Final Project

Grade System

Alan Menciangelo and Chris Erwin
Project Summary

- Assignment
- Use case
- UML Class Diagram
- Database Tables (mapping to classes)
- SQL Query Implementation
- Execution Walkthrough
- Multiple Layer Design and its Implementation Mapping
- Conclusion
Assignment

Develop a password-protected database for student grading. The system needs to handle all the information for a person role:

- A student who can view his/her grade of any homework, project, quiz, test, and final letter grade in any class he/she took, and only the classes he/she took.
- Any attempt of student to change any of the scores should be forbidden.
- The teacher who can update the above student information in the system, including adding all student records at the beginning of class.
Use Case - Student

OK Case

Exception Case

Login → View Courses → View Grades

Login → View Courses → View Grades

Invaliaded

Relogin

DB Connection Failure

Error Page
Another Case - Teacher
UML Class Diagram

- **Person**
  - int personID
  - String firstName
  - String lastName
  - String email
  - String personType
  - int personTypeID

- **Course**
  - int courseId
  - Teacher teacher
  - String title
  - int year
  - String season

- **Student** extends Person

- **Teacher** extends Person

- **GradesItem**
  - int gradeItemID
  - Student student
  - Course course
  - String title
  - int possiblePoints
  - int earnedPoints
  - String type
Database Tables
## Login

Passwords hashed and salted:

<table>
<thead>
<tr>
<th>lastname</th>
<th>firstname</th>
<th>email</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Bob</td>
<td><a href="mailto:bsmith@wcupa.edu">bsmith@wcupa.edu</a></td>
<td>86ccfc2dfc145c7fbd2238e5e3f9c6ba3cd26af9193fd355a30e9eb195f6c7b54</td>
</tr>
<tr>
<td>Fin</td>
<td>Chris</td>
<td><a href="mailto:ccc326425@wcupa.edu">ccc326425@wcupa.edu</a></td>
<td>d97cc9f4c243f038fba74396d05d1355874047822cf81a0e4f875ca1b56c805c</td>
</tr>
<tr>
<td>Appel</td>
<td>Edna</td>
<td><a href="mailto:ekrabappel@wcupa.edu">ekrabappel@wcupa.edu</a></td>
<td>962a07c3567e4b6887a6ff4f72536ec372a8092198ca2061f8affd7f577d4b75</td>
</tr>
<tr>
<td>Gitter</td>
<td>Gabe</td>
<td><a href="mailto:gkotter@wcupa.edu">gkotter@wcupa.edu</a></td>
<td>2a98d0d11392329cfa56b1a5e6552a693cbada3f6c5593e7a7f60f9242ba980</td>
</tr>
<tr>
<td>Harry</td>
<td>George</td>
<td><a href="mailto:gfeeney@wcupa.edu">gfeeney@wcupa.edu</a></td>
<td>cedfcbf447f2684d4d4824d266502d9102a5270ef52d6933b057e0ac35ffed58</td>
</tr>
<tr>
<td>Jenson</td>
<td>Bart</td>
<td><a href="mailto:bs234659@wcupa.edu">bs234659@wcupa.edu</a></td>
<td>45958f0e12d9c3f9b53604b69a9b1a38aaba8b6459cab2957ae6c8638e7b5e28</td>
</tr>
<tr>
<td>Sano</td>
<td>Robert</td>
<td><a href="mailto:rs987655@wcupa.edu">rs987655@wcupa.edu</a></td>
<td>6d98a77ffce0d13d5a964b6e56ac515d94b64b5c5a31b5444f23dfebe04915ae00d</td>
</tr>
<tr>
<td>Mason</td>
<td>Hugh</td>
<td><a href="mailto:hj112679@wcupa.edu">hj112679@wcupa.edu</a></td>
<td>ba60cd22775c2c57b9adc9650c22e1489d027176c665f7c7f981b24c081009</td>
</tr>
<tr>
<td>Whitman</td>
<td>Eric</td>
<td><a href="mailto:ec987654@wcupa.edu">ec987654@wcupa.edu</a></td>
<td>3cfc6348638710c914daf79cdb55313757c30e987fc9ca2690c59003538991b0</td>
</tr>
</tbody>
</table>

set (0.00 sec)
Login

SELECT id as personid, email
FROM person
WHERE email = emailEntered and password=sha2(CONCAT(passwordEntered,firstname,lastname),256);
Queries – get one person

SELECT person.id as personid, lastname, firstname, email, persontype
FROM person
JOIN persontype ON person.persontypeid = persontype.id
WHERE person.id=?
WHERE person.id=?
Queries – get one course

```
SELECT course.id as coursed, teacherid, title, semester.year, semester.season
FROM course
JOIN semester ON course.semesterid = semester.id
WHERE course.id = ?
WHERE course.id = ?
```
Queries – get one gradeitem

```
SELECT gradeitem.id, studentid, coursed, title, possiblepoints, earnedpoints, gradeitemtype.type
FROM gradeitem
JOIN gradeitemtype on gradeitem.typeid = gradeitemtype.id
WHERE gradeitem.id = ?
```
Queries - other

SELECT courseid FROM studentcoursemap WHERE studentid = ?
SELECT studentid FROM studentcoursemap WHERE courseid = ?
SELECT id FROM gradeitem WHERE studentid = ?
SELECT id FROM gradeitem WHERE courseid = ?
SELECT id FROM gradeitem WHERE courseid = ?
SELECT id FROM gradeitem WHERE studentid = ? And courseid = ?
Teacher Execution

Teacher login and student add.

