1. Preamble

Last modified:	2014-05-09
Group members:	Tyler Hannan
	Nate Book
	Dmitri Ostapenko
Summary:	TA Management System: A database system for assigning TAs to CS courses.
Project idea link:	cs.rochester.edu/tars/
Project location:	betaweb/~nbook/tnd/

2. Domain Description

The old TARS (TA reporting system) is outdated. We propose to make a database to handle the sign up and management of lab TAs for CS courses. Upper level courses without labs with specific times will be handled. Professors will be able to approve specific students as TAs. Students who have completed the course will be able to sign up before the course begins to TA it. Administrators will be able to add, modify, and remove courses and other data. Integration with University of Rochester NetIDs may be possible. Students taking the course may be allowed to view the TAs approved for their course.

3. Use Cases

Case 1:
Paul is a senior in the University of Rochester computer science department who wants to TA a course that he had already taken. He signs up intending to get approved by the professor who teaches the course. Before the start of the course, the instructor, Dr. Boondoggle, can approve TAs.

Case 2:
Matt is a freshman at the University of Rochester who is taking an intro level CS course and wants to know who the TAs for his course is. The system should be able to show who are the approved TAs for the course.

Case 3:
Emily is the administrator assistant for the Computer Science department. She wants to schedule a workshop for CSC 173. She pulls the TA that is scheduled for the class and using their net-id she sends an email to him requesting times that he would be available to lead workshops. Once she hears back from them, she creates workshops that he will lead.

Case 4:
Emily is the administrator assistant for the Computer Science department. She needs to change the professor teaching a course for CSC 171 but wants to make sure that there will not be any conflicts for the time slot that she scheduled for that class and professor. She uses the system in order to fetch all the times that the designated professor is teaching next semester. She also uses the system to fetch all the 171 sections already scheduled. Using the System, she deletes the professor which is no longer teaching and adds the new professor to the system. If there are no conflicts, she leaves it alone, but if there is a conflict, she reschedules the section.

4. Sample English Queries

Case 1:

1. List the positions with times, room, and total positions for a specified course.
2. List the students wanting to sign up to TA a specific professor's class(es).

Case 2:

1. List the approved course TAs for a given course.
2. List the approved workshop leaders for a given course and the workshop weekday and time.

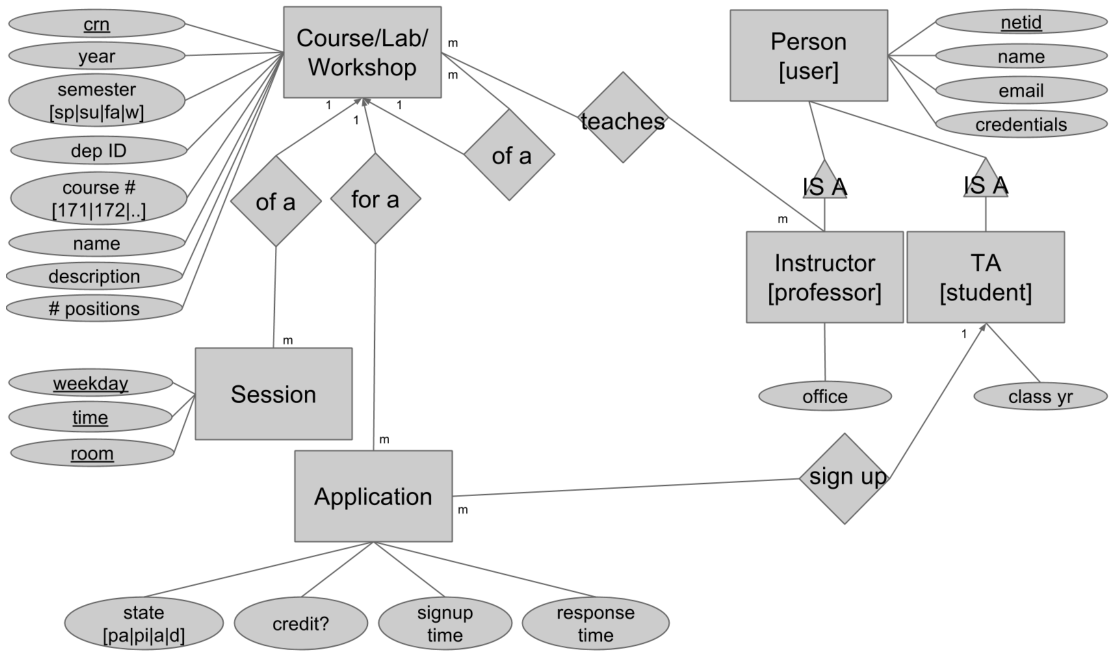
Case 3:

1. Retrieve email addresses for approved TAs for a specific course.
2. For the first workshop leader, create a workshop time, date and location.
3. Repeat for another workshop session if applicable.

