Continuous Assessment Instructions

Course Application Processing System (CAPS)

Contents

Objective

The objective of this project is to apply the knowledge and skills that you have gained from attending and practicing the lectures, demonstrations and workshops.

Management information systems for academic institutions cover major transactions like student applications, student scores and performance, scheduling, attendance monitoring, and course list development. All these transactions are done manually by officers and administrators, which may cause overheads on resources, time and cost.

You will demonstrate your team's software craftsmanship by building a web application for processing course enrolment. This application will be called **CAPS.**

The **CAPS** is used by three kinds of users: students, lecturers and administrators. For simplicity, let us assume the system is used to manage course enrolment for the current year and each course is run only once in the=is year.

Architecture

The team will use MVC Architecture for constructing CAPS. The purpose of building MVC applications is not only for the sake of the new knowledge you acquire, but also for the maintainability and the manageability of codes after project delivery.

Software Development tools

Version	Tool Name	Details
Java SE	JDK Version 1.8	You can download a copy on line or use from L: Drive
Java EE	Tomcat or Pivotal Server	You can download a copy on line or use from L: Drive
Data Base	JDBC API	Preferably MySQL. If your team is comfortable, you can chose any other valid RDBMS in the market.
Framework	Spring	Usage of framework is optional. The team can explore use of Spring framework if all members are comfortable
Client Side	JQuery Validation	The team can perform client side validation of forms as necessary – if deemed necessary in your design
Optional		Team can explore any additional codes for pagination, email etc for additional credits

You can obtain the mentioned software from shared drive or internet.

Design Consideration

The following are the design considerations:

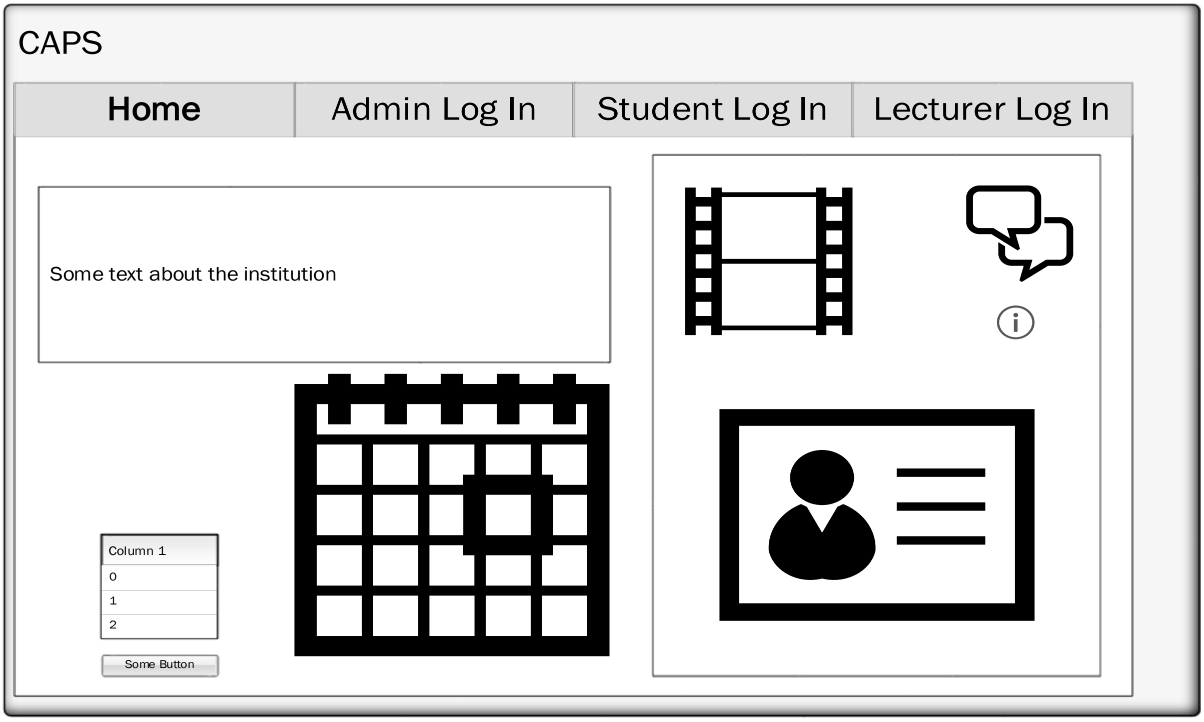
- The system is developed using Java EE components. Using Spring framework is optional.
- You should strictly follow the MVC Architecture.
- You should strictly follow DAO with JDBC or JPA based persistence architecture.
- The system uses MySQL Database to store the data. Do design appropriate data scripts and also populate the database with sufficient test data.
- Use of client side technologies such as Java Script, JSON and JQuery are optional.
- Credits would be awarded for any creative work your team decides to explore using additional frameworks or libraries.
- We are interested in versatility than volume, please create enough test cases with dummy test data to thoroughly test the system before the final presentations.

