**Project #2**

**Instructions:**

Follow the instructions below very careful. Points will be deducted if instructions are not followed.

**Project Options**

There are 2 options for Project #2 and you need to choose only one:

1. System Development, and
2. Research Paper.

**Evaluation**

*Evaluation for Option 1 (20%):*

* Functionality and Database Structure/10%
* User Interface/5%
* Tech. report/5% (This includes the Readme file, *see details in Option 1 project*)

*Evaluation for Option 2 (20%): Research Paper/20%*

**Due Date**

All deliverables for Project #2 are due on April 2. For details on deliverables see each option below.

**Project #2 – Option 1 (System Development)**

“Stephanos Kung Fu Club” (SKF) needs a database. SKF is a martial arts school with hundreds of students. SKF has these requirements:

* There is only one instructor who teaches all classes
* Students are given a student number when they join the school. This is stored along with their name, date of birth, the date they joined the school, their mobile number, email, and address.
* The school wants to track also the students’ parents (Mother and/or Father). Assume that students live with their parents. Not all students have parents (some students might be adults so need to keep track of their parents). We need to store also their mobile phone and email address. Parents may also be students.
* Students pay fees for membership, tests, purchasing products, etc. The system must be able to keep track of all money received from students. The system should be able to show this information for any given date, e.g. by month or within a range such as from Jan 1-Jan 20th.
* A class is offered for a specific level at a specific time, day of the week. For example, one class taught on Mondays at 5:00 pm is an intermediate-level class. Another class taught on Mondays at 6:00 pm is a beginner-level class. A third class taught on Tuesdays and Thursdays at 5:00 pm is an advanced-level class, and so on. There are different levels of classes. These provided here are only as an example.
* Students may attend any class at any level during each week so there is no expectation that any particular student will attend any particular class session. Therefore, the actual attendance of students at each individual class meeting must be tracked.
* Each student holds a rank in the martial arts. The rank name, belt color, and rank requirements are stored. Each rank will have numerous rank requirements. Each requirement is considered a requirement just for the rank at which the requirement is introduced. Every requirement is associated with a particular rank. All ranks except white belt have at least one requirement.
* A given rank may be held by many students. While it is customary to think of a student as having a single rank, it is necessary to track each student’s progress through the ranks. Therefore, every rank that a student attains is kept in the system. New students joining the school are automatically given a white belt rank. The date that a student is awarded each rank should be kept in the system. All ranks have at least one student that has achieved that rank at some time.
* The system should be able to produce dynamic (not static) reports. For example, the admin may want to see all current (active) students, students of a particular rank (e.g. red belt), Students who joined in a specific year and are black belt, etc.

You are asked to provide:

1. A *Readme* file in Word or PDF format with the following information:

* the names and student numbers of your team members
* how to run the system
* provide usernames/passwords (if any)
* A dependency diagram in 3NF
* A complete Crow’s Foot ERD or a UML Class Diagram

1. A complete system using any programming language for the interface and any DBMS of your choice. This includes all system files including source code.

A team representative must upload on Blackboard the above in a ZIP format. In the ZIP file, there should be the **Readme** file containing the (1) from above and one folder containing the (2) from above. The ZIP file should be saved with the name of the team representative (LastName\_FirstName), e.g. **Smith\_John**

Teams should be from 1-4 people.

**Project #2 – Option 2 (Research Paper)**

**Group Size**

The Project will be undertaken by a group of no more than 4 students. Individual projects are also allowed.

**Topics**

The Project will focus on any topic in the area of Databases. Some ideas include but not limited are in big data, data mining, data analysis, data visualization and interaction, NoSQL, etc.

**Deliverables**

Each Team Representative will submit through Blackboard an original research paper based on the ACM Template [ <http://www.acm.org/publications/proceedings-template> ].

The paper should include at least 15 references (0-2 textbooks, 15 or more journal and conference articles, *no websites*). References should be no older than year 2010.

**Paper Format**

The paper should be between 6-10 pages long. You should use the ACM Word template for proper formatting. Sample papers are also provided so you can understand how your paper should be organized and look like.

**Articles Search**

The Leddy library is a good place to search for related articles. You can get free access to the online databases using your *uwin* account. <http://leddy.uwindsor.ca/computer-science>

I recommend:

* ACM Digital Library
* Lecture Notes in Computer Science
* IEEE Xplore
* Google Scholar

**Academic Integrity**

Students, be aware of copyright issues and avoid plagiarism. <http://www1.uwindsor.ca/academicintegrity/avoiding-plagiarism>

All you work must be original and properly cited. For citation help visit the Academic Integrity Office website at: <http://www1.uwindsor.ca/academicintegrity/citation-help>

Read *Bylaw 31: Academic Integrity:* <http://www.uwindsor.ca/secretariat/sites/uwindsor.ca.secretariat/files/bylaw_31_-_academic_integrity_amended_150313.pdf>

The instructor reserves the right to give a grade of 0 and/or report for disciplinary action to any student who has provided any plagiarized text as part of this project.