Navigator for Integrated Learning Experience (NILE) Project Reports

September 1, 2020

This work was supported by the U.S. Advanced Distributed Learning (ADL) Initiative (HQ0034-18-C-0041). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ADL Initiative or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes.

Distribution Statement A

Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
09-02-2020	Documentation	

4. TITLE AND SUBTITLE
Navigator for Integrated Learning Experience (NILE) Project Reports

5a. CONTRACT NUMBER
HQ0034-18-C-0041

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0603769D8Z

6. AUTHOR(S)
Gooru

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Gooru

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OUSD Personnel & Readiness
Advanced Distributed Learning Initiative
13501 Ingenuity Drive, Suite 248
Orlando, Florida 32826

10. SPONSOR/MONITOR'S ACRONYM(S)
OUSD/P&R/FE&T/ADLI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Documentation that covers pedagogical and technical topics related to the operation of the Gooru learning technology system. Topics include the learning and competency model, the system's search indexing service, system's general architechture, various APIs methods, xAPI implementation, technical guidance and user instructions, and other implementation instructions. If other, more detailed instructions are desired, please reach out to the ADL Initiative.

15. SUBJECT TERMS
adaptive learning systems, xAPI, competency,

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Nick Armendariz
U	U	U	UU	180	19b. TELEPHONE NUMBER *(Include area code)* 407-381-5550

Table of Contents

NILE Overview ..... *- 10 -*

Navigated Learning ..... *- 11 -*

Navigator Competency Model ..... *- 11 -*

Competency Graph ..... **- 12 -**

Map to Local Norms ..... **- 13 -**

Learner Profile ..... **- 14 -**

Link to Learning Maps ..... **- 15 -**

Updating Competency ..... **- 16 -**

Relationship Constraints ..... **- 16 -**

Skyline ..... **- 18 -**

Mastery with confidence measure ..... **- 19 -**

Rolling back mastery ..... **- 19 -**

Extending the competency model ..... **- 21 -**

Search Overview ..... *- 22 -*

Search ..... *- 22 -*

Query Processing pipeline ..... **- 22 -**

Query Expansion ..... **- 23 -**

Query Rewriting & Filter processing ..... **- 23 -**

Search Flow Diagram ..... **- 24 -**

Ranking ..... **- 24 -**

Search Result Processing ..... **- 24 -**

Personalized Search Environment ..... *- 25 -*

Personalized search flow ..... **- 25 -**

Core Environment ..... - 26 -

Catalog Enricher Environment ..... - 26 -

Analytics Environment ..... - 26 -

Search Environment ..... - 27 -

Search Signals ..... - 27 -

Catalog Enricher Environment ..... **- 29 -**

Crawl And Maintain Catalog ..... - 29 -

Curate Activities ..... - 31 -

Transcript ..... - 32 -

Summarization ..... - 33 -

Phrase Cloud ..... - 34 -

Classification ..... - 36 -

Demo Accounts ..... *- 44 -*

Exploring as a Student: ..... *- 45 -*

Overview: ..... **- 45 -**

Your Profile: - 46 -
Entering Math Navigator: - 46 -
Course Map: - 47 -
Study Player: - 48 -
Class Activities: - 51 -
My Proficiency: - 52 -
My Portfolio: - 53 -
Exploring as a Teacher: - 54 -
Overview: - 54 -
Entering Math Navigator: - 54 -
Performance Overview: - 55 -
Making Suggestions: - 55 -
Option 1: Single Student Suggestion via Skyline - 56 -
Option 2: Student Suggestion by Domain - 57 -
Class Activities: - 61 -
Option 1: Assigning Class Activities from the Course Map - 61 -
Option 2: Assigning Class Activities from the Tenant Library - 63 -
Option 3: Assigning Class Activities from My Content - 64 -
Option 4: Assigning Class Activities from the Gooru Catalog - 64 -
Go Live - 65 -
Enabling Competency Mastery for Class Activities - 66 -
Multi-Grade Support (For Select Tenants Only) - 67 -
Accessing Student Data: - 67 -
Learning Journey and Course Map: - 70 -
Content Management: - 71 -
Glossary - 72 -
Creating a Classroom - 77 -
Accessing Your Classrooms - 77 -
Choosing a Course - 77 -
Enrolling Students - 80 -
Viewing the Course Map - 81 -
Adding a Co-Teacher - 82 -
Setting Math Level and Destination - 83 -
Multi-Grade Support (For Select Tenants Only) - 84 -
Remixing an Existing Course - 85 -
Remixing Existing Collections and Assessments - 87 -
Creating a New Unit or Lesson - 90 -

Adding Collections and Assessments to a Lesson - 91 -
Creating a Course from Scratch - 92 -
Adding Collaborators - 93 -
Creating a New Collection - 94 -
Writing a New Assessment - 98 -
Adding Activities to Your Classroom - 101 -
Option 1: Assigning Class Activities from the Course Map - 101 -
Option 2: Assigning Class Activities from the Tenant Library - 102 -
Option 3: Assigning Class Activities from My Content - 104 -
Option 4: Assigning Class Activities from the Gooru Catalog - 104 -
Enabling Competency Mastery for Class Activities - 105 -
Go Live - 105 -
Selecting Students for a Class Activity - 106 -
Performance Overview - 108 -
Option 1: Single Student Suggestion via Skyline - 109 -
Option 2: Student Suggestion by Domain - 111 -
Class Proficiency Report - 114 -
Class Progress Report - 115 -
Sorting Go Live Data by Score - 117 -
Interpreting a Skyline - 118 -
Glossary - 119 -
Overview - 121 -
Navigator Overview - 121 -
Navigator Platform Components - 122 -
Navigator Core Design Principle - 123 -
Technology Stack - 124 -
Architecture Overview - 125 -
Component Architecture - 125 -
Clusters - 126 -
Environments - 127 -
Data In & Data Out - 127 -
Data In - 128 -
Event Types - 128 -
Event Logger - 128 -
Event Transformer - 129 -
Event Processing - 129 -

Global Event Processors - 130 -
Event Fan-Out - 130 -
Discrete Event Processors - 130 -
Navigator XPI Transformer - 130 -
Data Out - 131 -
Search & Suggest - 131 -
Architecture - 132 -
Profile based Route Suggestion - 134 -
Profile based Rescope Suggestion - 135 -
Performance based Reroute Suggestion - 135 -
Navigator Application Design - 135 -
Deploy Architecture - 136 -
Overview - 139 -
Competency Framework enabling AI - 140 -
Progression Space of Competencies - 140 -
Auto-generating competencies and structuring in a progression space - 141 -
Learning Map for every competency - 142 -
Learner Identity embedding in every competency - 142 -
Relate to local norms - 143 -
Personalized Learning Pathways - 143 -
Activity Stream Data from Distributed Systems - 144 -
Curate Resources - 145 -
Update Learner Identity - 145 -
Principles of Learning - 145 -
Recommend Pathways - 147 -
Big-data operationalizes science of navigated learning - 147 -
Navigator Competency Model - 150 -
Competency visualization as Tree - 150 -
Classification - 150 -
Subject - 150 -
Framework - 150 -
Framework Creation Prerequisites - 151 -
Framework Creation: Ref FW - 151 -
Framework Creation: Mapped FW - 151 -
API List for NCG Core - 152 -
Entities - 152 -
Detailed APIs - 154 -
API Authentication and Authorization - 184 -

NILE Overview

Navigator for Interoperable Learning Experience (NILE) leverages the science of personalizing learning to offer teachers and students the most appropriate high-quality resources in each moment to meet all students exactly where they are in order to give students and teachers the ability to precisely visualize and understand each learner's unique needs and learning in real time. With that understanding, teachers can provide a full spectrum of learning resources and activities and individual learning pathways designed for each student. Students can learn in a way that inspires them and enables them to meet their learning goals. We will use the term Navigator and Learning Navigator interchangeably with NILE.

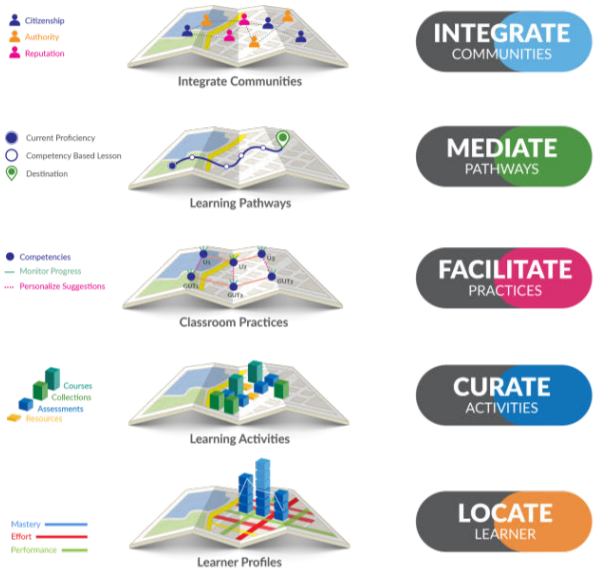


Figure 1: Exploded view of Navigated Learning elements

The Navigator works by locating every students' current knowledge and skills, providing them a personalized route for learning using open curated resources and making reroute suggestions based on their performance until they reach their learning destination as seen in Figure 1. (01) Locate: Learning Navigator "locates" each student — which means it pinpoints exactly which knowledge and skills they have already attained and which they have yet to learn. It locates them when they begin their learning and it continues to update their location in real-time as they progress. Whether students and teachers are utilizing the resources in the catalog, or are engaged in complex, interactive activities — like projects, essays, proofs, presentations, debates, etc. — they can create digital data from every learning activity. The Navigator can then use that data to provide real-time updates about each student's progress. (02) Curate: The Navigator also includes a catalog of millions of curated and organized resources that students and teachers can use to meet learner exactly where they are. It offers students personalized curated courses to their learning destinations with well tagged open resources and question. (03) Facilitate: As students study these courses, Learning Navigator helps teachers facilitate personalized learning by monitoring students' progress, engaging the class in live assessments for a real-time update on student understanding and suggesting additional activities to help students stay on track. (04) Mediate: Gooru uses principles of learning from the science of learning coupled with data science to suggest personalized pathways. Given a student's learning destination, the Navigator, based on the student's profile, personalizes the route to their destination and makes reroute suggestions based on their performance.

NILE: Navigator Competency Model

Navigated Learning

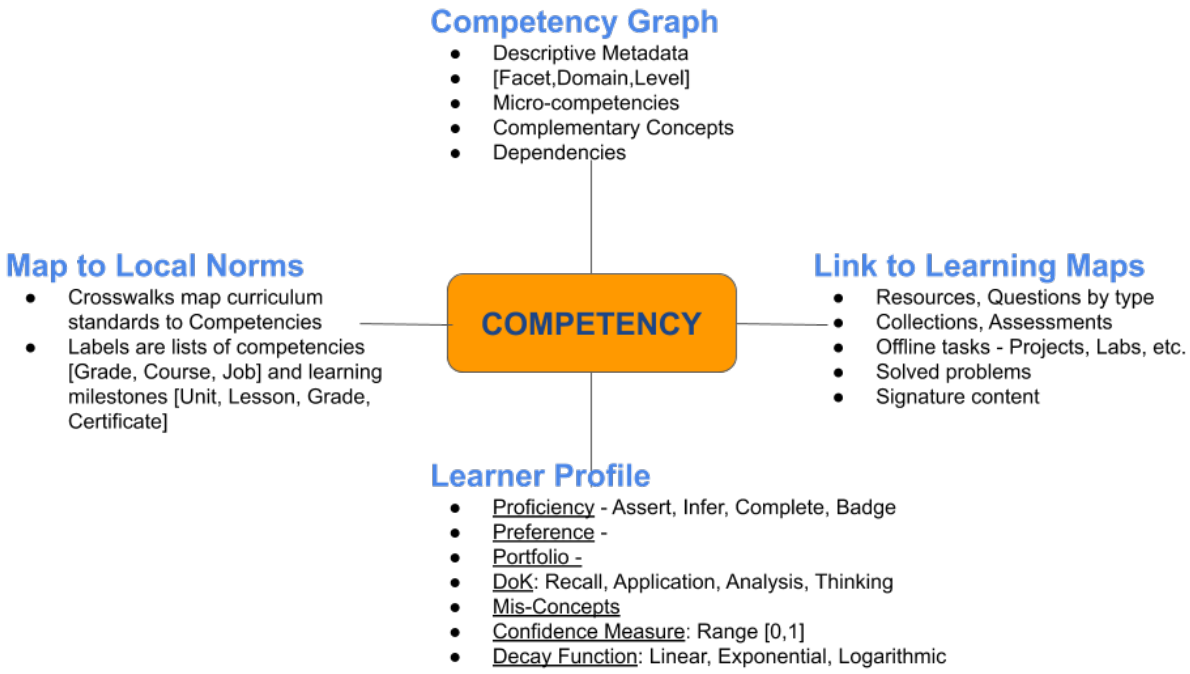
This document outlines Navigated Learning, describes Navigator Competency Model and the extensions that we will build to to cover additional categories such as Skills Training, Teacher PD and Habits of Success.

Navigated Learning is a pedagogical approach that combines the learning sciences and data sciences to foster learning across all learners and all settings. Navigated Learning enables students to achieve learning outcomes in a competency-based enviroment, where competencies comprise of academic are designed at local, state, and national levels. The construct of "competency" goes beyond traditional learning; it includes both cognitive and non--cognitive skills (e.g. critical thinking, creative thinking, collaboration, and metacognition)). Competency-based framework is motivated by the objective of ensuring that students achieve the desired competencies through the normal classroom process or where required, through additional instruction, practice time, and academic support.

Navigated Learning places the interaction of the learner, teacher, and system at the center of the learning environment. Pinpointing the learner's location based on the learner's prior knowledge, experiences, and complex identity informs all aspects of the Navigated Learning experience. The learner and teacher set a destination, the competencies the learner masters in a given period of time, to set expectations and goals for each student. Navigated Learning is premised in the principle that to navigate a learner, we have to first, locate them. The learner's location will then inform their learning "route". The learner can navigated and continuously "rerouted" based on their strengths, preferences and gaps learning styles until they reach their stated destination.

Navigator Competency Model

Gooru has developed an extensive competency model. The concepts are organized in a hierarchy of Facet(Subject) Domain, Competency and Micro-competency. The term competency is synonymous with learning standard. Examples of commonly used standards in the U.S. include the Common Core State Standards and Next Generation Science Standards. K-12 educators generally understand competencies and content developed at the competency level.



Each "leaf-node" is a first-class object that is associated with metadata, progression, and proficiency. Each concept is mapped to the NILE Universal Taxonomy (GUT) and cross-walked to standards from a variety of frameworks. All learning activities are mapped to a concept.

Competency Graph

The Competency Graph defines the competency in terms of its metadata, alternate conceptions and the different edges - complementary, supplementary and dependency edges.

- Descriptive metadata.
    - Every competency has metadata associated with it.
    - Competency is identified by a code, 2-3 word display, student-friendly description and full description
- The X, Y, Z or Domain, Level, Facet
    - The competency is identified uniquely in a metric space by its x-y-z coordinate.
    - The Z-axis represents the Facet (or subject), X-axis represents the Domains in the facet, Y-axis represents the pedagogy or the learning depth (level) in a domain
    - Facets can be from the core curriculum or non-cognitive skills or vocational and skills training and other areas
- Topics
    - A topic is a group of competencies which are in sequence
    - Mastery in a competency in a topic implies inferred mastery in competencies that come before that competency in sequence
    - One or more topics in sequence form a domain
- Micro-competencies
    - Each competency is further broken down into micro-competencies

    - Resources are tagged to competencies, whereas questions are tagged to micro-competencies
- Complementary competencies
    - The series of competencies in the same topic whose learning depth or level is below that of the competency
- Dependencies/Progression
    - A competency will generally be identified with a prerequisite list of competencies that are recommended to be mastered in order to be able to master the competency
    - The prerequisites can cut across domains
    - The dependencies for a competency at a certain milestone level cannot be of a higher milestone level
    - The dependencies cannot be circular
- Related competencies
    - A competency is related to other competencies without having a prerequisite dependency
    - These related competencies can be both within and across domains
- Complementary concepts or Alt-concepts
    - Each competency has a list of potential alt-concepts (or misconceptions) that a student may have when learning the competency
- Decay Function
    - The "decay" of knowledge over time varies from competency to competency
    - Can be linear or exponential
    - This will be used to determine when the student's mastery in a competency would need to be evaluated again and present learning activities to reinforce mastery

Map to Local Norms

Different classes of users can use variations of a competency model that could be defined by, for example their geography or cultural norms. Example, while CCSS Math serves as a standard, each state follows its own curriculum and need to map to the CCSS model. A state may or may not use all competencies in a model and may have their own descriptions for the competencies.

Similarly, users could also have their own definition for a group of competencies within and across domains. For example in India, the NCERT uses Learning Outcomes as a measure for student's progress. Learning Outcomes are a group of competencies that can span one or more domains.

These variations are captured via crosswalks and labels that define the different types of grouping

- Standards crosswalk.
    - The standards framework adopted vary vastly from country to country, and within the states in those countries.
    - The different standards can be crosswalked to the Navigator Competency Graph (NCG)
    - Topics in the standards framework
        - May or may not include all the topics in the NCG.

        - may have their own name for the topics (and domains)
    - Crosswalk involves
        - Crosswalking competencies
        - Crosswalking micro-competencies for a competency. Micro-competencies for one competency, FC1 in the framework that is crosswalked to competency C1 in NCG can be crosswalked to micro-competencies of C1 only, and not to those of any other competency
- Additional attributes
    - Each competency is identified as either core or supporting or additional for a given framework
    - Teachers are supposed to focus 65-80% of their time on the core competencies
    - Supporting or additional competencies are not complementary to a competency
    - The supporting or additional competencies have a "related" competency relationship with other competencies but not a prerequisite relationship
- Learning Milestones
    - Competencies can be grouped together across domains as learning milestones
    - The milestones are defined at the GUT level
    - Competencies cannot overlap across milestones
    - Competencies should not be skipped between milestones
- Labels
    - The labels serve as a way to group competencies
    - Examples
        - Learning Outcomes: Primarily used for display purposes. Help teachers map their current understanding to the competencies
        - Grades: A set of competencies taught in a particular grade

Learner Profile

Learner's proficiency, progress and portfolio can be defined for every competency across the different disciplines

- Proficiency
    - Learner's proficiency or mastery in a competency can be determined in multiple ways
        - Completed. Student completes learning activities for the competency, and establishes mastery by taking an assessment and successfully completing it
        - Badged. Student takes a higher level assessment for a competency within the Navigator to earn a badge
        - Inferred. Having successfully established mastery for a competency, the student's mastery in complementary competencies within that domain is inferred
        - Asserted. Student or some authority can specify that the student knows a competency and assert mastery for that competency
- Preference

    - Student's preference in terms of type of learning activity, source of content and other attributes can be specified or determined
    - The preference, while established at a facet or domain level can be inferred for a competency
- Portfolio
    - All learning activities that the student has completed for a competency can be tracked
    - These evidences are then presented as a portfolio for the competency
- Depth of Knowledge (DoK)
    - Student's mastery for a competency can be tracked at different DoK levels
    - Primarily determined by the level of questions in assessments that establishes the student's mastery
- Alt-concepts
    - Identifies the different misconceptions that the student may have for a specific competency
- Confidence measure
    - Identifies the level of confidence in the mastery for the competency
    - Assessments that evaluate a student's mastery need to cover all the micro-competencies for a competency. One of the aspects that defines the confidence measure is the extent of coverage of the micro-competencies in the assessments used to evaluate student's mastery in the competency
- Authority
- Reputation
- Citizenship
- Other profile attributes
    - Grit
    - Perseverance
    - Interest
    - Motivation
    - Self-confidence
    - Engagement

Link to Learning Maps

Learning maps identifies all the different types of curated learning activities required and available for a competency

- Resources and Questions
    - Includes all the different types of resources and questions
- Collections and Assessments
    - Ideally collections should include resources that teach one competency
    - Assessments should evaluate a single competency and should cover all the micro-competencies for that competency
- Offline activities
    - Learning activities for a competency are not all digital and online

    - Typically Includes activities that are done offline in the real world, such as projects, presentations or using other systems such as simulators outside of the Navigator
- Signature collections and signature assessments.
    - Are of very high quality, curated by pedagogy experts to provide a very engaging learning experience.
    - The signature assessments are designed to establish badged mastery for the competency
    -

Updating Competency

    - The competency model can be extended to include additional competencies or a competency can be decomposed to include additional competencies or micro-competencies
    - A competency in the model can not be deleted
    - The metadata for a competency can be updated
- Dependency graph
    - Competencies also have dependency across domains that are called pre-requisites and establish a dependency graph

Relationship Constraints

- There is an implicit dependency for a competency based on the sequence of competencies in the domain. If the competencies for Topic T1 in Domain D1 are C1, C2, C3 in sequence, C3 is dependent on C2 and C2 is dependent on C1. A student mastering C3 would lead to inferred mastery for C2 and C1 even if the student does not show any evidence of mastery for C2 and C1
- Prerequisite dependencies span domains
- The pre-requisite dependencies cannot be circular.
    - If the prerequisite sequence for C3 belonging to D4-T1 is C4 of D3-T2, C2 of D2-T3, the C2 of D2-T3 cannot be dependent on C4 of D3-T2
- Milestones are defined at a GUT level only and do not have any specification at Framework level
- Dependencies for a competency at a milestone level cannot be at a higher milestone level
- Dependencies for a competency of a domain at a milestone level cannot be from domains at a higher sequence for the same milestone level, but can be of any domain at a lower milestone level.
    - For example, dependency for 3.NBT.4 (Numbers and Operations in Base 10) cannot be from 3.F domain (Fractions is higher in sequence than Numbers and Operations in Base 10)
- Milestone levels must have contiguous competencies within a domain and must be in the sequence of the competencies within the topics of the domain.
    - Let's consider an example of domains, topics and competencies in sequence as below
        - D1
            - T1.1, T1.2, T1.3
            - T1.1

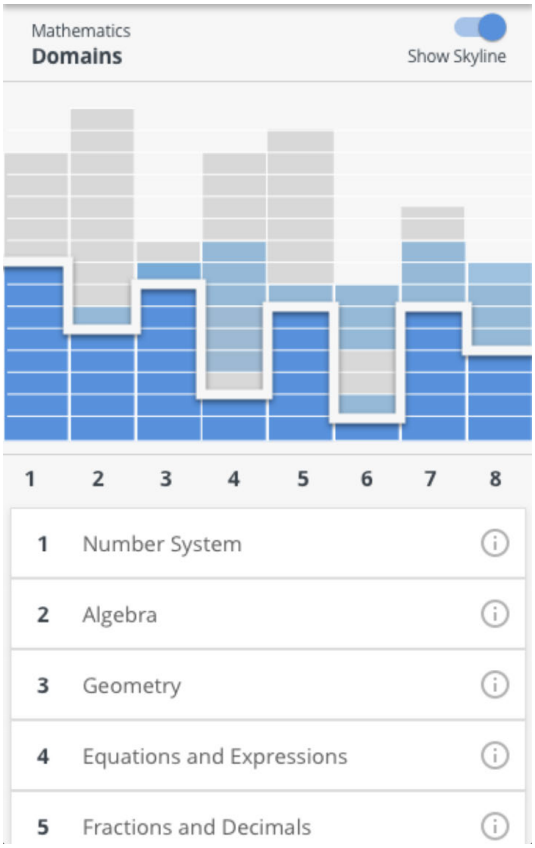
            - C1.1.1
            - C1.1.2
        - T1.2
            - C1.2.1
            - C1.2.2
            - C1.2.3
        - T1.3
            - C1.3.1
    - D2
        - T2.1, T2.2
        - T2.1
            - C2.1.1
        - T2.2
            - C2.2.1
            - C2.2.2
    - D3
        - T3.1, T3.2
        - T3.1
            - C3.1.1
            - C3.1.2
            - C3.1.3
        - T3.2
            - C3.2.1
            - C3.2.2
- One possible scenario for the milestones for this model could be as given below
    - Milestone 1
        - C1.1.1, C1.2.1, C1.2.2
        - C2.1.1
        - C3.1.1
    - Milestone 2
        - C1.1.2, C1.2.3
        - C2.2.1
        - C3.1.2
    - Milestone 3
        - C1.3.1
        - C2.2.2
        - C3.1.3, C3.2.1, C3.2.2
- And they cannot be out of sequence as below
    - Milestone 1
        - C1.1.1, C1.2.2
        - C2.2.1
        - C3.1.2

- Milestone 2
  - C1.1.2, C1.2.1, C1.2.3
  - C2.1.1
  - C3.1.3
- Milestone 3
  - C1.3.1
  - C2.2.2
  - C3.1.1, C3.2.1, C3.2.2

Skyline

NCM helps locate the learner's proficiency precisely on the learning map. A learner's mastery status of the competencies is plotted on a competency proficiency matrix. The proficiency matrix shows all the competencies in a domain stacked vertically along the y-axis, with the domains placed next to each other along the x-axis. The student's proficiency in a competency is marked on the proficiency matrix.

Every student has a Skyline for each subject. The skyline represents the highest competency that the student has mastered in each of the topics across all the domains. The skyline is represented by a line joining these competencies.



Mastery with confidence measure

We need to track mastery status at micro-competency level. While an assessment is tagged to a competency, the questions within the assessment are tagged to micro-competencies. An assessment may or may not cover all the micro-competencies for the competency. We track mastery for a competency based on performance in an assessment. If the student scores more than 80% in the assessment, the student has gained mastery and the status is set to completed. In this case, the student may not have successfully answered questions for one or more micro-competencies. Cases to consider:

- Assessment is for competency C1 which has 3 micro-competencies MC1.1, MC1.2, MC 1.3
- Assessment covers only MC1.1 and MC1.2
- Assessment has 10 questions, 5 for MC1.1 and 5 for MC1.2
- Option 1
    - Student gets 5 of MC1.1 and 3 of MC1.2 right, and scores 80%
    - Student's mastery status for C1 is now completed
    - Confidence measure is 66.67% since only 2 out of 3 micro-competencies are covered in the assessment
    - In this case, the student gets only 60% score for MC1.2.
        - Should we consider that the student has mastered MC1.2?
        - Or should we factor that into the confidence measure?
        - Each micro-competency in this case contributes 33.33% to the overall mastery
        - Student gets 100% in MC1.1 and 60% in MC1.2 and has no score yet in MC1.3. So, confidence measure is 100% of 33.33 + 60% of 33.33 = 33.33 + 19.99 = 53.32%
    - Factors to consider
        - How many questions per micro-competency should we consider as a minimum to contribute to mastery confidence measure?

Rolling back mastery

- Mastery status can be rolled back along the following lines
    - As the student continues to study and is continuously assessed for a competency, the performance in micro-competencies should contribute to the confidence measure
    - If the confidence measure drops below the threshold, the mastery status will be rolled back from completed to in progress
    - The decay function for the competency contributes to the (negative) change in the confidence measure.
    - Factors to consider
        - Since the decay function is for the competency, should the decay apply equally to all the micro-competencies?
- The mastery status for a competency over time should be tracked, along with reasons for the change

- Mastery status can be rolled back by assertion. Teacher or student can identify competencies that are mastered as not mastered. Depending on the competency whose mastery is being rolled back, competencies in sequence will also need to be rolled back
- Competencies that have inferred mastery should also be rolled back as the competency contributing to the inference is rolled back

Visualizing confidence measure

- Option 1
    - The extent to which the competency cell is filled with dark blue is based on the confidence measure for the mastery

Extending the competency model

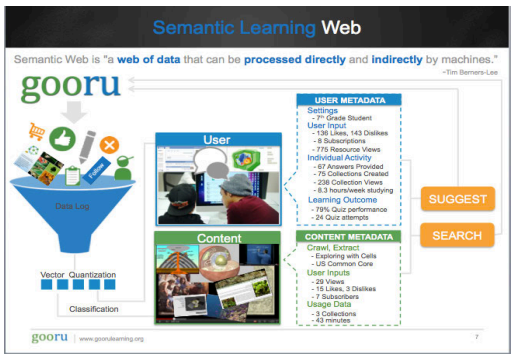
Today, Gooru has developed an extensive competency model to provide a framework for organizing all learning concepts. The Navigator Competency Model (NCM) includes concepts organized into a 5-level a hierarchy. The hierarchy is Subject -> Course -> Domain -> Competency (Standard) -> Micro-competency (Concept). The current competency model is defined for different categories such as K-12, Higher Education and Professional Development. The steps to follow to add or extend the competency model are detailed in the Partner's guide

Search Engine For Learning

Search Overview

Gooru aims to enable Personalized Learning. We would like every student to be able to leverage all the content, experts and community on the web to always learn based on their interest and at their pace. A central measure of success for Gooru is positive learning outcome for users at scale. We borrow from the efforts of semantic web community toward implementing a semantic learning web, where the web of content, experts and community (collectively called resources) annotated with machine readable metadata can be readily interpreted by systems to find, combine and act upon the metadata associated with the resources.

Core Search entails major functions of indexing, querying and ranking. Gooru has developed learning infrastructure with the content architecture and learning analytics. Gooru blends core search techniques with learning infrastructure to implement Suggest -- a personalized recommendation for every topic in any context.



