

HIPO Functions list

Braydon Pacheco,

March 26, 2021

1 Course List

function: Display Course List

Inputs: None

Outputs: List (*@Course_Name, @Course_Code*)

Pseudocode:

Connect to the database

Query = SELECT Course_Name, Course_Code FROM Course

Parse Query

Execute Query

Close connection to the database

function: Get Course Description

Inputs: *@Course_Name, @Course_Code*

Outputs: *@Units, @Description, @Faculty_Name*

Pseudocode:

Connect to the database

Query = SELECT c.Course_Name, c.Course_Code, c.Units, c.Description, c.Faculty_Name
FROM Course AS c WHERE c.Course_Name = @Course_Name AND c.Course_Code = @Course_Code

Parse Query

Execute Query

Close connection to the database

2 Schedule

function: Generate Schedule By Program

Inputs: *@UCID, @Schedule_Name, @Program_Name, @Faculty_Name*

Outputs: List (*@Course_Name, @Course_Code, @Section_Number, @Building,
@RoomNumber, @Days_Of_Week, @Time_of_Day, @Professor_Name*)

Pseudocode:

```

Connect to the database
Query = SELECT c.Course_Name, c.Course_Code FROM Program_Requirments AS c
        WHERE c.Program_Name = @Program_Name
Parse Query
Execute Query

```

```

For each course_name and course_code {
    If (!Student already took the course) {
        Find Course_instance in database that matches name and code
        If (Course_instance does not conflict with schedule)
            Add course instance to schedule
        Else
            Look for other course instance
    }
}

```

```

Query = SELECT (*) FROM Courses_for_semester AS c
        WHERE c.Schedule_Name = @Schedule_Name
Parse Query
Execute Query
Close Connection to the database

```

Display generated schedule to the user in GUI.

function: Delete Schedule from Database

Inputs: @UCID, @Schedule_Name

Outputs: None

Pseudocode:

```

Connect to the database

```

```

Query = DELETE FROM courses_for_semester AS c
        WHERE c.UCID = @UCID AND c.Schedule_name = @Schedule_name

```

```

Parse Query
Execute Query
Close connection to the database

```

function: Add Course to Schedule

Inputs: @UCID, @Schedule_Name, @Season, @Year, @Course_Name,
@Course_Code, @Section_Number

Outputs: None

Pseudocode:

```

Connect to the database

```

```
Query = INSERT INTO courses_for_semester(@Season, @Year, @Schedule_Name, @Course_Name,
    @Course_Code, @Section_Number)
```

Parse Query

Execute Query

Close connection to the database

function: Remove Course from Schedule

Inputs: @UCID, @Schedule_Name, @Season, @Year, @Course_Name,
@Course_Code, @Section_Number

Outputs: None

Pseudocode:

Connect to the database

```
Query = DELETE FROM courses_for_semester AS c
    WHERE c.Schedule_Name = @Schedule_Name AND c.Season = @Season AND
    c.Year = @Year AND c.Course_Name = @Course_Name AND
    c.Course_Code = @Course_Code AND c.Section_Number = @Section_Number AND
```

Parse Query

Execute Query

Close connection to the database

3 Programs

function: Display Program List

Inputs: None

Outputs: List (@Program_Name, @Faculty_Name, @Major, @Minor)

Pseudocode:

Connect to the database

```
Query = SELECT Program_Name, Faculty_Name, Major, Minor FROM Program
```

Parse Query

Execute Query

Close connection to the database

function: Show program requirements

Inputs: @Program_Name

Outputs: List (@Course_Name, @Course_Code)

Pseudocode:

Connect to the database

```
Query = SELECT p.Course_Name, p.Course_Code
    FROM Program_Requirements WHERE p.Program_Name = @Program_Name
```

Parse Query

Execute Query

Close connection to the database

4 Previous Education

function: Get Courses From UofC

Inputs: @UCID

Outputs: None

Pseudocode:

Apply web-scraping algorithm on @UCID to get list of UofC courses from UofC website to get inputs (@Course_Name, @Course_Code, @Grade, @Season, @Year)

```
For each course {
    Connect to the database {
        Query = INSERT INTO Student_Takes_Course(@UCID, @Course_Name,
            @Course_Code, @Grade, @Season, @Year)
        Parse Query
        Execute Query
        Close connection to the database
    }
}
```

function: Add UofC Course

Inputs: @UCID, @Course_Name, @Course_Code, @Grade, @Season, @Year

Outputs: None

Pseudocode:

```
Connect to the database
Query = INSERT INTO Student_Takes_Course(@UCID, @Course\_Name,
    @Course\_Code, @Grade, @Season, @Year)
Parse Query
Execute Query
Close connection to the database
```

function: Delete UofC Course

Inputs: @UCID, @Course_Name, @Course_Code, @Season, @Year

Outputs: None

Pseudocode:

```
Connect to the database
Query = DELETE FROM Student_Takes_Course AS c
    WHERE c.UCID = @UCID AND c.Course_Name = @Course_Name AND
        c.Course_Code = @Course_Code AND c.Season = @Season AND
        c.Year = @Year
Parse Query
Execute Query
Close connection to the database
```

function: Add non-UofC Education

Inputs: @UCID, @Previous_Education

Outputs: None

Pseudocode:

Connect to the database

Query = INSERT INTO Previous_Education(@UCID, @Previous_Education)

Parse Query

Execute Query

Close connection to the database

function: Display Courses Taken

Inputs: @UCID

Outputs: @Course_Name, @Course_Code

Pseudocode:

Connect to the database

Query = SELECT c.Course_Name, c.Course_Code

FROM Student_Takes_Course AS c WHERE c.UCID = @UCID

Parse Query

Execute Query

Close connection to the database