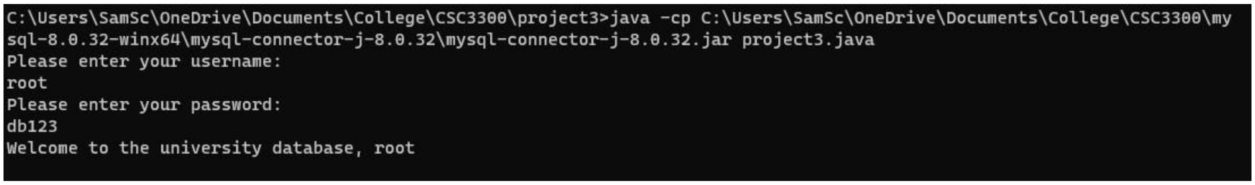
USER MANUAL FOR PROJECT 3

How to Build and Run the Application:

1. Install a java connector for MySQL using this link: https://dev.mysql.com/downloads/connector/j.
2. Make sure the MySQL server is running.
3. In your project directory, execute the following command: *javac <name>.java*
4. Execute the command using the .jar file from the connector downloaded in step 2. Replace <jar_path> with your pathway: *java -cp <jar_path> <name>.java*

Connecting to the Server

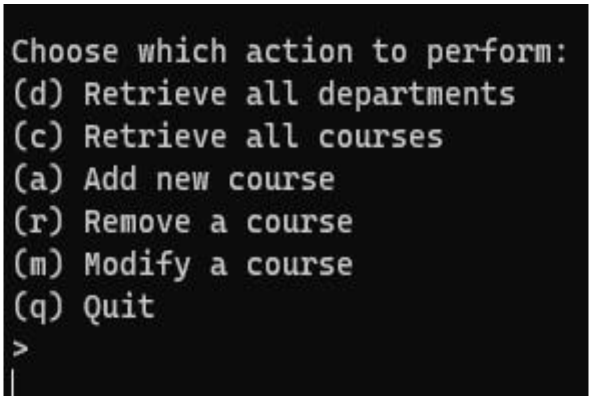
The application first asks the user for a username and a password. If the user inputs incorrect information, the connection fails, and the user is prompted to either re-input the information until they successfully connect or quit the program. If the connection is successful, the user is greeted.



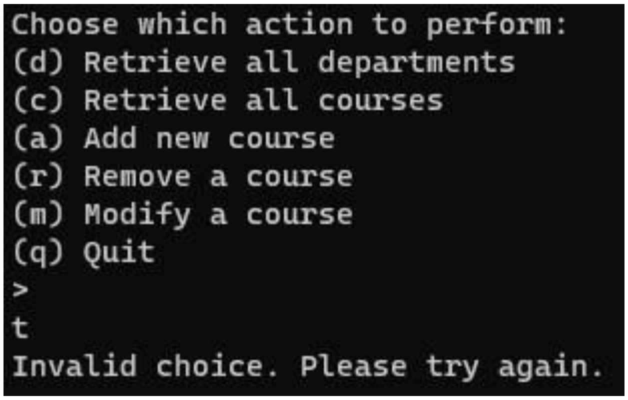
```
C:\Users\SamSc\OneDrive\Documents\College\CSC3300\project3>java -cp C:\Users\SamSc\OneDrive\Documents\College\CSC3300\my
sql-8.0.32-winx64\mysql-connector-j-8.0.32\mysql-connector-j-8.0.32.jar project3.java
Please enter your username:
root
Please enter your password:
db123
Welcome to the university database, root
```

Main Menu

The main menu gives the user 6 options. The user must input the corresponding upper- or lowercase letter for the action to work; if the user inputs anything other than one of the designated letters, the program prompts the user to try again.