Functional Requirements

This **CAPS** software is a typical management information system for course enrolment and award of grades in academic institutions that has automated selective administrative functions. Through integrating the concept of relational database management systems such as MySQL, the accuracy, and integrity, of simultaneous and redundant access to these data, are monitored and audited periodically.

General Features

This wireframe shows the façade of the CAPS. To attract aspirants and students, we plan for this page to contain some marketing concepts, like the presence of images, news scoops, and other campus-related images. Teams can design this as a simple static web page. Teams can also use style sheets to standardize the look and feel across the application.



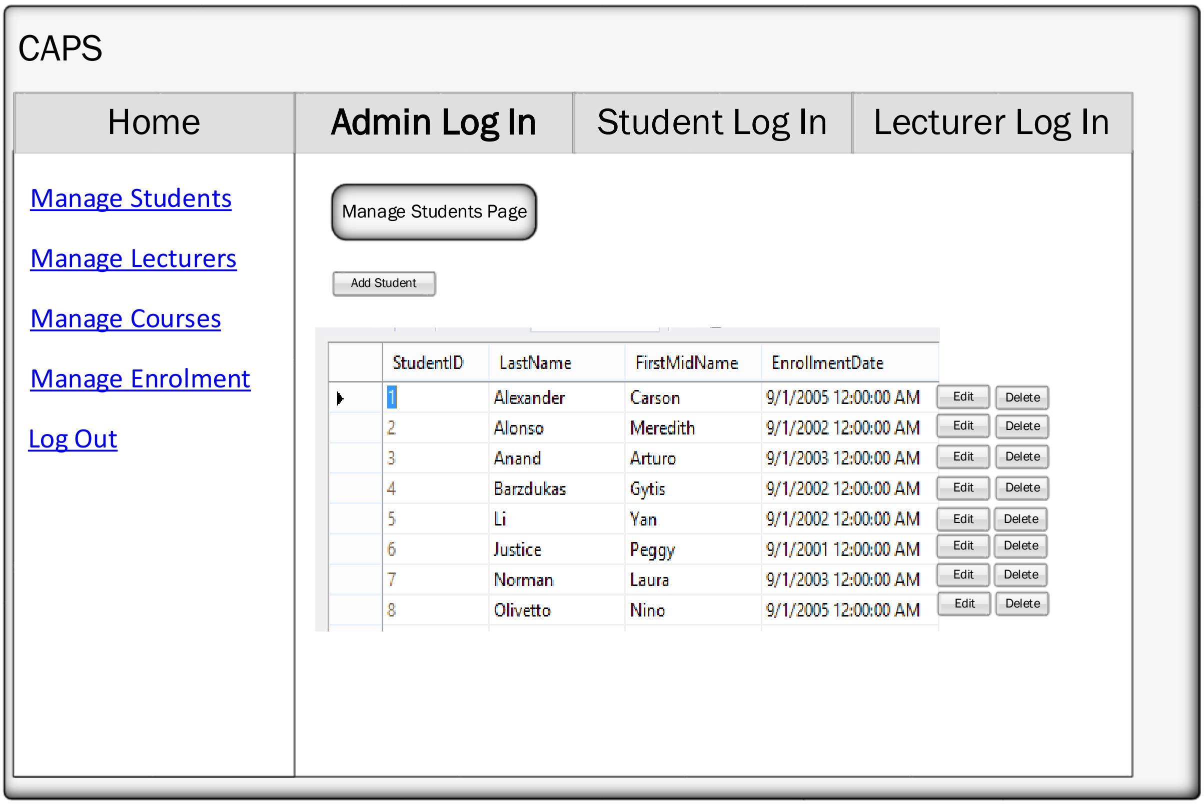
Some text about the institution

Column 1
0
1
2

- An actor can be admin or student or staff.
- The teams can choose to have three different log in pages or one integrated page with automated role selection.
- **CAPS** application has a centralized data store to manage student, lecturer and course and enrolment information.
- The **CAPS** system has a common logout page.
- Once a user with proper credential logs into the system, the system authorizes that user and displays appropriate welcome page befitting the role. Also menu functionalities must be relevant to that particular user role.

The Administrator Module

This module of **CAPS** adds, removes, updates and retrieves all master lists of courses, students, lecturers and current year enrolment. Only administrators can access this module. The wireframe could be like the following:

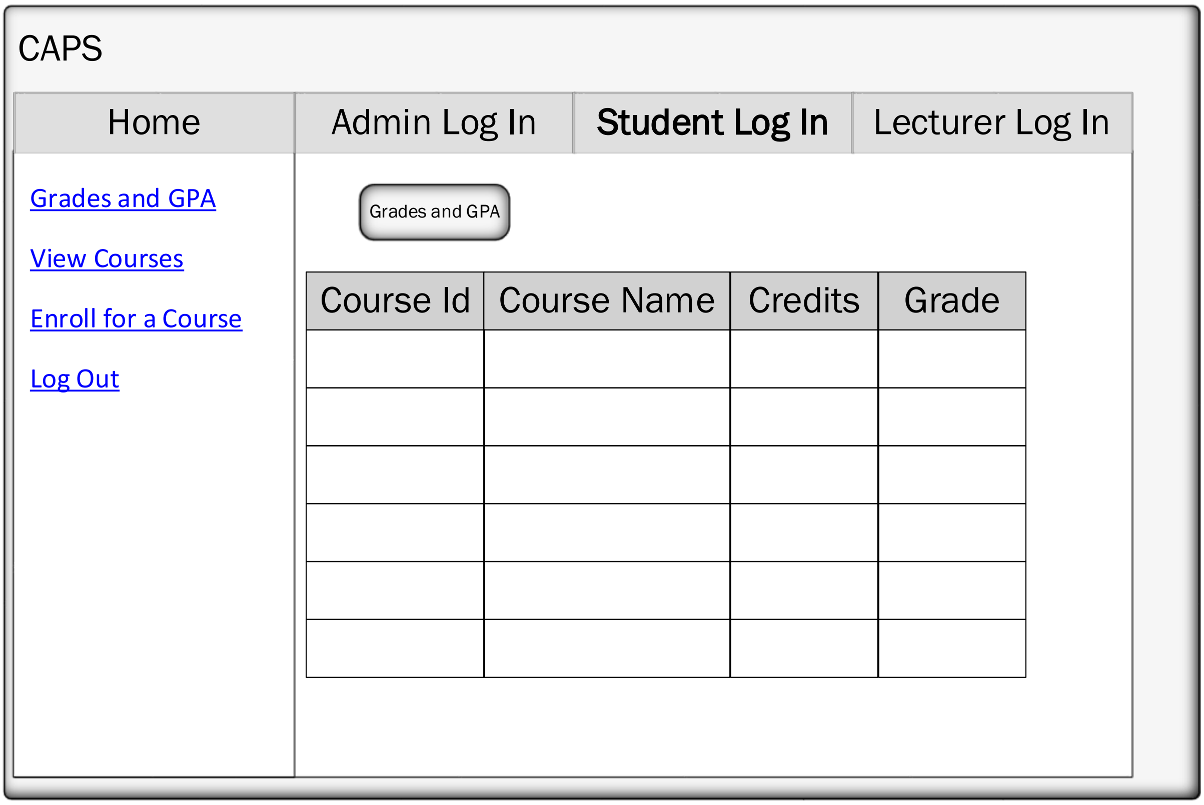


StudentID	LastName	FirstMidName	EnrollmentDate		
1	Alexander	Carson	9/1/2005 12:00:00 AM	Edit	Delete
2	Alonso	Meredith	9/1/2002 12:00:00 AM	Edit	Delete
3	Anand	Arturo	9/1/2003 12:00:00 AM	Edit	Delete
4	Barzdukas	Gytis	9/1/2002 12:00:00 AM	Edit	Delete
5	Li	Yan	9/1/2002 12:00:00 AM	Edit	Delete
6	Justice	Peggy	9/1/2001 12:00:00 AM	Edit	Delete
7	Norman	Laura	9/1/2003 12:00:00 AM	Edit	Delete
8	Olivetto	Nino	9/1/2005 12:00:00 AM	Edit	Delete

- Administrators create and manage student related information.
- Administrators create and manage lecturer related information.
- Administrators create and manage course and enrolment related information such as class size etc.
- Administrators update the status of enrolment provided there is a request from the lecturer to do so. Normally students enrol themselves to courses of their interest. Administrators are empowered to override this selection.

The Student Module

This wireframe below is dedicated to the page that manages student profiles and scholastic information. The student's individual grades, courses they are enrolled in, and the overall courses they took with the GPA, are all the concerns of this area.



