**URL** : <u>http://access.engr.oregonstate.edu:4462/</u>
**Name**
A student volunteer registration systems:

## Summary:

At the beginning we design to makes 5 entities(Phone,Students,Grades,College and Language) and 3 intersection table (Skill_table, StudentColleges_table,and StudentGrades_table ) in our database, however with the deep research and check the relation of each part we change the entity to these 5 (StudentID, Students, Grades, Genders and Luangages). This change because of considering the practicability and the correction of the relationship. We agree that the attribute of student phone numbers is redundant as we have a discussion. Because the student ID is much more important than the phone number in the real world university. Schools can find the person by their ID. And we don't need to ask students to provide their own phone numbers. However we think we still need to have a 1:1 relationship between 2 entities, so we add the StudentID entities. Then we start to recheck the lowercase letter, and the capital letter. We may need to make them consistent.  After that we reconsider the intersection table based on our project goal. We delete the college entities and the intersection table and also we use more tables to represent the more complex relation.Somebody gave us some very good suggestions in the review. Because the outline of each of our entities does not clearly describe their purpose. We add a sentence after the entity name to describe the purpose of the entity.  Finally we design the search part case on the new entities, and it can select the students with the StudentsID, or First name or Last Name or gender or grades or language they can speak.

## 1) Overview

A college student social activity organization wants to hold a  long term social activity to receive foreign visitors and provide the translation service, this system will record the 200 volunteer registration.
The database driven website will record  the student basic information, Grades, Name , Gender,Language skill, StudentID to Students of 200 volunteers from college each term, and this Language skill is the Language they can speak. The web can select the student's name and StudentID by selecting different language skills they can speak or Grades or Gender or Student ID.

## 2) Database Outline

**Entities:**

(StudentID is works like the University student's ID)

- StudentID:
    - personID:int, unique, not NULL, PK
    - studentID:int , unique,  not NULL
    - Relationship:
      a 1:1 relationship between Students and StudentID is implemented with personID as a FK inside StudentID


- Students:
    - personID:int, auto_increment, unique, not NULL, PK
    - fname:varchar, not NULL
    - lname:varchar, not NULL
    - studentGenders: int, not NULL, FK
    - studentGrades: not NULL, FK
    - studentColleges(not implement) :not NULL, FK
    - skills :not NULL, FK
    - Relationship:
      a 1:1 relationship between Students and StudentID is implemented with personID as a FK inside Phones
      A M:1 relationship between Students and Genders is implemented with genderID as a FK inside Genders
      a M:1 relationship between Students and Grades is implemented with gradeID as a FK inside studentGrades-table
      A M:M relationship between Students and Languages is implemented with personID and the languageIDas a FK inside skill-table


(Grades has 4 choice freshman,sophomore, junior,    and senior)
- Grades:
    - gradeID: int, auto_increment,unique,not NULL,PK
    - levelName: varchar, not NULL
    - Relationship:
      a 1:M relationship between Grades and Students is implemented with gradeID as a FK inside studentGrades

(Language is what the student's can speak: English Chinese, French, Spanish, Arabic)
- Languages
  - languageID: int,auto_increment,unique,not NULL, PK
  - languageName:varchar, not NULL
  - Relationship:
    a M:M relationship between Language and Students is implemented with personID and languageID as a FK inside skills


