School Management System (SMS) App

CMPT 661 Project Phase – WebApp Design and Implementation

The project submission is due by midnight Sunday 3rd December.

1. Requirements

You are requested to design and implement the School Management System (SMS) Web App for managing various school processes including admission, registration, grading and absence. The application will allow the school staff and parents to follow-up the progress of their kids and help teachers engage parents and easily communicate with them.

The main SMS modules to be designed and developed are described below.

All modules include **Login** to allow the user (i.e., *Principal*, *Teacher* and *Parent*) to login to use the application.

1.1. Module 1 - Admissions Management

This module should support the following use cases:

- **Applicant Registration:** Allows applicants (typically parents) to register to be able to submit an Admission Application for one or many students to join the school. The registration should include the following minimum details: Father first name and last name, National Id, Home Phone, Mobile, Email, Occupation, Name of Employer (same details should be entered for the mother). The applicant can enter a password to be able to login later using their email and their selected password.
- **Admission Application**: Once registered, the applicant (i.e., parent) can submit an Admission Application for one or many students. For each application the student details must be entered including: First name, Last name, Date of Birth, Gender, Current School Grade, Grade applying for, Name of Current School. The applicant can also enter comments such as the fact that they are applying from overseas. Note that each student needs a separate application. The Admission Application should allow attaching documents such as previous school reports. Once submitted, the initial application status should be 'Submitted' and the system assigns it a unique Student Identifier. The application should be associated with the academic year flagged as open for admission (typically the next Academic Year).
- Update **Admission Application**: Upon login the system should show the list of available applications so that the applicant can either view (e.g., check Admission Application status), update or withdraw an application. If an application is withdrawn, then its status should be change to 'Withdrawn' and the Principal and the Applicant should be notified.

  The applicant can also update and resubmit a withdrawn or rejected application. In such case, upon submission it should be auto-assigned to the academic year flagged as open for admission (not the initial academic year at the time it was first submitted).

The Applicant can view an application details including attachment and notes that flagged as visible to applicants.

Note that applications with status Accepted, Rejected or Waiting list can longer be updated.

- The Principal should have the ability to perform the following:
  - Get the list of applications with the ability to filter by status (including get all) and by academic year (by default it should be set to the academic year configured with 'open for admission').
  - View an application details including attachment and notes o Update an application details including:
    - ✦ Change the application status to Pending, Accepted, Rejected, Withdrawn, or Waiting list.
    - ✦ Attach/Delete an attachment
    - ✦ Add/Update an Admission Test and its date. When an Admission Test is entered or updated then the applicant should be notified.
    - ✦ Add/Update Admission a Test Result and Comments (there could be many test results per application). When an Admission Test results are entered/updated then the applicant should be notified.
    - ✦ The applicant & coordinator should be able to view the test results associate with an application.
    - ✦ Add/Update a note such as a summary of conversation with parents, summary of discussion of the admission committee. The coordinator can decide whether the added/updated note can be visible to the applicants. By default notes are visible only to the Principal.
    - ✦ View previous notes attached to an application.
    - ✦ Generate various reports such as the number of applications per grade and by application status (e.g., for particular academic year for each grade list the number of received applications, the accepted, rejected, pending, waiting list, withdrawn). Comparative reports of admissions for the last 3 academic years with graphs.

Note that an applicant can have one or many students. Each student can have one or many admission applications. A student can have of the following admission status: pending, accepted, rejected, withdrawn or waiting list. Student has first admission academic year property.

1.2. Module 2 - Registration Management

This module allows the Principal and her team to achieve the following use cases:

- Record a student registration: registration date, academic year, the amount paid (to book a seat) and payment mode (cash, bank card, credit card, cheque, direct bank deposit).
- Cancel a registration (for a student who decided to leave the school or was sacked from the school).
- Get pending registrations (admitted or current students who did not register yet).
- Send pending registration reminders.
- Record tuition and registration fees received from the parents
- Get pending payments

- Send reminders about pending payments
- Manage students taking the school bus and the school bus payments
- Generate various financial reports such as received payments during a date range (by default for the current month).

Note that a student has Last Registered academic year property, Registration Status: current, moved

1.3. Module 3 - Attendance Tracking

The Attendance Tracking module records student and teacher absences including leaves approved by the school. Generate reports such as students average monthly absences per grade level. Teachers absence report.

1.4. Module 4 - Behavior Management

Using this module, Teachers/Principal document and track student disciplinary incidents, maintain related records to help improve discipline by ensuring that students are held accountable for their actions. Record information such as:

○ Type of incident ○ Location and time ○ Student and staff involved

○ Grade level ○ Teacher remarks

○ Interventions, follow-up actions / penalties

○ Notifications to parents

Notify parents of infractions. Make information available online for students, parents and staff. Instantly access any student's complete disciplinary history when speaking with parents.