One important use of Gooru Search is to seek recommendation of learning resources. This is used to navigate learners through personalized learning paths by suggesting learning resources, social objects and people in the community to interact. Gooru aims to recommend resources using:

1. Learning Architecture: Resources are tagged with Metadata, organized using taxonomy and enriched with user activity data.
2. Analytics: Tracking all activity stream data, and using an analytics pipeline to compute relevance, engagement, efficacy of resources, and proficiency of the learner.
3. Personalization: Extracting concepts and classification labels and associating them with learning resources, semantic processing of the query, and using user's learning path as a signal to rank results uniquely make Gooru search experience relevant.

Search

Gooru processes the queries in stages in the Query Processing pipeline. The stages are chained and can be configured to run in a different order or run in parallel.

Query Processing pipeline

Query processing pipeline is an assembly of various stages - e.g. expansion, rewrite, etc - of processing that is required on a given search call. On receiving a search call, the call goes through a Query Processing pipeline, and returns the result.

Query Expansion

Query Expansion processor expands the queries to improve recall.

Query Rewriting & Filter processing

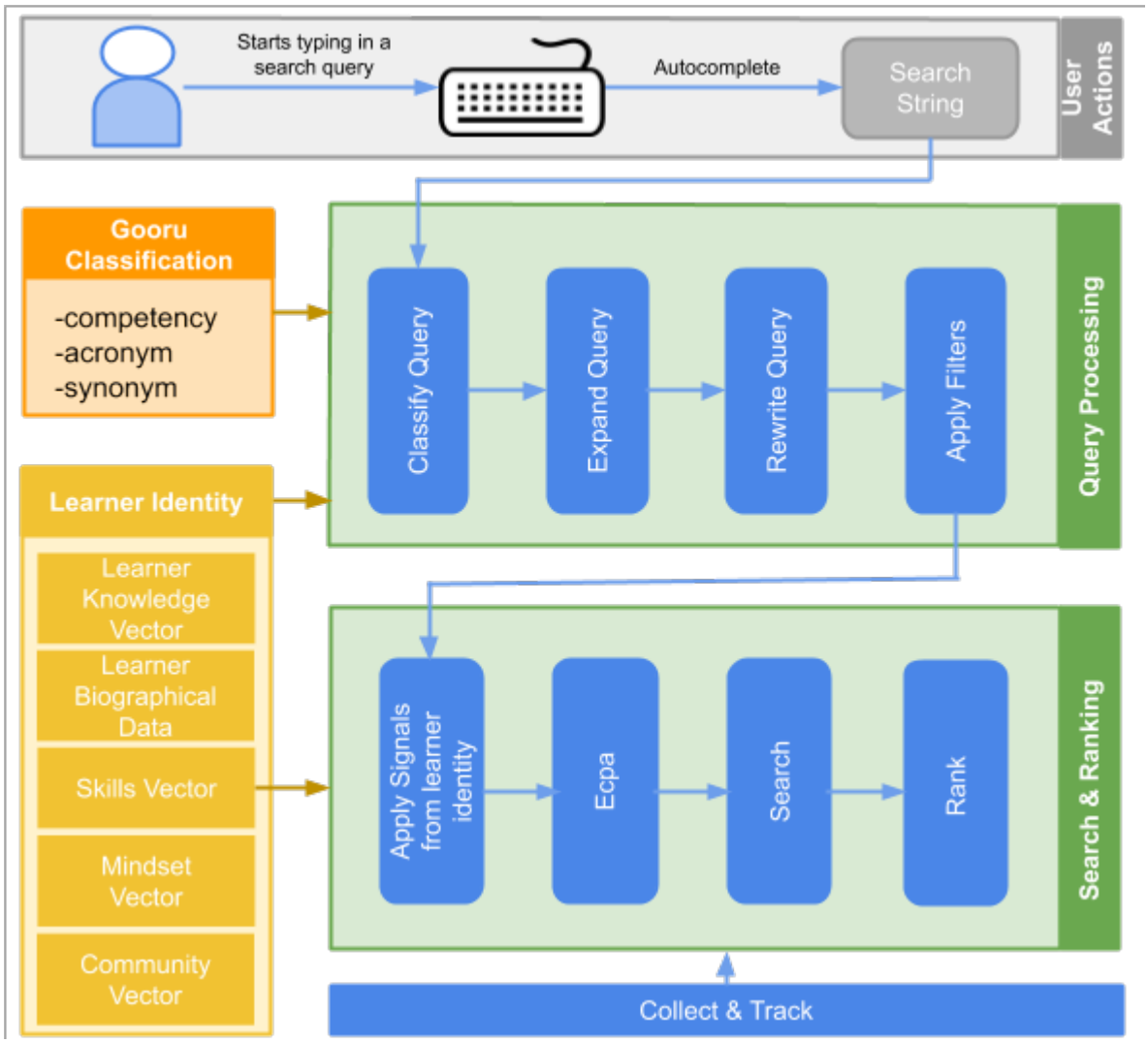
The queries are parsed and rewritten to treat different type of queries differently. Any potential filters detected are rewritten as filters too (see table below).

Table below lists some queries and the way they would be rewritten

Query	**Rewritten as**
genetics quiz	genetics category:quiz
Algebra videos	algebra category:video
quadratic system tools	quadratic system category:tools

Based on personalization signals, further filters may be added to the query to return the most relevant results for the user.

Search Flow Diagram



Ranking

The search results are ranked based on various signals about the content and/or the user. A partial list of signals used is provided below.

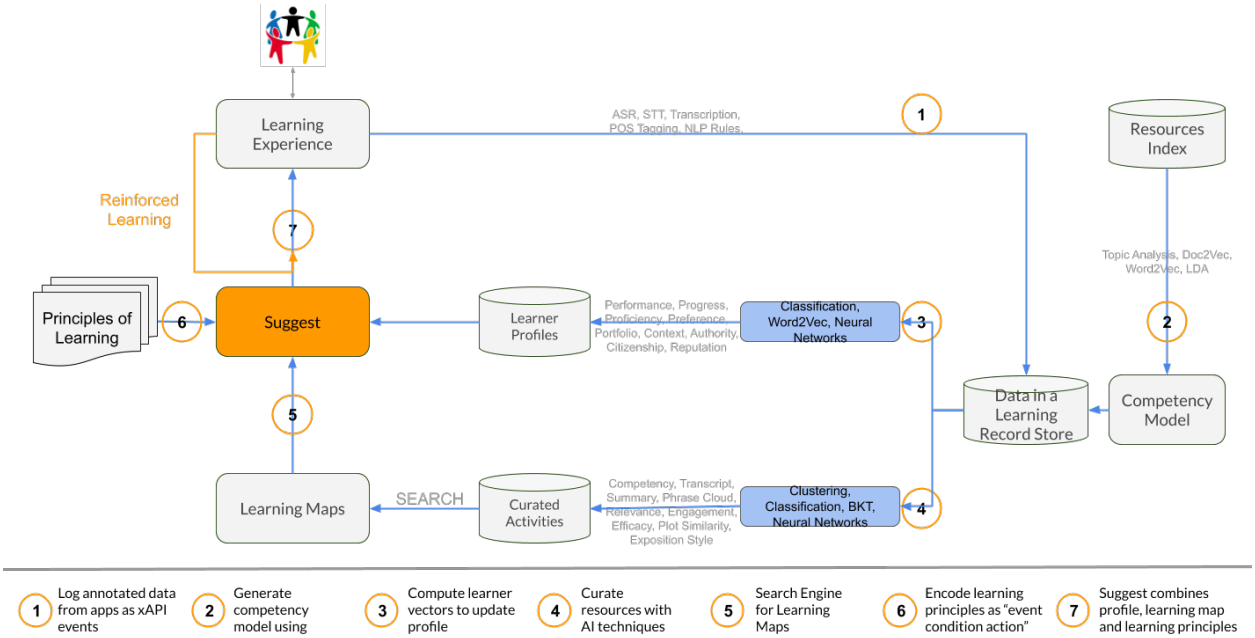
Search Result Processing

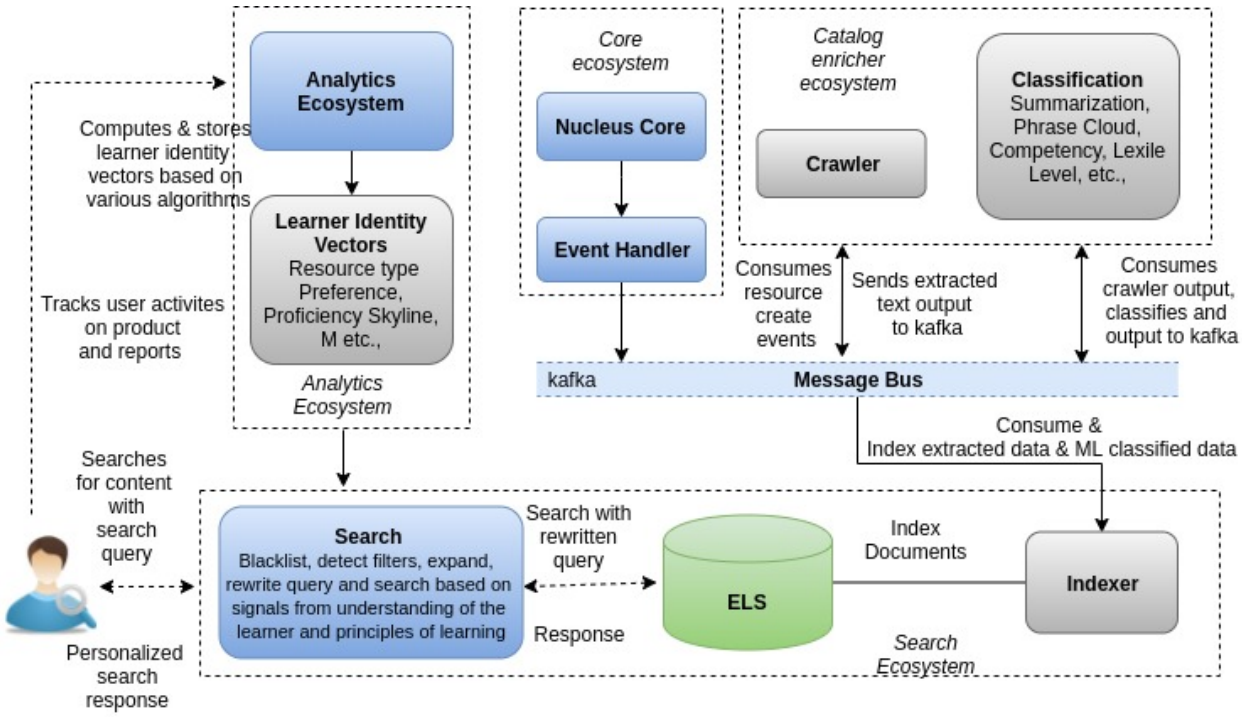
At the completion of search an event is sent on the data-bus for Gooru Insights infrastructure to consume.

The results returned from search are updated to include any look-up data, serialized and sent as a JSON back to the API caller.

Search response structure:
https://docs.google.com/spreadsheets/d/1-2Ph4pnOGaFY1WeKEnHdaHZ023nXlsd1tYy7WjyeAck/edit?ts=5b349ac5#gid=0

Personalized Search environment





Personalized search flow

Personalised search happens with the help of inputs from various environments of Gooru.

Core Environment

The resources appearing in our search are either created by Gooru admin or by teachers via our product content builder. Core Environment is where the different types of resource creations in our product are handled, and once they are created event handler produces the creation events and sends to kafka to notify other environments for consumption.

Catalog Enricher Environment

Catalog enricher is where the created resources or new resources are crawled/ recrawled and enriched with necessary metadata and using machine learning algorithms contents will be classified with additional data about the resource, which will be used by search.

Catalog enricher comprises of Crawler, Content analysis engine and Classification engine.

Crawler

Crawler consumes the creation events from kafka, crawls the url, extracts text and metadata of the resource and stores them to data store. Also it tracks the status of crawling. This engine is also responsible for recrawling to maintain the catalog with the latest metadata.

Content Analysis Engine

This engine checks the crawler status table and fetches the crawled output from crawler table for further processing for transcript, summarization and word cloud services.

Transcript Service uses the crawled output (extracted text) of normal resources, for other types like embedded sites, audio resources, etc., transcript service crawls, extracts text and stores them with completion status.
Summarization Service uses the output of transcript (final extracted text) to generate the overall summary of the resource and persists it.
Phrase Cloud Service with the help of transcript and summarization of the resource, generates keywords and persists with confidence of each.

Classification Engine

This engine checks the status of the content analysis engine for completion and starts the classification process with the help of content's own metadata and metadata obtained from content analysis.
Here, the resources are classified with taxonomy, lexile level and accessibility feature of the resource using various machine learning algorithms for each of them. After classification process is complete, the outputs are persisted and sent as event to kafka for search indexer to consume and update respective documents at index.

Analytics Environment

Analytics Environment tracks the user activity on our product serves the near real time reports. Computation includes micro-aggregations, periodic aggregations across various dimensions for different pivots like competency based performance reports, class specific reports, district based performance reports, etc.,

Analytics data points are useful to identity the learner's interests, proficiency, grade level of learner, etc., which can in turn help recommend personalized content for the learner.

Learner Activity & Identity Services

Here, based on user activity on our product and Learner Identity vectors are computed to identify the learners preference, proficiency, grit, perseverance, etc.,

Search Environment

When a user searches in our product, query is processed in stages of query processing pipeline which is an assembly of blacklisting non-educational query, query expansion processor, filter detection processor, query is rewritten.

Personalized search uses the inputs from api caller's biography, signals from other environments (signals used at search are mentioned below), includes necessary filters, ranks the response accordingly and returns the response to user.

Search Signals

Along with actual text relevance, below are the additional signals used for ranking.

Resource Ranking Signals

Gooru Resource Search uses the following list of content quality and usage signals for ranking.
Signals used for demoting

- Resources without descriptions or with description length less than 80.
- Resources without thumbnail.
- Resources which are frame breakers.
- Wikipedia resources are demoted slightly.

Signals used for promoting

- Resources used in many collections.
- Resources aligned with standards.
- Resources with more views.
- Resources tagged with 21st Century skills.
- Resources with editorial tags.
- Featured Resources.
- Library Resources.

Collection Ranking Signals

Gooru Collection Search uses the following list of content quality and usage signals for ranking.
Signals used for demoting

- Collections without descriptions are demoted.
- Collections without thumbnail are demoted.
- Collections without contents (resources/ questions) are demoted.

Signals used for promoting

- Collections with more views are ranked accordingly.

- Original collections (not copied) are ranked higher.
- Collections aligned with standards are ranked higher.
- Collections with resource count greater than 2 and less than 8 are ranked higher.
- Collections with at least one question are ranked accordingly.
- Featured Collections are ranked higher.
- Library Collections are ranked higher.

In addition to current signals used, below are the signals to be used for search ranking.

Content relevance/quality signals

Content's extracted/classified text relevance fields to use.

- Editorial tags
- Content Summary
- Phrase cloud
- Competency
- Lexile Level / Grade

Usage Signals

Rank contents with high REEf, with an inference that it will be of good use.

- Engagement
- Efficacy
- Relevance
- Resource Type Preference
- Skyline proximity with content's competency

User declared Signals

Ranking should consider api caller's content preference on language (user can declare language preference, default is english)

- Language preference
- Taxonomy preference - used for xform after crosswalk

Social Signals

Content author:

Rank authoritative contents slightly higher, with an inference that contents from this author will be of good quality.

- Authority
- Citizenship
- Reputation

Signals for Suggest

- Proficiency

- EcPA rules
- Grit
- Perseverance
- Motivation
- Self-Confidence

Catalog Enricher Environment

Crawl And Maintain Catalog

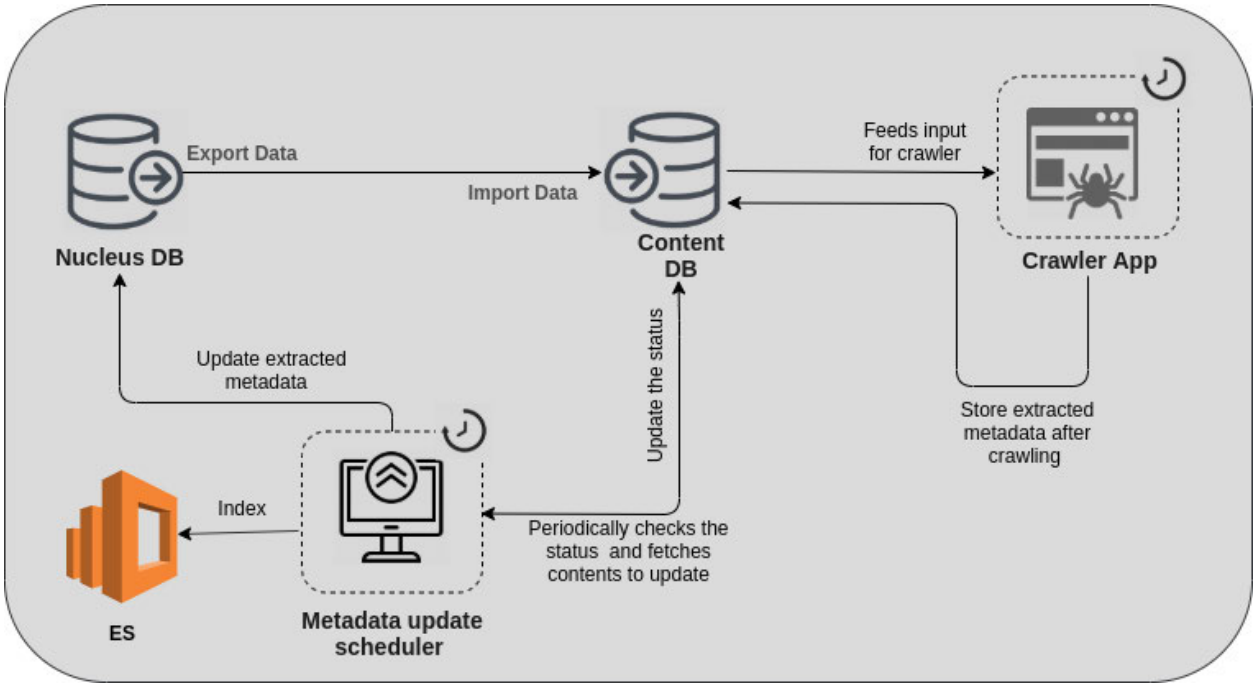
Crawler consumes the resource creation events from kafka, crawls the url, extracts text and metadata of the resource and stores them to data store. Also it tracks the status of crawling. This engine also handles the crawling and recrawling.

Crawl

Crawling includes the crawling of open resources from whitelisted domains, premium resources with the use of login credentials given by tenants and ingests these new resources to nucleus db.

Recrawl

To maintain the existing catalog, this engine recrawls periodically(less frequent) to update catalog with the latest metadata.



Input Data

Field	Sample Value
Resource url	https://www.livestrong.com/article/448466-tart-cherries-and-arthritis/

Output Data

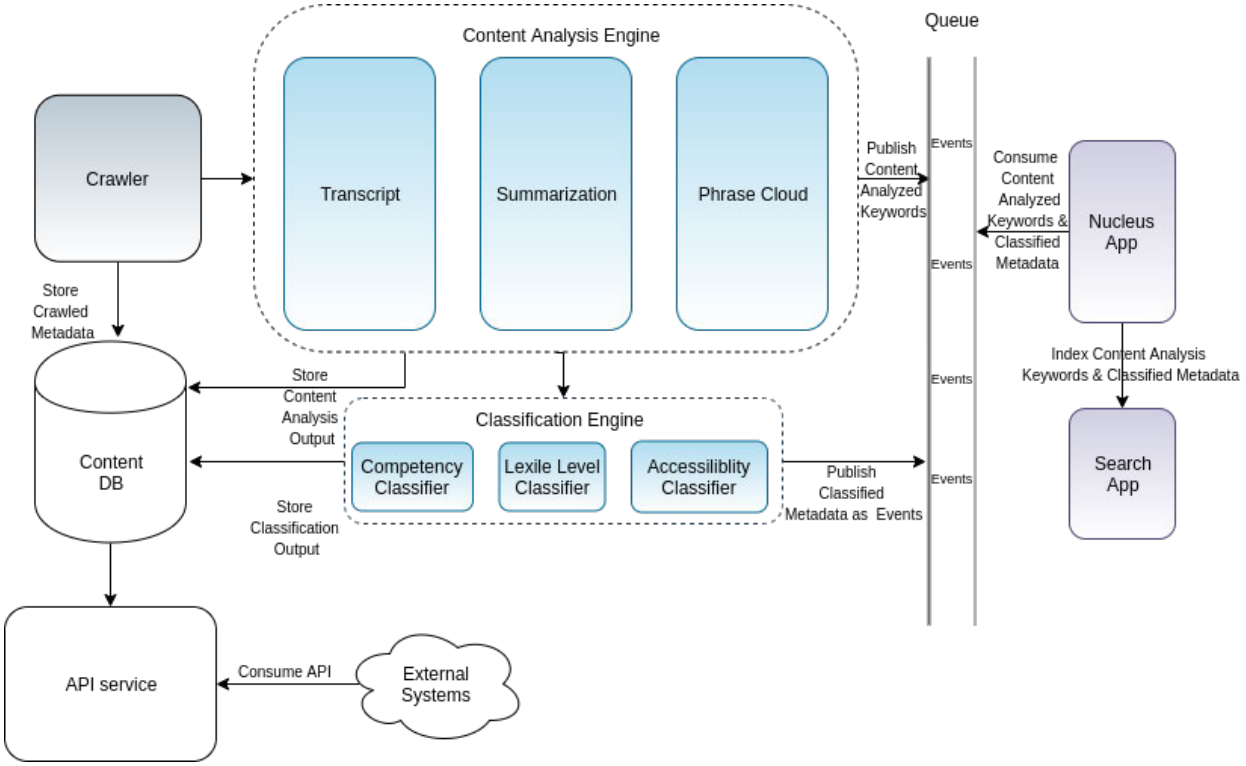
Field	Sample Value
Resource url	https://www.livestrong.com/article/448466-tart-cherries-and-arthritis/
Extracted Text	Arthritis is an inflammatory condition that affects the joints, causing swelling, stiffness, limited mobility and pain. PubMed Health says approximately 37 million people in the U.S. have arthritis, which equates to almost 1 out of every 7 individuals. Treatment for arthritis varies, depending on the type of arthritis you have. It can involve prescription drugs, over-the-counter medication, topical creams and herbal remedies. Diet might also play a role in relieving arthritic symptoms. Cherries have both anti-inflammatory and antioxidant properties that might help arthritis. Arthritis occurs when the cartilage of the joints breaks down. Without proper cartilage, which is connective tissue that protects the joint, the bones will rub together. This can cause stiffness, pain, pressure, redness and swelling. Arthritis can develop due to an injury such as a broken bone, autoimmune disease, normal wear and tear from physical activity, and from an infection. Sometimes, if the arthritic pain is being caused by an injury or an infection, once the condition heals, the arthritis will dissipate.
Reference links	[“https://www.livestrong.com/cat/nutrition/”, ”https://www.livestrong.com/scat/fruits-and-vegetables/”]

Metadata	{
	"title":"Tart Cherries and Arthritis",
	"description":"Arthritis is an inflammatory condition that affects the joints, causing swelling, stiffness, limited mobility and pain. PubMed Health says approximately 37 million people in the U.S.",
	"license":{
	"url":"https://www.livestrong.org",
	"code":null,
	"label":"PARTNER & LICENSEE OF THE LIVESTRONG FOUNDATION"
	},
	"author":[
	"Kimberly Riggins"
	],
	"primary_language":"eng",
	"advertisement_count":2,
	"copyright":[
	"LIVESTRONG"
	],
	"is_mobile_friendly":false,
	"subject":null,
	"grade":null,
	"publisher":[
	"LIVESTRONG"
	],
	"accessibility":{
	"accessibility_control":null,
	"accessibility_hazard":null,
	"accessibility_api":null,
	"access_mode":[
	"textual"
	]
	},
	"country_code":1
	}

Http Status	200

Curate Activities

Content Analysis and Classification pipeline is as illustrated below.



1. Transcript service reads the crawler index to create resource transcripts. If the crawled Index content has non-textual data like an embedded video or weblink, the transcript service will crawl the embedded content and creates transcript.
2. Output of the transcript will be used by the summarization service to create a textual summary from the transcript.
3. The Phrase Cloud service will extract important keywords from the transcript and summarization of the content. These keywords will be used by the Search Enviroment to enhance search results.
4. The output of each of these Services is stored and indexed so that it can be accessed asynchronously as need be.
5. Classification Engine will classify the resources based on competency, lexile levels and accessibility features using Machine Learning Algorithms.
6. The Metadata created by Resource Classification Engine will be used by the Search Enviroment to enhance search and boost rankings.
7. API Orchestration Service will enable partners, admins to access data generated by these services for their content.

Transcript

Transcript service reads crawled resources output from Content DB. Here is the sample input and output data.

Input Data

These are input data points transcript service needs from crawler service

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Crawled Text	Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.
Embedded Links	[]
Crawled Metadata	{"”}
CrawledHttp Status	200

Output Data

These are sample data of transcript service output

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Transcript Text	Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.

Summarization

Summary service reads the transcripted text and using Machine Learning libraries to create a summary of the resource. Here the details of the input and output data

Input Data

These are sample data of summary service input

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Transcript Text	Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.

Output Data

These are sample data of summary service output

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b

Summary Text	[
	{
	"text":"These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types",
	"score":0.8
	},
	{
	"text":"Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given",
	"score":0.7
	},
	{
	"text":"Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered",
	"score":0.6
	},
	{
	"text":"Compatibility of systems of linear constraints over the set of natural numbers",
	"score":0.5
	}
	]

Phrase Cloud

Phrase cloud service reads the summary text and using Machine Learning libraries to create a phrase cloud. Here the details of the input and output data

Input Data

These are sample data of phrase cloud service input

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b

Summary Text	[
	{
	"text":"These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types",
	"score":0.8
	},
	{
	"text":"Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given",
	"score":0.7
	},
	{
	"text":"Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered",
	"score":0.6
	},
	{
	"text":"Compatibility of systems of linear constraints over the set of natural numbers",
	"score":0.5
	}
	]

Output Data

These are sample data of phrase cloud service output

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Keywords	[{ "keyword":"systems", "occurrences":4, "score":1 }, { "keyword":"solutions", "occurrences":3, "score":0.8 }, { "keyword":"minimal", "occurrences":3, "score":0.8 }, { "keyword":"inequations",

```
                "occurrences":2,
                "score":0.6
        },
        {
                "keyword":"linear",
                "occurrences":2,
                "score":0.6
        }
]
```


Queue Event Structure

The idea is here to use the extracted important keywords of the resource to improve search results. In the context of the more relevant search results and boost the search results based on keywords score. So, content analysis engine will send out the processed results to message queue as an events. Sample event message will be as shown below


```
{
        "event.name":"resource.keywords",
        "data":[
                {
                        "keyword":"systems",
                        "occurrences":4,
                        "score":1
                },
                {
                        "keyword":"solutions",
                        "occurrences":3,
                        "score":0.8
                },
                {
                        "keyword":"minimal",
                        "occurrences":3,
                        "score":0.8
                },
                {
                        "keyword":"inequations",
                        "occurrences":2,
                        "score":0.6
                },
                {
                        "keyword":"linear",
                        "occurrences":2,
                        "score":0.6
                }
        ],
        "id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b"
}
```


Classification

Classification engine uses the output of content analysis engine(transcript, summary and phrase cloud) classify resource metadata such as competency, lexile level, accessibility feature. Here the details of the input and output data of this classification.

Input Data for Classification

These are sample input data for competency, lexile level and accessibility classification

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Transcript Text	Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.
Summary Text	[{ "text":"These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types", "score":0.8 }, { "text":"Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given", "score":0.7 }, { "text":"Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered", "score":0.6 }, { "text":"Compatibility of systems of linear constraints over the set of natural numbers", "score":0.5 }]
Keywords	[{"keyword":"systems","occurrences":4,"score":1}, {"keyword":"solutions","occurrences":3,"score":0.8}, {"keyword":"minimal","occurrences":3,"score":0.8}, {"keyword":"inequations","occurrences":2,"score":0.6}, {"keyword":"linear","occurrences":2,"score":0.6}]

Output Data - Competency

These are sample data of competency classification

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Subject	{"title":"Math","code":"MA"}
Course	{"title":"Algebra 2","code":"MA-A2"}
Domain	{"title":"Systems of Equations and Inequalities","code":"MA-A2-SEI"}
Competency	[{"code":"MA-A2-SEI-01.01","score":"1"}, {"code":"MA-A2-SEI-01.02","score":"0.9"}, {"code":"MA-A2-SEI-01.03","score":"0.8"}, {"code":"MA-A2-SEI-01.04","score":"0.8"}]

Output Data - Lexile Level

These are sample data of Lexile level classification,it's based on Flesch-Kincaid algorithm

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Lexile Level	{"score":89,"grades":"10,11"}

Output Data - Accessibility

These are sample data of Accessibility classification.

