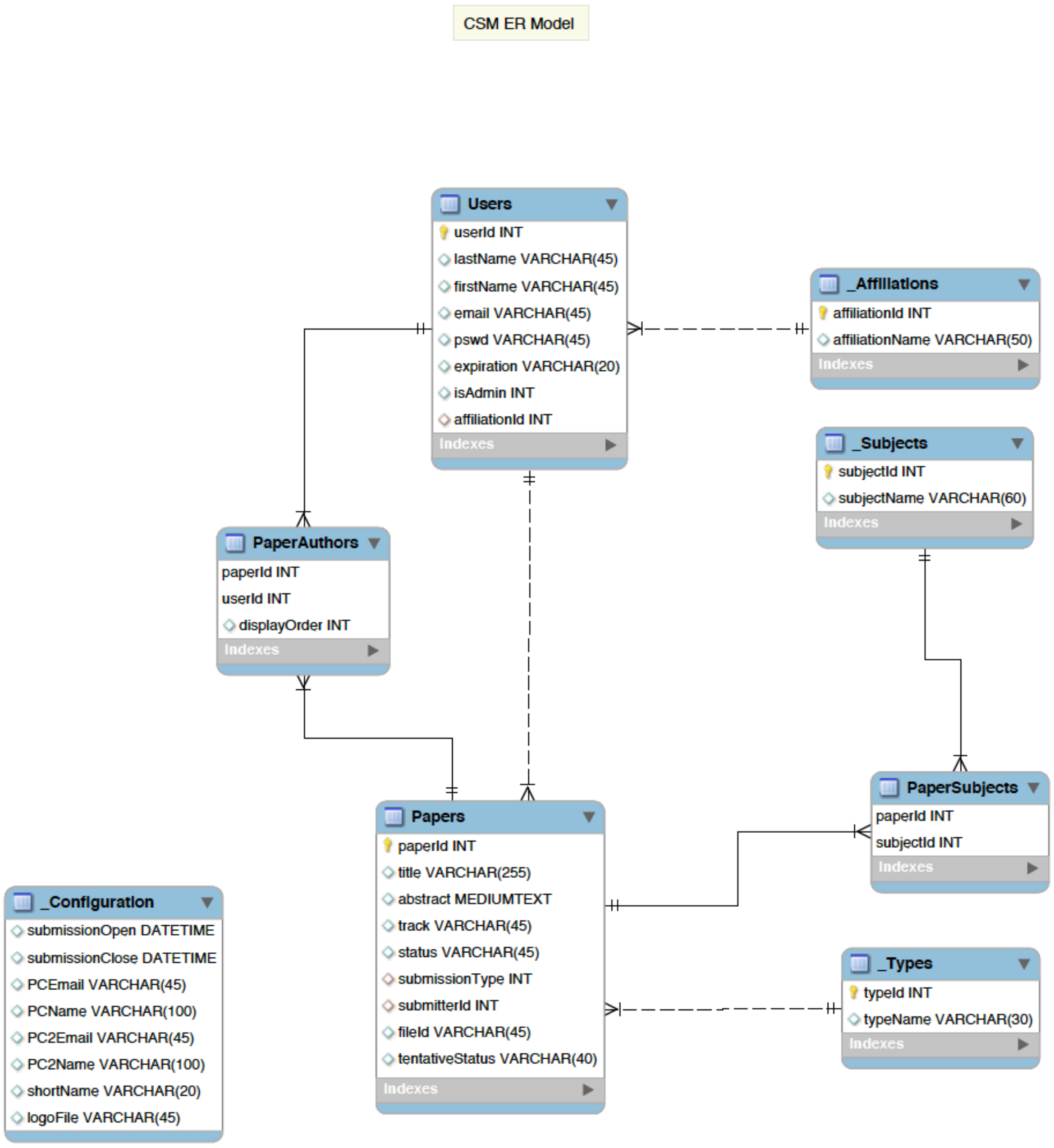
resetPassword(String email) creates new password and emails to specified address. Good for 5 minutes. The Users table has a field labeled "expiration". This field is used to indicate how long the password is valid. It should be checked when a user logs in. When a user sets their password, this date is set far off into the future. However, when a user has forgotten their password and is emailed a temporary one, this field is set to 5 minutes in the future. This allows the user 5 minutes to log in with the temporary password and set it to a permanent one.

setPassword(String password) changes password of currently logged in user to specified value.

login(String email, String password) If good credentials, returns token

Authorization

Someone who is not an admin can only create, read, and update their own papers and associated information, or their own profile information. Only the admin can delete entries or have full CRUD rights to all tables.


CSM ER Model
Users
userId INT
lastName VARCHAR(45)
firstName VARCHAR(45)
email VARCHAR(45)
pswd VARCHAR(45)
expiration VARCHAR(20)
isAdmin INT
affiliationId INT
Indexes
_Affiliations
affiliationId INT
affiliationName VARCHAR(50)
Indexes
_Subjects
subjectId INT
subjectName VARCHAR(60)
Indexes
PaperAuthors
paperId INT
userId INT
displayOrder INT
Indexes
PaperSubjects
paperId INT
subjectId INT
Indexes
Papers
paperId INT
title VARCHAR(255)
abstract MEDIUMTEXT
track VARCHAR(45)
status VARCHAR(45)
submissionType INT
submitterId INT
fileId VARCHAR(45)
tentativeStatus VARCHAR(40)
Indexes
_Configuration
submissionOpen DATETIME
submissionClose DATETIME
PCEmail VARCHAR(45)
PCName VARCHAR(100)
PC2Email VARCHAR(45)
PC2Name VARCHAR(100)
shortName VARCHAR(20)
logoFile VARCHAR(45)
_Types
typeId INT
typeName VARCHAR(30)
Indexes