Case 4:

1. List all the courses titles, lecture time, and lecture location that a specific professor is teaching.
2. List all sections time and lecture locations of a course being taught this semester.
3. Delete the name of the professor that is no longer teaching a section of the course.
4. Add the name of the new professor who is teaching the section of the course.
5. Delete and re-add new class times and lecture locations for the section if necessary.

5. Entity-Relation Diagram



6. Relations

Courses(<u>crn</u>, year, semester, deptId, courseNumber, name, description, numPositions, parentcrn)
Sessions(<u>wkday</u>, <u>time</u>, <u>room</u>, coursecrn)
Instructors(<u>netid</u>, name, email, credentials, office)
TAs(<u>netid</u>, name, email, credentials, classYear)
Applications(<u>coursecrn</u>, <u>tanetid</u>, signupTime, responseTime, state, forCredit)

Teaches(<u>course</u>, <u>instructor</u>)

7. Sample Relational Algebra Queries

Case 1:

1. List the positions with times, room, and total positions for a specified course:
   "for CSC 171 next semester"

   $R1 = \sigma_{\text{year=2014 AND semester='fall' AND depId='CSC' AND courseNumber=171}} (\text{Courses})$

   $\rightarrow$ $R2 = \pi_{\text{S.wkday, S.time, S.room, C.numPositions}} (R1 \bowtie_{\text{C.crn=S.coursecrn}} \text{Sessions})$

2. List the students wanting to sign up to TA a specific professor's class(es):
   "Dr. Boondoggle's courses, in the fall"

   $R1 = \sigma_{\text{I.netid='boondoggle ' AND C.year=2014 AND C.semester='fall'}} (\text{Courses} \bowtie \text{Teaches} \bowtie \text{Instructors})$

   $\rightarrow$ $R2 = \pi_{\text{T.name, C.depId, C.number, C.name}} (\sigma_{\text{A.state='pending'}} (R1 \bowtie \text{Applications} \bowtie \text{TAs}))$

Case 2:

1. List the approved course TAs (without labs or workshops) for a given course:
   "for CSC 173 in the fall"

   $R1 = \sigma_{\text{year=2014 AND semester='fall' AND depId='CSC' AND courseNumber=173}} (\text{Courses})$

   $R2 = \sigma_{\text{A.state='approved'}} (R1 \bowtie \text{Applications})$

   $\rightarrow$ $R3 = \pi_{\text{T.name}} (R2 \bowtie \text{TAs})$
2. List the approved workshop leaders/Lab TAs for a given course and the workshop weekday and time:
   "for CSC 172 in the spring"

   $R1 = \sigma_{\text{year=2015 AND semester='spring' AND depId='CSC' AND courseNumber=172}} (\text{Courses})$

   $R2 = \sigma_{\text{A.state='approved'}} (R1 \bowtie_{\text{C.parentcrn=R1.crn}} \text{Courses} \bowtie \text{Applications})$

   $\rightarrow$ $R3 = \pi_{\text{T.name}} (R2 \bowtie \text{TAs})$

Case 3:

1. Retrieve email addresses for approved TAs for a specific course:
   "for CSC 173 in the fall"

   $R1 = \sigma_{\text{year=2014 AND semester='fall' AND depId='CSC' AND courseNumber=173}} (\text{Courses})$

   $R2 = \sigma_{\text{A.state='approved'}} (R1 \bowtie \text{Applications})$

   $R3 = \sigma_{\text{A.state='approved'}} (R1 \bowtie_{\text{C.parentcrn=R1.crn}} \text{Courses} \bowtie \text{Applications})$

   $\rightarrow$ $R4 = \pi_{\text{T.email}} ((R2 \bowtie \text{TAs}) \cup (R3 \bowtie \text{TAs}))$

Case 4:

1. List all the course titles, lecture time and location that a specific professor is teaching: "Dr. Boondoggle's courses this semester"

   R1 = $\sigma_{\text{netid='boondoggle'}}$ (Instructors)

   R2 = R1 $\bowtie$ Teaches $\bowtie$ Courses $\bowtie$ Sessions

   $\rightarrow$ R3 = $\pi_{\text{C.name, S.wkday, S.time, S.room}}$ ($\sigma_{\text{semester='spring' AND year=2014}}$ (R2))