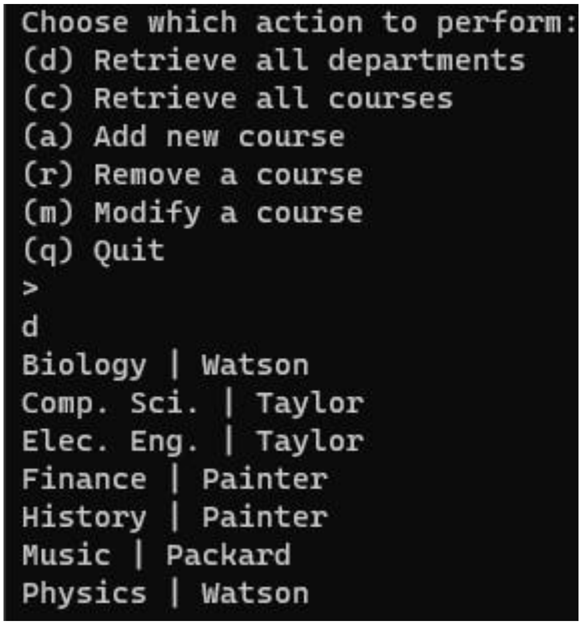
```
Choose which action to perform:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add new course
(r) Remove a course
(m) Modify a course
(q) Quit
>
```



```
Choose which action to perform:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add new course
(r) Remove a course
(m) Modify a course
(q) Quit
>
t
Invalid choice. Please try again.
```

(D) – Retrieving All Departments

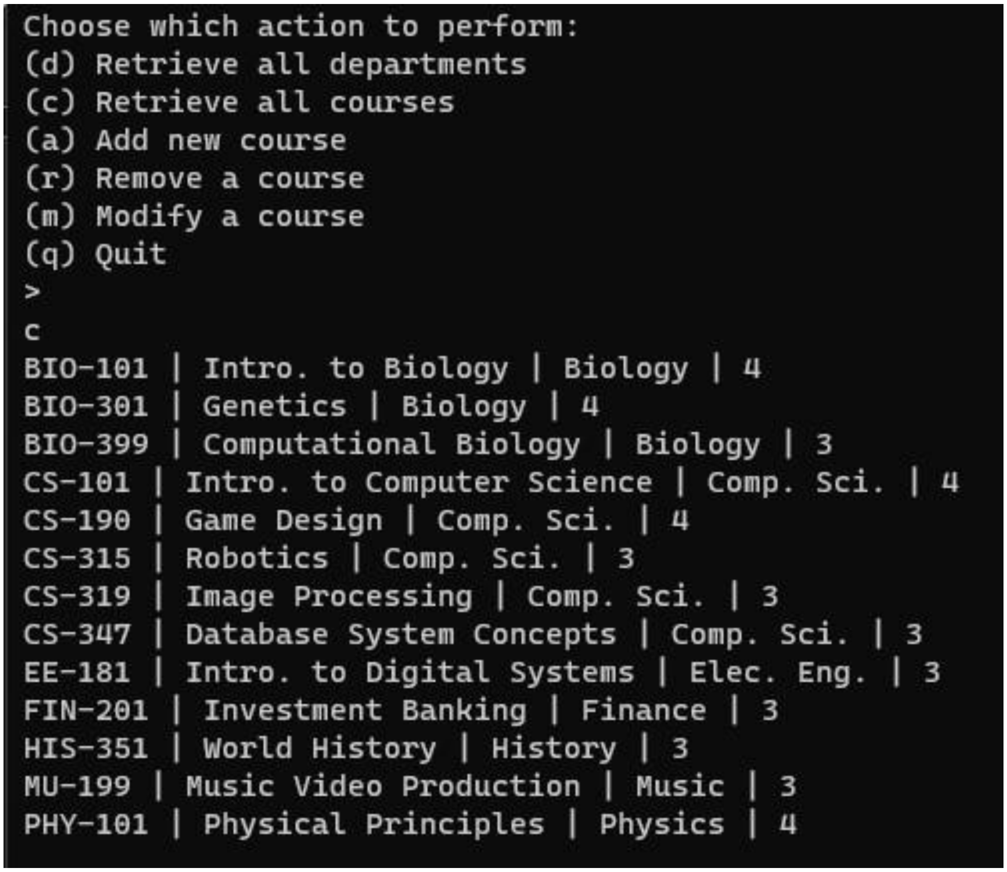
If the user chooses “D”, the program displays information related to the departments within the database, excluding any budgetary information. The user is not prompted to enter any other related information. The information is displayed in the following way: “department name | building name”.



```
Choose which action to perform:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add new course
(r) Remove a course
(m) Modify a course
(q) Quit
>
d
Biology | Watson
Comp. Sci. | Taylor
Elec. Eng. | Taylor
Finance | Painter
History | Painter
Music | Packard
Physics | Watson
```

