

This document defines a hierarchy of functional requirements for Gradebook. They are based partly on requirements identified and processed by the following students of CS305-71 in Fall 2018: Adrian Brocke, Andrew Figueroa, Cristian Fitzgerald, Elly Griffin, Jonathon Middleton, Beth Psenicnik, and Shelby Simpson. The Gradebook product itself is created and maintained by members of the Data Science & Systems Lab (DASSL).

Work Breakdown Structure: The hierarchy lists only the functions some end user performs. It does **not** list tasks such as creating API and UI at any level, because such tasks should be included in the Work Breakdown Structure (WBS) for each functional requirement where that is relevant. In general, only requirements go in this hierarchy. Tasks needed to complete any requirement are added to the hierarchy as part of project planning when creating a WBS.

User interface (UI): There is no requirement that each functional requirement in the hierarchy have its own dedicated UI. Instead, the UI should suitably integrate and separate interaction elements to collectively support the required functionality.

Viewing and modification: Requirements that refer to viewing, modifying, or updating an object expect the UI to display all attributes of a selected object and if appropriate alter the value of one or more attributes of the object.

Reports: The expectation for “Report” and “List” type of requirements is that the system lists all or a subset of the instances of the appropriate kind. (The term “list” is not meant to limit the UI to tabular presentation.). The UI should show values for one or more appropriate attributes of each instance listed and possibly provide a means to view or modify (more) details of a selected instance. It may also include a means to navigate to a related report. For example, the “Report on courses” could show course number and title, and link to a report on past and present offerings of a selected course.

Regardless of their presentation, each report should support appropriate filters and sorters: A filter should let the user limit or expand the instances included in the report; a sorter should let the user order or reorder the instances presented. Additionally, a report could also include “selectors” which let users choose what details they wish to see for each listed instance.

Each report could optionally provide an “Export” and/or “Print” feature to save the current results to an external file in raw form such as CSV and JSON, or in “pretty form” such as PDF and HTML.

1. Manage group roles (a group role is a server role like Instructor, Student; cannot login)
 - 1.1. Add group role
 - 1.2. Remove group role
 - 1.3. Report on group roles (which group roles are available)
2. Manage users (a user is a human or an app subject to authentication: must login)
 - 2.1. Add user
 - 2.2. Assign group role (a user belongs to one or more group roles)
 - 2.3. Revoke group role
 - 2.4. View and modify user
 - 2.5. Report on users (also see “Monitor user activity”)
3. Authenticate user: all system functionality should be available only to authenticated users

4. Monitor user activity
 - 4.1. Report on user connection and disconnections
 - 4.2. Report on user activities within the system (“journaling”)
5. Manage terms
 - 5.1. Manage years
 - 5.1.1. Add a year
 - 5.1.2. Remove a year
 - 5.1.3. List years
 - 5.2. Manage seasons
 - 5.2.1. Add a season (a season is a label such as Fall and Spring)
 - 5.2.2. Remove a season
 - 5.2.3. Modify a season (should amount just to changing spelling)
 - 5.2.4. List seasons
 - 5.3. Add a term (a section is a combination of a year and a season)
 - 5.4. Remove a term
 - 5.5. Update a term
 - 5.6. Assign timeline: start and end dates, enrollment deadline, due dates for mid-term and final grades
 - 5.7. Assign holidays and closures (includes updating and removing)
 - 5.8. Report on terms (including aggregate information such as #courses)
6. Manage departments
 - 6.1. Add department
 - 6.2. Remove department
 - 6.3. Assign chairperson (a department has one instructor as its chairperson)
 - 6.4. View and modify department
 - 6.5. Report on departments
7. Manage instructors (also see “Manage users”)
 - 7.1. Add instructor (every instructor has a unique e-mail address)
 - 7.2. Assign department (instructor belongs to one or more department, one being “home”)
 - 7.3. Remove from department
 - 7.4. View and modify instructor
 - 7.5. Report on instructors (also see “Monitor user activity”)
8. Manage courses
 - 8.1. Add a course (a course has a name/number, title, #credits, and offering departments)
 - 8.2. Remove a course
 - 8.3. View and modify a course
 - 8.4. Report on courses
9. Manage students (also see “Manage users”)
 - 9.1. Add student (every student has a unique ID and a unique e-mail address)
 - 9.2. Assign major (a student has zero, one, or more majors)
 - 9.3. Remove a major
 - 9.4. View and modify a student
 - 9.5. Report on students (also see “Monitor user activity”)

10. Manage grade tiers (instructors have own grade tiers; may (not) be same across their courses)
 - 10.1. Add grade tier (see CS305 syllabus for examples)
 - 10.2. Remove a grade tier
 - 10.3. View and modify a grade tier
 - 10.4. Copy grade tiers (useful in customizing a grade tier for a specific course)
11. Manage sections (a section has a term, course, number, capacity, schedule, letter grade?)
 - 11.1. Add a section
 - 11.2. Remove a section
 - 11.3. View and modify a section
 - 11.4. Assign instructors (add and remove; a section has one or more instructors)
 - 11.5. Enroll students (add and remove up to the enrollment deadline)
 - 11.6. Assign term grade to students (including withdraws)
 - 11.7. Report on sections
12. Manage assessments
 - 12.1. Manage assessment kind (such as Assignment, Test, Exam, and Quiz)
 - 12.1.1. Add an assessment kind (name, description, weightage)
 - 12.1.2. Remove an assessment kind
 - 12.1.3. View and modify an assessment kind
 - 12.2. Manage assessment items (say Test 1 and Test 2: assessment item belongs to a kind)
 - 12.2.1. Add an assessment item (name, base points)
 - 12.2.2. Remove an assessment item from a kind
 - 12.2.3. View and modify an assessment item
 - 12.2.4. Assign score to student with optional notes (includes changing score)
 - 12.3. Report on assessment kinds (aggregate, by kind and student)
 - 12.4. Report on assessment items (detailed, by student)
13. Manage attendance (section specific; marked each time the class meets)
 - 13.1. Mark attendance (present, absent, explained absence, excused absence, tardy)
 - 13.2. Update attendance
 - 13.3. Report on attendance
14. Import course schedules: parse a file containing course schedules to populate the following information to the extent possible: department, course, instructor, section
15. Import roster: parse a file containing a section's roster to populate the following information to the extent possible: student, enrollee (an enrollee is a student a specific section)