Produce statistical reports including statistical analysis of discipline data to track trends.

1.5. Module 5 – Sections and Schedule Management

This module provides use cases to maintain the following structure:

- Each Grade Level has 1 or many sections. Add remove sections.
- Ability to copy sections and schedule setup from previous year then modify it.
- Each Grade Level has many courses
- Each section has a set of courses scheduled at a particular time and classroom. Section-course has an assigned teacher. A teacher can teach one or many section-course.
- Manage grade levels, sections, courses per grade level. Assign teaching to sections.
- Assign students to sections
- **Integration with Google Classroom**: Create classes, enroll students and synch classes and students.

1.6. Module 6 - Grade Center

This module allows the following use cases to help in measuring and communicating achievements of individual students:

- Add/update an assessment activity:
    - ▪ Select the Course to do the grading for.

- Enter the assessment activity details: a type (e.g., Homework, Midterm, Final Exam, Quiz, Project, Report, Presentation), a title and a weight.

- Add/Update Grading for an assessment activity of a particular course section:
  - Get the list of students to be graded.
  - Assign/update the grade assigned to each student. If grades were previously entered for the selected section then the system should allow the instructor to change them otherwise an empty grading sheet should be provided.
  - Enter optional overall comment for each graded assignment ▪ Store the grading details and compute the overall course grade.
  - Navigate backward and forward through the list of students to add/update their grading or the instructor can select a particular student to add/update their grading (without leaving the grading view).
- Allow parents and students to view grades details for a particular student and academic year. First, they get a grades summary per course. Then they can drill into the details grading per course.
- Produce reports of grade achievements for individual student, whole section or sections of the same course using tables and graphs.
- Make the grading details as service to be able to easily extract the grades from the system.
- Integrate with Google Classroom to Import/Export grades to Google Classroom.

The modules assigned to each team are shown in the table below:

First Name	Last Name	Project Module	Group
Nada	Aboueata	Module 1 - Admission	G3
Rahma	Ali		
Sara	Al-Rasbi		
Fatima	Haouari	Module 5 - Sections & Schedule	G4
Sarah	Sheikh		
Tooba	Salahuddin		
Ealaf	Hussein	Module 5 - Sections & Schedule	G2
Ranim	Faraj		
Salma	Shalaby		
Hisham	Yassin	Module 6 - Grade Center	G5
Shabir	Moosa		
Israa	Alsarsour	Module 2 - Registration Management	G1
Nesreen	Jboor		
Nour	Haj Ahmad		
Mays	Abdulwahab	Module 3 - Attendance	G6
Salman	Raeisi		

2. Deliverables

1) Use case diagram for the module(s) assigned to your group. This will help refine and clarify the requirements during the initial progress meeting with the instructor. Note that further important clarifications and features maybe added to your assigned module(s).
2) Architecture Design, Classes Design and the UI design for the use cases of the module(s) assigned to your group.

   The design documentation should include at least the following:
   - Application Architecture Diagram
   - Class Diagram showing Entities, Repositories and Services and Controllers.
   - UI Design and navigation.
   - Discussion of design rationale (i.e., justification) of key design decisions.

   During the weekly project meetings with the instructor, you are required to present and discuss your design with the instructor and get feedback. You should only start the implementation after addressing the feedback received about your design.
3) Data management using MongoDB.
4) Generate test data to implement and test your assigned module
5) Implement the client-side and the server-side Web components to deliver the use cases of your module(s) based on your previously developed and validated design.
6) Perform and document unit testing (using Mocha and Chai) and integration testing (using screen shots) The implementation should use the last JavaScript features and Node.js. The Web UI should use HTML 5 and bootstrap. The pages should comply with Web user interface design best practices. Also remember that 'there is elegance in simplicity'.

Push your implementation and documentation to your group GitHub repository as you make progress.

3. Grading rubric

Criteria	%	Functionality*	Quality of the implementation
Application Design Architecture Design, Classes Design and the UI Design to deliver SMS use cases. The design documentation should include at least the following: • Application Architecture Diagram • Class Diagram showing Entities, Repositories and Services and Controllers. • UI Design and navigation. • Discussion of design rationale of key design decisions.	20		
Complete and correct implementation of the use cases of your assigned module(s)	74		

Testing documentation with evidence of correct execution using snapshots illustrating the results of testing.	6		
Total	100		
Copying and/or plagiarism or not being able to explain or answer questions about the implementation	-100%		

[*] **Possible grading for functionality**: ***Working*** (get 70% of the assigned grade), ***Not working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation. In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation. Design quality includes **correct usage of MVC**, meaningful naming of identifiers, no redundant code, simple and efficient design, clean code without unnecessary files/code, use of comments where necessary, proper white space and indentation.

Marks will be reduced for code duplication, poor/inefficient coding practices, poor naming of identifiers and unnecessary complex/poor user interface design.