Executive Summary

Leon Tran tranleo@oregonstate.edu
Michael Tornatta tornattm@oregonstate.edu

Project URL: http://flip3.engr.oregonstate.edu:5312/

Fixes based on Feedback From Step 1

- Missing the identification of foreign and primary keys, fixed this by identifying the keys we missed.
- Field names changed to all lowercase.

Fixes based on Feedback From Step 2

- We're adding more details about the entities which was a suggestion made in all of the reviews.
- We're specifying if an attribute is NULL or NOT NULL based on feedback from all of the reviews.
- Made all of the entities singular and capitalized them in the overview based on comments from all of the reviews.
- Underlined the keys in our outline based on reviews.

Changes made based on Peer Feedback from Step 3

- Created a school page along with a field to insert school data based on the comments from our first submission on canva.
- Included an option to delete from tables based on reviews from everyone mentioned in the above section.
- All of the comments we got mentioned that we did not have an option to update an entity which is not entirely correct. All of the entity pages allow you to "add or update" an entity. When updating an entity you would just need to enter the name of the entity you are trying to update. We'll change it so that the updating takes place within the displayed table and you can take an action to update an item.
- We changed the student count in classes to an attribute that's allowed to be null based on all of our feedback.
- Added headers in order to make the site more navigable based on the comments we received.

Feedback from Step 4-6

- Most of the feedback from this step was to just get our site working properly, our links were broken for most of the last few project iterations.

Project Outline and Database Outline - Updated Version

Project Name: School Organization System (S.O.S.)

Overview

We want to create a school website that operates using a database in the back end. The website would be used by the school to track students and teachers by an ID number and would establish relationships between the people in the school and the physical resources (classrooms, buildings, ect.) the school has to offer. The goal of the system would be to allocate teachers to classes, classes to rooms, and students to staffed classes. We'll be dealing with a large higher education school with 20,000 students and 1,250 professors (16:1 student:teacher ratio). There will be a system to search for classes, teachers, and students. The user can also add classes to the school, as well as students and teachers to a class.

- Student: All students are identified by a student_id number. A student can be enrolled by course ID in more than one class but should not exceed 5.
    - first_name: varchar, NOT NULL
    - last_name: varchar, NOT NULL
    - student_id: int, NOT NULL
    - year: int, NOT NULL
    - major: varchar, NOT NULL
    - email: varchar, NOT NULL
    - school_name: varchar, NOT NULL
    - M:M relationship with classes
    - M:1 relationship with school
    - Primary key: {student_id}