Field	Sample Value
Resource Id	d5a8b0e2-2ce1-11e3-9ab2-12314001917b
Accessibility	{"accessibility_features":["Audio","Text"]}

Queue Event Structure

The idea is here to use the classified values of the resource to improve search results. So, classification engine will send out the processed results to message queue as an events. Sample event message will be as shown below. These events will get consumed by Nucleus system and updates the resource with classified values, after the successful update of the resource search indexer will get triggered, it will pick the updated resource and re-index it. So, eventually classified value reaches the search index.


```
{
	"event.name":"resource.classification",
	"data":{
		"competency":[
			{
				"code":"MA-A2-SEI-01.01",
				"score":"1"
			},
			{
```

```
                        "code":"MA-A2-SEI-01.02",
                        "score":"0.9"
                },
                {
                        "code":"MA-A2-SEI-01.03",
                        "score":"0.8"
                },
                {
                        "code":"MA-A2-SEI-01.04",
                        "score":"0.8"
                }
        ],
        "lexile_level":{
                "score":89,
                "grades":"10,11"
        },
        "accessibility_features":["Audio","Text"]
    },
    "id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b"
}
```


Get APIs

Get Transcript

This API will return transcript of the given resource ID.

API Endpoint	api/ds/transcript/resource/{resourceId}
Method Type	GET
Request Param	

Parameter name	Parameter type	Is Required	Description
resourceId	Pathvariable	Yes	Id of the resource

| **Request Headers** | |

Name	Value	Description
Authorization	Token{auth-token}	Auth token obtained from the login API

| **Sample Response** | |


```
{
        "id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b",
        "transcript":"Compatibility of systems of linear constraints over
        the set of natural numbers. Criteria of compatibility of a system
        of linear Diophantine equations, strict inequations, and
        nonstrict inequations are considered. Upper bounds for components
        of a minimal set of solutions and algorithms of construction of
        minimal generating sets of solutions for all types of systems are
        given. These criteria and the corresponding algorithms for
        constructing a minimal supporting set of solutions can be used in
        solving all the considered types of systems and systems of mixed
        types. "
}
```

Http Response Code	

Code	Description
200	Ok - Success, returned expected results
403	Forbidden - User don't have access to use this API
404	NotFound - Transcript for given resource ID not found

Get Summary

This API will return summary of the given resource ID.

API Endpoint	api/ds/summary/resource/{resourceId}
Method Type	GET
Request Param	

Parameter name	Parameter type	Is Required	Description
resourceId	Pathvariable	Yes	Id of the resource

Request Headers	

Name	Value	Description
Authorization	Token{auth-token}	Auth token obtained from the login API

Sample Response


```
{
        "id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b",
        "summary":[
                {
                        "text":"These criteria and the corresponding
                        algorithms for constructing a minimal supporting
                        set of solutions can be used in solving all the
                        considered types of systems and systems of mixed
                        types",
                        "score":0.8
                },
                {
                        "text":"Upper bounds for components of a minimal
                        set of solutions and algorithms of construction of
                        minimal generating sets of solutions for all types
                        of systems are given",
                        "score":0.7
                },
                {
                        "text":"Criteria of compatibility of a system of
                        linear Diophantine equations, strict inequations,
                        and nonstrict inequations are considered",
                        "score":0.6
                },
```

<table>
<tr><td></td><td><pre>
{
"text":"Compatibility of systems of linear
constraints over the set of natural numbers",
"score":0.5
}
]
}
</pre></td></tr>
<tr><td>Http Response Code</td><td>
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>200</td><td>Ok - Success, returned expected results</td></tr>
<tr><td>403</td><td>Forbidden - User don’t have access to use this API</td></tr>
<tr><td>404</td><td>NotFound - Transcript for given resource ID not found</td></tr>
</table>
</td></tr>
</table>

Get Phrase Cloud

This API will return phrase cloud of the given resource ID.

<table>
<tr><td>API Endpoint</td><td>api/ds/phrase-cloud/resource/{resourceId}</td></tr>
<tr><td>Method Type</td><td>GET</td></tr>
<tr><td>Request Param</td><td>
<table>
<tr><th>Parameter name</th><th>Parameter type</th><th>Is Required</th><th>Description</th></tr>
<tr><td>resourceId</td><td>Pathvariable</td><td>Yes</td><td>Id of the resource</td></tr>
</table>
</td></tr>
<tr><td>Request Headers</td><td>
<table>
<tr><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td>Authorization</td><td>Token{auth-token}</td><td>Auth token obtained from the login API</td></tr>
</table>
</td></tr>
<tr><td>Sample Response</td><td><pre>
{
"id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b",
"keywords":[
{
"keyword":"systems",
"occurrences":4,
"score":1
},
{
"keyword":"solutions",
"occurrences":3,
"score":0.8
},
{
"keyword":"minimal",
"occurrences":3,
"score":0.8
},
{
</pre></td></tr>
</table>

<table>
<tr><td></td><td><pre>
                        "keyword":"inequations",
                        "occurrences":2,
                        "score":0.6
                },
                {
                        "keyword":"linear",
                        "occurrences":2,
                        "score":0.6
                }
        ]
}
</pre></td></tr>
<tr><td>Http Response Code</td><td>
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>200</td><td>Ok - Success, returned expected results</td></tr>
<tr><td>403</td><td>Forbidden - User don’t have access to use this API</td></tr>
<tr><td>404</td><td>NotFound - Transcript for given resource ID not found</td></tr>
</table>
</td></tr>
</table>

Get Classification

This API will return classification results of the given resource ID.

<table>
<tr><td>API Endpoint</td><td>api/ds/classification/resource/{resourceId}</td></tr>
<tr><td>Method Type</td><td>GET</td></tr>
<tr><td>Request Param</td><td>
<table>
<tr><th>Parameter name</th><th>Parameter type</th><th>Is Required</th><th>Description</th></tr>
<tr><td>resourceId</td><td>Pathvariable</td><td>Yes</td><td>Id of the resource</td></tr>
</table>
</td></tr>
<tr><td>Request Headers</td><td>
<table>
<tr><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td>Authorization</td><td>Token{auth-token}</td><td>Auth token obtained from the login API</td></tr>
</table>
</td></tr>
<tr><td>Sample Response</td><td><pre>
{"id":"d5a8b0e2-2ce1-11e3-9ab2-12314001917b",
        "competency":[
                {
                        "code":"MA-A2-SEI-01.01",
                        "score":"1"
                },
                {
                        "code":"MA-A2-SEI-01.02",
                        "score":"0.9"
                },
                {
                        "code":"MA-A2-SEI-01.03",
                        "score":"0.8"
</pre></td></tr>
</table>

```
                },
                {
                        "code":"MA-A2-SEI-01.04",
                        "score":"0.8"
                }
        ],
        "lexile_level":{
                "score":89,
                "grades":"10,11"
        },
        "accessibility_features":[
                "Audio",
                "Text"
        ]}
```


Http Response Code

Code	Description
200	Ok - Success, returned expected results
403	Forbidden - User don't have access to use this API
404	NotFound - Transcript for given resource ID not found

Navigator

Exploration Guide

What is Navigator?

Think of Navigator as a GPS for student learning. It locates each learner's current knowledge and skills and then designs a personalized route to their destination.

Using Navigator, teachers have access to thousands of curated and standard-aligned resources to enrich and support students' personalized learning journeys. And like any good GPS, Navigator uses data to adjust each student's path to their destination, making re-route suggestions in real-time.

What is this document?

This document is your map to Navigator. It will provide step-by-step instructions to a sample course called Math Navigator, allowing you to explore as both a student and a teacher. To access this information in Google Slides, click here.

Start exploring!

Go to gooru.org and log in using one of the demo accounts below. Once you log in, access the class listed for the account you have chosen. Feel free to explore on your own or by following the steps provided in this document. All we ask is that you do not delete any content or classrooms in the demo accounts. Happy exploring!

Demo Accounts

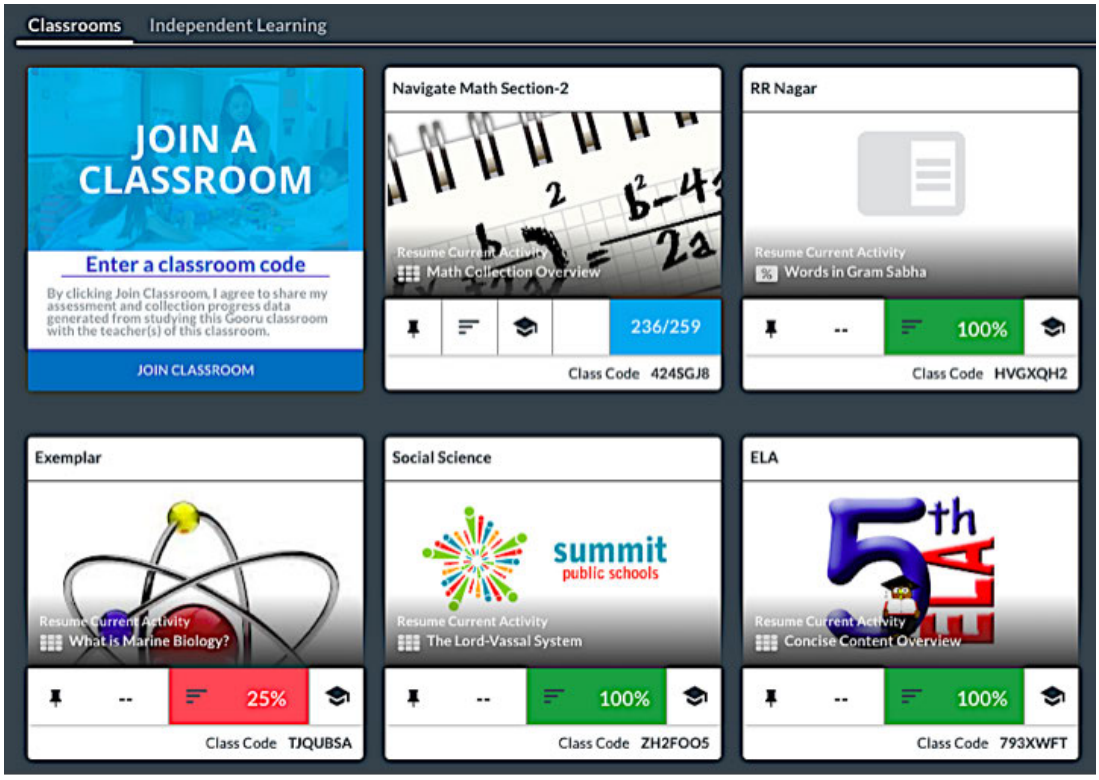
Student: Michael Caine	**Teacher**: Mr. Feeny
Account: Username: mcaine Password: learn123 **Class:** Math Navigator Demo (**95QJIMQ**)	**Account:** Username: MrFeeny Password: 12345 **Class**: Math Navigator Demo (**95QJIMQ**)

Note: Please do **NOT** delete any of the classrooms or content in these accounts.Thank you!

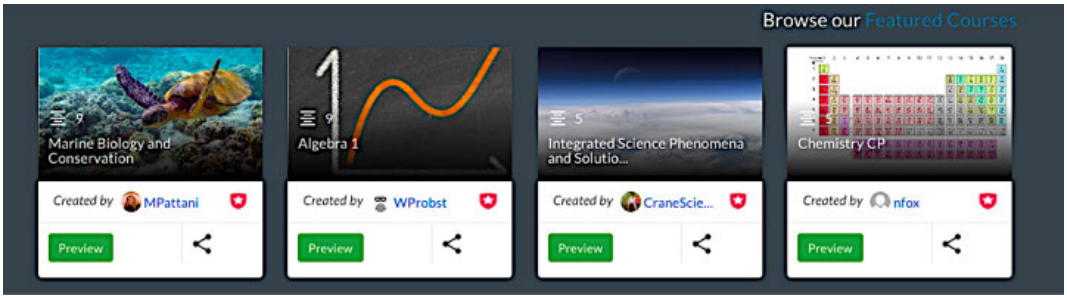
Exploring as a Student:

Overview:

Once you log in, you'll see the **My Study** page, showing all of the classes you're enrolled in. Enter any class by clicking on its picture.



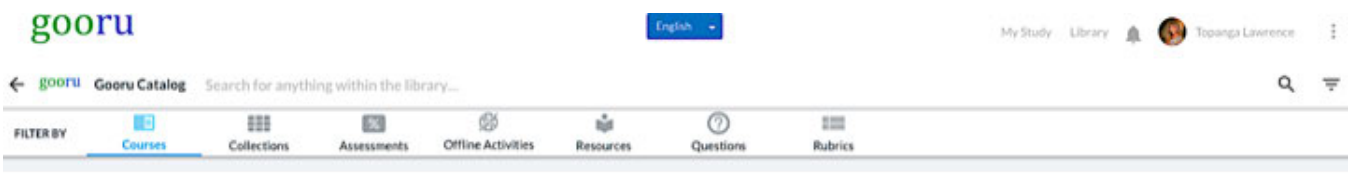
You can also **browse** through any course in Gooru's catalog. We call this **Content Discovery**. Simply click on the blue "Featured Courses" link and then the Gooru Catalog button.



Once you're in the Gooru Catalog, you can **search** for courses, collections, assessments, offline activities, and even individual resources and questions by typing in a keyword.

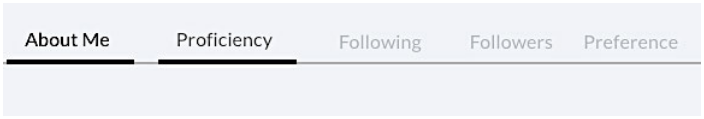
You can also join a course for independent study by previewing and clicking the green "Study" button.

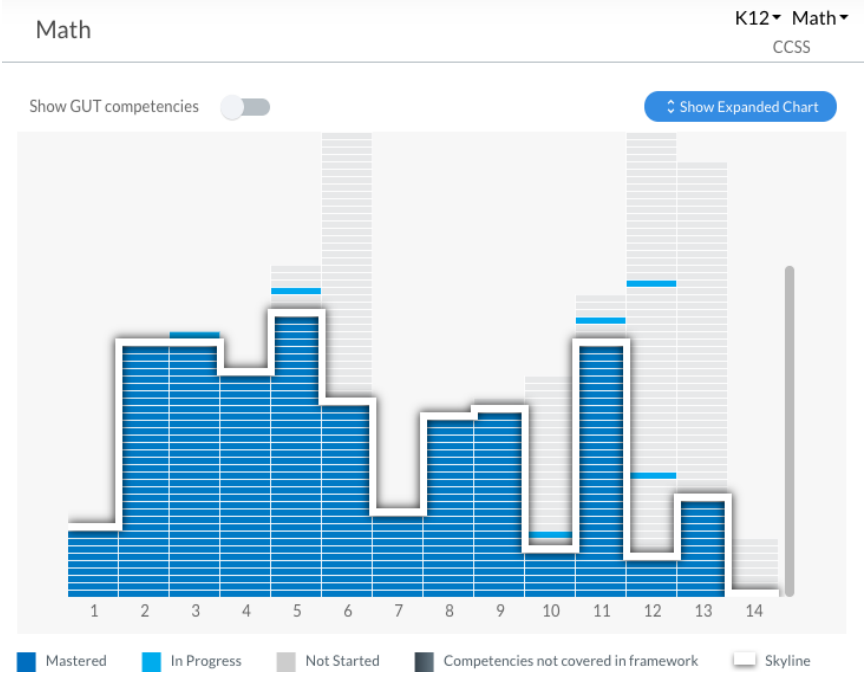
The catalog includes materials made by educators, pedagogy experts, researchers, and partner schools. It is always expanding, so check back often!



Your Profile:

Clicking on your name at the top right of the screen will open your **Profile** page. Here, you can adjust your settings and preferences. By selecting “Proficiency” you can view your **skyline**, which shows the highest competency levels you have achieved across all of a subject’s domains.





This is a sample math skyline. The white line shows the highest competency levels achieved so far. The numbers along the bottom represent each domain in the math curriculum.

Initial skyline data are derived from standardized assessments. We use these data to roster students and build their personalized learning journeys.

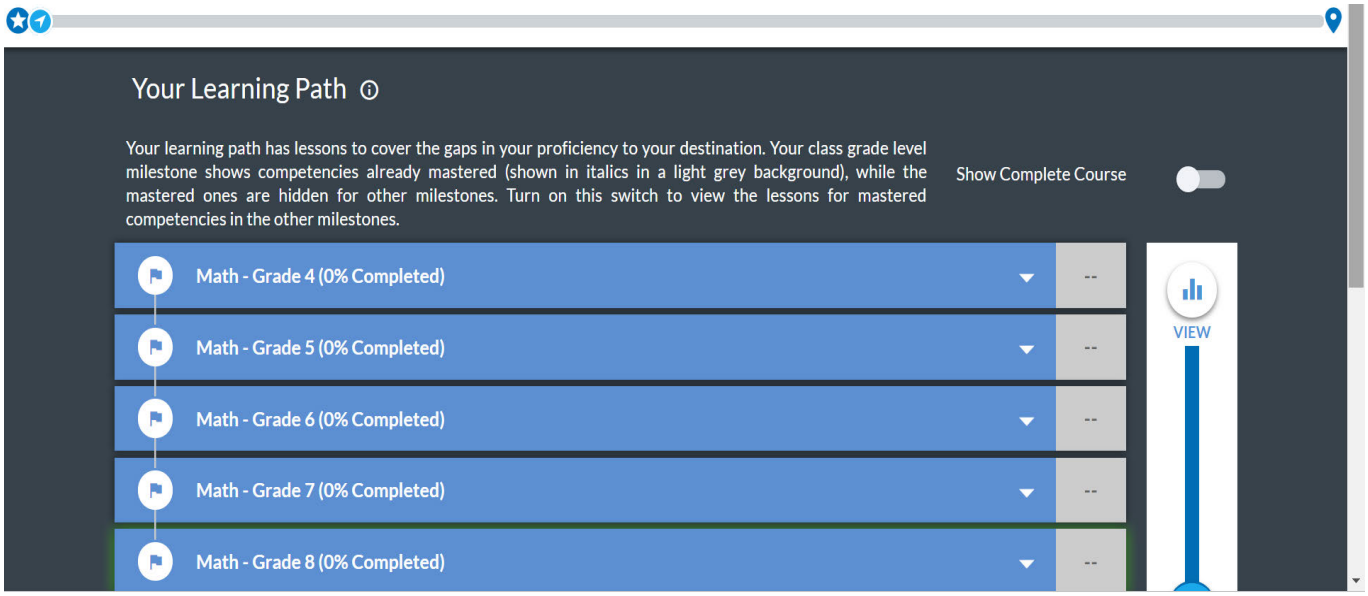
The skyline is **dynamic**, growing block by block as competencies are mastered along your learning journey.

Entering Math Navigator:

Enter Math Navigator by clicking on its picture on your My Study page. Inside, you’ll find three tabs at the top of your screen: **Class Activities, Course Map,** and **My Proficiency**.

Course Map:

Clicking on the **Course Map** tab brings you to your personalized learning journey. Learning journeys typically include several grade-levels of resources and content. It all depends on what your starting point is, what your skyline looks like, and what gaps need to be filled.

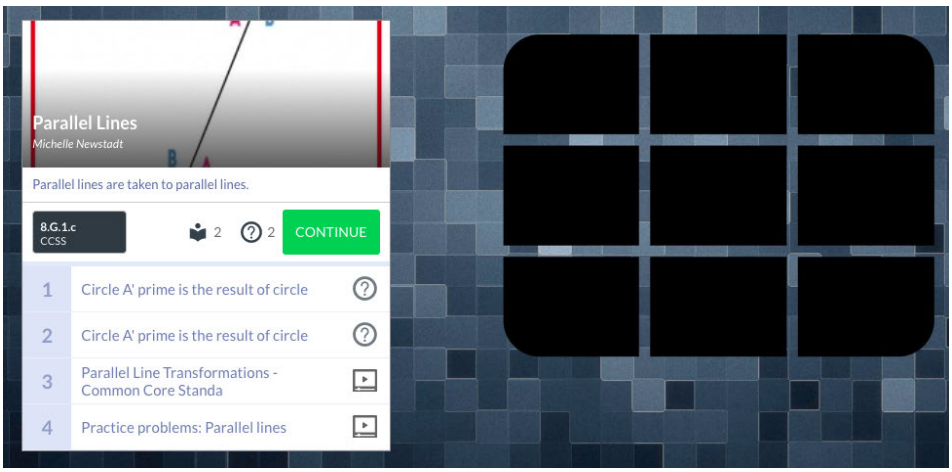


You can click on your Learning Journey to get a closer look at units, lessons, and even individual collections and assessments. You can also track your progress using the scores posted next to each grade level, unit, lesson, and assessment. Everything is color-coded to help you see what topics you might need to revisit.

To view a collection or work on assessment, just hover over it and click the play button. This launches your **Study Player.**

Study Player:

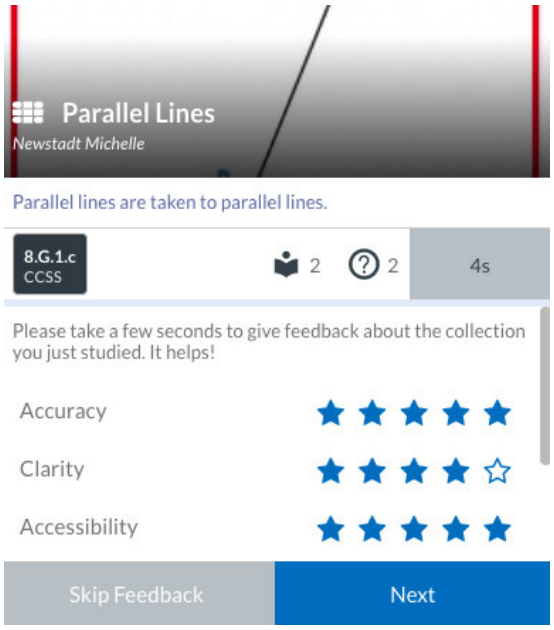
The **Study Player** launches collections and assessments in your learning journey. By entering from the Course Map, you can pick up right where you left off.



Once you select a collection or assessment, click the green “Continue” button to launch.

A collection can include a range of resources, including videos, interactive web pages, and slideshows. These resources have been curated to supplement what you are learning in class and to help you succeed on the assessment afterwards.

As you navigate through an assessment, you can let your teacher know how you feel about a question or concept by clicking on the emojis at the bottom of the screen. These range from “I can explain this topic” to “I need help.”

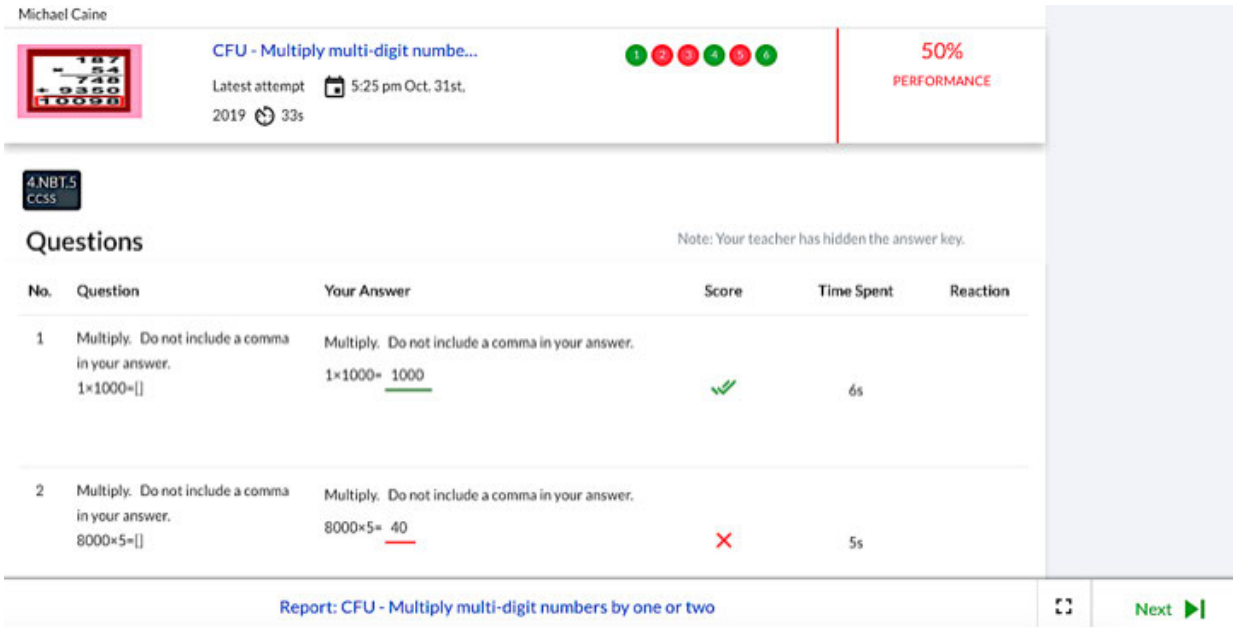


After each collection and assessment, you have the opportunity to provide feedback on its accuracy, clarity, and accessibility.

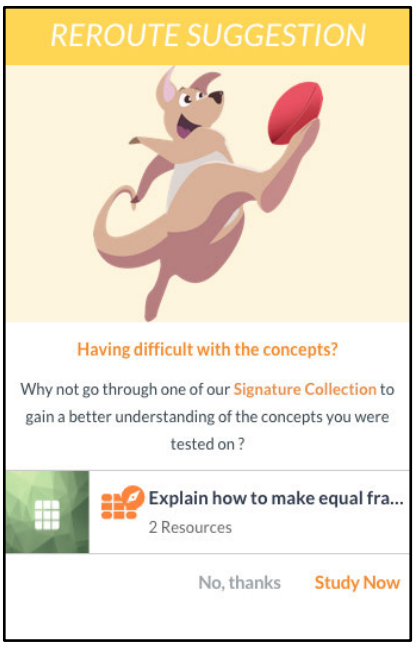
After you assign each category 1-5 stars, click the blue “Next” button. Gooru uses the data you provide to make improvements.

You also have the option to skip this step by clicking the gray “Skip Feedback” button.

Math Navigator will automatically give suggestions of additional resources and support after any assessment on which you score less than 80%. Think of this like a **re-route**, when your GPS adjusts to get you back on track.



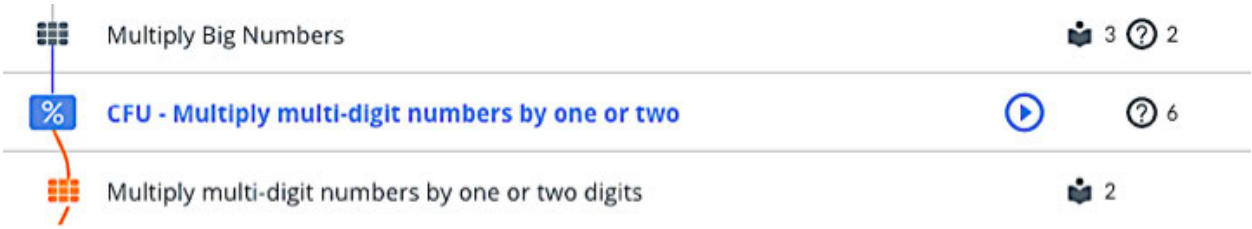
You can access these suggestions by clicking “Next” (on the bottom right of your screen) once you receive your score.



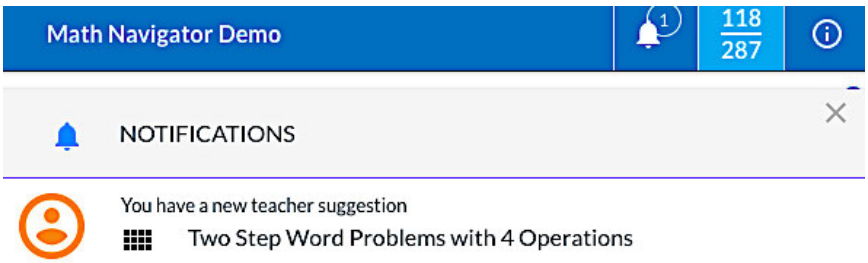
Reroute suggestions look like this. You can choose to take the suggestion (click “Study Now”) or not. In this example, you can see that the suggestion includes two resources on explaining how to make equal fractions.

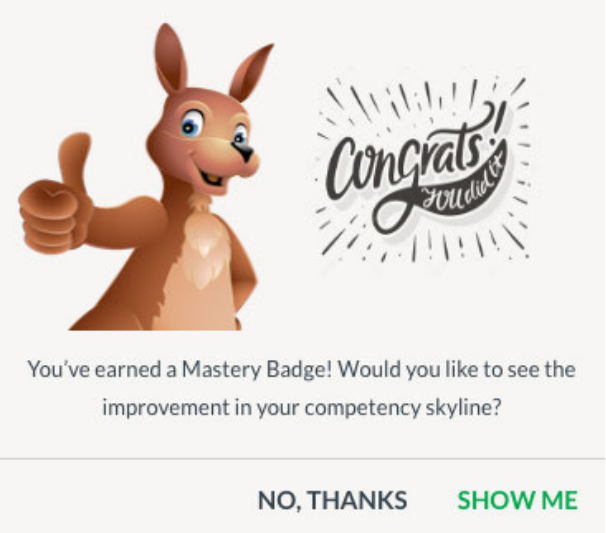
Once you work through the reroute suggestion, Math Navigator will bring you back to the assessment to give it another try.

Every reroute suggestion you take is added to your course map in orange, so you can keep track of the steps you have taken towards your destination.



Your teacher can use Math Navigator to offer you suggestions too. You can access these by clicking the bell icon in the upper right corner of your screen.