Username (email): bsmith@wcupa.edu
Password: CSC321
Hello Bob Smith
Welcome to grading system
~~~~Teacher Menu~~~~
  Add a Student
  Add an Assignment
  Update an Assignment
  View a Course
  Enroll a student in a course
  Exit

Adding a Student
  First Name: Tom
  Last Name: Gregory
  Email: tgreg@wcupa.edu
  Student added.

Adding an assignment.

2
Adding an assignment
Assignment title: test2
Student ID: 3
Course ID: 1
Possible Points: 100
Earned Points: 92
Assignment type (homework,quiz,test): test
Assignment successfully added.
Course Grades

Chris Erwin

- homework1: Earned Points: 15, Possible Points: 20, Score: 75.0
- homework2: Earned Points: 10, Possible Points: 8, Score: 125.0
- homework3: Earned Points: 10, Possible Points: 8, Score: 125.0
- test: Earned Points: 10, Possible Points: 10, Score: 100.0
- test2: Earned Points: 90, Possible Points: 120, Score: 75.0
- Final Score: 98.571434
- Course Grade: A

Michael Corleoni

- homework1: Earned Points: 9, Possible Points: 10, Score: 90.0
- homework2: Earned Points: 14, Possible Points: 15, Score: 93.333336
- test1: Earned Points: 82, Possible Points: 100, Score: 82.0
- Final Score: 88.444444
- Course Grade: B

Robert Sacamano

- hw1: Earned Points: 4, Possible Points: 5, Score: 80.0
- test: Earned Points: 45, Possible Points: 50, Score: 90.0
- test2: Earned Points: 0, Possible Points: 100, Score: 0.0
- Final Score: 56.666668
- Course Grade: F

Tom Gregory

- test1: Earned Points: 75, Possible Points: 100, Score: 75.0
- Final Score: 75.0
- Course Grade: C
Intro to Mathonomics

Assignment: homework 1  Earned Points: 15  Possible Points: 20  Score: 75%
Assignment: homework2  Earned Points: 10  Possible Points: 8  Score: 125%
Assignment: homework3  Earned Points: 10  Possible Points: 8  Score: 125%
Assignment: test     Earned Points: 10  Possible Points: 10  Score: 100.0%
Assignment: tester   Earned Points: 10  Possible Points: 10  Score: 100.0%
Final Score: 102.5
Course Grade: A

1. View Courses
2. Quit

Courses

- Intro to Mathonomics  Teacher: Smith  Semester: spring 2014
- Recursion II    Teacher: Krabapple  Semester: fall 2013

Student Execution

e (email): ce326425@wcupa.edu
d: csc321
login
e (email): ce326425@wcupa.edu
d: CSC321
hris Erwin
to grading system
udent~~~~~

Courses~~~~~
Student Execution

- Showing updates

Intro to Mathonomics

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Earned Points</th>
<th>Possible Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>homework 1</td>
<td>15</td>
<td>20</td>
<td>75.0</td>
</tr>
<tr>
<td>homework 2</td>
<td>10</td>
<td>8</td>
<td>125.0</td>
</tr>
<tr>
<td>homework 3</td>
<td>10</td>
<td>8</td>
<td>125.0</td>
</tr>
<tr>
<td>test</td>
<td>10</td>
<td>10</td>
<td>100.0</td>
</tr>
<tr>
<td>tester</td>
<td>10</td>
<td>10</td>
<td>100.0</td>
</tr>
<tr>
<td>test</td>
<td>90</td>
<td>100</td>
<td>90.0</td>
</tr>
<tr>
<td>test2</td>
<td>90</td>
<td>120</td>
<td>75.0</td>
</tr>
</tbody>
</table>

Final Score: 98.571434

Course Grade: A

Intro to Mathonomics

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Earned Points</th>
<th>Possible Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>homework 1</td>
<td>15</td>
<td>20</td>
<td>75.0</td>
</tr>
<tr>
<td>homework 2</td>
<td>10</td>
<td>8</td>
<td>125.0</td>
</tr>
<tr>
<td>homework 3</td>
<td>10</td>
<td>8</td>
<td>125.0</td>
</tr>
<tr>
<td>test</td>
<td>10</td>
<td>10</td>
<td>100.0</td>
</tr>
<tr>
<td>tester</td>
<td>90</td>
<td>100</td>
<td>90.0</td>
</tr>
<tr>
<td>test</td>
<td>90</td>
<td>120</td>
<td>75.0</td>
</tr>
</tbody>
</table>

Final Score: 97.14286

Course Grade: A
Multiple Layer Design
Mapping Classes - DBConnector

Executes prepared statements against database

Returns resultsets to be fed to object constructors

```java
public ResultSet getPerson(int personID) throws Exception{
    cx = connection();
    ResultSet rs = null;
    String query = "SELECT person.id as personid, lastname, firstname, email, persontype " +
                    "FROM person " +
                    "JOIN persontype ON person.persontypeid = persontype.id " +
                    "WHERE person.id=?";
    PreparedStatement st = cx.prepareStatement(query);
    st.setInt(1, personID);
    rs = st.executeQuery();
    return rs;
}
```
Mapping Classes - Mappers

Returns ArrayLists of Objects

```java
public ArrayList<Teacher> teachers(ResultSet rs) throws Exception {
    DBConnector db = new DBConnector();
    ArrayList<Teacher> data = new ArrayList<>();
    while (rs.next()) {
        data.add(new Teacher(db.getPerson(rs.getInt("personid"))));
    }
    return data;
}
```
Summary

- Most time consuming – circling back to fill in functionality in classes when implementing them (better planning)
- Mainly worked remotely using GitHub – worked very well
- Don’t pass around ResultSet – relies on garbage collection