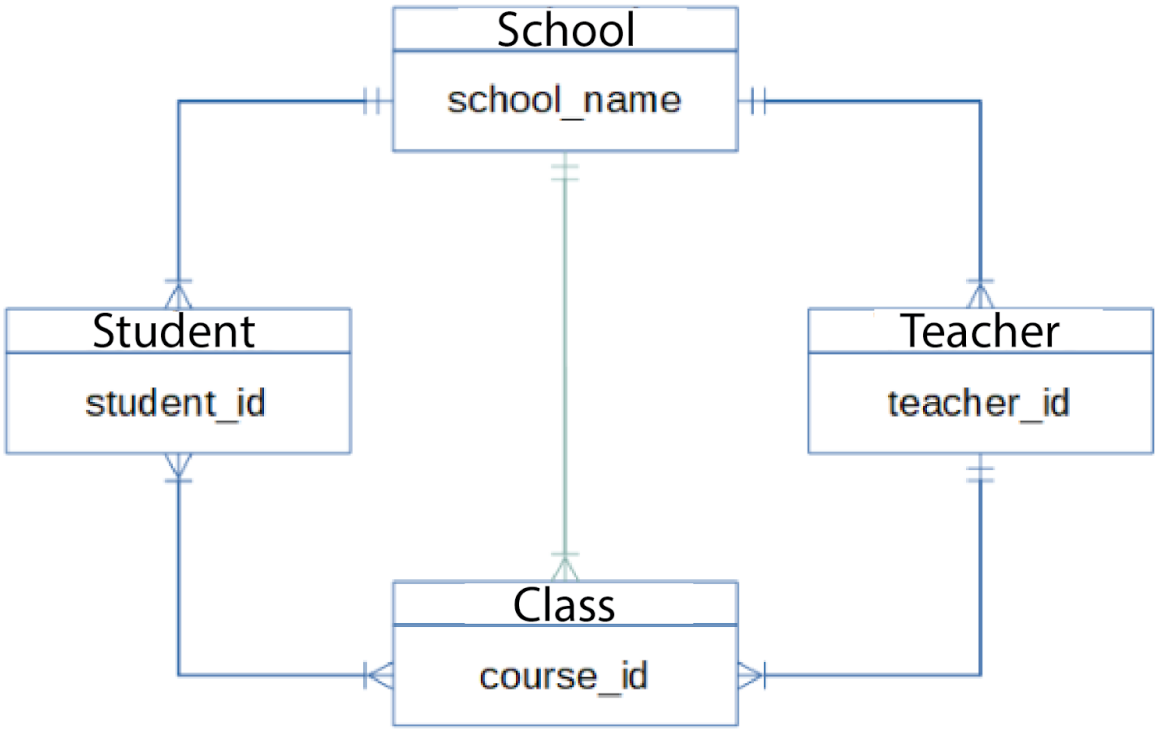
- Class: Classes are identified by course_id. When available_seats is 0 no more students can be added.
    - course_id: int, NOT NULL
    - number_of_students: int
    - available_seats: int, NOT NULL
    - course_name: varchar, NOT NULL
    - school_name: varchar, NOT NULL
    - M:1 relationship with teachers

    - Primary key: {course_id}

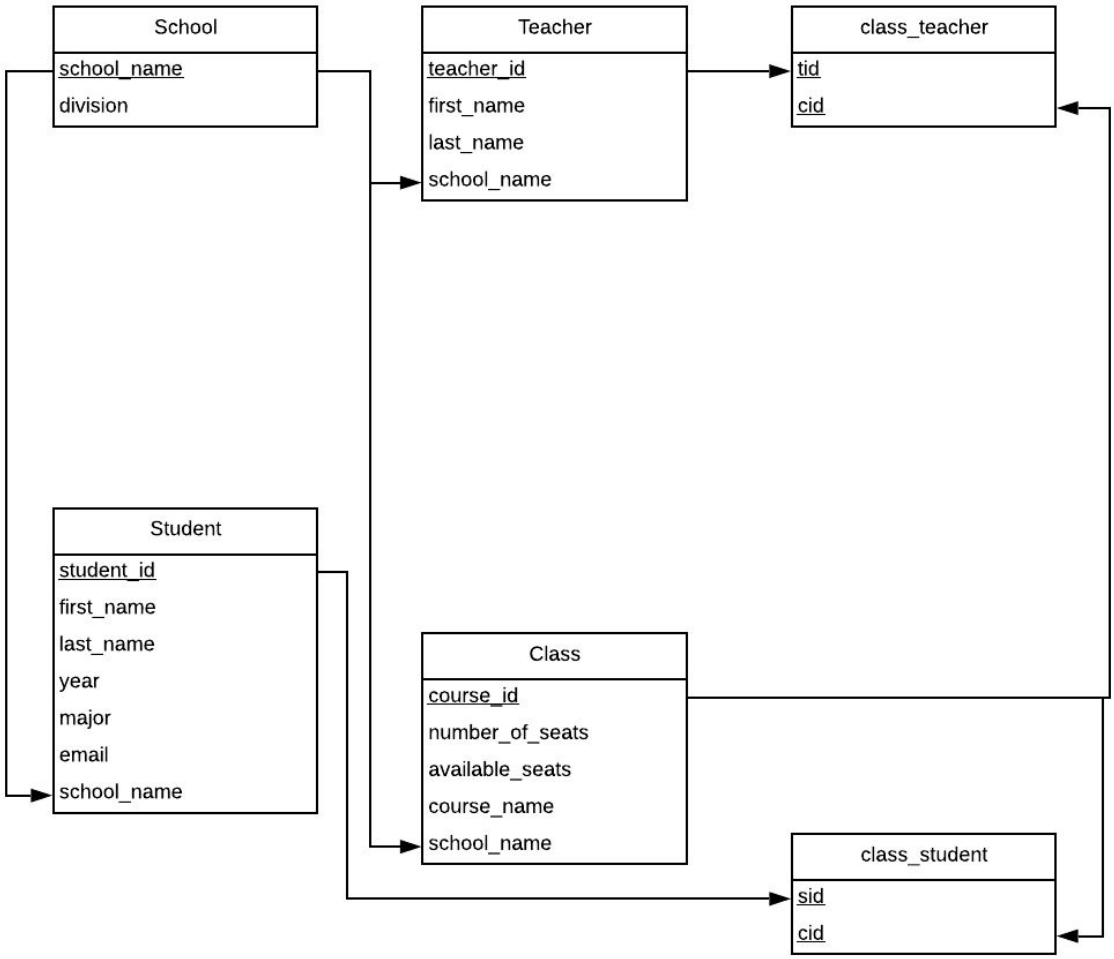
- School: Identified by school_name. All entities are linked to School.
    - school_name: varchar, NOT NULL
    - division: varchar, NOT NULL
    - 1:M with all attributes
    - Primary key: {school_name}
    - FK: school_name with all attributes