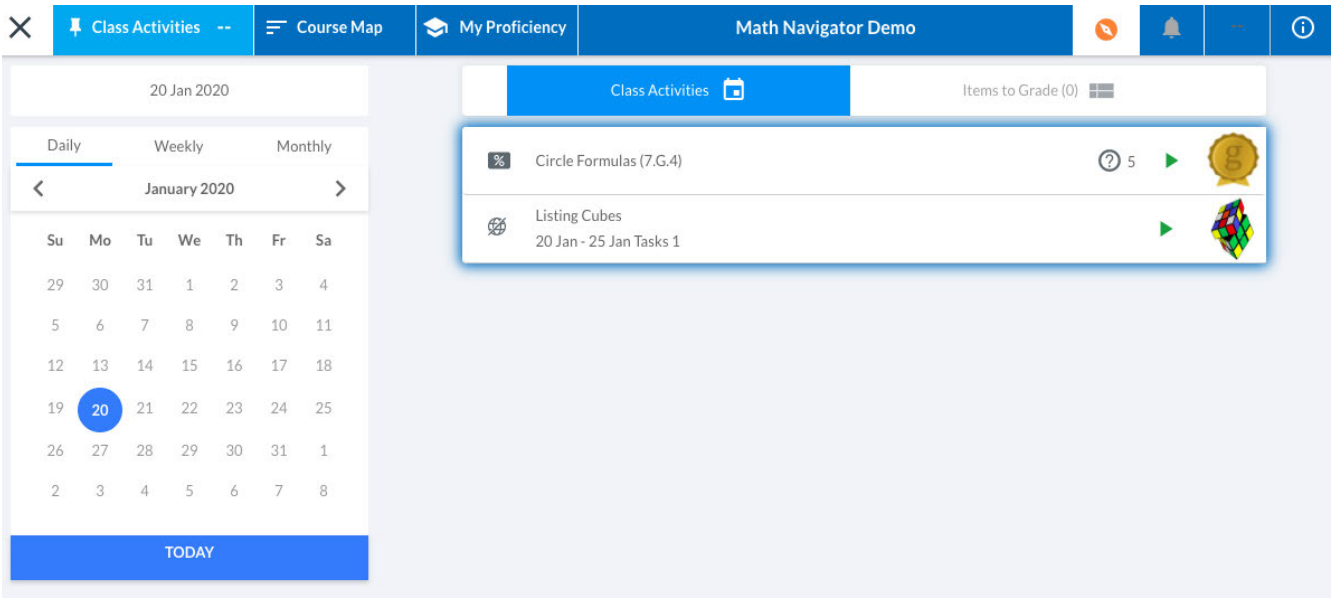


Scoring an 80% or higher on an assessment will earn you a Mastery Badge.

For each competency you master, a block is added to your skyline. Click “Show Me” (in green) to watch your skyline grow, bringing you one step closer to your destination.

Class Activities:

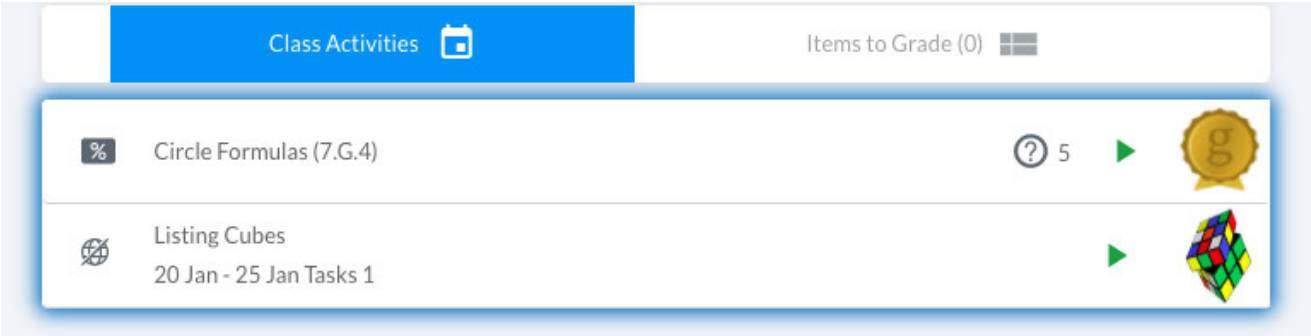
The **Class Activities** tab allows you to access activities your teacher has assigned to supplement your Learning Journey.



Your teacher can post resources, assessments, and offline activities on your Class Activities page.

The assessments can be scheduled to be completed in the future or in real-time. We call this **Going Live**; you can see your score as soon as the activity is over.

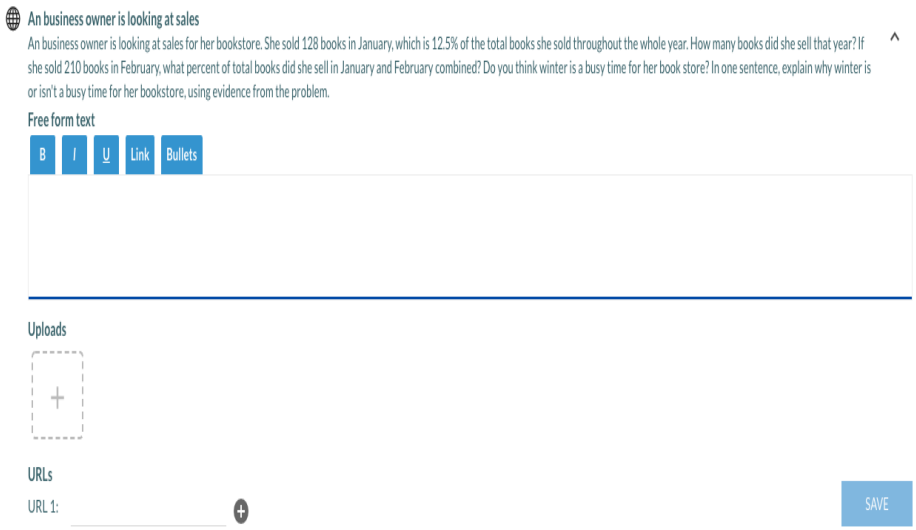
Some class activities are **Offline Activities**. In the example below, there are two Class Activities assigned: Circle Formulas (a five-question assessment) and Listing Cubes (an offline activity).



To begin work on any Class Activity, click the green play arrow.

Offline activities provide a typed explanation.

They also upload websites before grading.

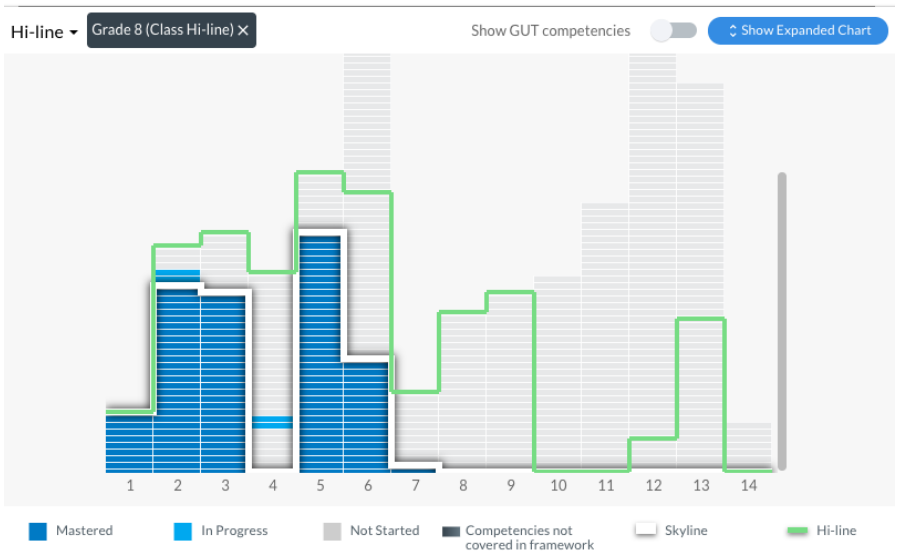


often require you to response or

enable you to attachments or link submitting them for

My Proficiency:

Clicking on the **My Proficiency** tab brings you to your skyline.



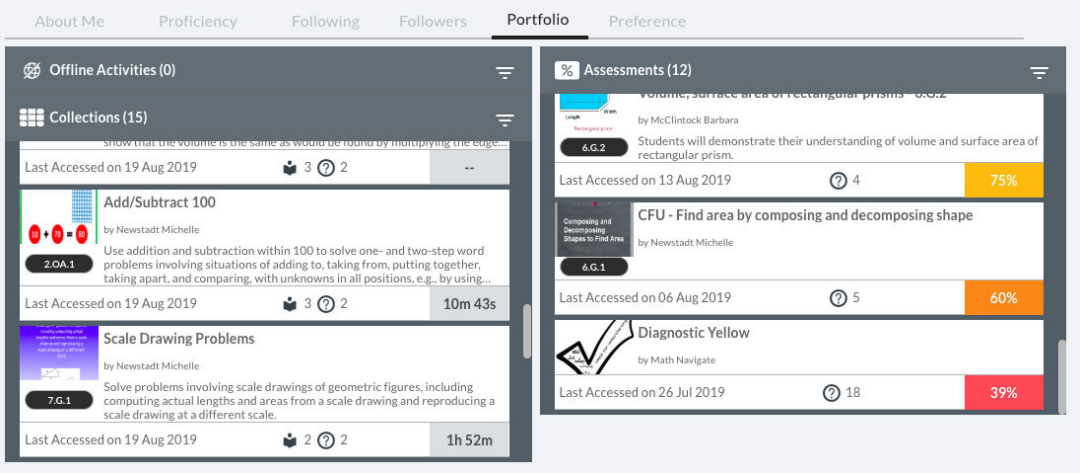
Your skyline shows you how many competencies you have mastered so far.

The green line is your **destination**.

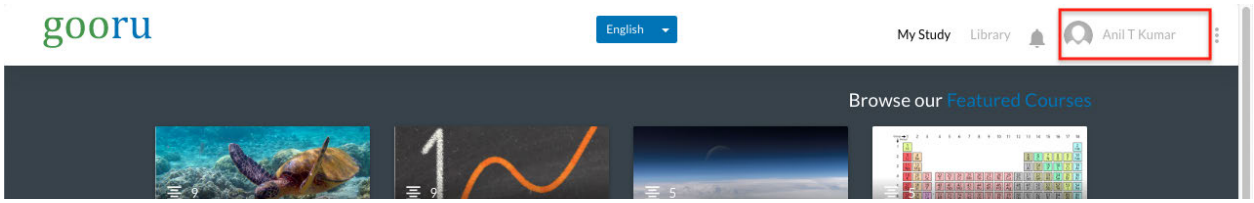
You can click the blue “Show Expanded Chart” to zoom into your skyline and click on individual blocks to find out what competencies they represent.

My Portfolio:

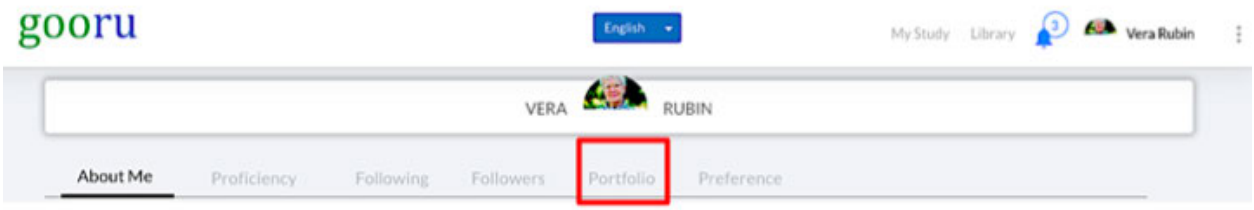
You can see all of the collections, assessments, and offline activities you have already completed by accessing your portfolio.



You can access your portfolio by clicking on your name in the upper right corner of the home screen.



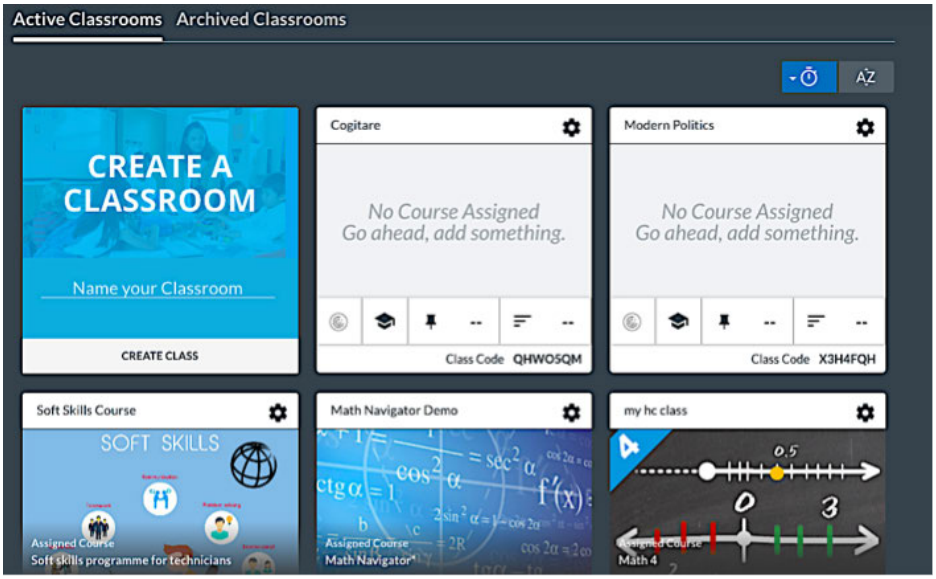
Then, click the “Portfolio” tab.



Exploring as a Teacher:

Overview:

Once you log in, you'll see your **Classrooms** page, showing your active and archived classrooms. Access any class by clicking on its picture. You can also create a new classroom from this page by clicking "Create Class."



Once created, each class has a **class code**.

One way to enroll students in your class is to give them this code. On the My Study page, students can enter the code and click "Join Class."

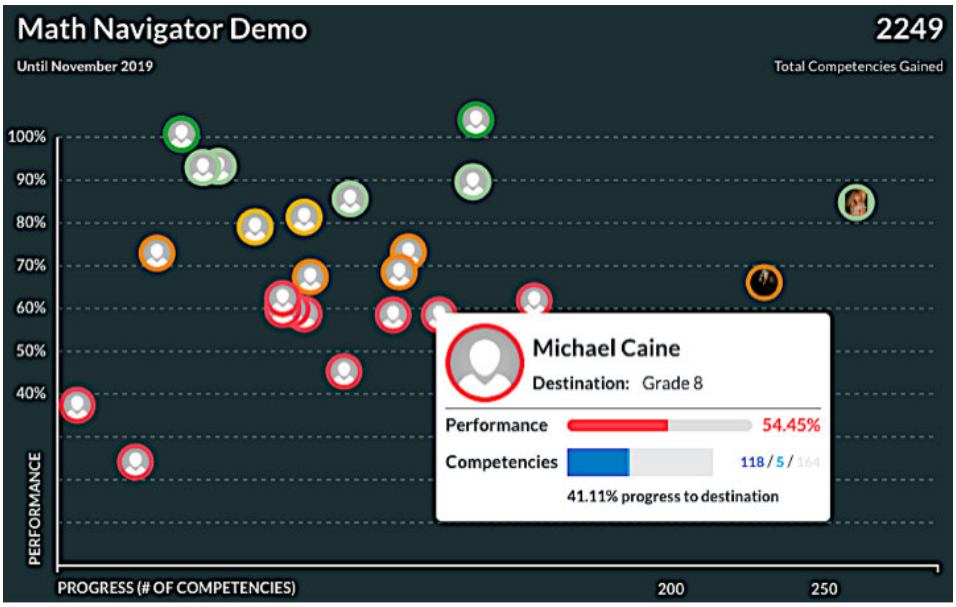
If Math Navigator is being implemented schoolwide, Gooru will roster all of your students and their data for you.

Entering Math Navigator:

Enter Math Navigator by clicking on its picture on your Classrooms page. Inside, you'll find four tabs at the top of your screen: **Performance Overview, Student Locator, Class Activities,** and **Learning Journey.**

Performance Overview:

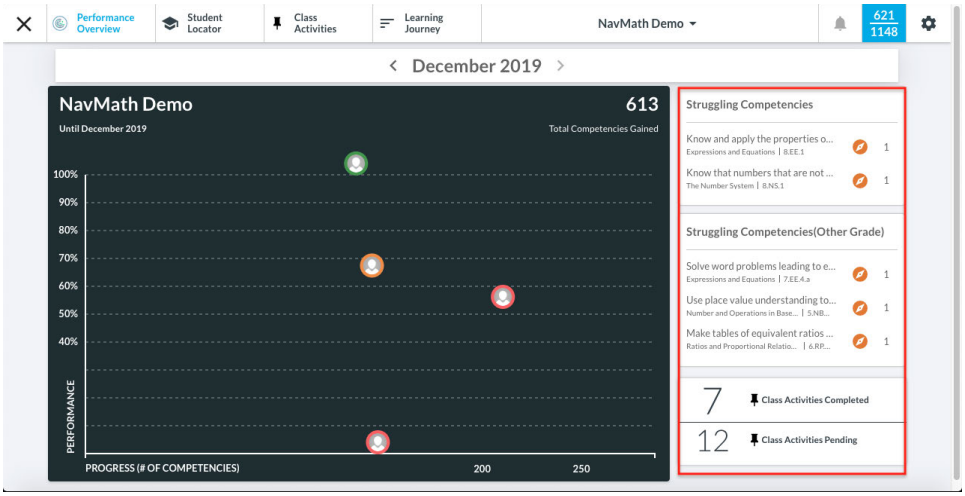
Clicking on the **Performance Overview** tab provides a quick snapshot of how your students are doing. You'll see a plot charting performance and progress for each student. The plot is dynamic, updating as your students work.



Hovering over a student's avatar will produce a pop-up with more details about the student's performance.

Performance is measured by the student's mean score on assessments, and progress is measured by the number of competencies mastered.

Competencies with which students are struggling (scoring <80%) are listed on the right side of the Performance Overview page. Suggestions can be assigned to struggling students directly from this box by clicking the orange suggestion icon.



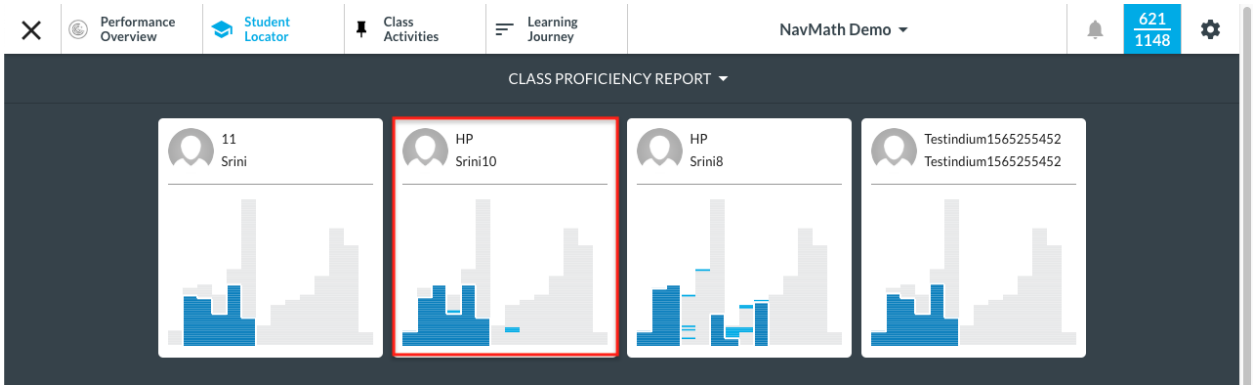
Making Suggestions:

Even the best drivers need to be rerouted once in a while. The same is true for students.

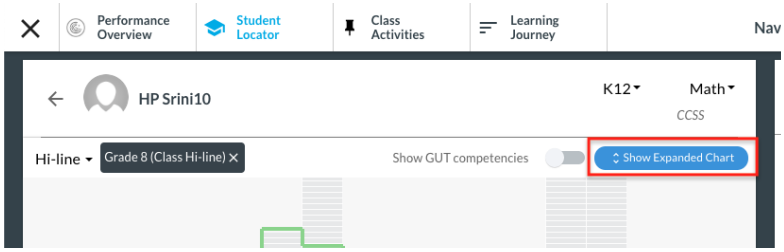
Gooru automatically makes suggestions of additional resources and assessments when students are struggling with a concept. You can choose recommended collections or assessments from a list of suggestions available in multiple locations:

Option 1: Single Student Suggestion via Skyline

Click on the “Student Locator” icon on the classroom card and then select the learner to whom you want to assign the resources or assessments.

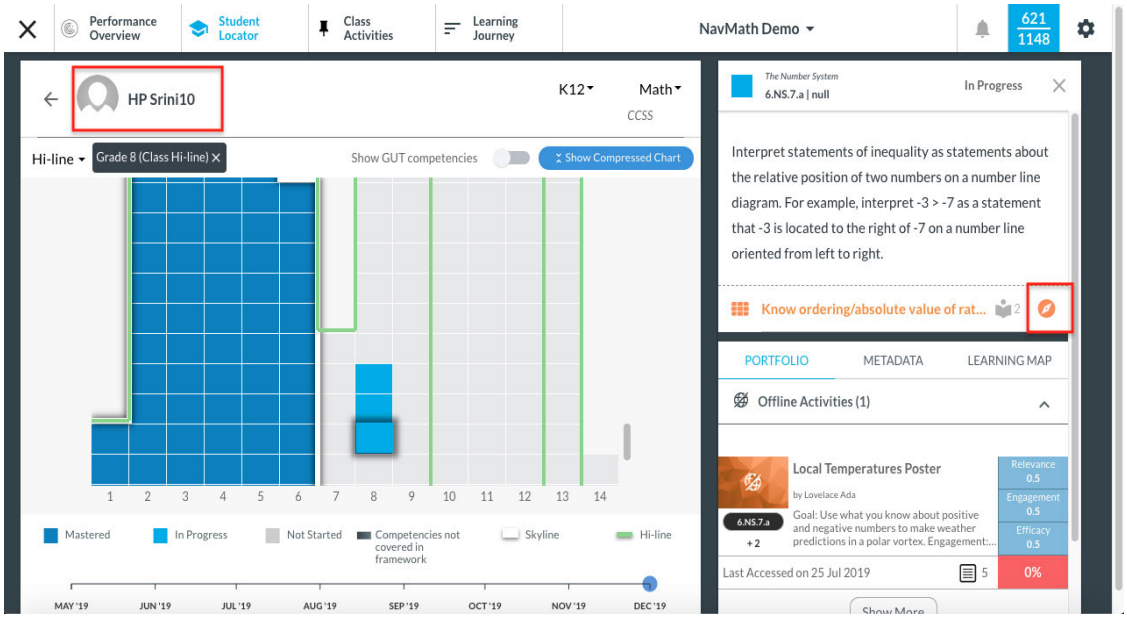


You can click on “Show Expanded Chart” button to get an enhanced view of the student’s skyline.



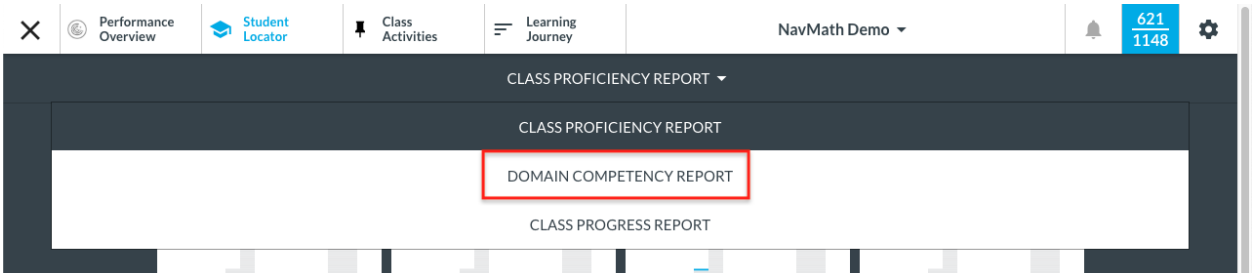
Then, select the competency from the skyline with which the learner needs more support by clicking on the corresponding block.

Click on the orange suggestion icon (on the right side of the screen) to send the suggested collection or assessment to the student.

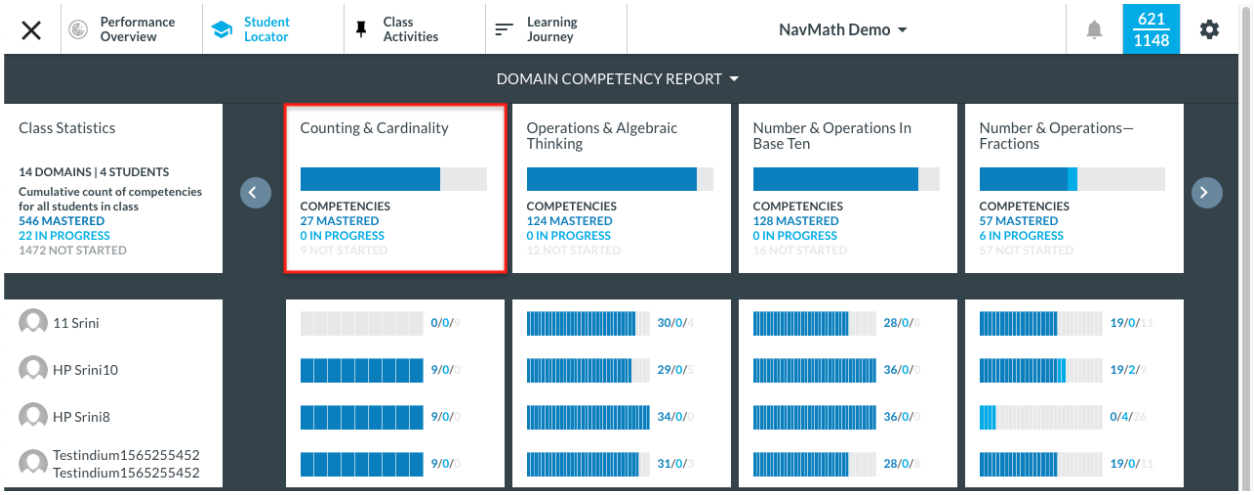


Option 2: Student Suggestion by Domain

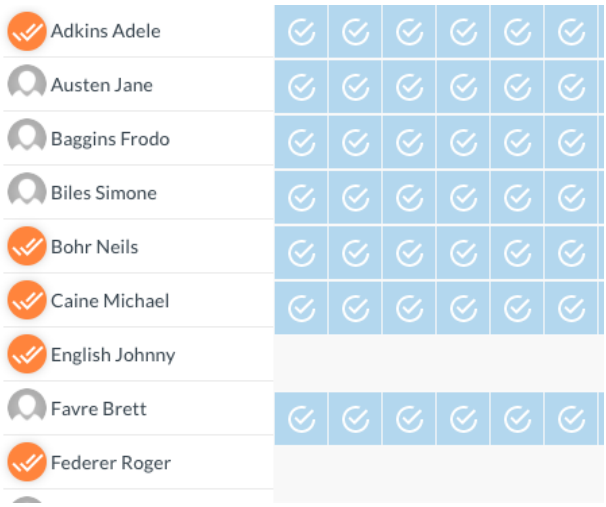
Click on the “Student Locator” icon on the classroom card and then select “Domain Competency Report” from the dropdown menu.



Select a domain. You can scroll through the course’s domains using the left/right arrows on the screen.

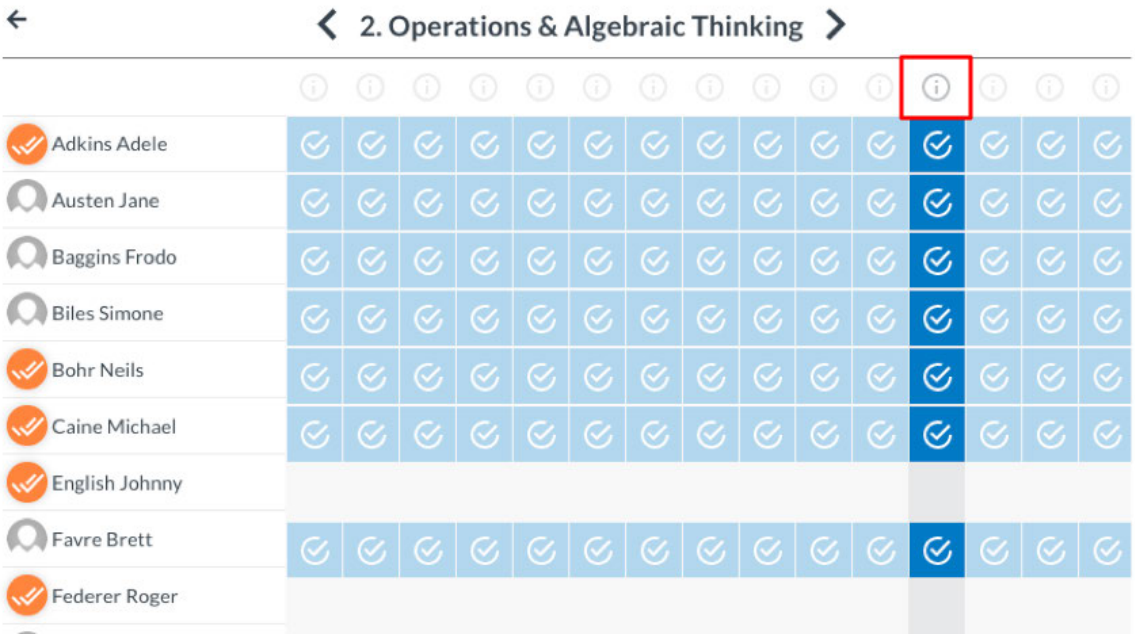


Identify which students you would like to receive suggested supplemental resources by clicking on their avatars. Orange check marks indicate that a student has been selected.

