Progress Report 1 for ConGrad

Group members: Jieyu Ren, Qi Cheng, Zijun Liu, Denglin Zhang, Hao Yang, Tao Huang

1. Project Description

1.1. Major Pieces Description

Our project has several major pieces: User Interface, Administrator Interface, Databases, and the recommendation algorithm. The User Interface is where users login into their account, enter their information, and receive their recommended list of programs. The Administrator Interface is where the administrator creates, reads, updates or deletes information in the database. The Database is the place where we store the information we have collected so far. The recommendation algorithm is the algorithm designed to provide the recommended list of programs to users. Our expectation is to connect the database and the recommendation algorithm to calculate the possibility that a certain user successfully applies for a specific program.

1.2. Users Classes and Characteristics

The target users for this project are the students who have the intention to apply for a graduate/PhD program but do not have a specific plan or idea about which program they should apply for. In this case, our product would provide a recommended list of programs to the users.

1.3. Expected Function

Our product would have several functions: 1. It allows users to search admission information from previous years and use our sort and time filters to refine the results. 2. It generates a recommended program list based on users' application information.

2. Background and related work

ConGrad focuses on providing users with the function of searching admission information from previous years and using our sort and time filters to refine the results. And it can generate a recommended program list based on users' application information. Using our system, the user can get recommendations for the school/programs that they are probably interested in and have a larger chance of getting accepted given their academic status.

Gradcafe: https://www.thegradcafe.com/
Offerduoduo: https://offer.1point3acres.com/

3. Completed Work

3.1. Information Collection

An essential part of our project is the dataset that we can use to generate acceptance patterns and to be displayed on the search page. In addition, only after we have collected some information can our team test and debug our programs. Therefore, the first thing that our team did was to collect attributes of information designed for the databases.

There are two sets of data that we need: the first set is the information about the graduate schools and their programs; the second set is the past applicants' details and their application results. Our team has collected the attributes for over 60 graduate schools and the information of at least 10 programs for each of them. For the students, we have collected over 600 applications' details. In addition, we have entered all the data collected into our database.