- Teacher: Identified by teacher_id. Can teach many classes.
    - first_name: varchar, NOT NULL
    - last_name: varchar, NOT NULL
    - teacher_id: int, NOT NULL
    - school_name: varchar, NOT NULL
    - 1:M relationship with classes
    - M:1 relationship with school
    - Primary key: {teacher_id}

ER Diagram

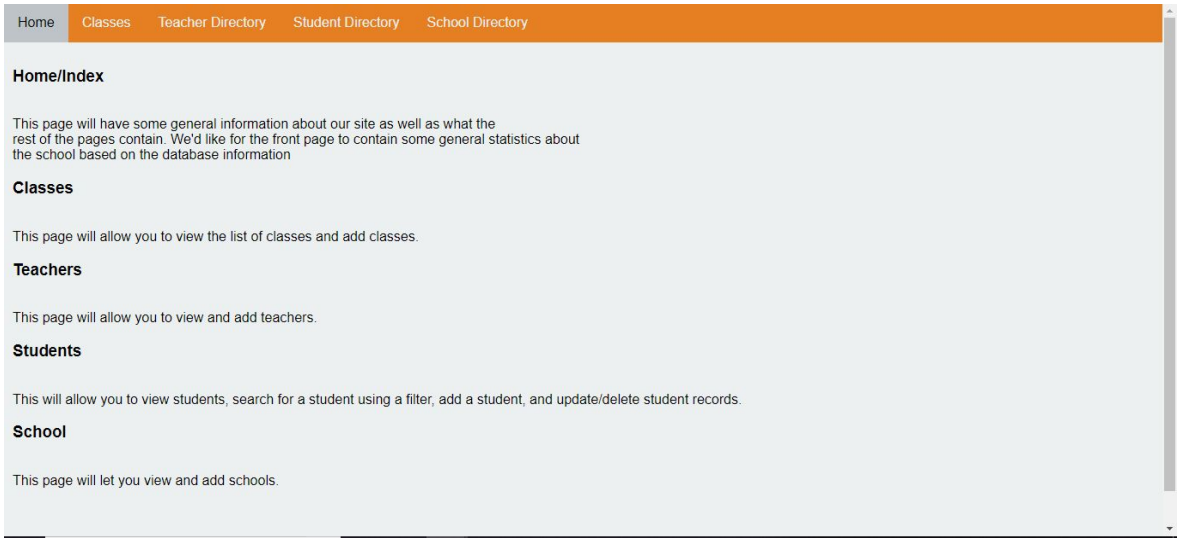


Schema

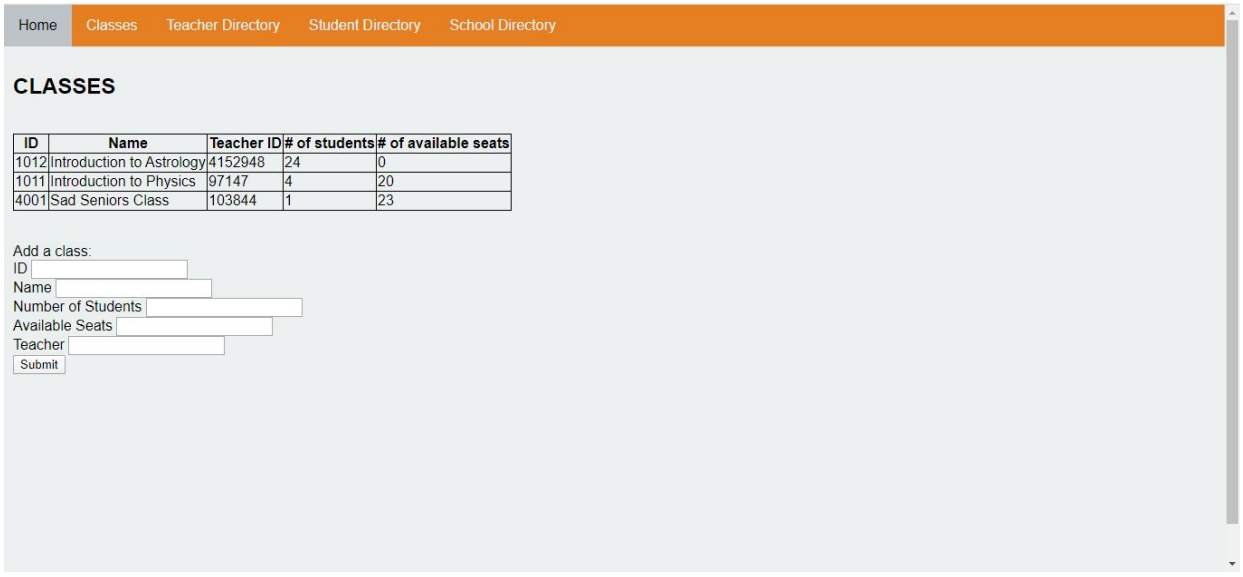

School
school_name