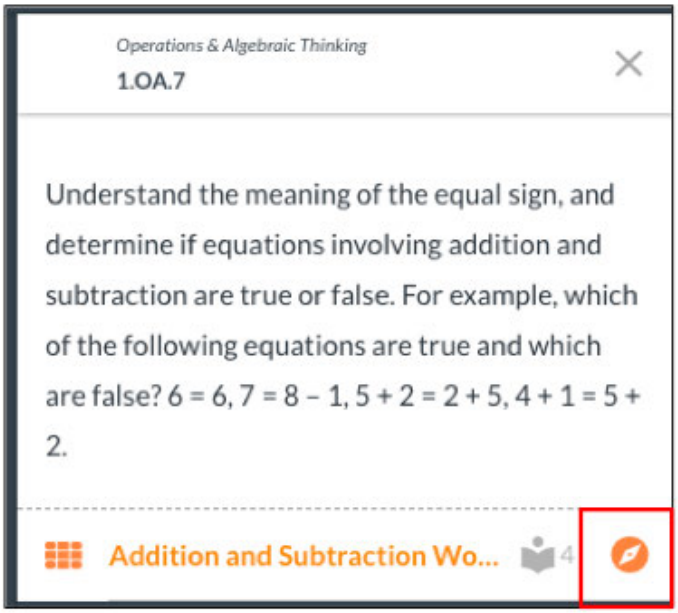


In this example, Adele, Neils, Michael, Johnny, and Roger have all been selected to receive the suggestion.

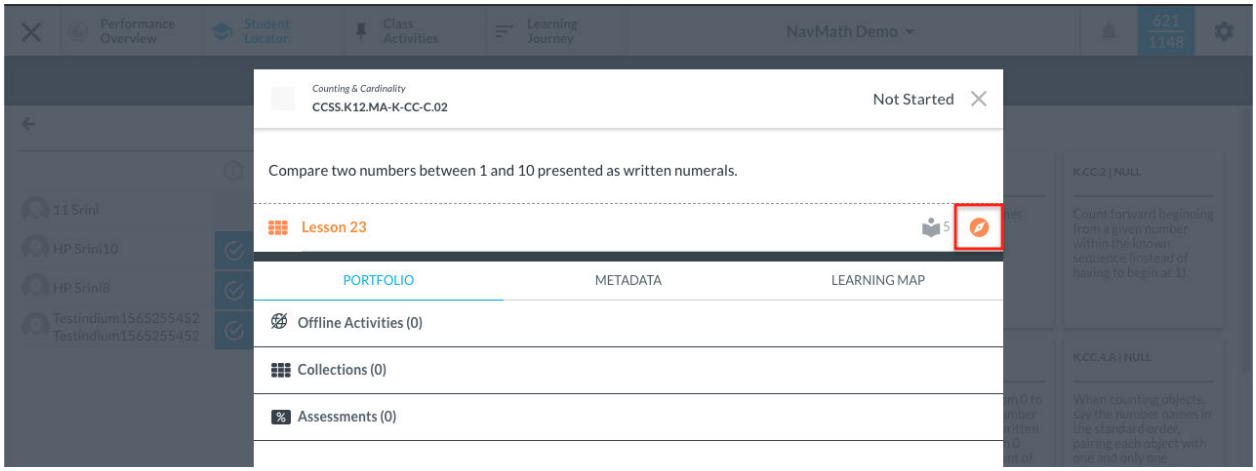
Hover over each “i” symbol to get a description of the competency. Click on the “i” of the targeted competency. This will highlight the column in dark blue (shown below).

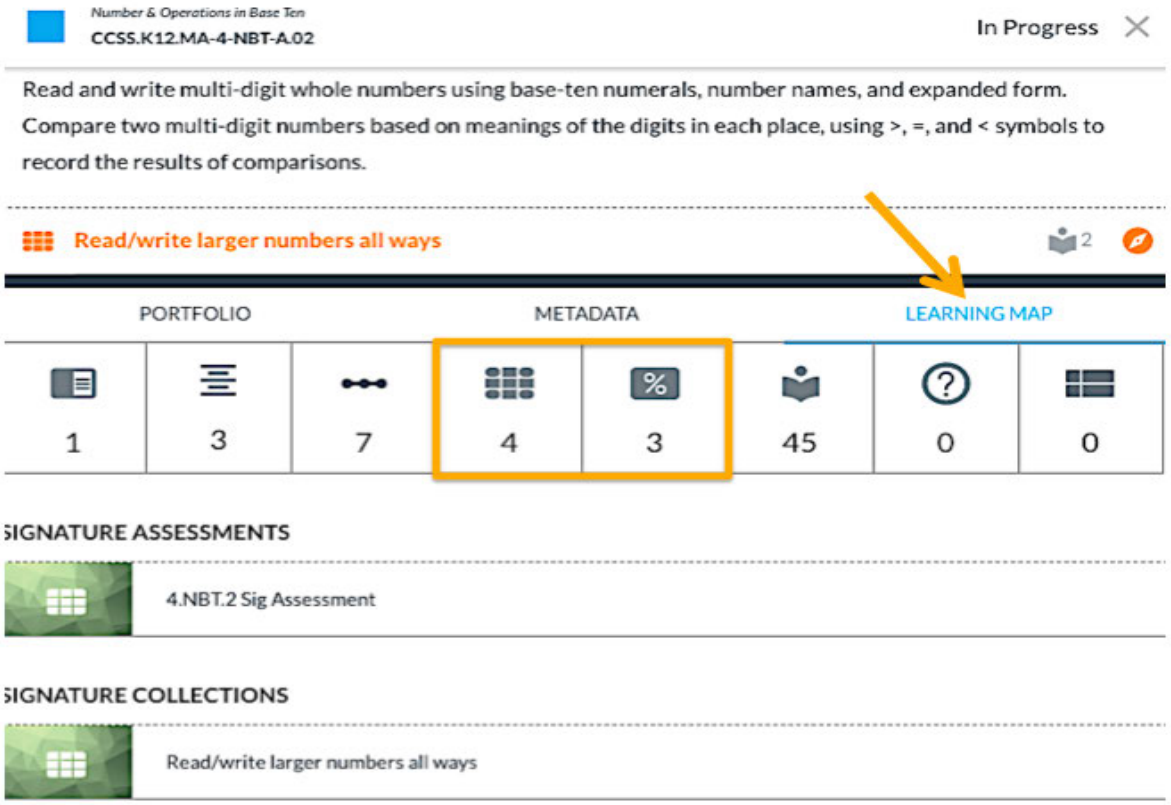


On the right side of the screen, click on the orange suggestion icon.



In the pop-up window, click on the orange suggestion icon to assign the additional resources and assessments to the cohort of students you selected.



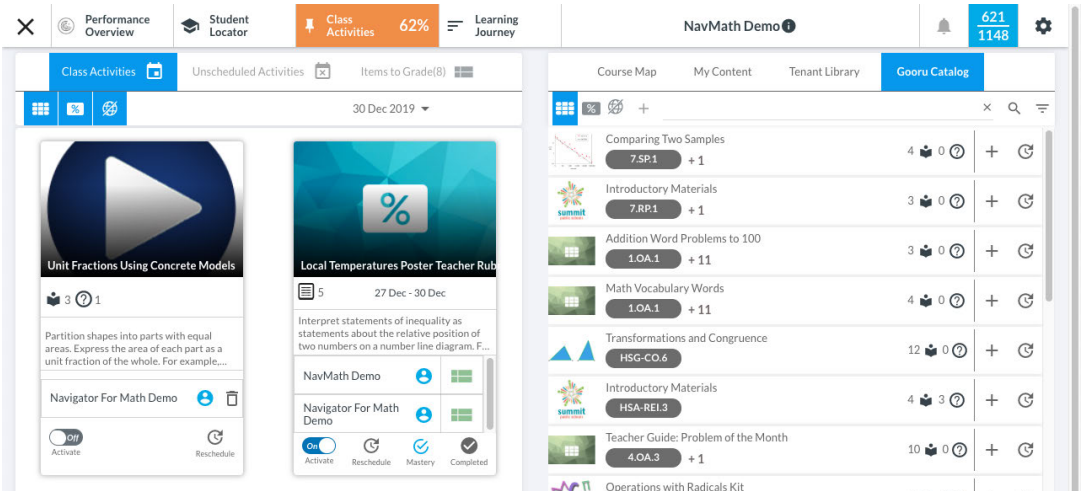


You can also select the "Learning Map" tab in the pop-up window, and then click on the collection or assessment tabs (highlighted in the image to the left) to assign any collection or assessment from the list to the student.

Class Activities:

Clicking your **Class Activities** tab brings you to your activities dashboard, where you can search for and assign collections, assessments, and offline activities.

Class activities are a great way to formatively assess understanding, fill in gaps, and challenge accelerated learners. You can use all or part of an existing activity from Gooru's catalogs or you can create your own from scratch.



Once posted, Class Activities will appear on the left side of the page. Class Activities can be added from multiple sources to fit your needs:

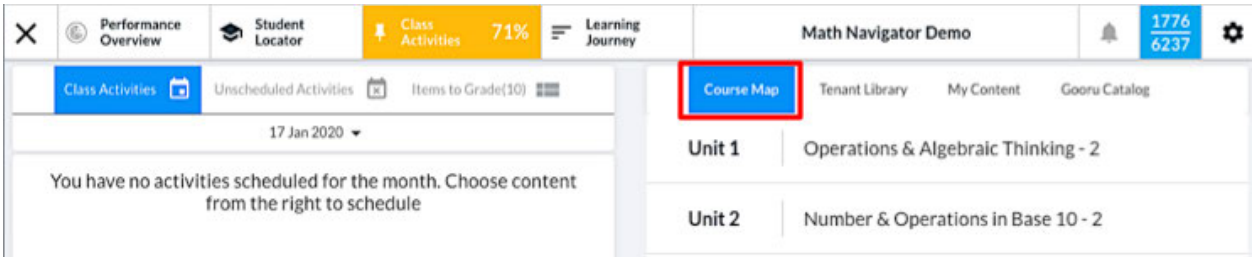
Option 1: Assigning Class Activities from the Course Map

The Course Map allows you to assign collections and assessments from anywhere in the course. This is an easy way to find gaps in students' understanding.

For instance, if you are teaching 6th grade but you're not sure how well your students really know a particular 2nd grade standard, select a 2nd grade assessment from the course map and assign it as a class activity. This is a great way to assess prior knowledge and to help kids fill in gaps.

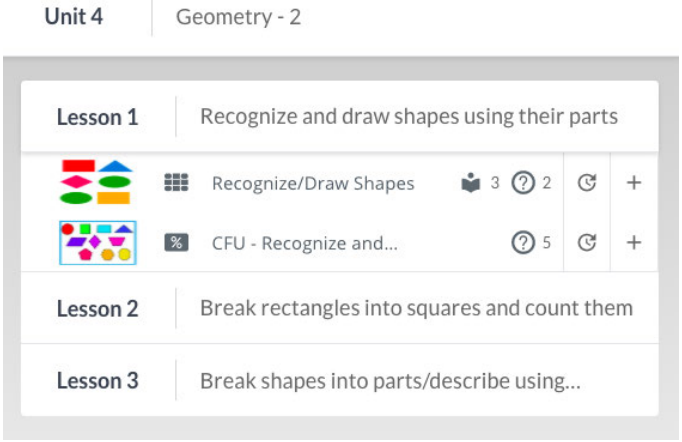
To view the course map, click on the "Class Activities" tab at the top of the classroom page.

On the right side of the page, select “Course Map.” This will surface <u>all</u> of the units in the course.



Click on a unit and a lesson to view the associated collections and assessments.

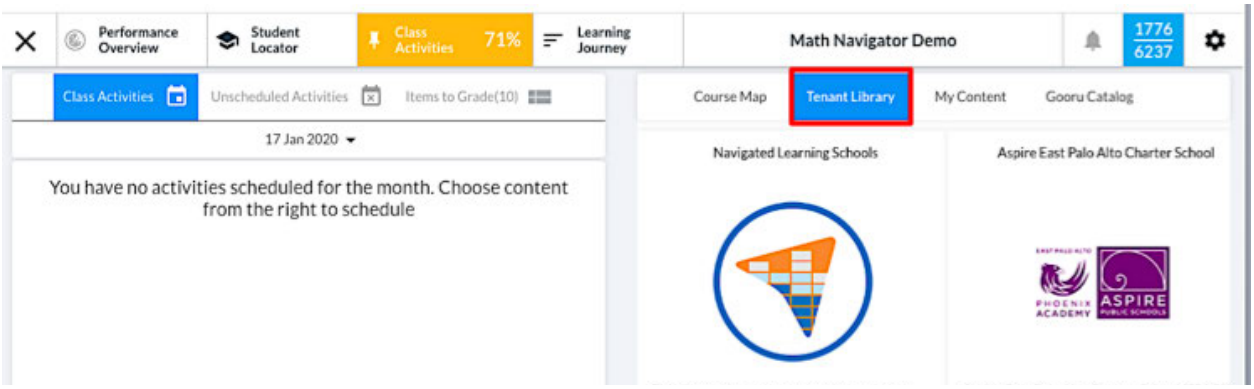
To preview the collection or assessment, click on its name. To assign, click the “+” icon. To schedule for later, click on the clock icon and select a date.



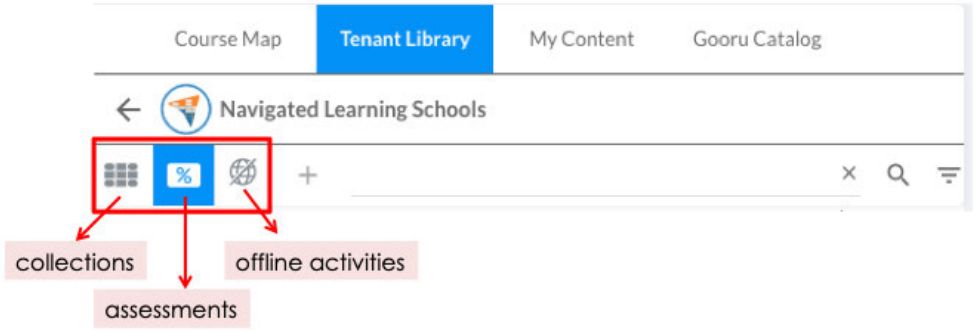
Option 2: Assigning Class Activities from the Tenant Library

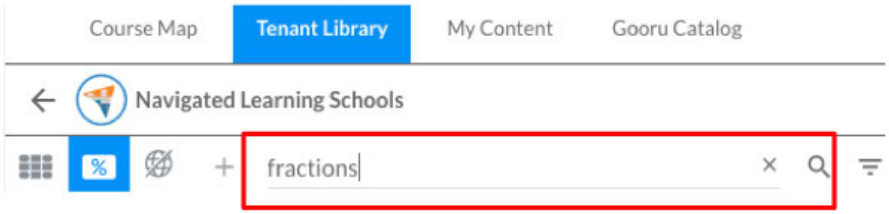
The Tenant Library is a set of curated resources made by educators and Gooru content creators. It contains assessments that are not included in the Course Map.

On the right side of the page, select “Tenant Library”. Choose the desired library from the cards shown.

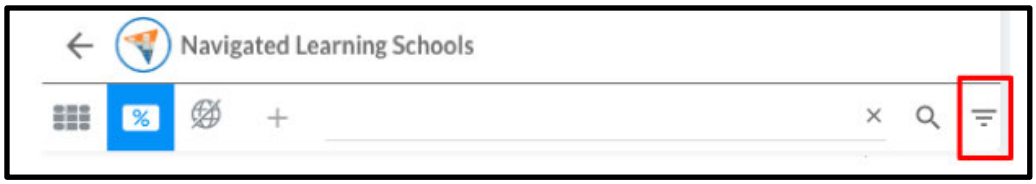


Use the icons (shown below) to narrow the search by activity type.

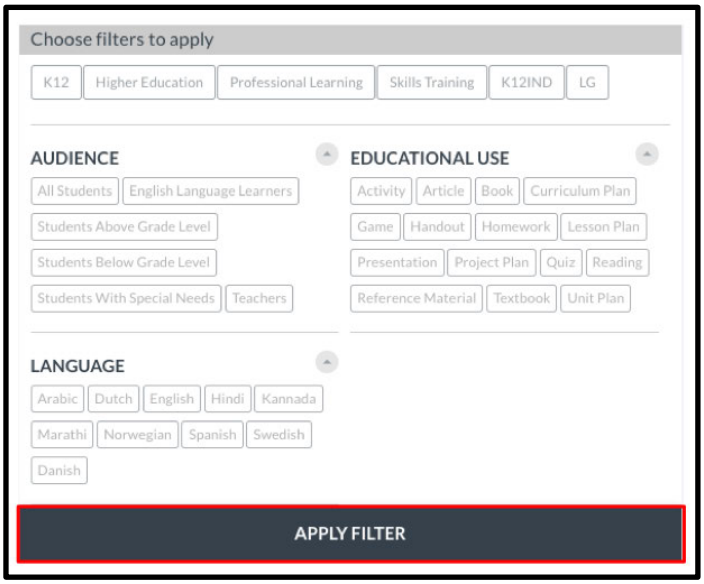




Use the search bar to narrow results.

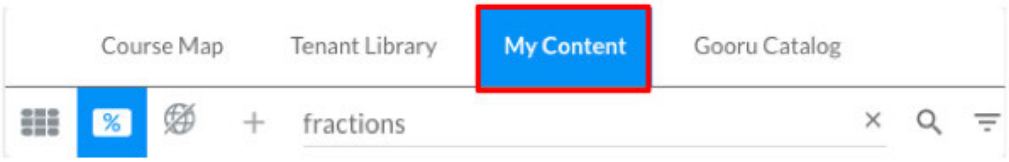


You can narrow the results even further by clicking the filter icon. Select a framework and competency, and click “Apply Filter”.



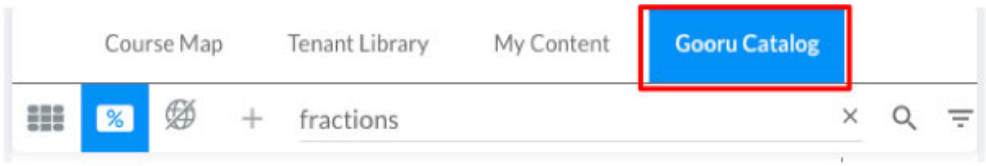
Option 3: Assigning Class Activities from My Content

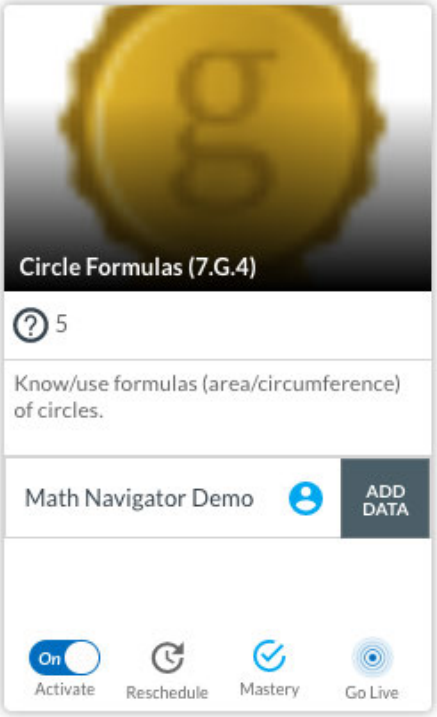
“My Content” includes collections and resources that you have already created or edited.



Option 4: Assigning Class Activities from the Gooru Catalog

The Gooru Catalog includes *all* collections and assessments created by Gooru users.



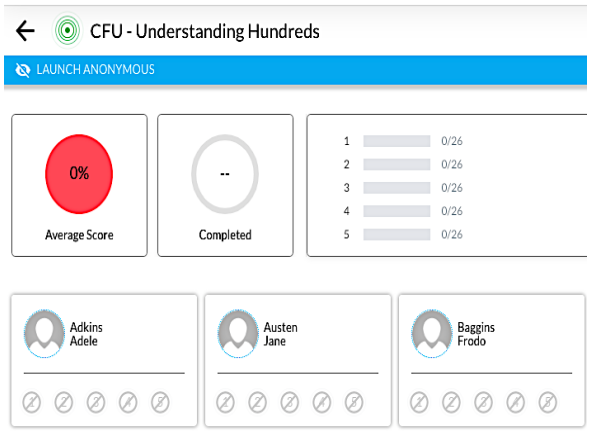


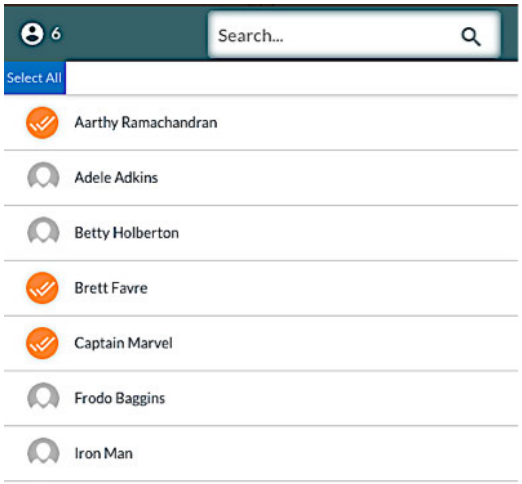
Once added to your class, activities can be viewed by day, week, or month. To activate, toggle to “On.”
You can also reschedule the activity and allow it to be used for mastery (to build students’ skylines). You also have the option to select the students the activity assigned to by clicking the blue student icon next to the activity name.

Go Live

Clicking the Go Live icon opens a new, dynamic window. As the activity runs, the dashboard updates in real-time, giving an in-the-moment snapshot of what topics you need to revisit and with which students.

Go Live can be run using student names or anonymously.





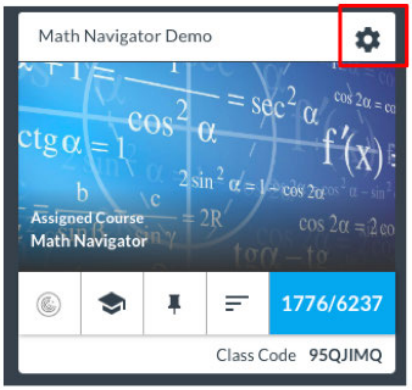
Teachers also have the option to select the students to whom each class activity is assigned. This allows you to provide personalized support and enrichment.

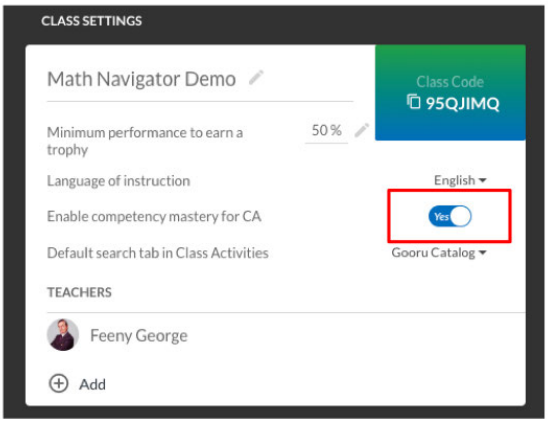
When you click the blue student icon next to the activity name, the pop-up window shows the selected audience for your activity. You can deselect students or select them by clicking on their avatars.

Enabling Competency Mastery for Class Activities

When you post a Class Activity, you have the option to use it to determine mastery of the competency it covers.

To enable this, click on the wheel icon to access your Class Settings.

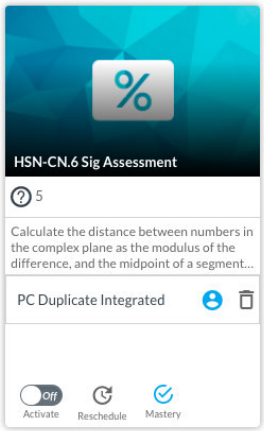




Under the Class Settings section, competency mastery can be enabled or disabled (below):

When enabled, all Class Activities you post will automatically count towards mastery and building a student's skyline (right).

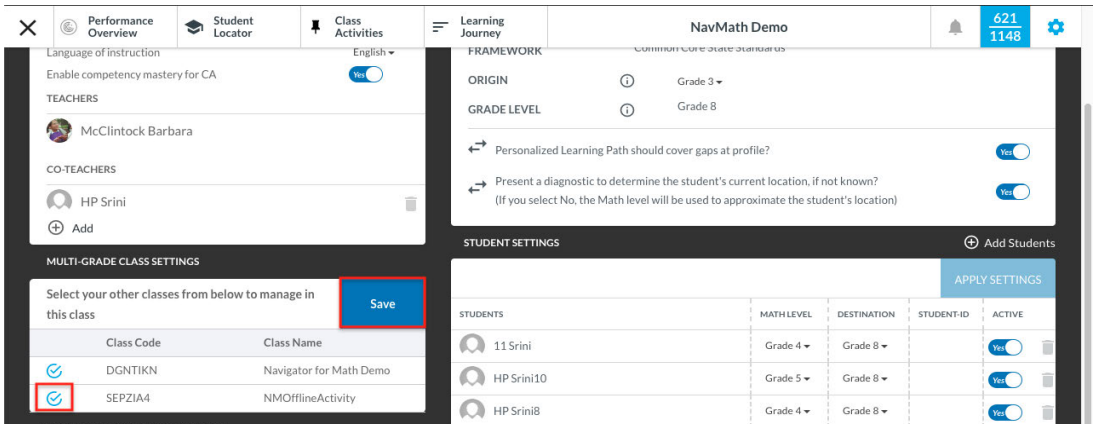
You can un-check mastery or turn off the setting at any time.



Multi-Grade Support (For Select Tenants Only)

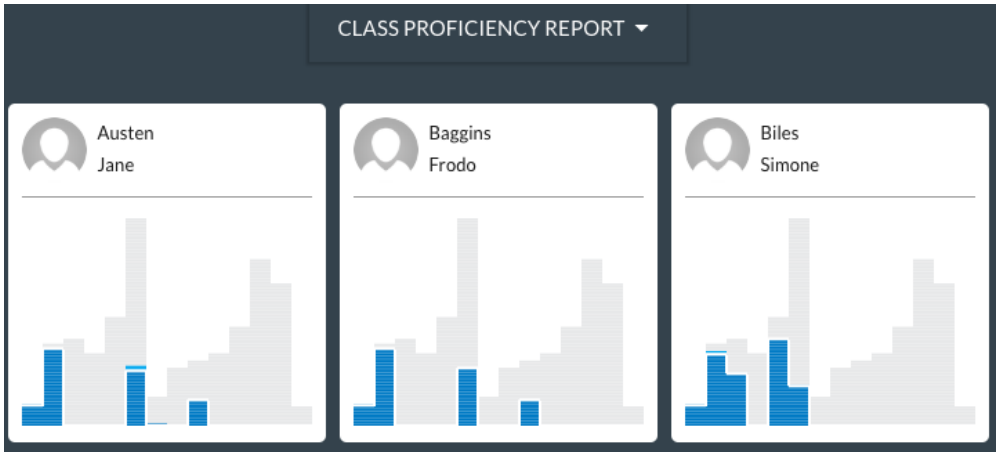
The objective of this feature is to support a combined class view, allowing instructors to add and manage multiple classes without the need to click in and out of each one.

All classes of the same subject and using the same framework will appear under Multi-Grade Class Settings, as shown on the Class Settings page (below). To move between classes, click on the class name and press save.



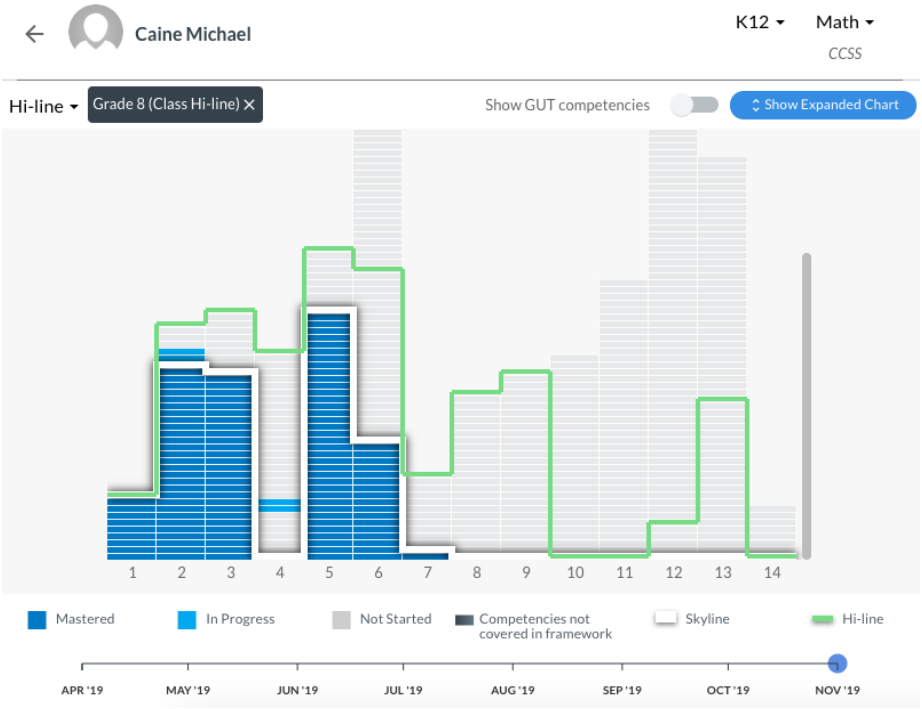
Accessing Student Data:

Clicking your **Student Locator** tab brings up your data dashboard. From here, you can access class- and individual-level reports on progress and proficiency.



