



## COMS3002: Software Engineering - III Enhanced Project Titles and Descriptions

August 6, 2018

# 1 Students Marks/Record Management System

An academic unit such as the School of Computer Science, wishes to implement a software system to manage the tentative (Unofficial) students marks as students complete assignments, tests, laboratory work, etc., through out the year. The general idea here is that for all the courses offered by the school (call this SchXYZ), the course coordinators for a particular course, say CrsABC, enters the marks obtained by the students for the each assessment done - tests, assignments, etc., into the system. Each student, at any time, can retrieve the marks accumulated so far, for each course. In this manner every student can track his/her tentative progress through out the year. This helps in catching errors in recorded marks as early as possible and further it alerts students that are failing to invest more time into their learning session or seek early remedial help.

At the end of the semester, or the end of year, the school administrative officer retrieves, for each course, in a spread-sheet format, the relevant summaries of marks and reports this formally to the **official marks system** of the faculty after this has been first certified by the course coordinator and then approved by the Head of School. The database is retained for an operational window of ten years (10 yrs). This means the first year marks/records of the courses taken by a student will still be available for ten years, after which it will be off-loaded to an archival system.

The front-end of the system involves interacting, over the Internet, with the system by:

**The Course Coordinator:** He/She interacts with the system to:

- Register himself/herself,
- Adds the various assessment method for the course, and the weighting for each assessment,
- Enters the students' marks for each assessment in a user-friendly interface,
- Display or print out the table of students and their marks
- Generate a summary statistics of the performance of the students - maximum, minimum, average and standard deviation or variants of each assessment.
- The projected pass rate based on the assessment marks accumulated by the students in the class so far.

**The School Administrator:** All the functions here are also executable by the head of school or his/her designate. The School Administrator should be able to:

- Register himself/herself,
- Register students for their courses
- Display or print out the table of students and their marks;
- Generate a summary statistics of the performance of the students.
- Generate a comparative chart of the assessment marks of selected courses being taken students of a particular group, e.g., charts of COMS2004, COMS2001 and COMS2003, ordered by student numbers.
- Histogram of assessment marks of all courses taken by a specific student.
- Any recorded offences (e.g., plagiarism) for a student.
- The performances in the same course across different years may be compared.

**A Student:** A student can retrieve information of himself/herself alone. A student can interact with the system to:

- Display assessment mark for a course and the statistics for that assessment.
- Display assessment marks for all courses;
- Based on the current assessment mark, give what performance goals are needed to pass the course.

The back-end of the system involves following:

**Database Administrator (DBA):** He/She is responsible for setting up the database, managing and implementing the relevant API functionalities for accessing the database from a client. The DBA is also responsible for backups and re-configuring the databases, etc. In particular, transactions and access logs are also secured persistently and possibly also on remote machine, for auditing.

**System Administrator** This person exercising this role is responsible for systems' (hardware, networks and software) configurations and also ensuring that the system's security is functioning and is fully monitored.

You may also consider the use of R or Matlab for carrying out the data analytics.

## 2 Examiners' Board Meeting: Synchronizing Marks Reviewing Process with Laptops, Tablets, etc

At the end of each year the academic staff members of the Science Faculty meet to approve, the pass, fail, exclusion, etc status of each student. Lets call this gathering, *the Examiners' Board Meeting*. Participant from the various schools of the faculty, arrive at this meeting with a document (a very fat document) of the names and tentative grades for the year, of each student in the faculty. The Registrar and the Dean lead the discussion for each student, to approve or modify the tentative status of the student recorded in the document. Where a student meets the requirement for a pass, there is normally no objection and the vote is typically unanimous. When a student is a boarder line case, some discussion ensues to either fail or pass the student depending on the justification given by the School's representative. For a student that appears to have failed, his/her status may change depending on the justifiable reasons forwarded by the school's representative. For a change of status of a student the entire faculty votes approve or disapprove the acceptance or change of status of the student concerned.

As each student comes up for discussion, a screen display of the student and his/her grades is done. However, the participants synchronise the name in question with the file/document being held. Voting is consequently done by show of hands. The main objective in this project is to replace the paper held document of students' names and the eventual voting to be done by a mobile device such as a laptop, tablet, smart-phone, phablet, etc. While the overall display can still be done over a large screen projector, participants can view the same name under discussion on their mobile device. After arguments are presented to keep or change the status of a candidate, voting can be done on the mobile devices. The result of the voting can be immediately tallied online, recorded in the database seen right away by the entire participants.

Feature considerations of this projects include:

- i. An architecture over the Internet;
- ii. A back-end database on an Web-server
- iii. A front-end App or Browser Interface that is synchronised with the display being shown by the Registrar and the Dean.
- iv. Tight security issues to be addressed by system's administrator.

### 3 Fast Food on Wheels Delivery System

A client is requesting the implementation of custom tailored fast food delivery software management system to run on a server (or on cloud computing service). The kinds of fast food normally to be served include, Pizza, Hamburgers, Hot-dogs, etc. You can safely assume that the restaurant has preset limited types of fast foods, except that clients can choose, add or refuse, condiments and dressings.