division
Teacher
teacher_id
first_name
last_name
school_name
class_teacher
tid
cid
Student
student_id
first_name
last_name
year
major
email
school_name
Class
course_id
number_of_seats
available_seats
course_name
school_name
class_student
sid
cid

UI Walkthrough

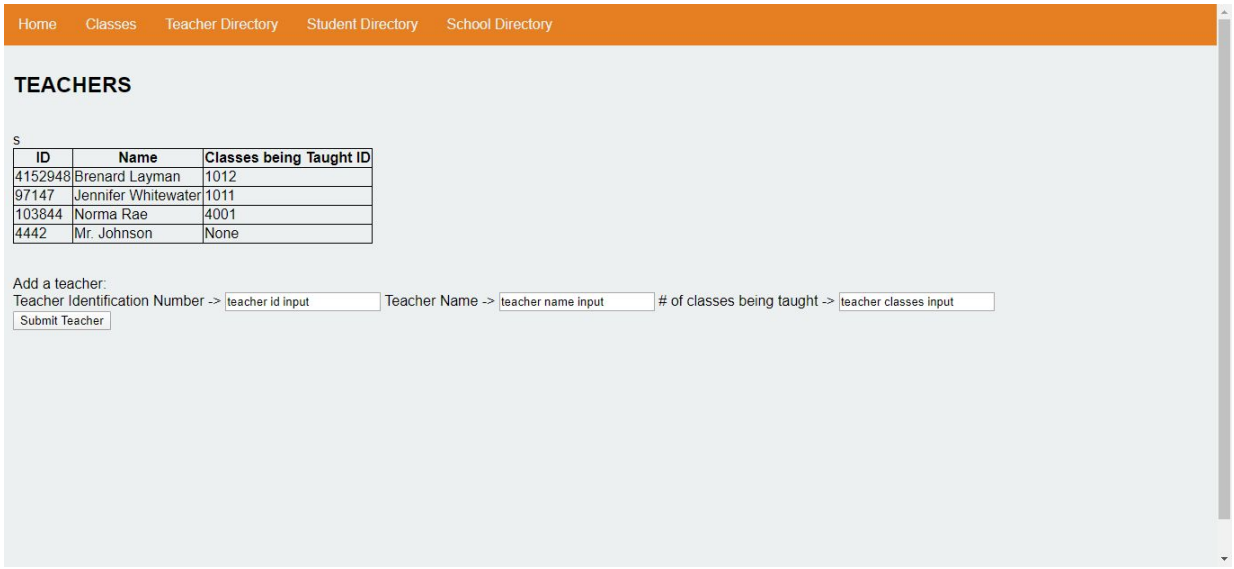
This page gives you a description of each page on the site and has a navbar so you can get to all of our pages.



