Note:

All scenarios below should be discussed, where appropriate, with an eye towards instructors and students using their mobile devices or a desktop/laptop computer. Additionally, as Purdue’s four campuses sometimes act as one and other times act independently, please describe how the scenarios below can be accomplished in a single or multi-tenant environment.

Case Study #1 - LMS Transition

Professor Ivanov has spent the last four years cultivating his course content. With the transition to a new LMS, he is worried that he will lose all of his hard work and spend undue hours manually moving and recreating content. Ideally, he wants the university as a whole to pick up and move everything into the new LMS. Any conflicts identified during migration should be sent not only to him but also support staff so they can work together to resolve those issues. He would appreciate even more if those issues were automatically taken care of by the vendor.

Dr. Traore doesn’t want to wait for the university to move her course. She is an early adopter and wants to move her course’s structure and content independently. Dr. Traore needs the appropriate vendor support and documentation to do this on her own successfully.

Inevitably, there will be problems with the transition and instructors and support staff will need the help of the vendor to move forward. They will need high levels of support during heavy transition periods as well as after hours and post-transition support as new problems are identified. The best scenario is to have multiple avenues of support available including but not limited to chat, phone, email, and on campus and in person gatherings for instructors, support staff, and students.
Case Study #2 - Multiple Sections

The new LMS should support the academic units to offer an equivalent experience to students across sections of a course, in courses that have a significantly high number of sections, students and instructors. The LMS should also expedite and simplify the work of course coordinators across campus, who support these large-enrollment, multi-section courses. James Williams is the instructional coordinator of Zoology 101, a course with approx. 3,500 students, 75 sections, and 40 instructors. He is also the instructor for 4 of those sections.

Part A: In his role of Course Coordinator, James needs to:

1. Have access and edit privileges for all sections of the course, including their gradebook.
2. Create a course template (from scratch or copying from a previous semester) with common content for all sections, and use it to populate all, or selected sections, of the course at the same time.
3. Configuration of elements or content such as visibility, due dates, grading, as well as any ties between elements (for instance, assignment and rubric) should be the same in the sections as in the course template. If copied from the previous semester, content should have dynamic dates connected to the course calendar, so copying from a previous semester would automatically update due dates for assignments and activities to reflect the current (or future) semester schedule.
4. Add, delete or edit an element or content in the course template (or in a section or elsewhere) and disseminate the changes to all, or selected, sections of the course, at the same time, at any time during the semester.
5. See grades quickly and easily across multiple sections, by instructor, and/or by day/time that the sections meet, for comparison purposes. Create reports (plots, statistics) using this information.
6. Create assignments, quizzes, and exams using a pool of questions. Randomize the questions -- and generate different assignments, quizzes, and exams per section, then upload them to the corresponding section at once, including their configuration features. Students in different sections may have different due dates for the same assignment.
7. Download all student submissions of an assignment at the same time (batch download) for one or more sections.
8. Run a “plagiarism detection” tool for students’ work for an assignment in multiple sections, at the same time.
9. Discussion boards should be easily filtered/organized, allow for instructors/peers to acknowledge reading and/or give quick feedback (e.g., thumbs up button), provide notification of new posts, and offer an option preventing students from reading posts until they make initial post which consists of an instructor selected number of minimum characters/words. Instructors should be able to view aggregate participation by student for easy grading with feedback via a rubric or inline grading. Peer-review is also needed for discussions, with an option for anonymity.
10. Set criteria for the gradebook such as dropping the 2 lowest quiz scores throughout the semester or the highest and lowest quiz scores, etc.
11. Ability to schedule messages to be sent to students individually, across specific sections, or the entire course.
12. Release content based on date/time, student viewing of specific content (files, video, etc.), and student performance.

Part B: In his role of Instructor, James needs to do the same things as a course coordinator -- but only for the one or more sections of which he is the instructor. Specifically, James needs to do points 4 to 12 above -- but instead of having a master course, he needs to make changes in one of his sections, and then populate the rest.