2. List all session times and lecture locations of a course being taught this semester: "Dr. Boondoggle's Intro to AI, CSC 240"

   R1 = $\sigma_{\text{depId= 'CSC' AND number=240 AND year=2014 AND semester='spring'}}$ (Courses $\bowtie$ Sessions)

   $\rightarrow$ R2 = $\pi_{\text{S.wkday, S.time, S.room}}$ (R1)

8. Functional Dependencies 1

For Courses:	crn → year	crn → semester	
	crn → deptID	crn → courseNumber	
	crn → name	crn → description	
	crn → numPosititons	crn → parentcrn	
For Sessions:	weekday, room, time → coursecrn		
For Instructors:	netid → name	netid → email	netid → office
For TAs:	netid → name	netid → email	netid → classYear
For Applications:	coursecrn, taNetid → signupTime	coursecrn, taNetid → responseTime	
	coursecrn, taNetid → state	coursecrn, taNetid → forCredit	
For Teaches:	course → instructor		

9. Proof of Normal Form 1

Our schema is in fourth normal form, because all multivalued dependencies are trivial.

10. Relational Schema 2

Since our schema is already in fourth normal form, we do not need to modify the schema.

11. Functional Dependencies 2

Since our schema is already normalized, we are done.

12. Proof of Normal Form 2

Again, our schema is normalized, so this proof is the same as the above.

13. Sample Queries 2

Our relational algebra is identical to the previous.

14. Database Implementation Status

Database setup script:	betaweb/~nbook/tnd/dbsetup-tuesdaynight.sql
Test data insert script:	betaweb/~nbook/tnd/datasetup-tuesdaynight.sql
Test query script:	betaweb/~nbook/tnd/testqueries-tuesdaynight.sql
Test query output:	betaweb/~nbook/tnd/testqueriesout-tuesdaynight.txt

15. Data Abstraction Layer

TA DAL:	betaweb/~nbook/tnd/ta.php (source)
Instructor DAL:	betaweb/~nbook/tnd/instructor.php (source)
Application DAL:	betaweb/~nbook/tnd/application.php (source)
Course DAL:	betaweb/~nbook/tnd/course.php (source)

16. Web Site DB Query Status / Web Site Status

(6) Index/login page: betaweb/~nbook/tnd/index.php (source)

This website is the front page/portal for the entire application. Allows you to log in as either a TA or Instructor by netid and password. When logged in, lists pages relevant to the current logged in user type. Uses the TA and Instructor DALs.

(1) TA lister: betaweb/~nbook/tnd/ta-list.php (source)

This website lists the TAs in the TA table, and allows you to start at a specific place in the data, using pagination. Uses the TA DAL.

(2) TA lister for course: betaweb/~nbook/tnd/ta-list-for-course.php (source)

This webpage requests the course number in the CSC department for which TAs have been approved. Once the course has been given, it returns the netid, name, email and class year of the students who applied and were approved to TA for the course that was given. It uses the TA DAL.

(3) Course lister for professor: betaweb/~nbook/tnd/get-courses.php (source)

This website requests the netid of the professor and the year that the professor was teaching and it lists the courses that the professor was teaches. It returns the name of the course, the weekday the course was taught, the start time of the lecture and the room the lecture was taught in. Use the Course DAL.

(4) Professor teaching updater: betaweb/~nbook/tnd/set-professor.php (source)

This website updates the professor teaching a class. So, the user gives the crn of the course being updated and the professor's netid, and the change is made in the database. CRN's are different from year to year, so that deals with the issue of changing professors for each year. Uses the instructor DAL.

(5) TA application inserter: betaweb/~nbook/tnd/apply-course.php (source)

This website inserts applications for a TA for a class. The user gives the TA's netid and whether or not this is for credit, along with the course department, course number, year and semester. The page then finds the crn number associated with that course based on the information provided and inserts a new application into the applications relation. This is done using the Application DAL, the Course DAL to find the associated crn, and the TA DAL for the actual apply function.

17. Data Abstraction Layer 2

Added Application and Course DAL classes.

18. Web Site DB Query Status 2

Added two websites, (3) and (4).

19. Web Site DB Modification Status

Added three websites, (5), (6), and (7).

20. Additional Database Capabilities

21. Additional Web Site Capabilities

Uses this utility script for DAL: betaweb/~nbook/tnd/utils.php (source)

22. Project Demo

Location: betaweb/~nbook/tnd/design/presentation.pdf