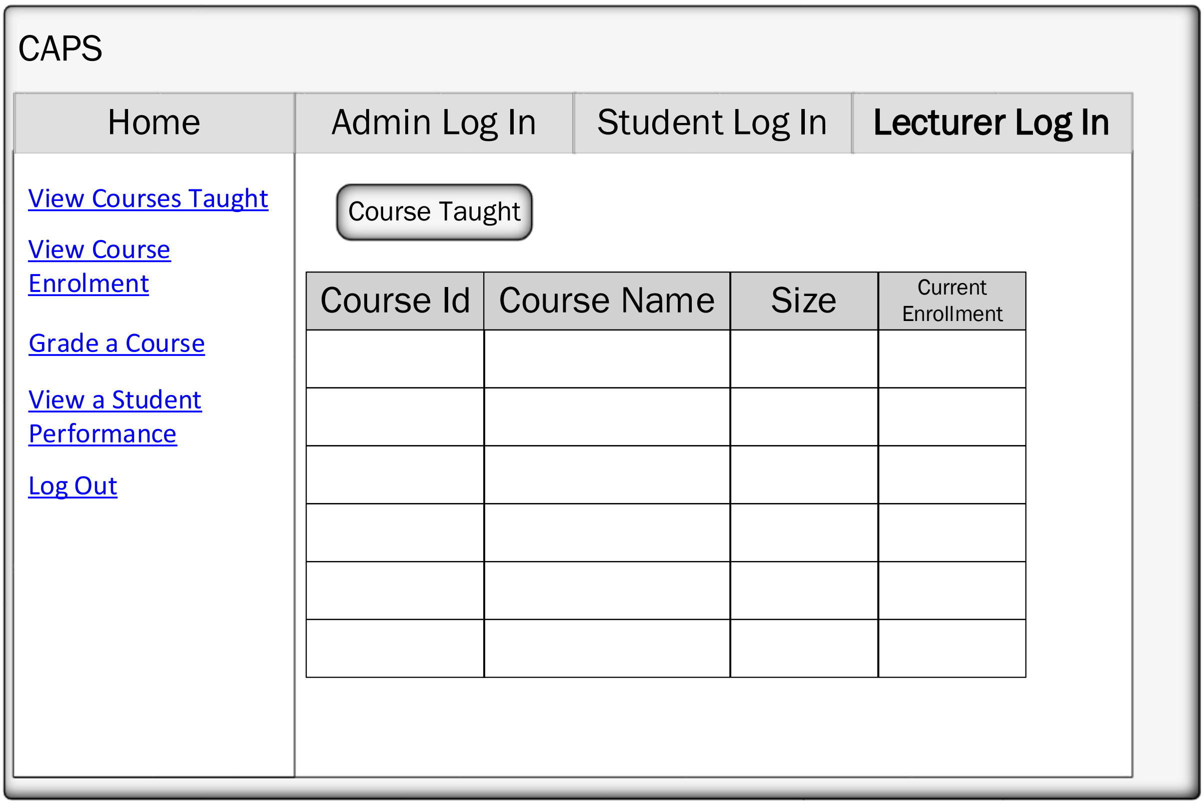
Course Id	Course Name	Credits	Grade

- Any student can enrol for a class. Enrolment is on a first-come-first-served basis. Each course has a capacity. When a student attempts to enrol for a course that is full, an appropriate directive message is provided.
- A student can view list of courses that he/she can enrol. This list must not show courses that the student has completed.
- A student can also look at his courses, grades and the Grade Point Average. Teams can devise their own formula for such calculations.

The Lecturer Module

This wireframe is to be accessed by any valid faculty member (lecturers) of the institution wherein they can input grades, manage students' performance, and manages their course loads and schedules.

- A lecturer can look at the list of enrolled students for a particular course that he teaches.
- A lecturer can input scores for individual student for a particular course he teaches.



Course Id	Course Name	Size	Current Enrollment

Additional Features

- ➢ Pagination in View Course Enrolment Page.
- ➢ Email student after they enrol to confirm.
- ➢ Use calendar and time table facility to improve course details
- ➢ Use Style Sheets.
- ➢ A lecturer can view a particular student's CGPA performance.
- ➢ Write Automated Test Cases.

Evaluation Criteria

The followings are sources the evaluation criteria.

1. Workable Solution [eclipse workspace]
2. Java implementation best practices; for example, object encapsulation, entity modelling, layering of architecture etc. Credits would be allocated for proper exception handling implementations, validation logic, test cases and other utility classes.