The default screen is the **Class Proficiency Report.**

Click on individual student's graph to access their **skyline**. The skyline is dynamic and updates as the student works.

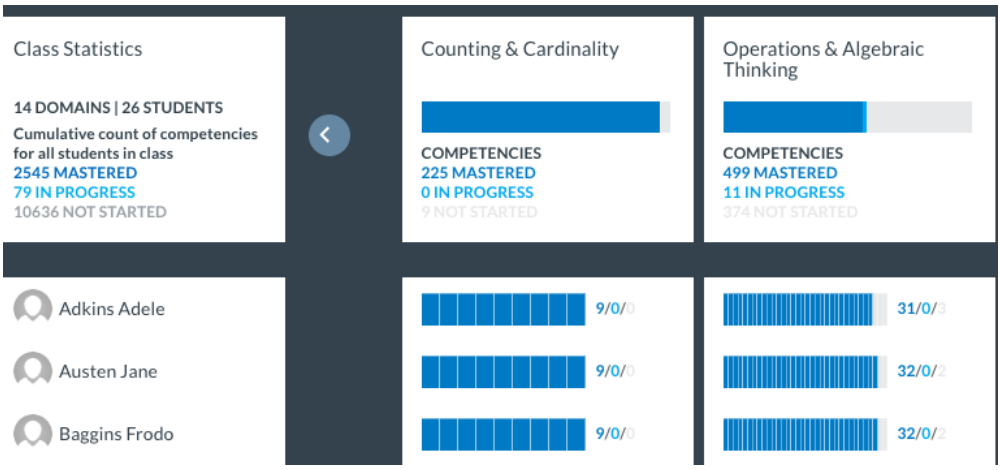


Recall that on a student's skyline, the white line shows their current level of proficiency while the green line is their destination.

If you click the blue "Show Expanded Chart" button, you can zoom into a student's skyline to click on any competency block for descriptions and data.

Selecting **Domain Competency Report** from the dropdown menu bring up class and individual data broken down by the standards or competencies in your course.

will

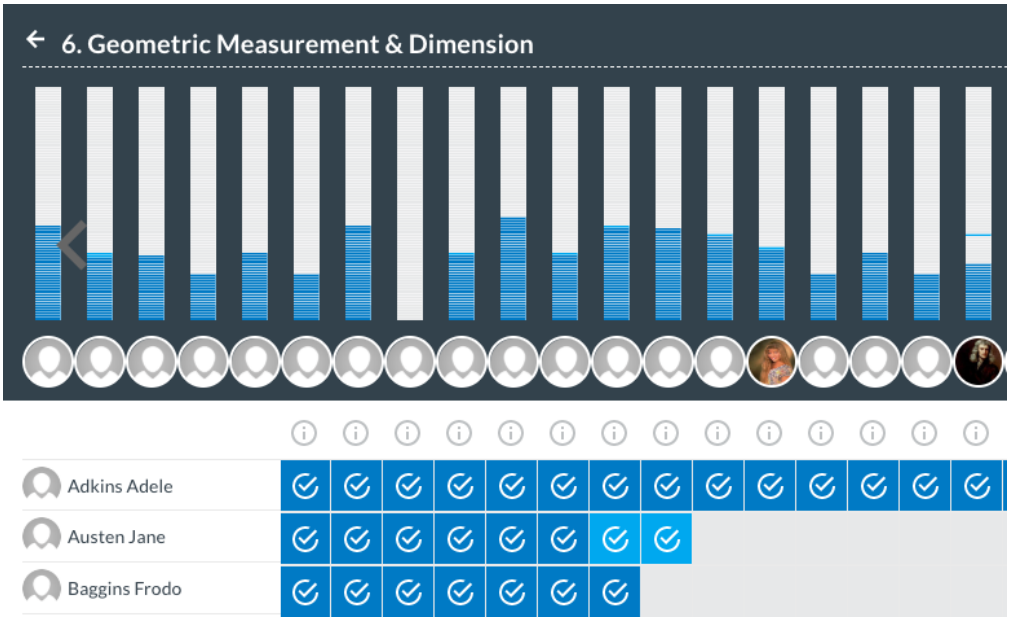


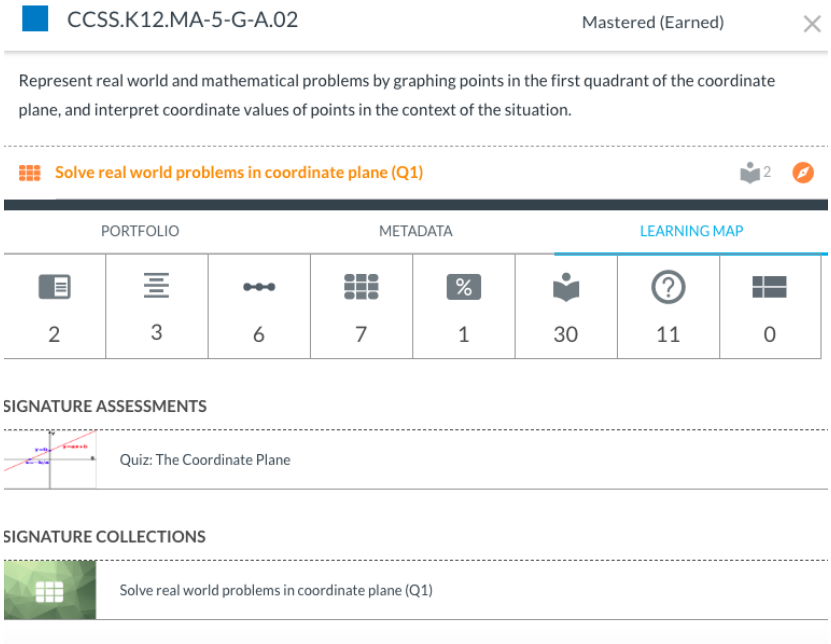
In this partial Domain Competency Report, we can see that our 3 students (Adele, Jane, and Frodo) have each mastered all 9 competencies in the Counting and Cardinality domain.

That means that all 3 students will have 9 blocks in their skylines for this domain.

By clicking on an individual domain, you can access data broken down by competency.

Hovering over the “i” symbol produces a description of the competency. A dark blue check indicates mastery.



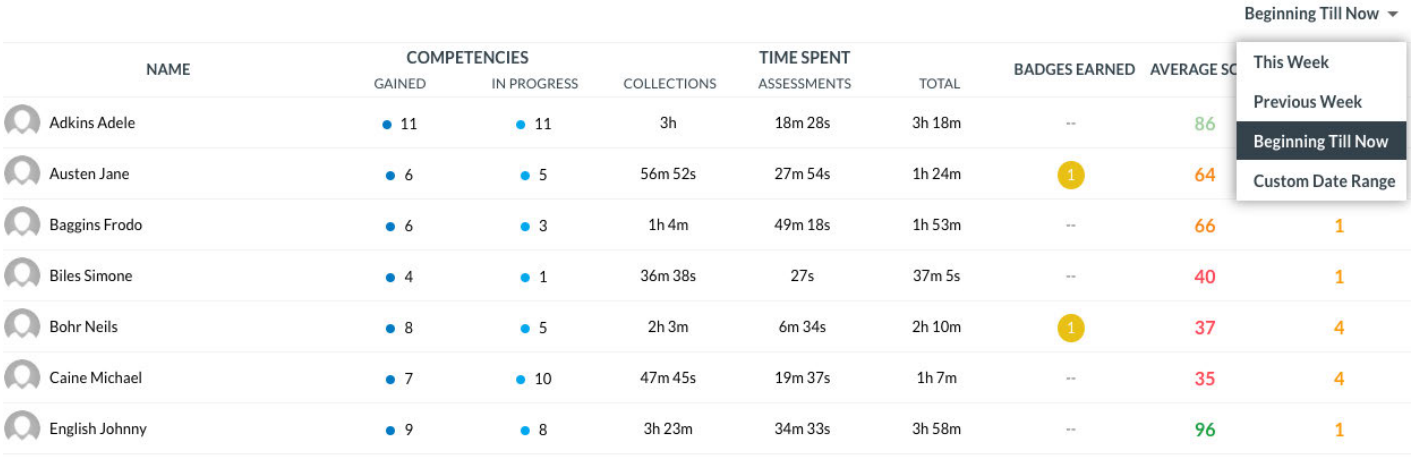


Clicking on any of the blue checkmarks will bring up a new window like this one, showing a description, student performance data, and the resources associated with the specific competency.

Class from the dropdown menu shows progress in terms of competency time spent on resources, mastery badges earned, and suggestions

Selecting **Progress Report** gains, taken.

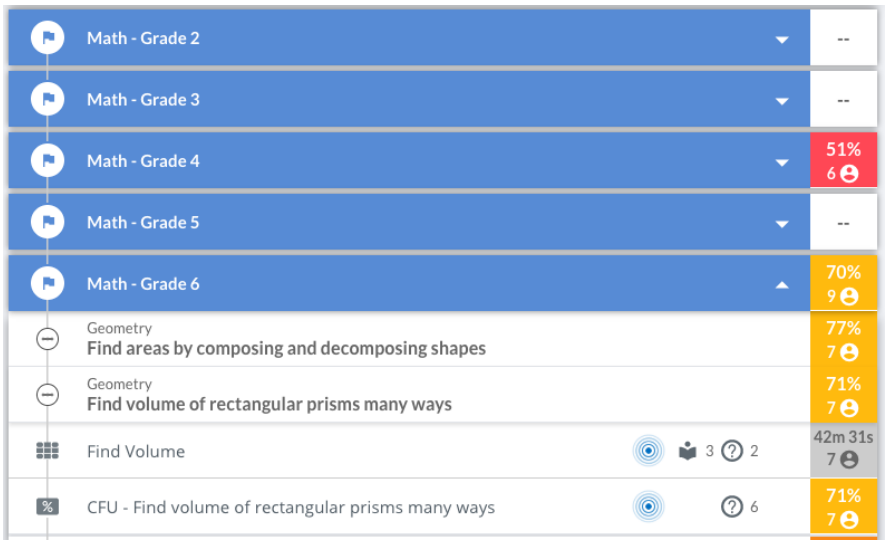
These data can be presented for the current week, the previous week, from the beginning of the school year, or over a custom date range.



NAME	COMPETENCIES GAINED	COMPETENCIES IN PROGRESS	TIME SPENT COLLECTIONS	TIME SPENT ASSESSMENTS	TIME SPENT TOTAL	BADGES EARNED	AVERAGE SC	
Adkins Adele	11	11	3h	18m 28s	3h 18m	--	86	
Austen Jane	6	5	56m 52s	27m 54s	1h 24m	1	64	
Baggins Frodo	6	3	1h 4m	49m 18s	1h 53m	--	66	1
Biles Simone	4	1	36m 38s	27s	37m 5s	--	40	1
Bohr Neils	8	5	2h 3m	6m 34s	2h 10m	1	37	4
Caine Michael	7	10	47m 45s	19m 37s	1h 7m	--	35	4
English Johnny	9	8	3h 23m	34m 33s	3h 58m	--	96	1

Learning Journey and Course Map:

Click on the **Learning Journey** tab on your classroom page to view your course's content broken down by grade, domain, and competency. You can click into any competency to see the collections and assessments associated with it.



For example, here we have clicked on the competency "Find the volume of rectangular prisms many ways" in Grade 6 geometry.

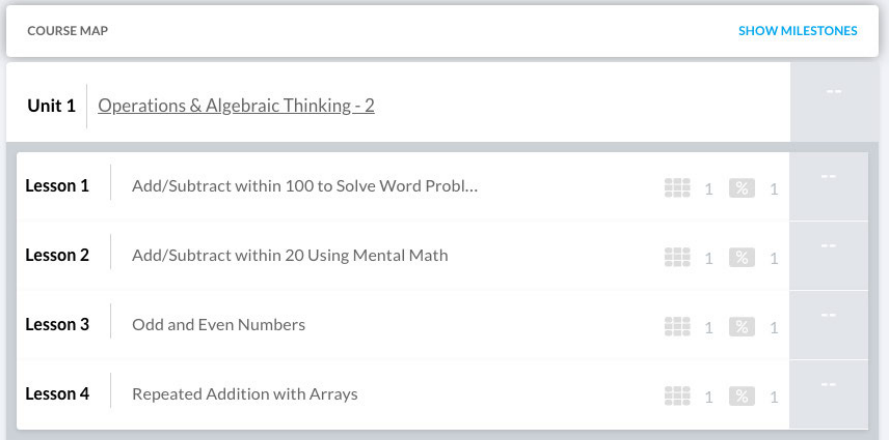
There is a collection of resources ("Find Volume") and an assessment ("CFU") linked to this competency.

Reviewing your course Learning Journey can give you a quick summary of your students' average performance per competency. In the example above, 7 students scored an average of 71% on the assessment ("CFU").

All scores have been color-coded to help you make quick, in-the-moment decisions about re-teaching.

Clicking on the blue “Show Course Map” text in the upper right corner of your Learning Journey will show all Math Navigator content from grades 2-8.

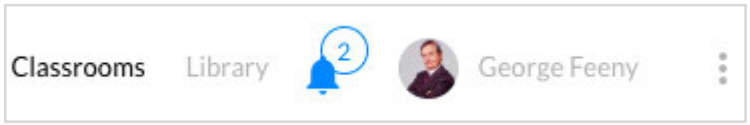
The complete course covers all 268 Common Core Standards across 10 domains and 8 mathematical practices.

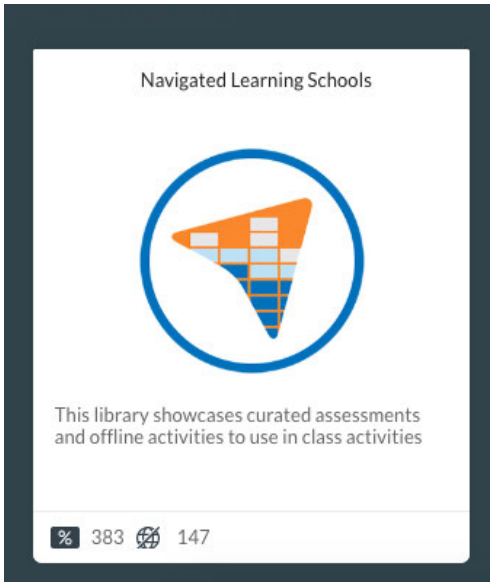


Content Management:

Teachers can also **search, remix,** and **create** their own content. This includes collections of resources, assessments, and activities.

To begin, return to your homepage and click on “Library” in the top right corner of your screen.

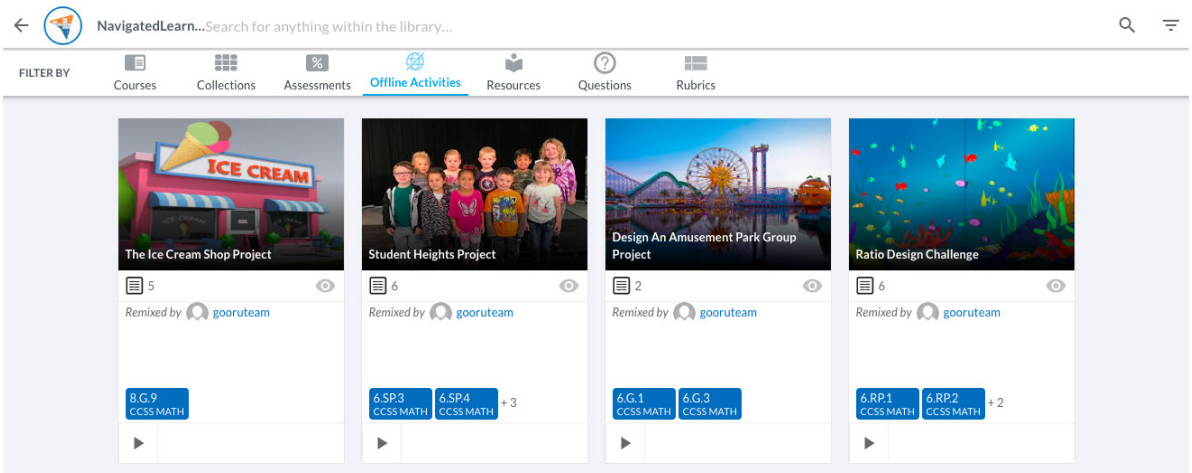




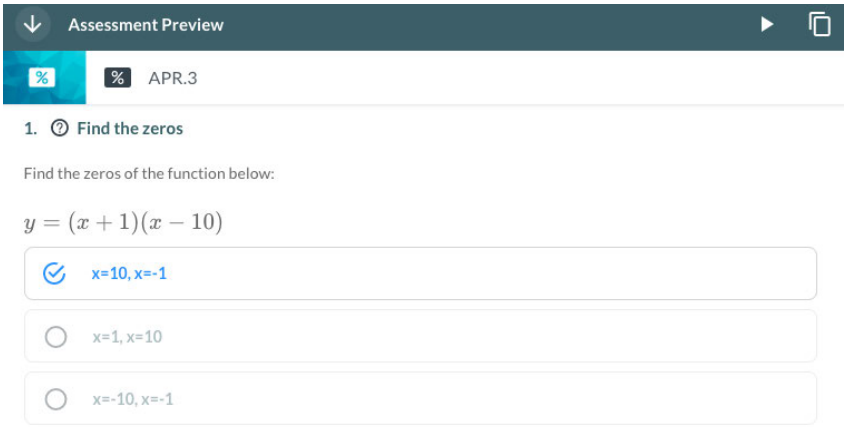
To search for new content, you can click on **Gooru Catalog**, which includes resources made by Gooru users. You can also click on **Navigated Learning Schools**. This will give you access to a curated selection of standards-aligned materials.

You can search by keyword or the material type (course, collection, assessment, offline activity, etc.)

By clicking the inverted pyramid to the right of the search bar, you can also filter your results by competency.



Preview anything you like by pressing the play button.

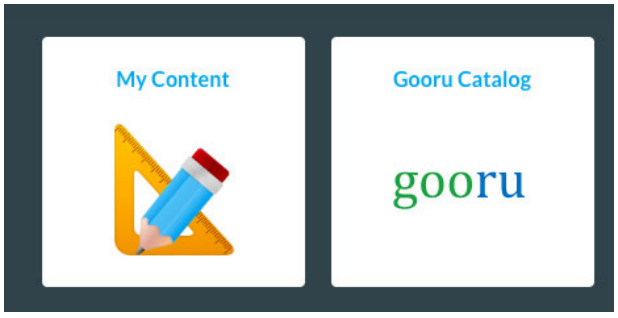


This is a preview of an assessment called “APR.3”.

If you want to edit the assignment or make a copy for yourself, click the **Remix** icon in the upper right corner of the window.

The icon looks like two cards resting on each other.
Remixing adds the item to your account. From “My Content,” the item is yours and can be changed however you want.

To access content you have remixed, click on “Library” in the top right corner of your screen. Then, select “My Content.”



Glossary

Terminology	Definition
Performance overview	Quick view of the entire class plotted on a graph, depicting progress vs performance of each student.
Classroom Code	A code unique to each classroom that students can use to join the classroom
Collection	A playlist of multimedia resources (videos, website, images, etc) and questions
Competency	This term is synonymous with learning standard. K-12 educators generally understand competencies and content developed at the competency level. Much of Gooru's current content is developed around learning at the competency level.
Content Editor	Where a user can build and edit their content on Gooru. Also where they develop the course hierarchy (unit, lesson, and collection folders) and add and organize content
Content Settings	The status of content in terms of visibility - can be private, shared individually, assigned to a student or students, or publically visible.
Course	A "folder" that allows you to organize your learning content into units and lessons.
Course Map	View of all the content available in a classroom
Current Location	Where a student is in a Gooru course in terms of mastery of a standard with regard to the learning targets. Demonstrates understanding and progression through the collections, lesson, unit, and course.
Data	Data of student assessments, attempts, scores, etc for a classroom
Go Live	When the teacher selects "Go Live" for an assessment, the results for each question for the whole class is visible on the teacher's dashboard in real-time
Gooru Catalog	The set of all content available in Gooru, searchable by collection, resource, and standard.
Grade Line	Line connecting the highest competencies in each domain within a subject that is expected to be covered at a specified Grade level.
Learner Profile	Biography, Progress, Performance, Proficiency, Portfolio, Preference and Content
Lesson	A lesson is a sub-folder in the course folder hierarchy.
Library	The content on a user's profile--their courses, collections, assessments, questions, and uploaded resources
Metadata	Information associated/tagged on resources, assessments, collections, courses; includes areas such as standard, grade level, type of media, etc.

Performance Data	View of progress within a unit, lesson, or collection
Proficiency	Student skill level.
Profiles	A collection of information on the user and the user's content (the user could be an individual or an organization) that the user can update. Profiles are publicly visible.
Reaction	Student response to a resource via a likert-type scale
Real-Time Performance	Visual representation of completion and mastery within a collection, lesson, unit and course
Remix	Copy and save content to your account
Resource	A resource is multimedia content in a variety of formats such as videos, interactives, websites, images, Google docs, and more.
Route-0	This is a list of lessons to help the learner gain knowledge in competencies not covered by the course, but, required to learn the competencies that the course covers
Search (Collection)	User enters keyword in search box to find and filter Gooru's catalog to find relevant collections
Search (Resource)	User enters keyword in search box to find and filter Gooru's catalog to find relevant resources
Signature Assessment	Assessment offered to students when they receive >= 80% on their Course Assessment, allowing them the opportunity to gain mastery of a competency.
Signature Collection	Additional resources offered to students to enhance their understanding of the concept if they receive < 80% on the Course Assessment.
Skyline	A white line on the Proficiency Chart that represents the highest-level concepts and skills they have mastered in each domain.
Standards	District, state or national frameworks that define content areas and skills. Well-known standards include Common Core, Next Gen Science Standards
Teacher Suggestion	Additional collections or assessments offered to students to reinforce concepts where students may be struggling.
Time Spent	Amount of time spent on a question, resource, assessment, or collection
Unit	A unit is a sub-folder in the course folder hierarchy.

Gooru

Training and How-To Guide

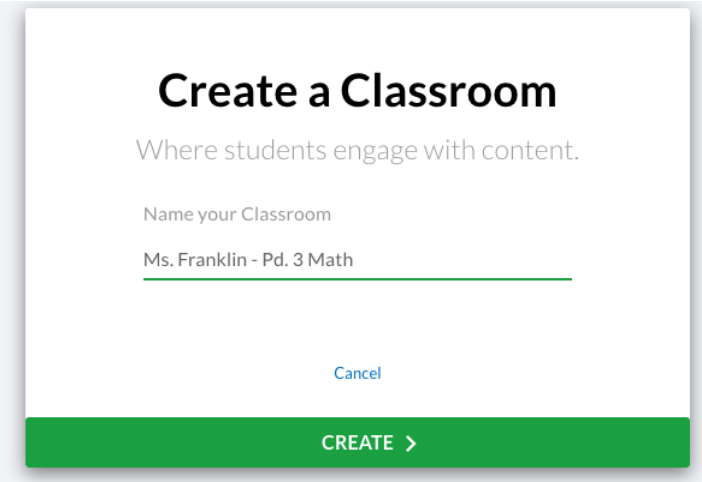
Updated January 2020

Getting Started

Creating a Classroom

When logging into Gooru for the first time:

1. You will be prompted to create a classroom.



Suggestion: make your classroom name specific, with information like your name, the subject, and/or the period.

2. Click the green “Create” button.

3. Add a course to your classroom. For instructions, see *Choosing a Course.*

At any time:

1. When you log in, your homepage will show all classrooms you have created. To make a new one, type the name into the “Create a Classroom” box.

Click “Create Class.”

2. Add a course to your new classroom. For instructions, see *Choosing a Course.*

Accessing Your Classrooms

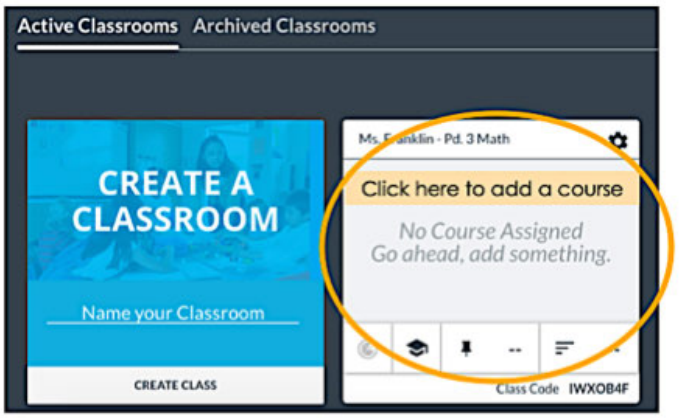
You can access your classrooms at any time by clicking “Classrooms” in the upper right corner of your screen.

Choosing a Course

Note: These instructions detail how to choose from an existing course in Gooru's catalog. For instructions on how to create your own course from scratch, please see Creating a Course from Scratch.

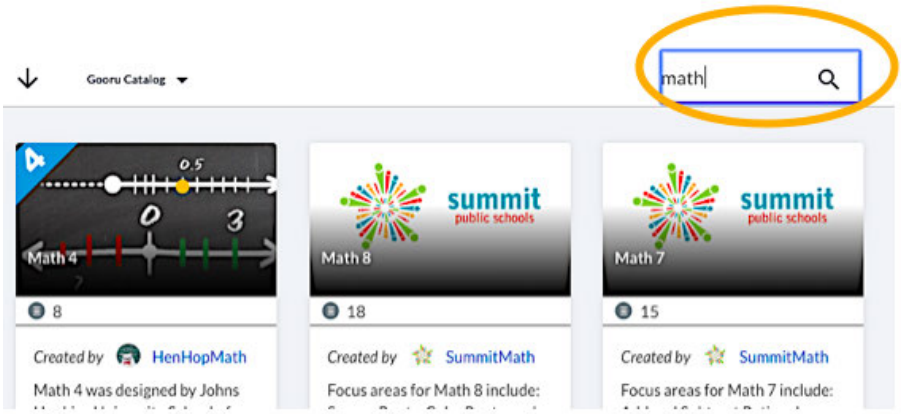
1. If you have just created a classroom, Gooru will prompt you to select a course.

   You can also click on the empty classroom on your homepage to add a course.

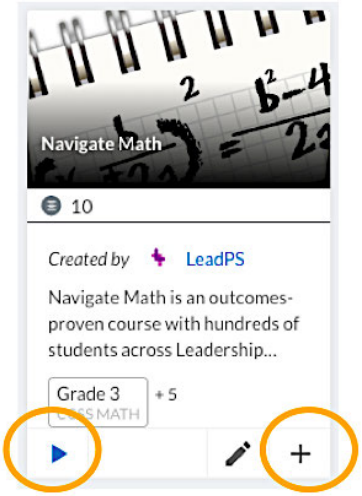


2. To browse all courses, click on the "Gooru Catalog" box near the bottom of the page.

3. In the new window, narrow your results by using the search bar.



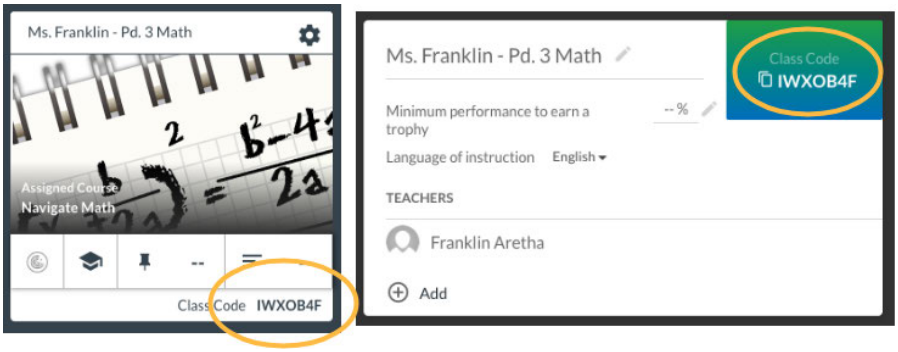
4. To preview a course, click on its play arrow. If you want to add the course to your classroom as is, click the + sign.



5. To use the course but make edits to it, click the pencil icon. This will allow you to **remix** the course. For further instructions on remixing, see *Remixing an Existing Course*.

