

# CS591L1 Final Project Report

Group Member: Hexuan Zhang, Nianyi Zhang

Github Link: [https://github.com/Helenazhx/cs591flask\\_sqlalchemy](https://github.com/Helenazhx/cs591flask_sqlalchemy)

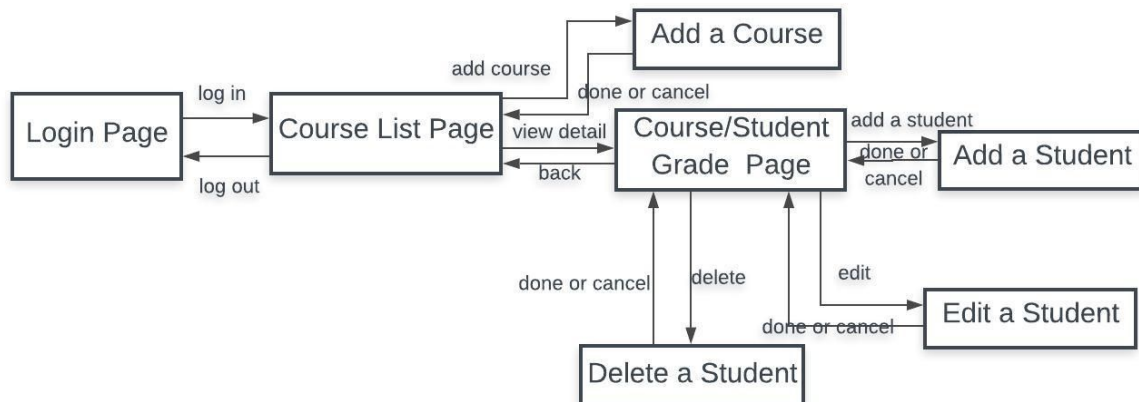
## 1. Web-Based Student Information Management Application

### 1.1 Introduction

For the final project, we build a webpage-based student information management application using existing embedded libraries. We already learned from lectures that sqlalchemy can be programmer-friendly that it makes us do the same thing as SQL statements without having to write SQL statements as strings. Instead, tables and schemas are represented as classes and objects in python, which are relatively easier for us to manipulate. On this basis, we intend to make a user-friendly application in which users can manipulate data easily on a user interface. Specifically, we use python as the programming language, flask as the web framework to build user interfaces and sqlalchemy as the logical support for the data schema and data manipulation. We want to explore the availability and advantages of using flask and sqlalchemy in building an application.

### 1.2 Architecture and schema

Architecture:



Data Schema: (yellow part indicate a many-to-many relation and it is not substantial)

Courses Table

Id	cname (String)	code (String)	seats (Int)	students
1	Network	CS655	80	

Students Table

id	sname (String)	snumber (String)	smail (String)	courses
1	Alan Bob	U111	<a href="mailto:alanbob@bu.edu">alanbob@bu.edu</a>	

Students/Courses Association Table

Association_id	sname(String)	cname(String)
1	Alan Bob	Network

Grades\_Record Table

id	sname(String)	cname(String)	grade(Int)
1	Alan Bob	Network	85

### 1.3 Usage

We provide instructions on how to run our application on GitHub as well as some screenshots. Having all the requirements fulfilled and you can enter the welcome page of our application. It requires you to login in. Here we set the user name as test and password as cs591. After login, you can see all the existing courses listed on the page. You can either add a course by click the button and enter the course information or access to specific course information by click on the course that you want to explore. If you click on a specific course to see detail information, you will see all the students who are attending this course and their names, student numbers and their grades on this course. You can also add a new student, delete a student or edit someone's information in this course.

## 2. Comparison with Java Implementation

We know that sqlalchemy uses the deep-embedding technique because it represents tables as classes in python and provides python objects such as a query for us to manipulate. In this project, we implemented this application by python, flask and sqlalchemy. We want to compare this method with java, Java Swing and MySQL implementation and list some similarities and differences.

### 2.1 Similarities:

Actually, in both implementations, there are deep embeddings used. In python, we use deep embedding in the backend as stated above. In java implementation, Java Swing is also a deep-embedding but in the front end. It is written in Java and provides objects like JFrame, JPanel, JButton and so on. Users can easily make use of them to design a user interface in java.

## 2.2 Differences:

Apparently, for us, python, flask, and sqlalchemy is a much easier way to implement it. It took us nearly 2 months to implement a similar system in Java and this project in python only cost us 2 to 3 days. This project is of only no more than 600 lines of code and the workload in java is 3 or 4 times of it. Although Java Swing is also a deep-embedding there is much redundant code when it comes to the graphic user interface in java. We have to listen to all kinds of events and enable many buttons. Instead in python, we build a webpage application and we write some HTML/CSS to implement the user interface part. Moreover, flask enables us to interact with HTML files easily. We use python decorator `@app.route` and `render_template` to bind a function to an HTML file and redirect the page. Flask and python allow us easily get the input from the webpage and pass them as parameters in between different HTML files.

## 2.3 Advantages and Limitations

Deep embedding provides us with much flexibility while programming. It allows us to make use of a new construct in the host language and these objects can be passed to or returned from functions. We can easily use `JFrame`, `JPanel`, and `JButton` in Java Swing to build a graphical interface and use the query in sqlalchemy without using SQL statements. But there are certain limitations. For example, while designing data schema in sqlalchemy we have to be very careful. We have to figure out whether it is a one-to-many relationship or many-to-many relationship and then define them. Queries should be handled very carefully too. It is a fact that SQL statements in database systems are compiled, optimized and executed very efficiently. However, if we transfer a huge amount of data from the database to python and do the query using python function, this could be a disaster because python is a slow interpreted language and much less efficient.