3. Demonstration must have minimal presentation slides (5 SLIDES ONLY).
4. You will be marked on the following attributes :
    a. Mandatory Use Cases
    b. Screen flow
    c. Business Rules and Validation
    d. Exception Handling
    e. Test Cases (both good and bad)

5. Additional marks will be awarded for any additional creative work. For example, use of frameworks, pagination of records, use of good open source, validation, use of templates, security, style sheets etc.,
6. Your code demonstration must walk through aspects such as:
   - Code Layering
   - Controller Layer
   - Model Layer
   - DAO Layer
   - Quality of JSP Pages
   - A decent exception management
7. You will also be awarded bonus points for any exemplary extensions demonstrated.

Stages of work

1. Stage One: Design of Data Base and Screen Flow
2. Stage Two: Building Prototype & Testing
3. Stage Three: Complete Development
4. Stage Four: Testing and Fine Tuning
5. Stage Five: Delivery

Deliverables

This assignment is part of the continuous assessment for this course. You will be evaluated for 25 marks on the whole. You will work in your team. No individual work will be accepted. The followings are the deliverables.

1) A 20 minute presentation, explaining the design and code of your team work.
   - Architectural diagram and (general class diagram or ER Model).
   - A CD-ROM with your workspace file containing all the source code, compiled classes, JSP, servlets and any other files required to run the application. If the team is using additional plug-in or library, it is to be bundled with deliverables. Write the names of your group members on the disk. You may wish to submit the workspace making it easier to evaluate.
   - Peer evaluation form.