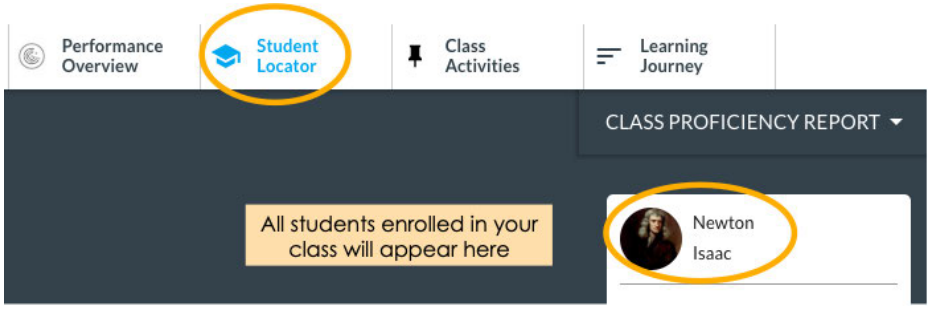
Enrolling Students

1. Identify your class code. This can be found in multiple locations, including on your homepage and in Class Settings.



2. Provide the code to your students. When they log into their Gooru accounts, direct students to enter your class code and click “Join Classroom.”

3. You can verify a student’s enrollment by clicking on your Student Locator tab. All enrolled students will appear on this page.

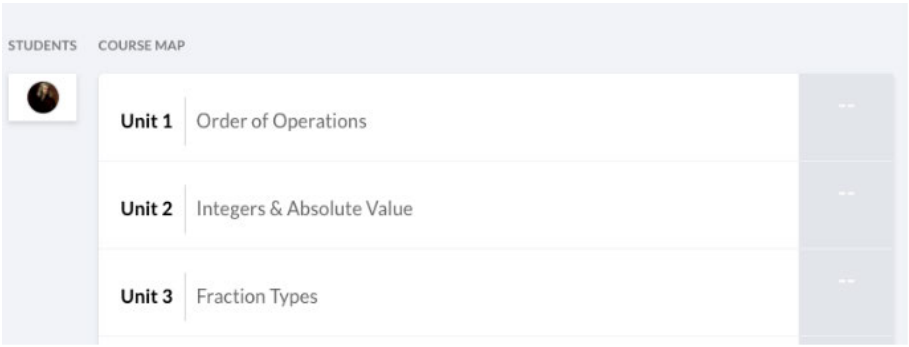


Viewing the Course Map

1. On your home page, click on the classroom whose Course Map you'd like to view.

2. Once on your Classroom page, click on the "Learning Journey" tab at the top of the page.

3. This will bring you to the Course Map. Clicking on a unit will show you the lessons it includes, as well as the materials included in each lesson.



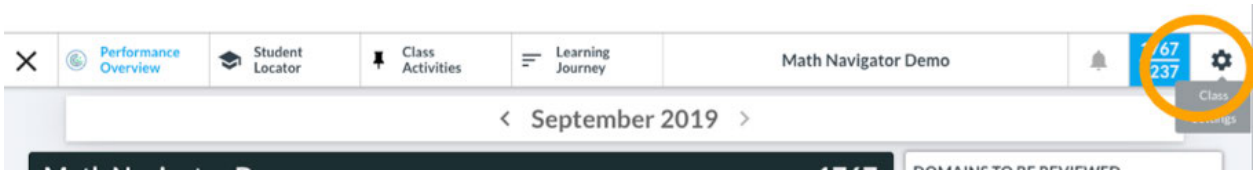
4. The Course Map also shows how many students have been assigned a given lesson or activity, as well as the mean score on each one. The scores are color-coded from green (≥ 80%) to red (≤ 60%).

Class Settings

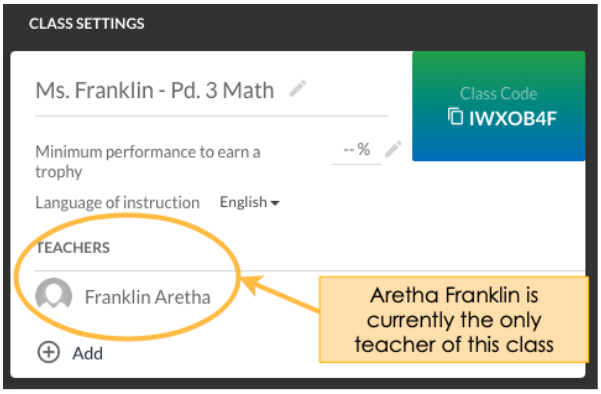
Adding a Co-Teacher

Note: A co-teacher must have a "teacher" account on Gooru. Co-teachers can view your course but cannot make edits. They can perform class management activities and view progress reports.

1. Once on your classroom's page, click the wheel in the upper right corner. This will bring up your **class settings**.



2. The course's teachers are listed in the Settings" box. To add a co-teacher, click next to "Add."

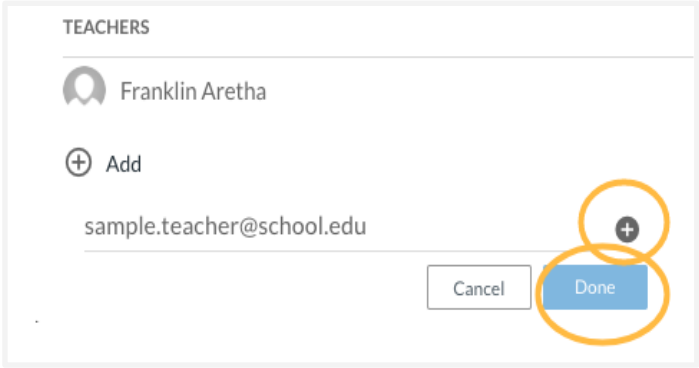


"Class the "+" sign

3. Type in the email address of the co-teacher. This must be the email they used to create their Gooru account.

   Then, click the "+" symbol. The co-teacher's name will appear in gray. "Done."

   You can add multiple co-teachers to a



address

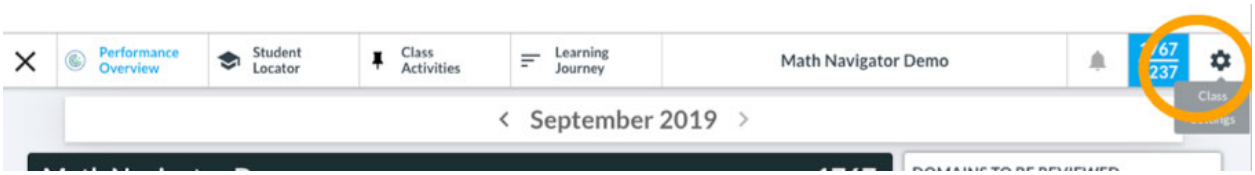
Click

course.

Setting Math Level and Destination

Note: A student's Math Level is their starting point and sets their learning journey. It is a good idea to use data (from state tests or benchmarks, for instance) to set Math Level. Destination indicates the highest level of material a student will have access to. These can both be changed at any time.

1. On your classroom page, click the wheel in the upper right corner. This will bring up your **class settings**.



2. On the Student Settings card, select the desired "Math Level" and/or "Destination" for your student from the drop-down menus. Then, click "Apply Settings."

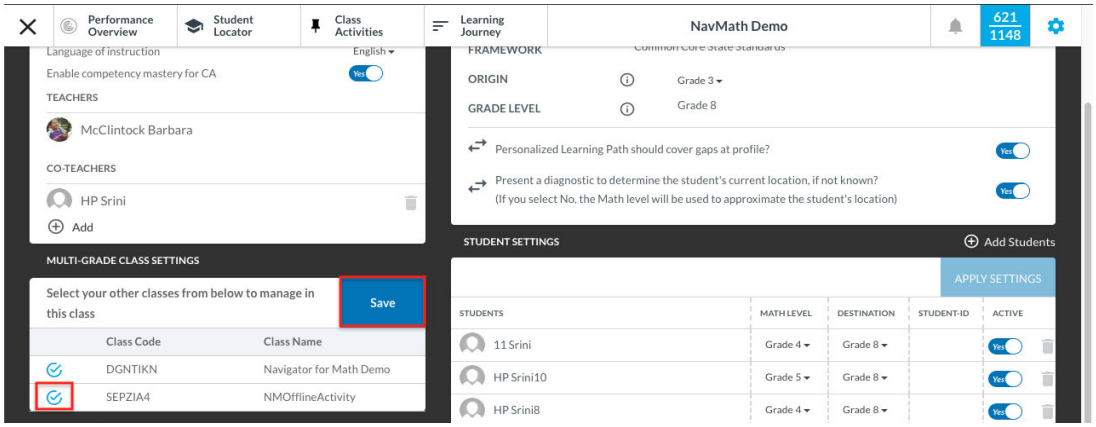
   It is a good idea to set the Destination a few grades ahead of your students so they have access to more challenging material as well.

Multi-Grade Support (For Select Tenants Only)

The objective of this feature is to support a combined class view, allowing instructors to add and manage multiple classes without the need to click in and out of each one.

1. Click on the wheel icon to access Class Settings.

2. All classes of the same subject and using the same framework will appear under Multi-Grade Class Settings (below).



3. To move between classes, click on the class name and press save.

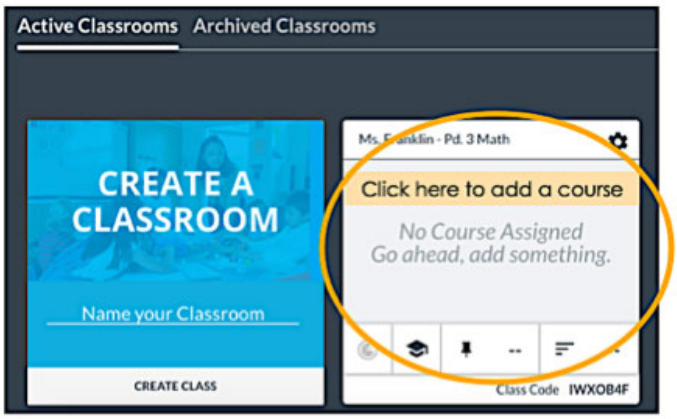
Content Creation

Remixing an Existing Course

Note: Remixing allows you to duplicate an existing course, adding it to your Library and giving you the ability to edit it as your own.

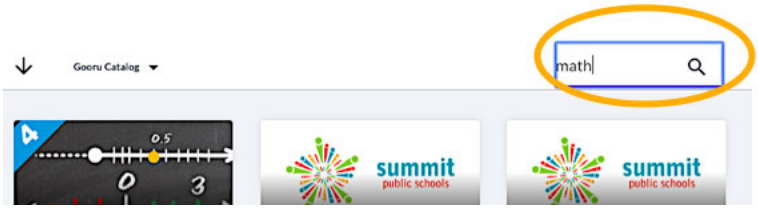
1. If you have just created a classroom, Gooru will prompt you to select a course.

   You can also click on the empty classroom on your homepage to add a course.

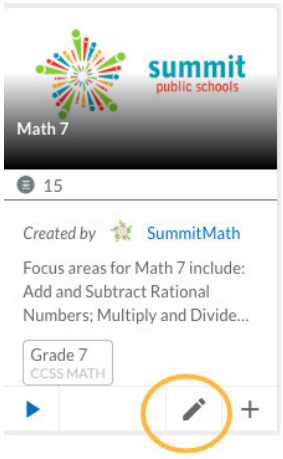


2. To browse all courses, click on the "Gooru Catalog" box near the bottom of the page.

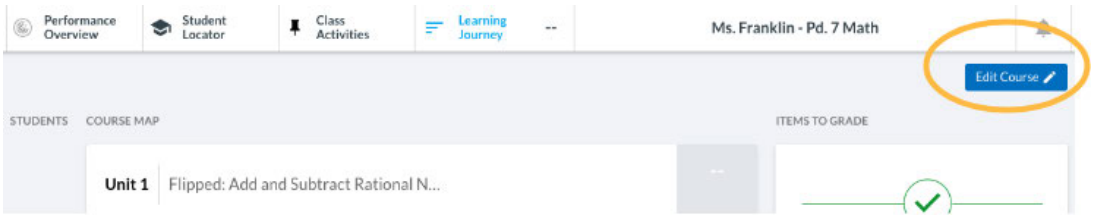
3. In the new window, narrow your results by using the search bar.



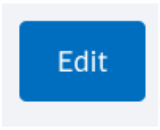
4. To preview a course, click on its play arrow. When you find a course you'd like to remix, click the pencil icon.

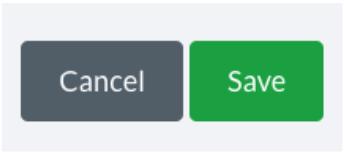


5. This will bring up the course map. To begin making edits, click on the blue "Edit Course" button in the upper right corner.

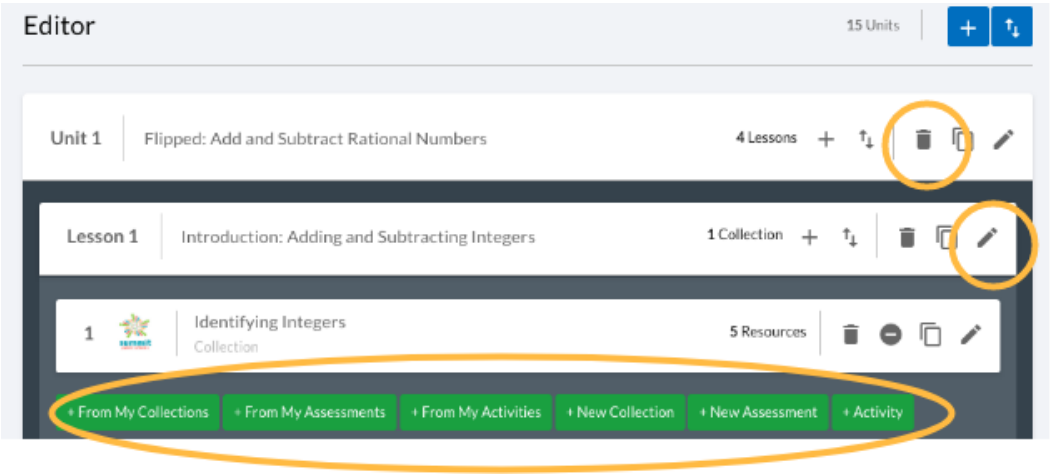


6. On the “Edit Course” page, you have the option to edit the course’s title, description, settings, contents, and collaborators. To edit the title and description, click the blue “Edit” button and save all changes with the green “Save” button.





7. Click on any unit or lesson to view the collections and assessments it contains. You can delete using the trashcan icon and make edits using the pencil icon. Click on the green buttons below each lesson to add new assessments or collections.



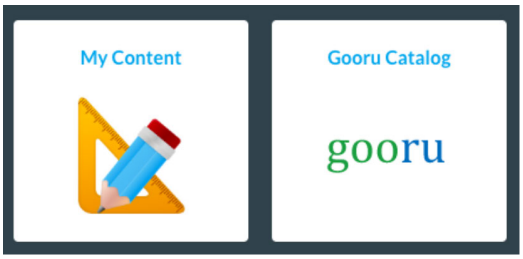
8. For instructions on creating new units or lessons, please see *Creating a New Unit or Lesson*.

Remixing Existing Collections and Assessments

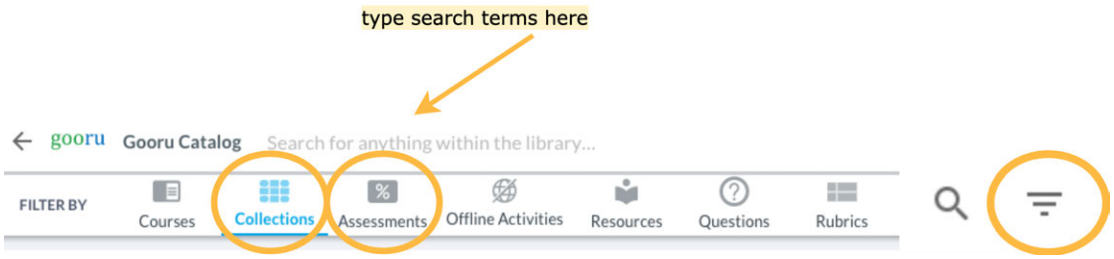
Note: Remixing allows you to duplicate existing materials, adding them to your Library and giving you the ability to edit them as your own.

1. From your homepage, click on “Library” in the top right corner of your screen.

2. If you would like to remix material that you have already created or remixed, click on “My Content.” Otherwise, click on “Gooru Catalog.”



3. In the catalog, narrow your results by selecting “Collections” or “Assessments.” You can further narrow your results by adding a search term or filtering by standard. This can be done by clicking on the inverted pyramid to the right of the search bar.



4. You can preview any of the materials by clicking on them. When you find one you like, click on it.

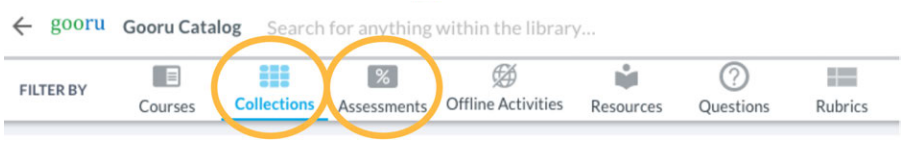
   Then, click on the “remix” symbol in the upper right corner of the new window.

5. Create a title for the remixed collection or assessment when prompted and then “Remix.”

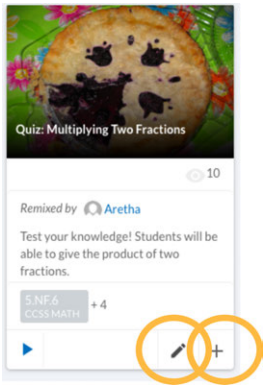
6. To access and edit your remixed collection assessment at any time, click on “Library” in the upper right corner of your homepage.

The remixed item can now be found in “My Content.”

7. Filter “My Content” by collection or assessment.

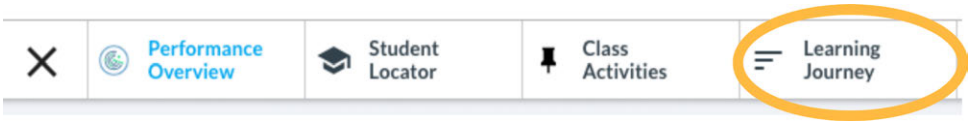


8. You can make edits by clicking the pencil icon. You can also post the collection or assessment to your Class Activities page by clicking the “+” icon.

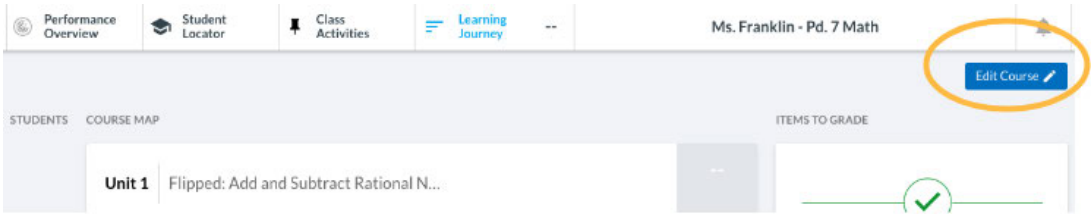


For more information on Class Activities, please see *Adding Activities to Your Classroom*.

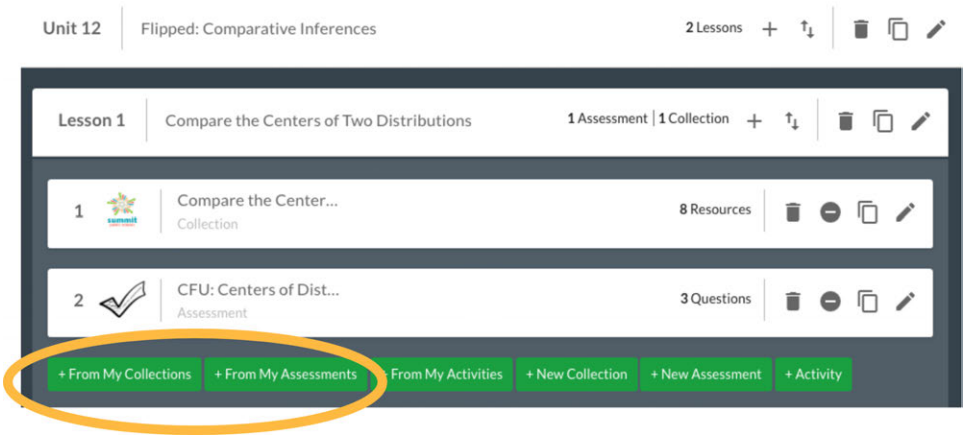
9. You can add the remixed item to a course you have remixed or created from scratch. To do this, click on your classroom (from your homepage), and then on “Learning Journey.”



10. Click on the blue “Edit Course” button in the upper right corner.



11. Scroll down and select the unit and lesson you'd like to add the remixed item to. Select "From My Collections" or "From My Assessments."

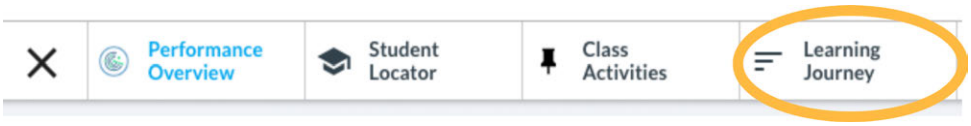


12. This will bring up a new window. Find your remixed assessment or collection, select it, and click "Add."

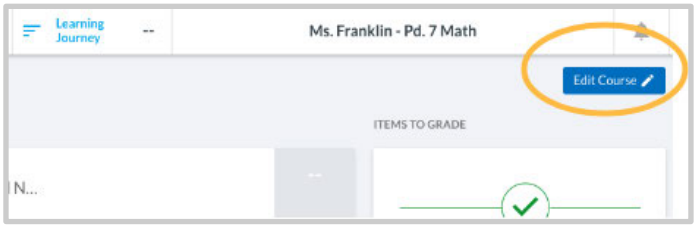
Creating a New Unit or Lesson

Note: The course must be editable for you to create a new unit or lesson. You can edit courses that you have remixed (see Remixing an Existing Course*) or created from scratch (see* Creating a Course from Scratch*).*

1. On your classroom page, click on “Learning Journey.”



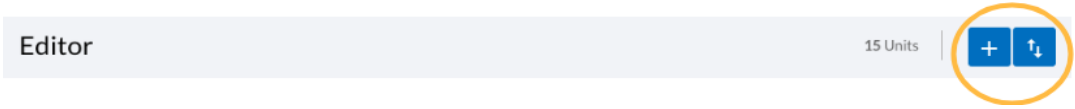
2. Click on the blue “Edit Course” button upper right corner.



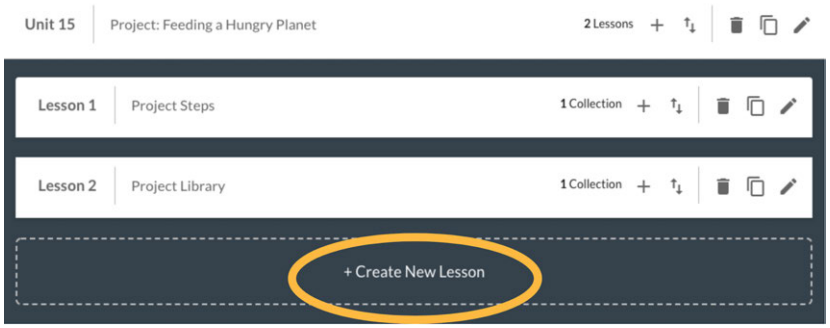
in the

3. To create a new Unit, click on the blue “+” icon. Give the unit a name and then save.

4. By default, the new unit will be placed last. To change the order, click on the up/down arrow icon to drag and drop units to the desired location.



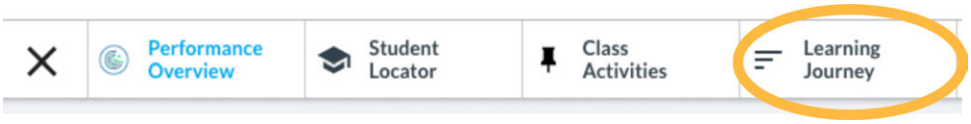
5. To add a new lesson, click on the unit in which you’d like to place it. Click on “Create New Lesson.” Give the lesson a name and click to save. For instructions on adding material to your lesson, see *Adding Collections and Assessments to a Lesson*.



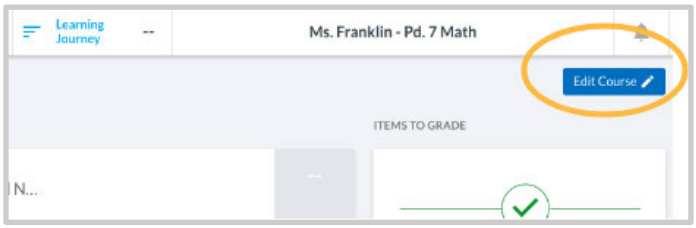
Adding Collections and Assessments to a Lesson

Note: The course must be editable for you to add collections or assessments. You can edit courses that you have remixed (see Remixing an Existing Course*) or created from scratch (see* Creating a Course from Scratch*).*

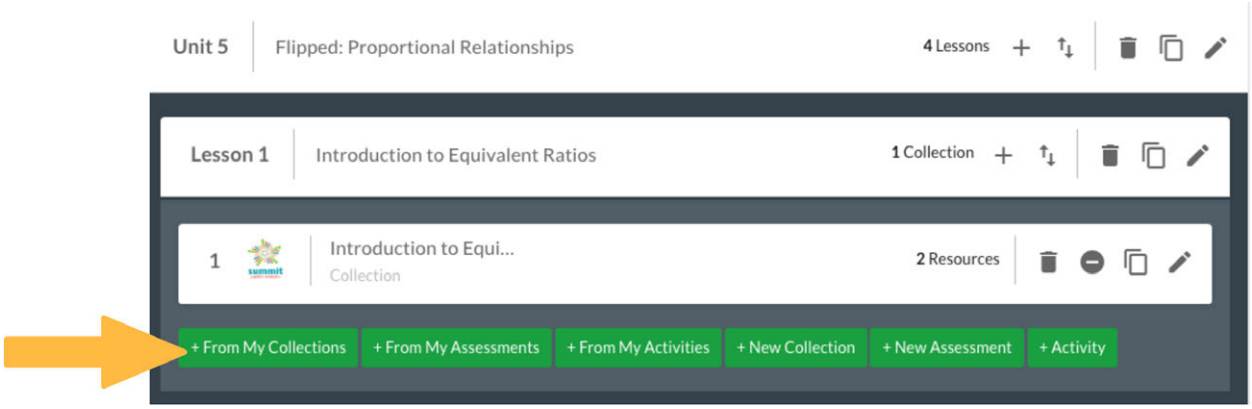
1. On your classroom page, click on “Learning Journey.”



2. Click on the blue “Edit Course” button in the upper right corner.



3. Click on the unit and lesson you’d like to add to. Select the green button that matches what you want to do.

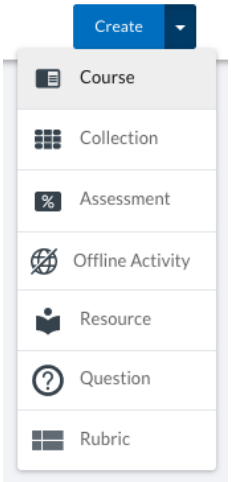


4. Clicking “From My Collections” or “From My Assessments” will allow you to select from materials you have created or remixed yourself.

   Clicking “New Collection” or “New Assessment” will prompt you to create your own materials. For instructions, see *Creating a New Collection* or *Writing a New Assessment*.

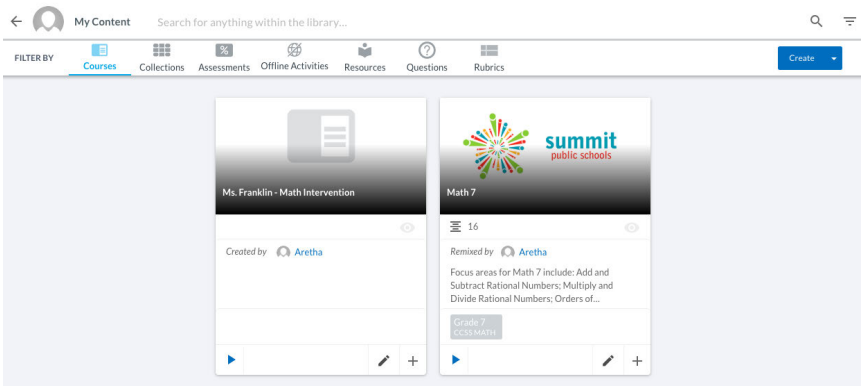
Creating a Course from Scratch

1. Click on “Library” in the top right corner of your homepage.
2. Select “My Content.”
3. Click the blue “Create” button in the upper right corner of the screen. Select “Course” from the drop-down menu.



4. Give your course a title, select “K12,” and then “Create.”
5. Once your course is created, you can edit its title, description, framework, audience, and thumbnail picture. Click the green “Save” button after any changes.
6. You can also add units and lessons to your course and resources to each lesson. For instructions, please see *Creating a New Unit or Lesson* and *Adding Collections or Assessments to a Lesson*.
7. Your course is accessible to you at any time in your Library.

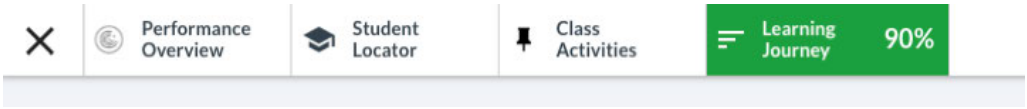
   Click “My Content” (as in Steps 1 and 2) to access courses you have created or remixed. Clicking on the course allows you to edit it.