Part C: In her role as Teaching Assistant, Amy is responsible for some grades (for example, recitation or lab work) -- but for other graded items (such as exams), should only be able to see the same content that students view, with no access to grades. These settings should be controllable by the course coordinator and/or the section instructor.
Case Study #3 – Usability

Dr. Watson is an older professor whose experiences with past LMS’s has been largely negative: they appear to him cluttered and unappealing with features he won’t use and unintuitive without a clear hierarchy. He needs an LMS that has a “beginner” or “simple” setting with just the basics that would later allow him to add features, including third party applications, as he becomes more comfortable with it.

On the other hand, Dr. Huang is an LMS all-star. Her courses take advantage of many of the LMS’s features and some 3rd party integrations. In teaching her course, however, she finds students need a couple weeks to get used to the interface and she loses out on quality instructional time. Ideally, she needs the ability to show and hide LMS features as they are needed in the course, so students are not overwhelmed when they first log in.
Case Study #4 – Accreditation

College deans, department heads and/or program coordinators need to present information supporting applications for accreditation. Dr. Jun is the program coordinator of an undergraduate program. For HCL accreditation, she needs to present evidence of development and/or mastery of skills in written communication, oral communication, and information literacy by students within each course, and across all courses within colleges, departments, and/or programs. Students develop these skills in the multiple courses throughout the program. In each course, there are specific learning objectives which are tied to skills through rubrics in assignments and exams.

Dr. Jun needs a system that allows her to do the following at the level of a section, a course (i.e., cross-sections), and/or across courses:

- Extract the information in the grading rubrics (grading points and feedback) per learning objective;
- Compile grading information from students to show the development and level of mastery of selected learning objectives;
- Generate reports (plots, statistics) about the development and level of mastery of selected learning objectives by the students;
- Compare development or level of mastery to program thresholds and create reports of key learning objectives that are being met and unmet.
- Aggregate/disaggregate data by a variety of different variables (student, program, rubric cell, multiple choice question, etc.)
- Reuse and share rubrics that align with student, program, and university level learning outcomes.
Case Study #5 - Group Work

Instructor Khan is teaching multiple courses that have students working collaboratively in groups across multiple projects. These groups need to have access to collaborative spaces within the LMS including: real-time content creation, communication tools like chat, video conferencing, discussion, email, file exchange/sharing, etc. These groups may/may not have different assignments/files/rubrics shared with each of them. Instructor Khan would like the flexibility to personalize and differentiate each group, if needed, or create a streamlined process of replicating the same settings for multiple groups at a time.

Additionally, Instructor Khan incorporates peer assessment into her course. The LMS needs to intelligently assign peer review so it can be done at the group or individual level. However, sometimes she notices that friends are a part of different groups and therefore she wants to assign the peer reviews herself instead of it automatically happening from the LMS.
Case Study #6 – Grading

Professor Grey teaches Zoology 201 each semester. He relies on a grading system that he has been using for the last 15 years, which he has found to be effective and fair for his students:

- He weights each of his three exams at 20% of the grade, all homework to 15%, quizzes at 15%, and participation is worth 10%.
- He drops 1 quiz, 1 homework, and drops 20% of the participation points.
  - Participation points are out of a total of 200, but this becomes 160 after he drops the 20%. If a student would receive more than 160 points, then this is just changed to a 100% for the participation column.
- He also takes into account any specific considerations that might impact what is fair for a particular student, such as dropping an additional quiz for an excused illness or other event.
- After the third exam, Professor Grey also offers extra credit opportunities where the points do not fit into any of the sections above.
- At the end of the semester, he displays the final grade as a letter (A, B, C, D, F) to each student based on a predefined cutoff.