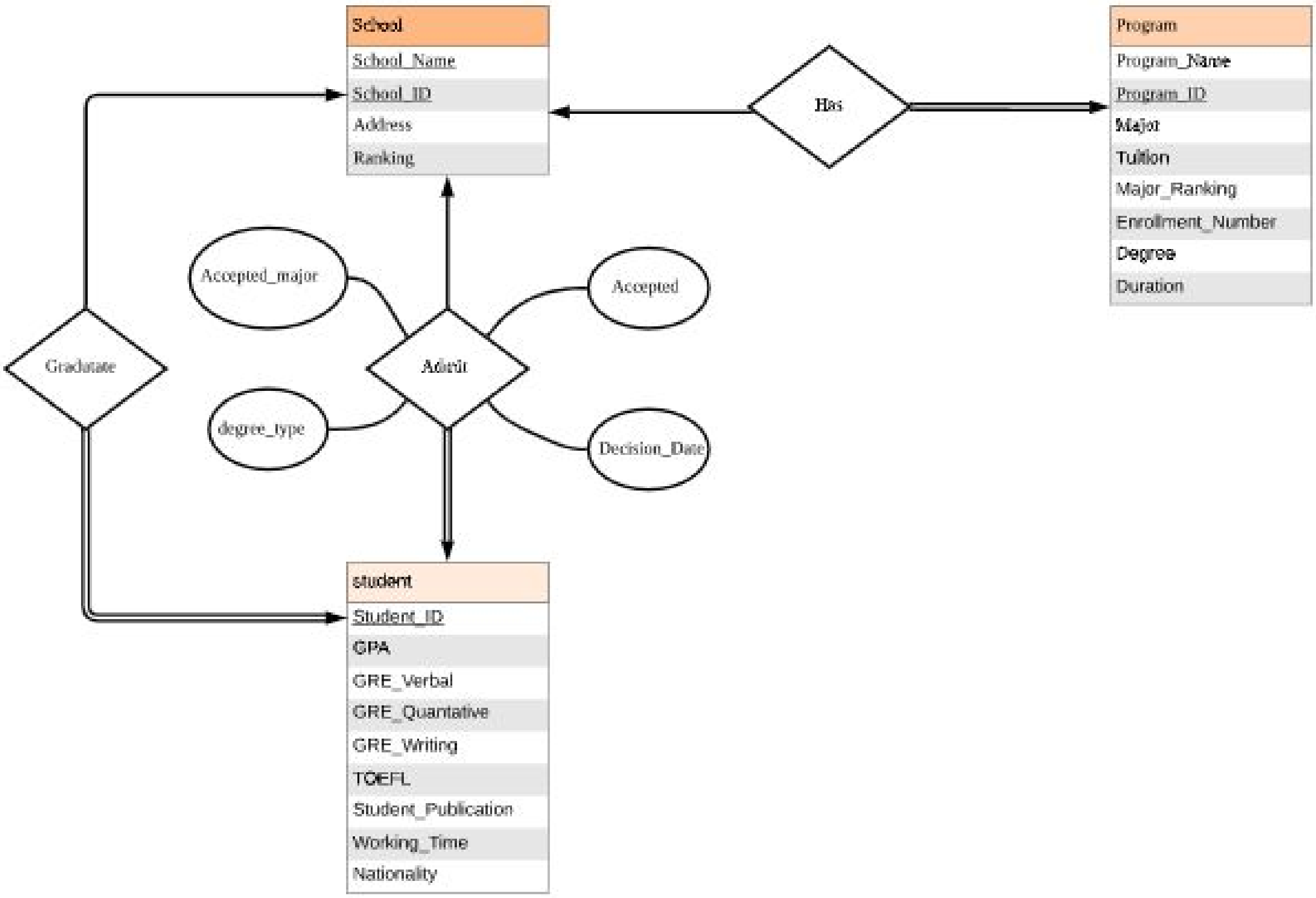
3.2. Database Design and Initialization

Our application requires a large amount of data about various schools, programs, and students, so we needed to design our database carefully so that all the information can be properly stored when we are doing data collection and easily retrieved in the later phase of the program when we need to analyze the data.

As a result, we designed six entities for our database. School entity to store school name, made-up school ID, address of schools, and U.S. News ranking of schools, with school ID being the primary key of the entity school. Entity program to store program name, made-up program ID, major of programs, tuitions, major ranking according to U.S. News, the total number of enrollment of programs, degree type provided by the program, with program ID being the primary key. To link each school with its programs, we designed a relational entity Has since each school has multiple programs. The entity Has got only two keys, school ID and program ID which are all foreign keys and it is a many-to-one relationship since each school can have many programs but each program can only belong to one school.

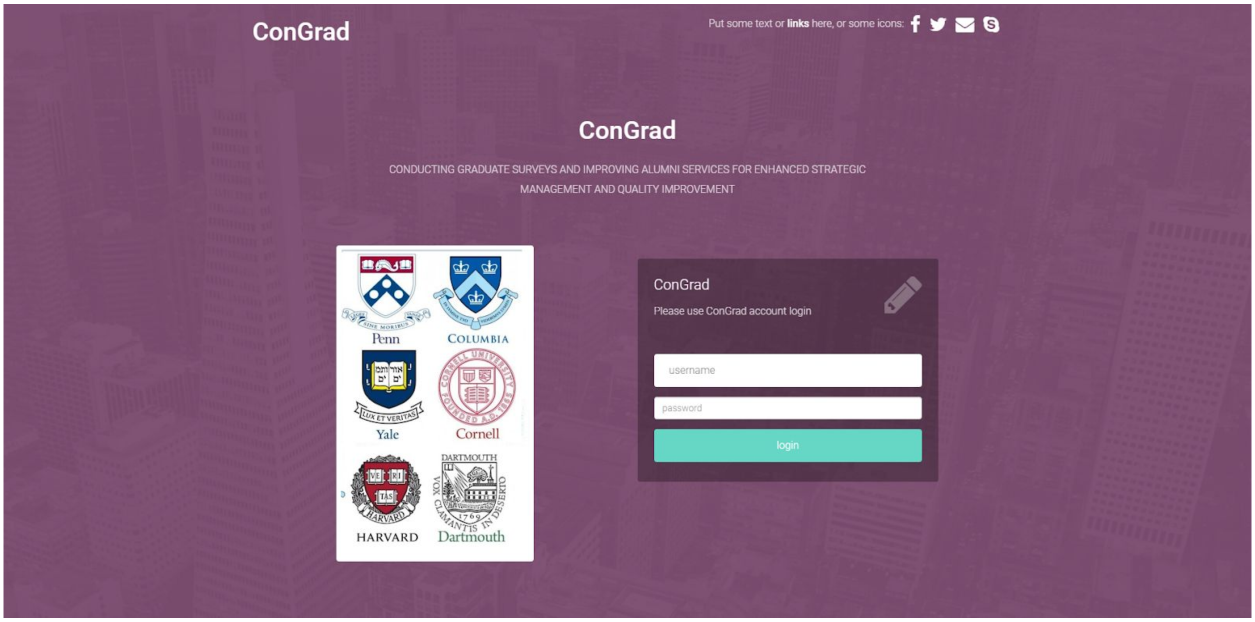
There is an entity Student which stores all the applicants' information we found in the data collection process. It stores made-up student ID, GPA, GRE scores for different sections, TOEFL score, a boolean called student publication to indicate whether the student had published any articles, work experience, and nationality of the student. To link students with the schools they applied and with the schools they graduated from, we designed two relational entities called Admit and Graduate. For relational entity Admit, it has two foreign keys which are student ID and School ID from entity Student and School. It also has a boolean called Accepted to indicate the decision made by schools, decision data to indicate when the student received the result, Accepted major to indicate which major the student applied and degree type to indicate which degree the student applied. The relational entity Admit has a many to many relationships since each student can apply for many schools and each school can have many applicants. Then there is a relational entity Graduate to indicate where the student graduated. It has only two foreign keys, student ID and school ID and many to one relationship since a student can only graduate from one school and each school can have many students graduate.