(Genders, male, female, unknown)
- Genders
  - genderID: int PK
  - genderName:varchar, not NULL
  - Relationship:
    A 1:M relationship between Students and Genders is implemented with genderID as a FK inside Genders
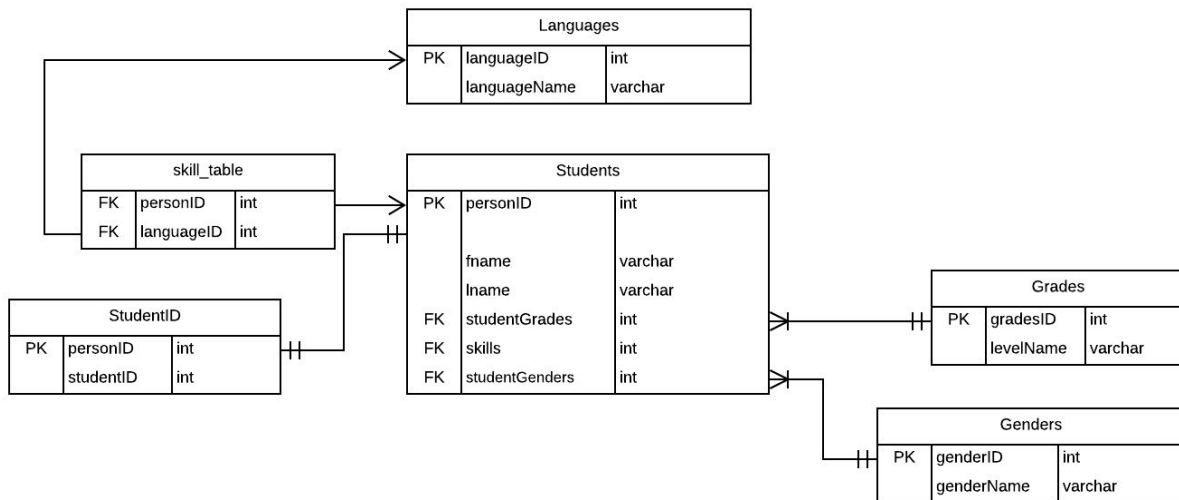
**relation tables :**

skills_table:
- personID: int, not NULL,PK
- languageID ; int , not NULL

This can implement the relationship between Students and Language which has a relationship of M:M. A Student can speak many Languages. And A Language can be spoken to Many students.


## c) Entity-Relationship Diagram:
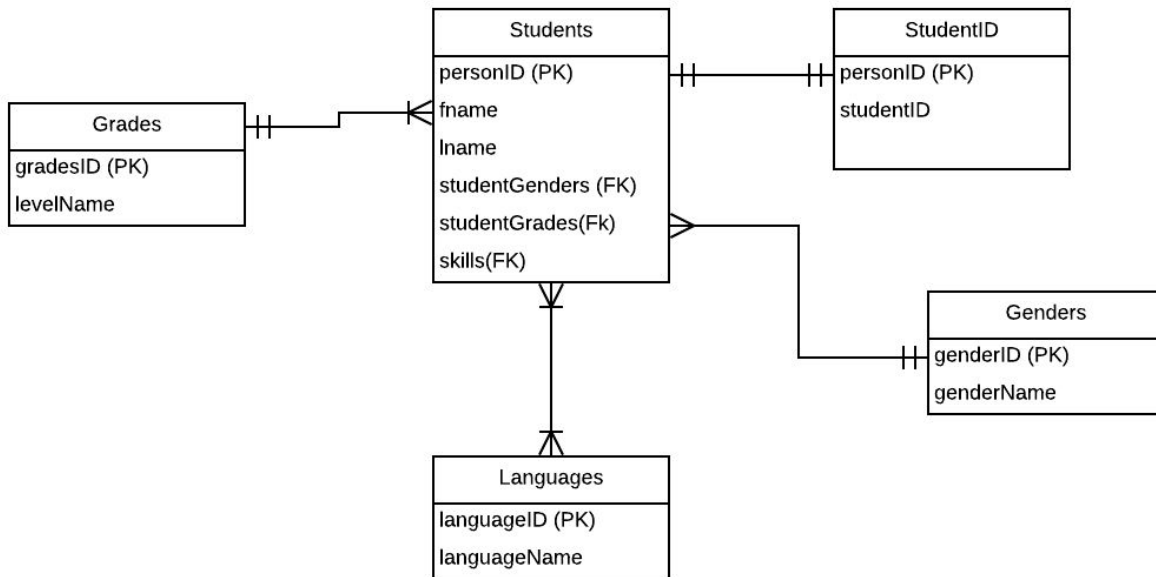
## d) Schema:



## e) Screen captures of each of the UI pages on your website

Home page:There are two buttons, one is to modify or add student information. The other is to find the students according to the specific student information.

# Student volunteer database system

modify
search

Modify page:There are five buttons, the first is to enter the interface of modifying or adding student information. The second is to enter the interface of modifying and adding voice options. The third is the interface to modify or add age. The fourth is to modify or add gender interface. The fifth is to return to the home page.

