* Entities-
  + Instructor
    - Attributes
      * instructor\_id
      * fname
      * lname
      * email
      * password
      * school\_id
    - User Stories
      * As an instructor I should be able to...
        + Create a course
        + Grade a student's assignment
        + Submit files/assignments to My WebSchool
        + Post/Edit Syllabus
        + After Max Capacity, can add a student to the course

Like Signing an RTF form

* + - * + View Student's Files that they upload
  + Student
    - student\_id
    - fname
    - lname
    - role\_id
    - course\_ranking
    - gpa?
    - email
    - password
    - school\_id
    - User Stories
      * As an student I should be able to...
        + Enroll in a course
        + Drop/Change a course
        + View assignments
        + Submit files/assignments to My WebSchool
        + View Syllabus
        + After Max Capacity, can ask professor for admittance
        + View Files that they upload
        + View Grades
        + Grade assignments if I am a TA
  + Course
    - course\_id
    - school\_id
    - instructor\_id
    - name
    - subject\_id
    - location
    - syllabus
    - start\_date
    - end\_date
    - num\_of\_credits
    - max\_capacity
  + School
    - school\_id
    - name
    - description
    - num\_of\_students
    - location
    - num\_of\_instructors
  + Assignment
    - assignment\_id
    - course\_id
    - assigned\_date
    - due\_date
    - name
    - file\_url
    - instructions
    - state: in class, out of class
    - type: group, individual
  + Grade
    - grade\_id
    - assignment\_id
    - student\_id
    - grade
    - file\_url
    - submitted\_date
    - submitted\_time
    - state: needs grading, graded
  + Semester
    - semester\_id
    - semester\_name
  + Role
    - role\_id
    - role\_name
* Computational Component
  + Course Ranking
    - Highest Average in the Course
  + GPA
    - Weighted Average Across all Semesters
  + Weighted Average
    - DUH
  + Average
    - DUH
* Technologies-
  + Java Spring Backend/Computational
    - Supports Hibernate and because it's AWESOME :-)
  + Hibernate ORM
    - Makes it a lot easier to map objects to database and makes coding a lot more simple because you don't have to wrtie a lot of different sql statements.
  + Angular Frontend
    - State of the Art. Efficient and easily connects with backend technologies. Component-based.
    - http://larseidnes.com/2014/11/05/angularjs-the-bad-parts/
  + MySQL Database
    - Very Tabular Data shown in many tables in many views of the application. Plus type safety prevents any stupidity from happening. NEVER TRUST USER INPUT.