ER Diagram for ConGrad



3.3. UI Design

The login page will provide users with the function of registering and logging. This page allows users to get into the account management page which users can edit their own information such as GRE scores, GPA, undergraduate schools and other info for the backend algorithm to recommend graduate school programs.



ConGrad

Put some text or **links** here, or some icons:

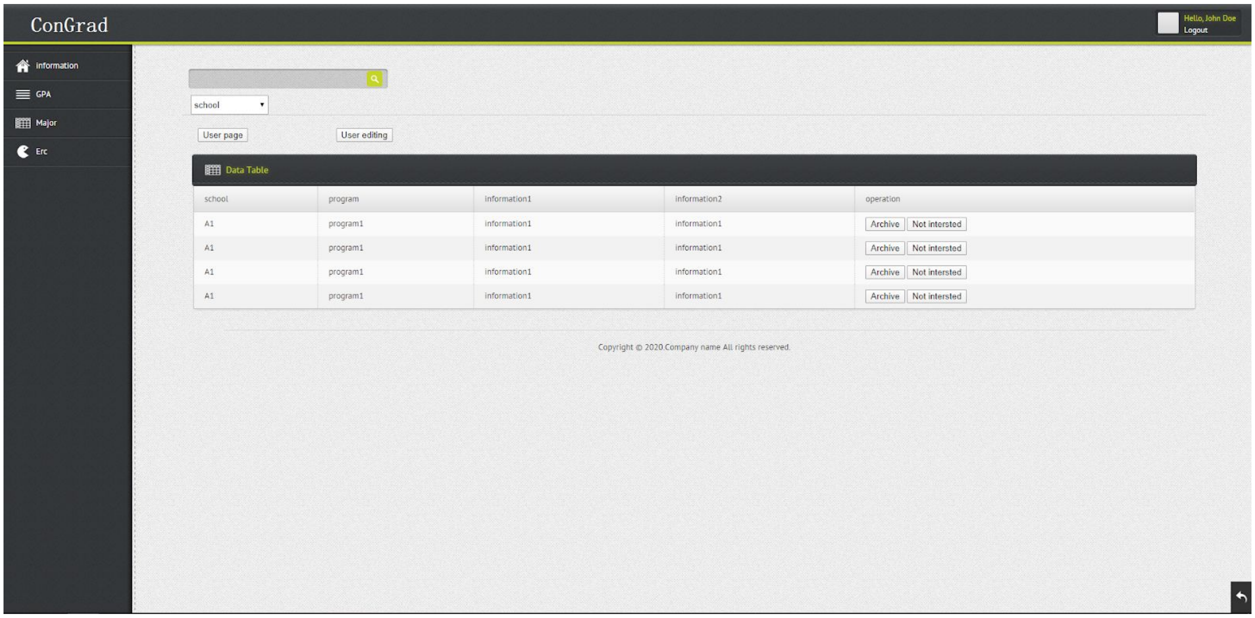
ConGrad

CONDUCTING GRADUATE SURVEYS AND IMPROVING ALUMNI SERVICES FOR ENHANCED STRATEGIC MANAGEMENT AND QUALITY IMPROVEMENT

ConGrad

Please use ConGrad account login

The homepage of ConGrad provides users with the function of showing the recommended programs based on the personal information(GRE scores, GPA, undergraduate schools) using back-end algorithm, users can edit the recommendation from the left side of the page using different types of categories(GPA, Major, etc), it will also provide user with a search box to manually search for programs. And it also allows the user to get to the account management page from the top right corner.



school	program	information1	information2	operation
A1	program1	information1	information1	Archive Not intersted
A1	program1	information1	information1	Archive Not intersted
A1	program1	information1	information1	Archive Not intersted
A1	program1	information1	information1	Archive Not intersted

Copyright © 2020.Company name All rights reserved.

3.4. Database Connectors

In order to utilize our database in Java, we developed a package called DBConnect to execute SQL statements. There are three main classes under DBConnect, which are called ProgramDBConnect, SchoolDBConnect and StudentDBConnect. Some typical methods include getAcceptedStudentGPA(school_id), getAllSchool(), and getAllProgramNames().

3.5. Recommendation Algorithm

The main recommendation method called Generate() is in the Generator class and under Generator package. It takes a student object as input and returns an arraylist of schools. At first, it calls the getAllSchool method to get all school information ordered by ranking and then calculate the average GPA, GRE score, Toefl score, working time and publication status of the previously admitted students of each school respectively. Then, it compares the input student's information with these average data and checks whether the input student has satisfied the requirements of each school. Of course, the student does not need to satisfy them all, but to get the school recommended by our algorithm, the student's data should not be much lower than average. Furthermore, If the student graduates from Top 50 colleges according to US News ranking, the requirements for getting recommendation will be less strict.

4. Project Completion Plan

Task	**Members in charge**	**Deadline**
Enlarge the dataset by collecting more information about past applications	Everyone	March 12nd
User page and information display page's frameworks ready	Tao, Hao, Zijun	March 19th
Account management page's framework ready	Tao, Hao, Zijun	March 21th
Code for the search information ready	Qi, Denglin	March 21st
Database and backend connection	Qi, Jieyu	March 23rd
Backend and frontend connection	Tao, Qi, Jieyu, Zijun, Hao, Denglin	March 23rd
Webpage for the users to update application progress ready	Tao, Hao, Zijun	March 26th

Progress report 2 submission	Everyone	March 27th
Functionality test	Jieyu, Zijun, Qi	March 27th
Web view beautification	Everyone	April 3rd
Final report draft	Everyone	April 10th
Poster Due	Everyone	April 15th
Final report ready	Everyone	April 19th
Final project submission	Everyone	April 20th

5. Member Contributions

Jieyu Ren:

- Co-designed system architecture
- Collected information about graduate schools, programs, and past applications
- Wrote Abstract, Proposal and Progress Report 1
- Constructed the database
- Inserted data into the database
- Wrote the database connectors
- Wrote the backend recommendation algorithm

Zijun Liu:

- Co-designed system architecture
- Collected information about graduate schools, programs, and past applications
- Wrote Abstract, Proposal and Progress Report 1
- Entered information into the database
- Web page UI mockup

- Code review and Testing

Hao Yang:

- Co-designed system architecture
- Collected information about admission information from previous years
- Code Review and Testing
- Designed and implemented the search page for the web system
- Inserted data into database
- Wrote abstract for the intersection, Progress Report 1 and Proposal.

Tao Huang

- Co-designed system architecture
- Collected information about admission information from previous years1
- Web page UI mockup
- Designed and implemented the login page for the web system
- Designed and implemented the search page for the web system
- Wrote Abstract Progress Report 1 and Proposal.

Qi Cheng

- Co-designed system architecture
- Collected information about graduate schools, programs, and past applications
- Designed database schema
- Inserted information into the database
- Wrote proposal, abstract for the intersection, and progress report 1

Denglin Zhang

- Wrote proposal
- Collected information about graduate schools, programs, and past applications