Interaction with the Lecturer

- Lecturer is available for discussion based on previous appointment for design review

Schedule

Presentation and Submission of Deliverables

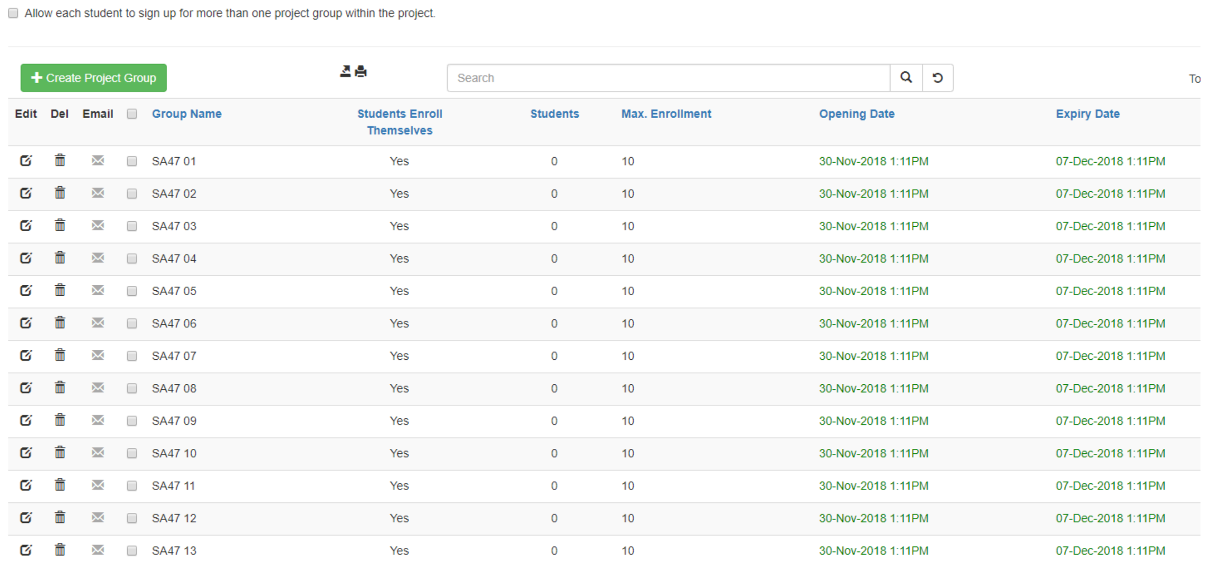
Task	Date
Presentation of Java Solution	**14th December 2018 (Friday)**
Submission of Deliverables • Eclipse or STS workspace • SQL Queries • PPT • Any other information essential to run the application • Peer evaluation form	**Upload to IVLE by the end of 14th December 2018.**
Individual Schedule for teams will be announced in IVLE later	

Plagiarism Notice: All students share the responsibility for upholding the academic standards and reputation of the University. Academic honesty is a prerequisite condition in the pursuit and acquisition of knowledge. Academic dishonesty is any misrepresentation with the intent to deceive or failure to acknowledge the source or falsification of information or inaccuracy of statements or cheating at examinations/tests or inappropriate use of resources. Please avoid sharing code, you will be penalized if found guilty. **Sharing of ideas, design and teaching your peers are most welcome.**

Everybody contributes and no excuses

APPENDIX 1

Please enrol into IVLE Project Groups Coming Week.



☐ Allow each student to sign up for more than one project group within the project.

Edit	Del	Email	☐	Group Name	Students Enroll Themselves	Students	Max. Enrollment	Opening Date	Expiry Date
✎	🗑	✉	☐	SA47 01	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 02	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 03	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 04	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 05	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 06	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 07	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 08	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 09	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 10	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 11	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 12	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM
✎	🗑	✉	☐	SA47 13	Yes	0	10	30-Nov-2018 1:11PM	07-Dec-2018 1:11PM