

The DBMS we decided to use for our implementation was mongoDB, which is a NOSQL database. We chose mongoDB because it has an easy-to-use GUI, and is easily implemented into a python/flask webserver using pymongo, which is exactly what we did. All of our code is either in the python web server “app.py” or in the html/css documents in the “templates” folder. Since we used mongoDB and pymongo for our application, we wrote our queries using the mongoDB queries language (MQL) instead of SQL. However, I will represent these queries in SQL for the implementation section of this report. The following is all queries used in our web server in SQL format:

1 SQL

function: get_courselist()

Inputs: None

Outputs: List (@department_code, @course_number)

QUERY:

```
SELECT department_code, course_code FROM Course
```

function: find_dep_courses()

Inputs: @department_code

Outputs: List (@department_code, @course_number)

QUERY:

```
SELECT department_code, course_code
FROM Course AS c
WHERE c.department_code = @department_code
```

function: get_course()

Inputs: @department_code, @course_number

Outputs: @department_code, @department_name, @course_number, @course_name, @course_description, @course_hours, @prerequisites, @antirequisites, @corequisites, @notes

QUERY:

```
SELECT c.department_code, c.department_name, c.course_number, c.course_name,
       c.course_description, c.course_hours, c.prerequisites,
       c.antirequisites, c.corequisites, c.notes
FROM Course AS c
WHERE c.department_name = @department_name AND
       c.course_number = @course_number
```

function: get_programlist()

Inputs: None

Outputs: List (@faculty_name, @program_name)

QUERY:

```
SELECT faculty_name, program_name FROM Program
```

function: get_faculty_programs()

Inputs: @faculty_name

Outputs: List (@faculty_name, @program_name)

QUERY:

```
SELECT faculty_name, program_name
FROM Program AS p
WHERE @faculty_name = p.faculty_name
```

function: get_program_accreditations()

Inputs: @faculty_name, @program_name

Outputs: @faculty_name, @program_name, @accreditation, @core_courses,
@outside_faculty_courses, @outside_major_courses, @option_courses, @alternative_courses

QUERY:

```
SELECT faculty_name, program_name, accreditation, core_courses,
       outside_faculty_courses, outside_major_courses, option_courses,
       alternative_courses
FROM Program AS p
WHERE p.faculty_name = @faculty_name AND p.program_name = @program_name
```

function: add_previous_education()

Inputs: @ucid, @previous_education

Outputs: none

QUERY:

```
INSERT INTO PreviousEducation (@ucid, @previous_education)
```

function: add_uofc_course()

Inputs: @ucid, @department_code, @course_code, @grade, @season, @year

Outputs: none

QUERY:

```
INSERT INTO StudentTakesCourse (@ucid, @department_code,
                                @course_code, @grade, @season, @year)
```

function: display_past_uofc_courses()

Inputs: @ucid

Outputs: List (@department_code, @course_code, @grade, @season, @year)

QUERY:

```
SELECT c.department_code, c.course_code, c.grade, c.season, c.year
FROM StudentTakesCourse as c
WHERE c.ucid = @ucid
```

function: delete_course()

Inputs: @ucid, @department_code, @course_code

Outputs: none

QUERY:

```
DELETE FROM StudentTakesCourse AS c
WHERE c.ucid = @ucid AND c.department_code = @department_code AND
      c.course_code = @course_code
```

function: make_schedule()

Inputs: @ucid, @schedule_name

Outputs: @ucid, @schedule_name

QUERY:

```
INSERT INTO Schedule (@ucid, @schedule_name)
```

function: display_schedule()

Inputs: @ucid, @schedule_name

Outputs: List(@department_code, @course_code,
@section_number, @season, @year)

QUERY:

```
SELECT s.department_code, s.course_code, s.section_number
      s.season, s.year
FROM CoursesForSemester AS s
WHERE s.ucid = @ucid AND s.schedule_name = @schedule_name
```

function: add_course_to_schedule()

Inputs: @department_code, @course_code, @section_number,
@season, @year, @ucid, @schedule_name

Outputs: none

QUERY:

```
INSERT INTO CoursesForSemester (@ucid, @schedule_name, @department_code,
                                @course_code, @section_number, @season, @year
```

function: delete_course_from_schedule()

Inputs: @department_code, @course_code, @section_number,
@ucid, @schedule_name

Outputs: none

QUERY:

```
DELETE FROM CoursesForSemester (@ucid, @schedule_name, @department_code,  
                                @course_code, @section_number)
```