(C) Retrieving All Courses

If the user chooses “C”, the program displays all information related to the courses within the database. The user is not prompted to enter any other related information. The information is displayed in the following way: “Course ID | Course Title | Department | credits”.



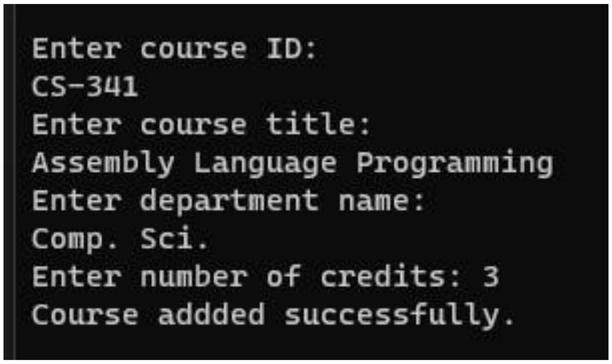
```
Choose which action to perform:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add new course
(r) Remove a course
(m) Modify a course
(q) Quit
>
c
BIO-101 | Intro. to Biology | Biology | 4
BIO-301 | Genetics | Biology | 4
BIO-399 | Computational Biology | Biology | 3
CS-101 | Intro. to Computer Science | Comp. Sci. | 4
CS-190 | Game Design | Comp. Sci. | 4
CS-315 | Robotics | Comp. Sci. | 3
CS-319 | Image Processing | Comp. Sci. | 3
CS-347 | Database System Concepts | Comp. Sci. | 3
EE-181 | Intro. to Digital Systems | Elec. Eng. | 3
FIN-201 | Investment Banking | Finance | 3
HIS-351 | World History | History | 3
MU-199 | Music Video Production | Music | 3
PHY-101 | Physical Principles | Physics | 4
```

(A) Adding A New Course

If the user chooses “A”, the program will prompt the user to input all related course information (a course ID, a title, an associated department, and credit number). None of this information can be empty and must be within the proper constraints affiliated with each value.

- A course ID must be of the form: 2-4 capital letters, followed by a dash, followed by 3 numbers. The ID must also not already exist in the system.
- A course title must be 50 characters or less.
- The department must be 20 characters or less and already exist in the database. - The credits must be an integer between 1 and 99.

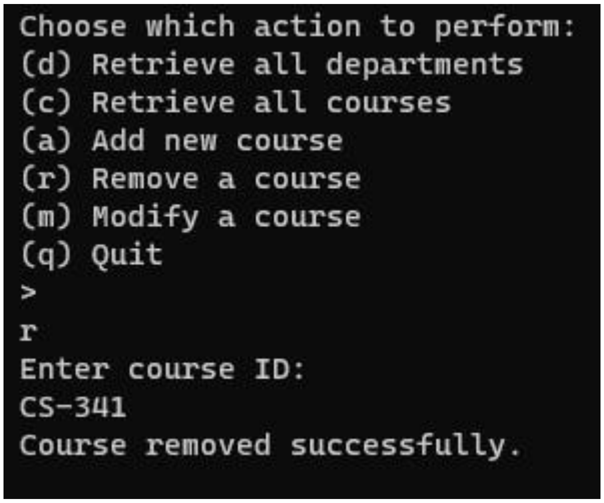
All functions associated with the user input correctly validates each user input to make sure the correct information is received.



```
Enter course ID:
CS-341
Enter course title:
Assembly Language Programming
Enter department name:
Comp. Sci.
Enter number of credits: 3
Course addded successfully.
```