Students in Professor Grey’s course need to be able to assess their current and/or projected course performance at various points in the semester. The students should be able to input “what-if” scores into the gradebook to see how their overall course grade is impacted.
Case Study #7 - Course Calendar

Instructor Chauhan needs a student and instructor course calendar that auto-populates from assignments, exams, discussion forums, etc. within the LMS. He needs the flexibility to see all course due dates from this course and all of his other courses in a single view or to select an individual course. His students need the same capability between their courses with the ability to automatically sync to the native calendar of their mobile device. The course calendar should adapt to the student or instructor location and display time zone differences accordingly.

One additional feature students need is for the calendar to remind them of upcoming key dates/events by text message, email, LMS announcement, and mobile app notification. Students need the ability to pick and choose how they are notified.
Marcy Franklin is part of a campus wide group of staff that support students from many different roles including academic advisors, student athlete tutors, and scholarship coordinators. This group needs day-to-day access to individual student information including but not limited to course progress, course content, assignments, calendar, instructor information, and grades. They need this not only for the set of courses their students are currently in, but also previous semesters information. They use this information to support student success and help plan short-term and long-term schedules. They would also need the ability to add events to the students’ calendars inside the LMS.
Case Study #9 - Non-Credit Courses

A university department offers a trio of non-credit short courses. These courses have open and rolling enrollment available to the public. The first course is a free offering, the second and third courses require payment. A potential outside student should be able to register for enrollment, provide payment for, and be granted access to either an individual course or the series as a package all without an existing Purdue career account. Students should be able to set their own update preferences and create a user bio. Each student would receive a certificate signifying completion of a unit or course in order to build a digital portfolio. The LMS should be capable of auto-generating a customizable certificate.

- **Course A**: Self-paced course. Since these learners are completing this on their own schedule, they might often get interrupted so the system should have a “start where you left off” feature. Moderation provided by Purdue faculty & staff.
- **Course B**: Hybrid course with fixed schedule. Instruction and administration provided by Purdue faculty and staff. Instructor should be able to easily view & require completion of online modules before the on-campus portion.
- **Course C**: Simulation exercise where a vendor has been contracted to provide the simulation. The vendor needs the ability to be provided with appropriate access to that course.

In addition, a corporate partner has approached the department with the desire to purchase 200 “seats” in the course series. These 200 users need to be bulk registered for the course outside of the individual registration route. The company’s HR manager also needs non-student read only access to the course to monitor progress and see user reports. The manager should only have access to the corporation’s student details.

The company would also like their employees to have an area where they could provide additional learning and discussions creating a community of practice for their employees only. Since the additional learning is self-paced and composed of modules that employees can choose to complete in any order based on their own perceived needs at the time, the LMS ideally would have some method by which employees can record their completion dates for each step of each module in one location such as a simple table. This would allow the HR Manager to tell several things at a quick glance: (1) who is working on which module, (2) how quickly employees are working, and (3) how long individual modules or activities tend to take. It would keep track of which modules they have already completed so that they can easily tell which are still available to them at a later date. The LMS needs to allow the HR Manager a lot of flexibility in adding, retaining, or deleting users. The course cannot time everyone out at the end of a semester, for instance, and the HR Manager needs to be able to turn employees’ access off and on without removing them permanently from the course.

Unfortunately, the day before Course C was to begin the simulation, two of the corporate users have to fly to Asia to troubleshoot a work emergency. The company would like to substitute two different employees for these seats. The new students need to be quickly added and the previous students removed.
Case Study #10 - Content Accessibility

Professor Garcia wants her class to be accessible. Her area of expertise is in chemical engineering. She needs to spend her time teaching students about chemical engineering, but she is spending inordinate amounts of time trying to figure out how to make her content accessible. She needs the LMS to automatically flag new uploaded content that is not accessible and point her to resources on how to make improvements. Ideally, Professor Garcia could edit the uploaded files within the LMS without having to download, change, and re-upload.

Students take advantage of accessible content, even if they do not require accommodations. Ideally, the LMS will automatically make any uploaded content (from the instructor or other students) available in multiple formats (audio, text, html) so students can select format in which they are most comfortable.