Add student information
Add language
Add Grade
Add Gender
back

Add student information page:

All buttons in this interface are available. You can add a new student, such as Yang Zhang. Its gender, age and at least one language are required. You can also add multiple languages. After clicking the Add button, the new student information will be displayed in the view list. It can also be modified through the modify and delete buttons in the viw list.

studentID

first name

last name

male ∨

frashman ∨

Chinese ∨

[add language] [remove language]

[add] [back]

# View List

- id: 100 | Bob | Young | trans ∨ | frashman ∨
  languages: Chinese ∨ English ∨ A ∨
  [modify] [delete]
- id: 101 | A | Young | male ∨ | junior ∨
  languages: Chinese ∨ A ∨
  [modify] [delete]
- id: 102 | B | Young | female ∨ | sophomore ∨
  languages: English ∨
  [modify] [delete]

[refresh]

---

123456

Yang

Zhang

male ∨

frashman ∨

Chinese ∨ Chinese ∨

[add language] [remove language]

[add] [back]

# View List

- id: 100 | Bob | Young | trans ∨ | frashman ∨
  languages: Chinese ∨ English ∨ A ∨
  [modify] [delete]
- id: 101 | A | Young | male ∨ | junior ∨
  languages: Chinese ∨ A ∨
  [modify] [delete]
- id: 102 | B | Young | female ∨ | sophomore ∨
  languages: English ∨
  [modify] [delete]
- id: 123456 | Yang | Zhang | female ∨ | senior ∨
  languages: Chinese ∨ English ∨
  [modify] [delete]

[refresh]

Add language page: All buttons are available.

language
[ add ] [ back ]

## View List

- id: 1 [ Chinese ] [ modify ] [ delete ]
- id: 2 [ English ] [ modify ] [ delete ]
- id: 3 [ A ] [ modify ] [ delete ]

Japanese
[ add ] [ back ]

## View List

- id: 1 [ Chinese ] [ modify ] [ delete ]
- id: 2 [ English ] [ modify ] [ delete ]
- id: 3 [ A ] [ modify ] [ delete ]
- id: 4 [ Japanese ] [ modify ] [ delete ]

Add Grade page: All buttons are available.

grade
[ add ] [ back ]

## View List

- id: 1 [ frashman ] [ modify ] [ delete ]
- id: 2 [ sophomore ] [ modify ] [ delete ]
- id: 3 [ junior ] [ modify ] [ delete ]
- id: 4 [ senior ] [ modify ] [ delete ]

Add gender page: All buttons are available.

gender

[ add ] [ back ]

## View List

- id: 1  male   [ modify ] [ delete ]
- id: 2  female [ modify ] [ delete ]
- id: 3  trans  [ modify ] [ delete ]

Search page: The list of students who meet the requirements can be found according to any one of the conditions. For example, I want to find students who can speak Chinese.

## filter conditions

| studentID | fname | lname | gender | grade |

language

[ search ]

## result:

_____

[ back home ]

## filter conditions

| studentID | fname | lname | gender | grade |

Chinese

[ search ]

## result:

studentID: 100, fname: Bob, lname: Young, studentGrades: frashman, studentGenders: trans, langs: Chinese,English,A

studentID: 101, fname: A, lname: Young, studentGrades: junior, studentGenders: male, langs: Chinese,A

studentID: 123456, fname: Yang, lname: Zhang, studentGrades: senior, studentGenders: female, langs: Chinese,English

[ back home ]

## f).Team evaluation form

**CS 340 TEAM EVALUATION FORM**
**July 12, 2020**
**RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.**

**1 = Strongly Disagree     2 = Disagree   3 = Agree 4 = Strongly Agree**

| **Group number** | 2 | |
|---|---|---|
| **Name of Group TEAM Members:** | Yang Zhang , Zan Zhang | |
| **SCALE AND COMMENTS** | RATING | ADDITIONAL COMMENTS |
| **HoW Prepared was your team?**<br><br>Research, reading, and assignment complete | 4 | We are both working on researching and reading more method we need to use in our project. We also try hard to finish every assignment whether the homework or the team assign work. |
| **How responsive & COMMUNICATIVE were you both as a team?**<br><br>Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc. | 4 | We are using email to talk and also the WeChat in order to talk and discuss the problem we meet efficiently |
| **Did both group members Participate equally**<br><br>Contributed best academic ability | 4 | We are using email to talk and also the WeChat in order to talk and discuss the problem we meet efficiently |

| | | |
|---|---|---|
| **DID YOU BOTH FOLLOW THE initial team CONTRact?**<br><br> Were both team members both positive and productive? | 4 | Yes, we discuss and assign small goals for each one. And we also work well together. |