The system runs a back-end service which manages all the essential databases, e.g., available dishes, customers and customer orders, stocks of ingredients for the dishes, employees - part-timers and full-timers, managers, suppliers of stocks, etc. The front-end services feature interfaces for interacting with the management systems. These include features for:

**Customers:** Customers can make orders to be delivered, etc. A unique feature of the system is that orders may include special requests to modify the content of the preset orders, e.g., no-salt, no-sugar, etc. The possible devices for interacting with the system include PCs (desktops), Laptops, tablets, mobile-phones, etc.

**Restaurant Manager:** The manager interacts with the system to manage:

- i.) Stock Inventory
- ii.) Orders
- iii.) Customer database
- iv.) Track delivery
- v.) Accounting/Financial Administration
- vi.) duties, off-days, leaves, etc.
- vii.) Generation of reports - Analytics

**System Administrator:** A system administrator is responsible for all computer related services such as setting-backups of databases, stopping/restarting system services, general system maintenance, etc.

The system should not be limited only to the features above, but include all features that are essential for a professionally managed restaurant. For example, for multiple orders, can a schedule, for delivery of all orders, be mapped for the most cost effect delivery?

## 4 A Dental Clinic Management System

A software management system is required for managing the services of a small dental clinic run collectively by group of about 1 - 5 dentists, each with his/her own speciality. Dental services are available for both routine, urgent care and referrals. A dentist can also serve as a primary dentist of a patients. Patients can drop in routinely, in which case they are triage or can make appointments. The clinic offers a number of dental routine services but no major speciality services like major surgery.

The software management system is expected to provide:

- i. Dentists' schedules
- ii. Patients' appointments, record keeping, prescriptions;
- iii. Laboratory Management,
- iv. Patient's Billing,
- v. Accounting and Financial Management,
- vi. etc.

One can develop and implement a prototype system from scratch. Alternatively one can custom tailor a software systems for Health-Care called *GnuHealth* for this purpose. The Website for this software is <http://health.gnu.org/>. It is based on the *Tryton* three-tier high-level general purpose application development framework. The Website for *Tryton* is <http://www.tryton.org/>. The front-ends and back-end services to be provided are easily configured from the *GnuHealth* installation process.

## 5 Work-flow Infrastructure for Postgraduate Application Approval

The School of Computer Science and Applied Mathematics wishes to conduct the approval process of applicants to their graduate program electronically using an on-line work-flow. The current manual process proceeds as follows:

- Step 1:** The postgraduate officer (PGO) receives, via email, a notification that there are completed applications ready for downloading and processing. After processing, the outcome is a decision to *Accept* or *Reject* the applicant with some explanation.
- Step 2:** The PGO logs in to the Students Information Management System (SIMS), and downloads all submitted applications onto her computer and prints each and every one. This is big paper work. For each new applicant a file is opened for the applicant.
- Step 3:** Depending on the program that the postgraduate application is for, e.g., Honours, MSc By research, PhD (always by research), etc., the application is forwarded manually, either to a research group, or an identified supervisor.
- Step 4:** The application is returned to the PG Officer who then forwards it to the PG Coordinator (PGC). Based on the recommendation of the evaluator and the final evaluation of the PGC who ensures that all the faculty rules and regulations and the standing orders of Computer Science and Applied Mathematics for acceptance are met, he/she signs of the application as *Accept* or *Reject* and sends the application back to the PGO.
- Step 5:** The PGO updates the application on SIMS and notifies, via email, the Postgraduate faculty officer (PGFO).
- Step 6:** The postgraduate faculty officer then sends to the applicant a letter notifying him/her whether he/she has been accepted. A copy of the letter is also forwarded to the school's (PGO) who then prints a copy and inserts it in the students file.

The exercise of this project is to develop an on-line systems that limits the paper work in the work-flow of the activities conducted in processing an application. Work on this project should begin by consulting with the Postgraduate Officer of the School to have a thorough understanding of the work-flow carried out in processing an application.

## 6 Courses Recommender for Postgraduate Students in Computer Science

It is becoming more and more difficult, particularly for students entering graduate schools, to make decisions on courses that subsequently impact on their successful completion of graduate studies. Often, students have to choose from a number of electives in their specific programmes. The main objective is to develop a prototype system that tries to recommend elective courses to students based on what other similar students have taken. The system then tries recommends courses and specifies expected grades for these courses. Accordingly, the student may take a course that is recommended by the system with an acceptable grade. For the system to be viable, it should apply *Data Mining* techniques, e.g., association rules of past-students, their, their grades, clustering techniques, etc. To recommend courses.



## 7 Employer-Student Job Matching System

Students get frustrated, in job seeking after graduation. The main objective of this project is to implement a prototype system for matching employers with job openings to prospective candidates with related skills. The system only matches prospective candidates to employers up to the interview stage. The system is expected to run over the Web.

Job seeker registers himself/herself over the Web and completes a form equivalent to what may be ideally on a Curriculum Vita (CV). Factors that may be asked from the employees and considered for matching purposes include:

- Qualifications and subject areas
- Data of birth,
- Nationality
- Work Experience
- Projects
- Marital Status
- etc.

Factors that may be asked from the employers and considered for matching purposes include:

- Job descriptions
- Qualifications required to be carried out on the job
- Age range of applicant,
- Nationality:
- Minimum Work Experience
- The time that the employee can begin,
- etc.

The system evaluates some metric of the applicant and also the metric as specified by the employer. A potential match occurs if the job-seekers measured metric falls within some percentage interval of the employer's metric for the job description. A recommendation is then made to the employer for a possible meeting for an interview.