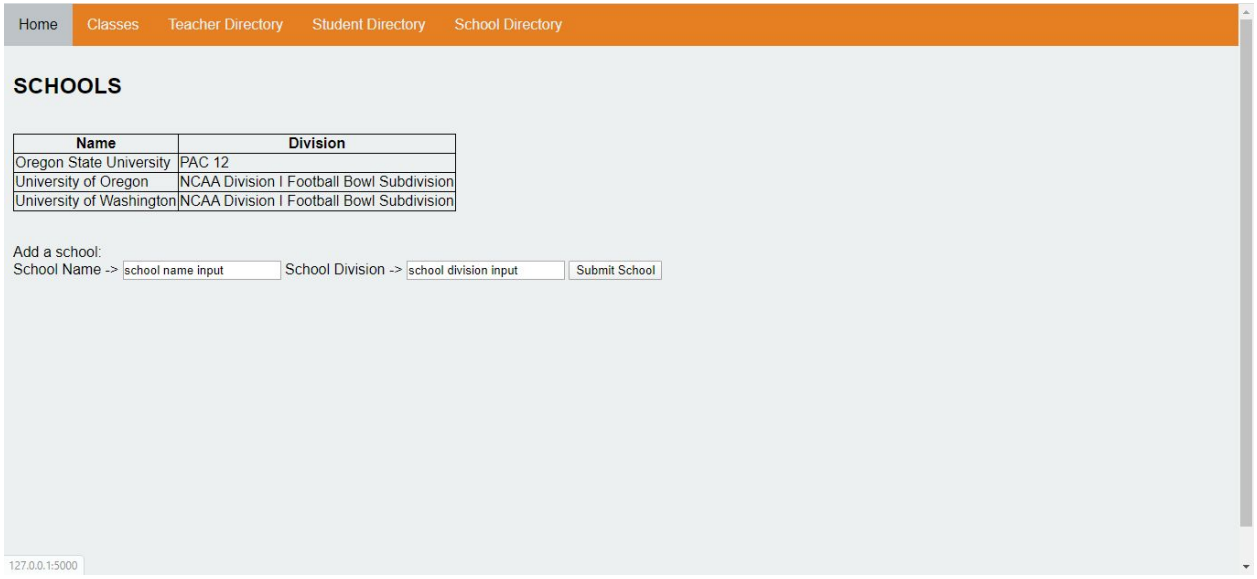
The class, teachers, and school pages all allow you to view data which is queried using SELECT and you can INSERT data into the databases.



The Teacher page contains SELECT and INSERT functionality



The school page also contains SELECT and INSERT functionality.



The student page contains the most functionality and allows you to view students and alter student records. You can delete a student by student ID using the text entry field and the “Delete” button. You can add or update (if their ID already exists) by entering student credentials into the bottom fields and pressing the “Add/Update” button. The Delete and Update functionality are a little glitchy and you have to leave the page and come back for the updates to show but it does work! The only thing we could not get working unfortunately was the filter and search functionality. But we have the interface for it.