(R) Removing A Course

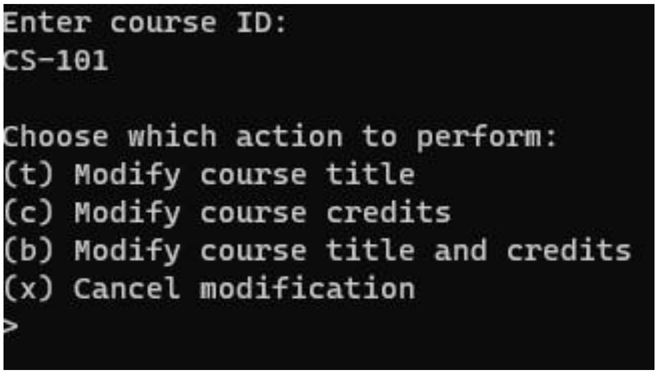
If a user chooses “R”, they will be prompted to enter the course ID they wish to delete. All constraints on the format of the course ID are the same in this case. Once the user inputs a valid Course ID, the program will remove it from the database.



```
Choose which action to perform:
(d) Retrieve all departments
(c) Retrieve all courses
(a) Add new course
(r) Remove a course
(m) Modify a course
(q) Quit
>
r
Enter course ID:
CS-341
Course removed successfully.
```

(M) Modifying a Course

If the user chooses “M”, they are prompted to enter the course ID that they wish to change; if the given ID doesn’t match, they are sent back to the main menu. If a valid course is input, the user is taken to another menu. This menu functions the same as the main menu in that it validates the input of the user until an upper- or lowercase version of one of the options is chosen.

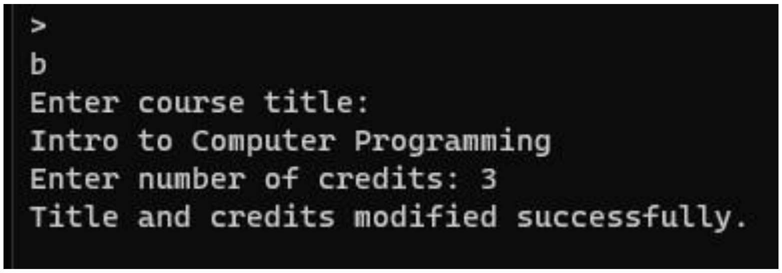


```
Enter course ID:
CS-101

Choose which action to perform:
(t) Modify course title
(c) Modify course credits
(b) Modify course title and credits
(x) Cancel modification
>
```

If the user chooses “T”, the user is asked to input the title; the same constraints are applied. If the user chooses “C”, they are asked to input the credit amount; the same constraints apply here as well. If they choose “B”, they are asked to input both the credit number and the course title.

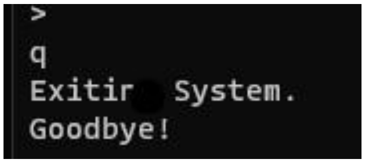
In the end, the program updates the course with the relevant information that the user input, and tells the user that the information has been successfully updated.



```
>
b
Enter course title:
Intro to Computer Programming
Enter number of credits: 3
Title and credits modified successfully.
```

(Q) Quitting the Program

IF the user inputs “Q”, the program terminates.



```
>
q
Exitir  System.
Goodbye!
```

Implemented Functionality:

- Use of Java Programming Language to connect to a MySQL database.
- User can retrieve information related to the departments and courses in the database.
- User can add and remove a course.
- User can update titles and credits of any course.
- Program validates user input.

- Command Line Interface is easy to navigate.
- Use of Prepared Statements for user input.

Assumptions:

- No user input can be empty.
- Course ID must be of the form: 2-4 capital letters followed by a dash followed by 3 digits.
- Course titles cannot be over 50 characters long.
- Department names cannot be over 20 characters long.
- Department names must already exist in the database.
- Credits must be a whole number between 1 and 99.
- Prepared Statements must be used when user inputs information for course modification.