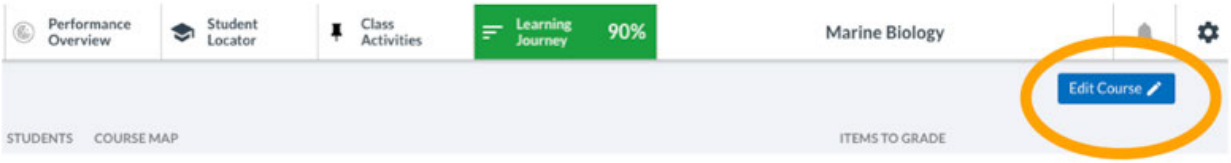
Adding Collaborators

Note: Collaborators can make edits, remix, and post activities to your course. A collaborator must have a "Teacher" account on Gooru. Collaborators can only be added to courses you have remixed or created from scratch.

1. On your classroom page, click the "Learning Journey" tab. This will bring up the course map.

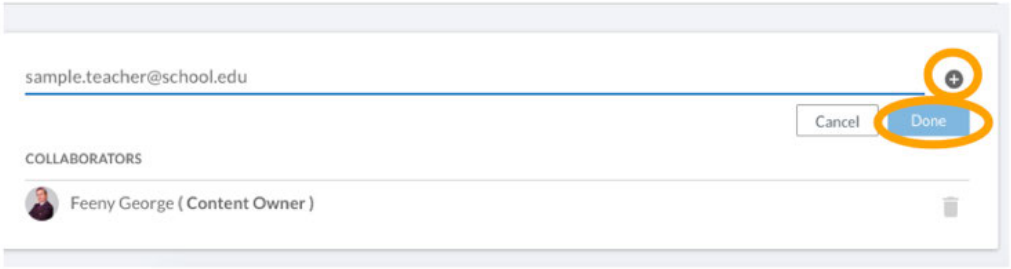


2. On the top right of the page, click the blue "Edit Course" button.



3. Scroll to the bottom of the page to find the "Add Collaborator" section. Click the blue "+" button to add a collaborator.

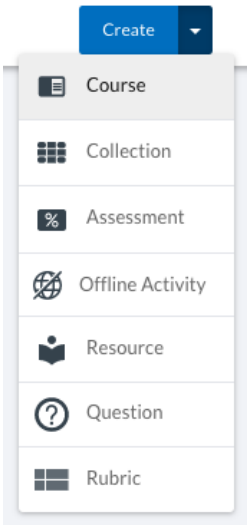
4. Type in the email address of the collaborator. This must be the email address they used to create their Gooru account. Then, click the "+" to the right of the email address. The collaborator's name will appear in gray. Then, click "Done." You can add multiple collaborators to a course.



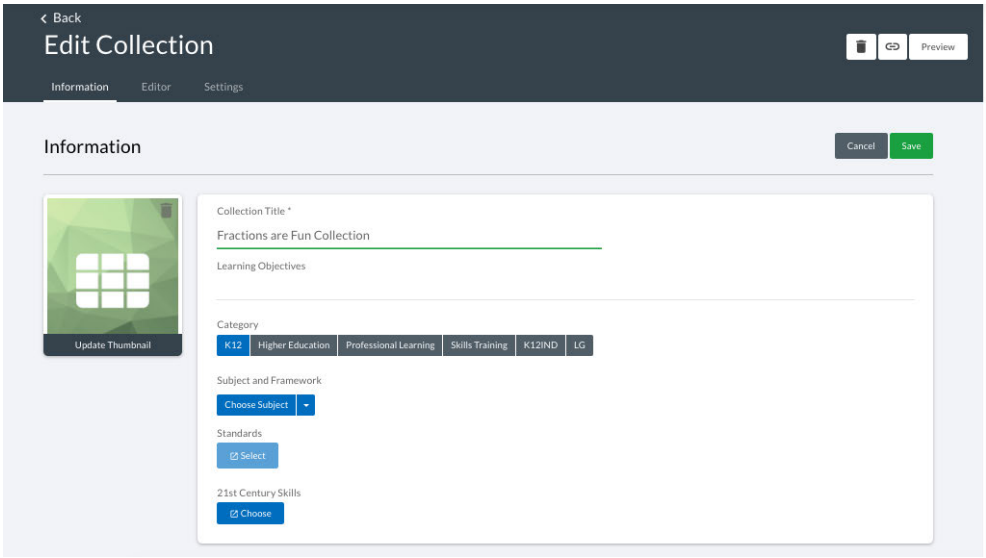
Creating a New Collection

1. Click on “Library” in the top right corner of your homepage.

2. Click on “My Content”
3. Select “Collection” from the blue “Create” drop-down menu.



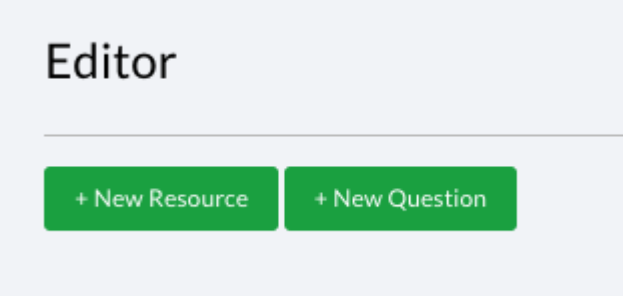
4. Give your collection a title and click “Create.”
5. This will bring you to the “Edit Collection” page.



To edit the collection’s settings, you can:

- Click “Update Thumbnail” and select an image from your computer to change the collection cover image.
- Edit the “Collection Title” if desired
- Add Learning Objectives
- Select “K12” then choose the “Subject and Framework,” and select standards to tag the collection
- Save any changes

6. To add a new resource to your collection, click the green “New Resource” button.

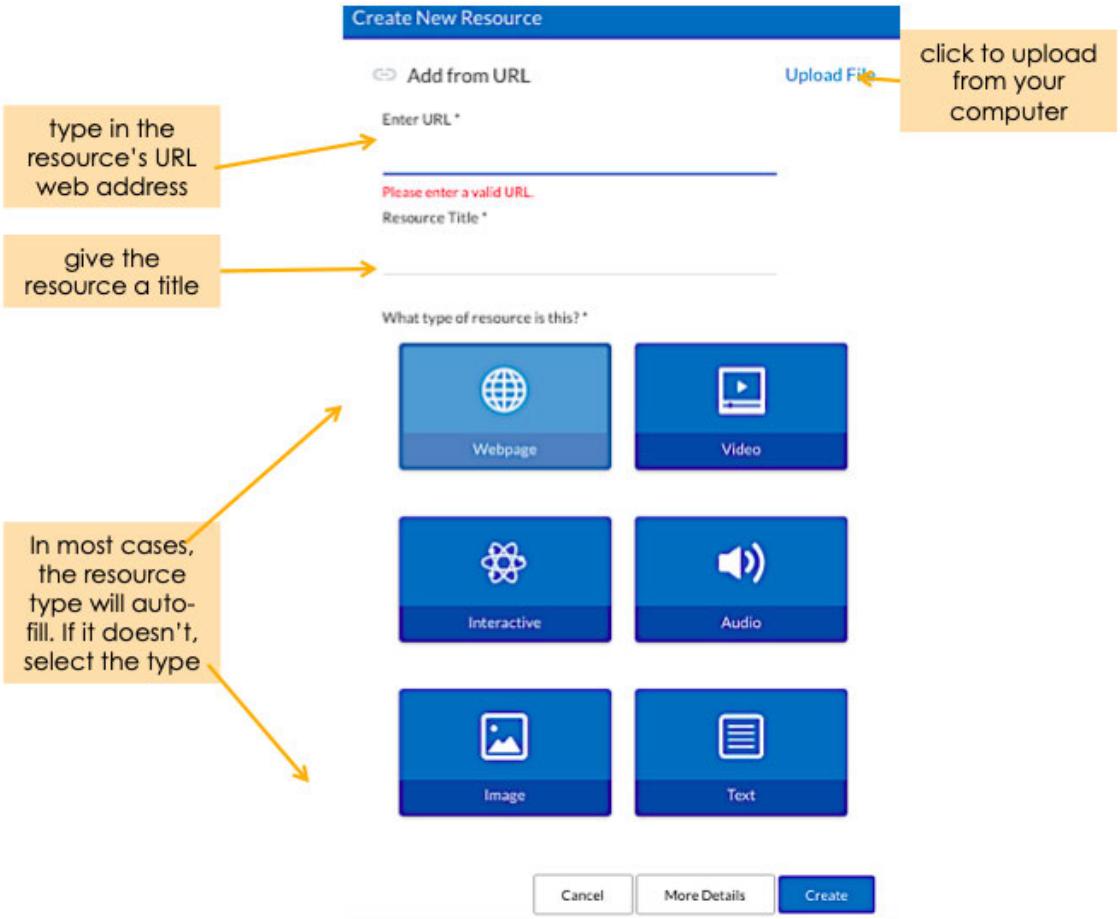


7. In the window that opens, enter the resource’s URL web address or upload a file from your computer.

   When prompted, give your resource a title.

   The resource type will auto-fill, but in the cases that it doesn’t, click on the correct resource type.

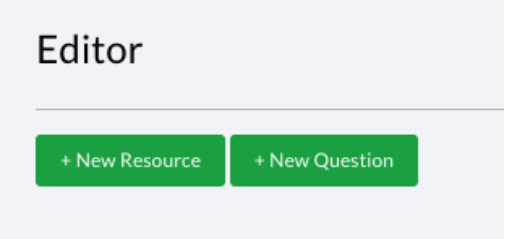
   Click “Create.”



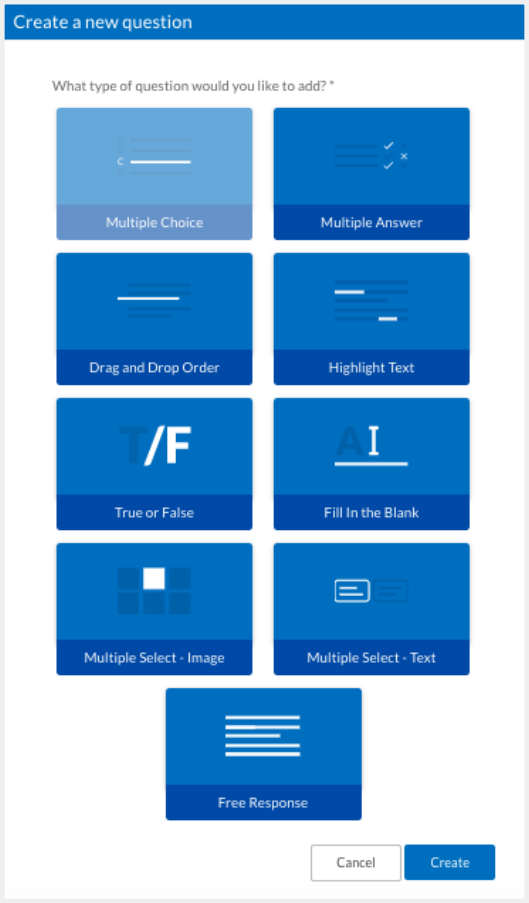
Note that your resources can be files uploaded from your computer, webpages, videos, interactive sites, audio files, images, or text.

You can add as many resources as you want to the collection.

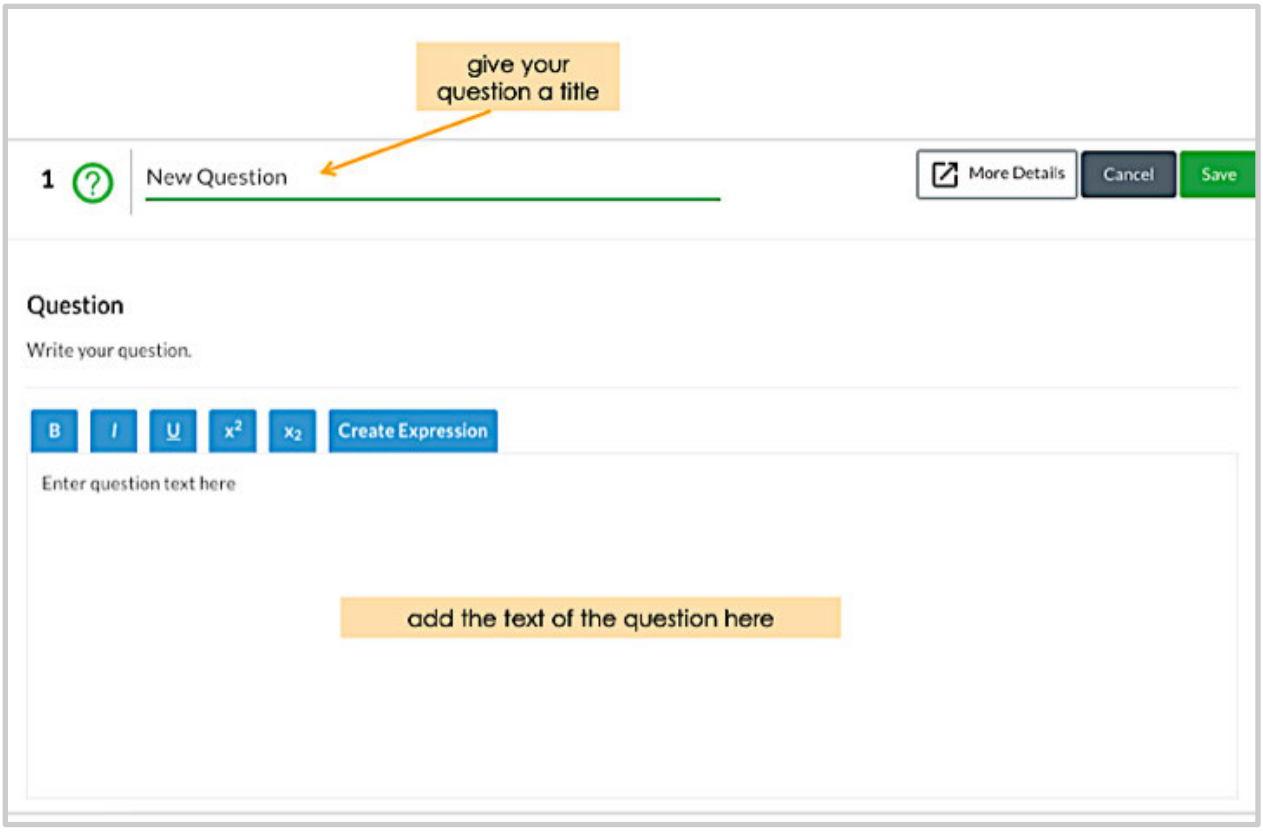
8. To add a question to your collection, click “New Question.”



9. In the window that opens, select the type of question you want to add. Click “Create.”



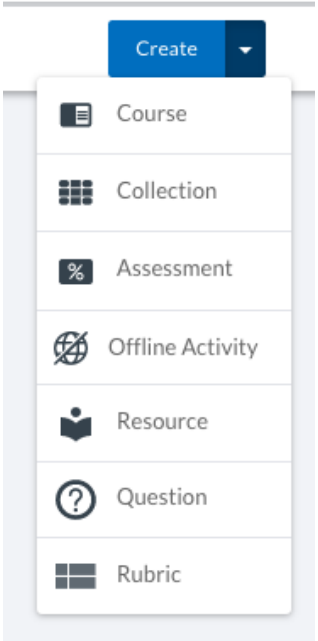
10. To create your question:
    - Enter a question title
    - Type in the text of the question
    - Click “Create Expression” to add mathematical expressions or symbols
    - Add question images by uploading from your computer
    - Add answer choices
    - Select the correct answer(s)
    - Add an exemplar
    - Click “Save”



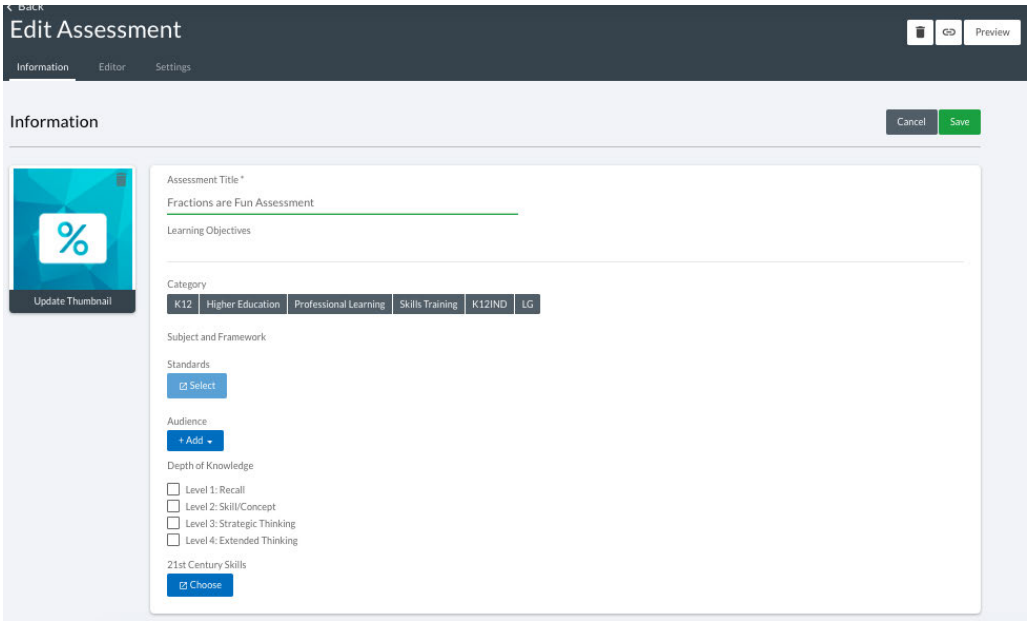
Note: You can add as many questions as you want to the collection.

Your collections are accessible to you at any time in your Library. Click "My Content" (as in Steps 1 and 2) to access collections you have created or remixed. Clicking on the collection allows you to edit it.

Writing a New Assessment



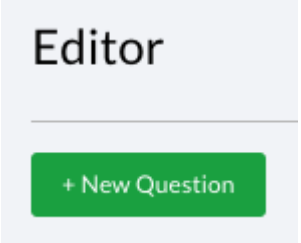
1. Click on “Library” in the top right corner of your homepage.
2. Click on “My Content.”
3. Select “Assessment” from the blue “Create” drop-down menu.
4. Give your assessment a title and click “Create.”
5. This will bring you to the “Edit Assessment” page



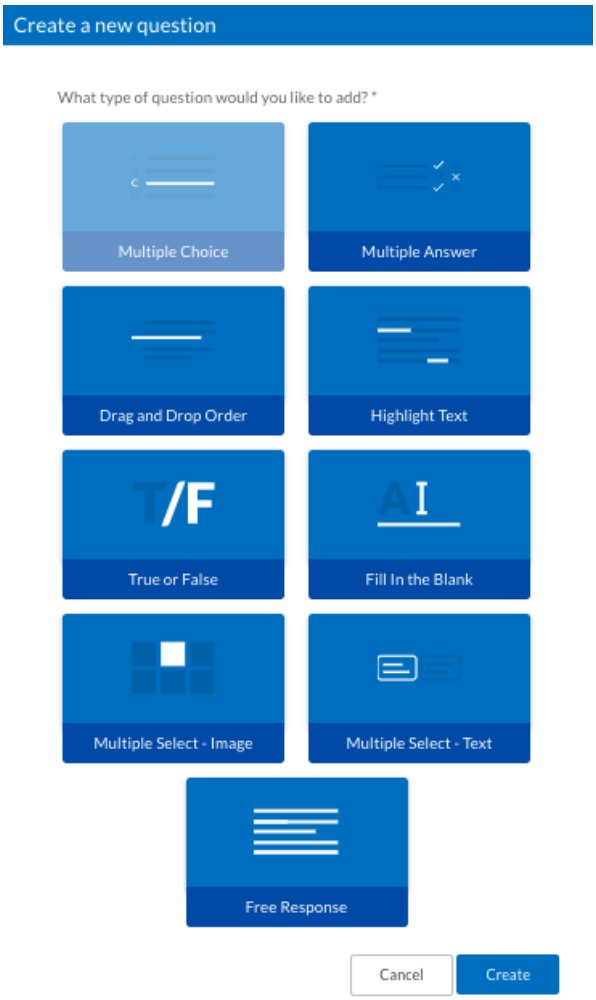
To edit the assessment’s settings, you can:

- Click “Update Thumbnail” and select an image from your computer to change the assessment cover image.
- Edit the “Assessment Title” if desired
- Add Learning Objectives
- Select “K12” then choose the “Subject and Framework,” and select standards to tag the assessment
- Indicate the Depth of Knowledge required to answer the question
- Save any changes

6. To add a question to your collection, click “New Question.”

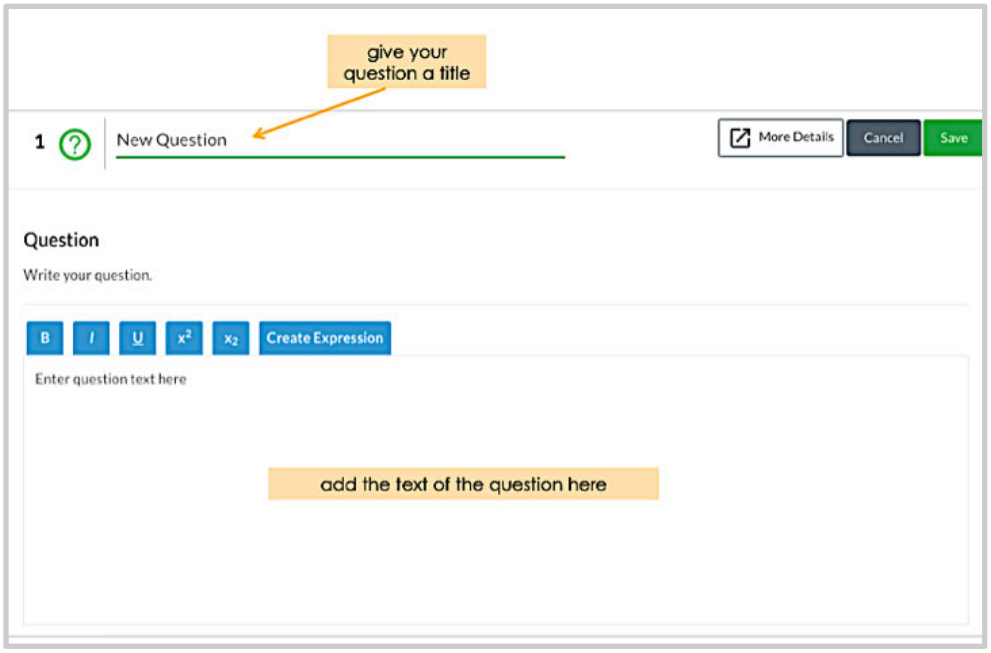


7. In the window that opens, select the type of question you want to add. Click “Create.”

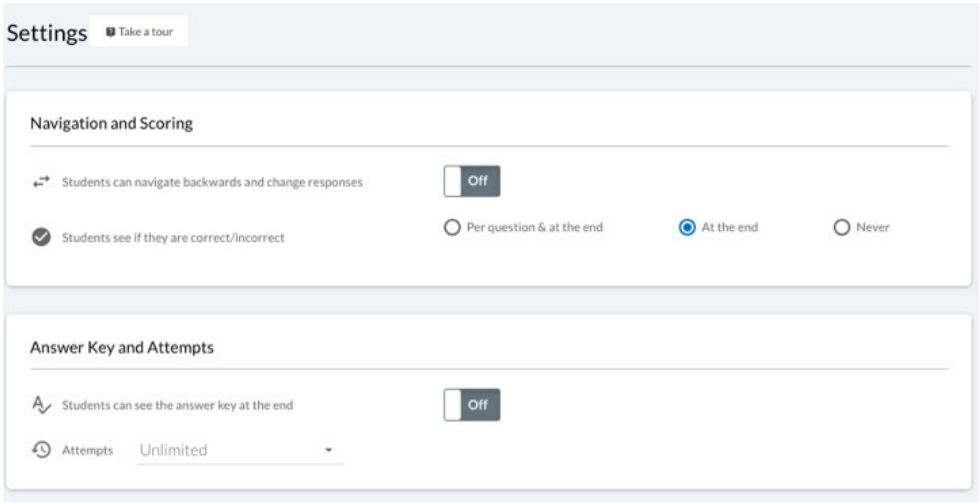


To create your question:

- Enter a question title
- Type in the text of the question
- Click “Create Expression” to add mathematical expressions or symbols
- Add question images by uploading from your computer
- Add answer choices
- Select the correct answer(s)
- Add an exemplar
- Click “Save”



8. Scroll down to edit the assessment's settings.



You can edit:

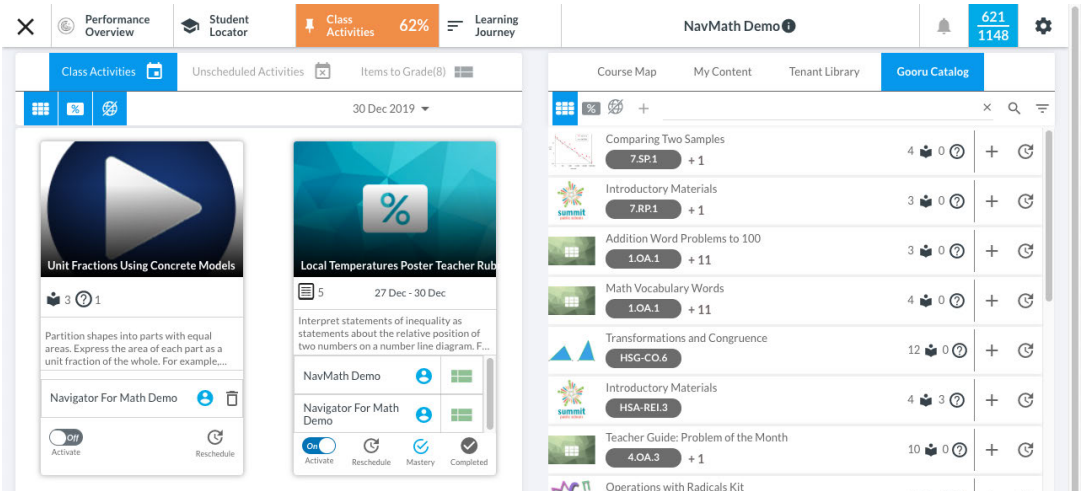
- Whether or not students can navigate backwards during the assessment (toggle to On/Off)
- When students see if their answers are correct or incorrect
- Whether or not students see an answer key after the assessment (toggle to On/Off)
- How many attempts students can have on the assessment

Your assessments are accessible to you at any time in your Library ("My Content"). Clicking on the assessment allows you to edit it.

Class Activities

Adding Activities to Your Classroom

Note: Class Activities (CA) are collections and assessments that you want students to complete on a specific day.

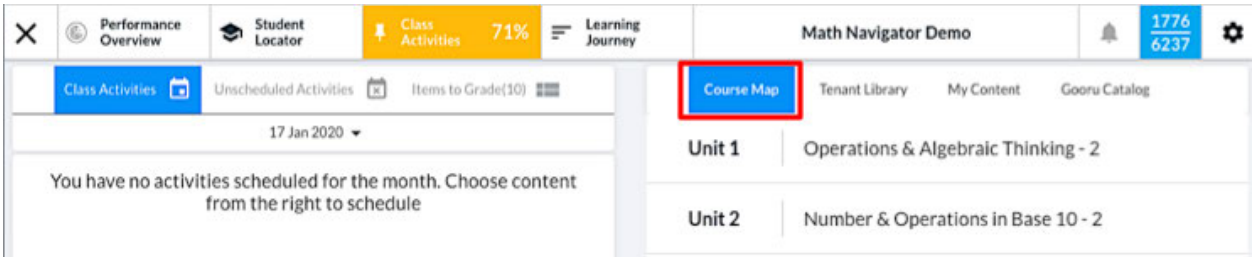


Once posted, Class Activities will appear on the left side of the page. Class Activities can be added from multiple sources to fit the instructor's needs:

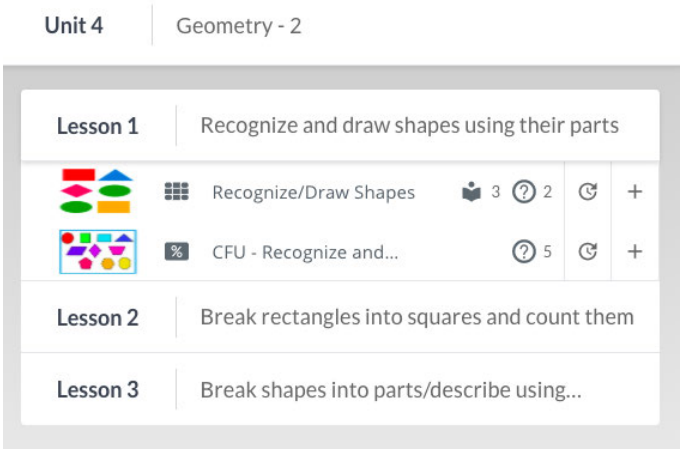
Option 1: Assigning Class Activities from the Course Map

1. Click on the "Class Activities" tab at the top of the classroom page.

2. On the right side of the page, select "Course Map." This will surface all of the units in the course.



3. Click on a unit and a lesson to view the associated collections and assessments.

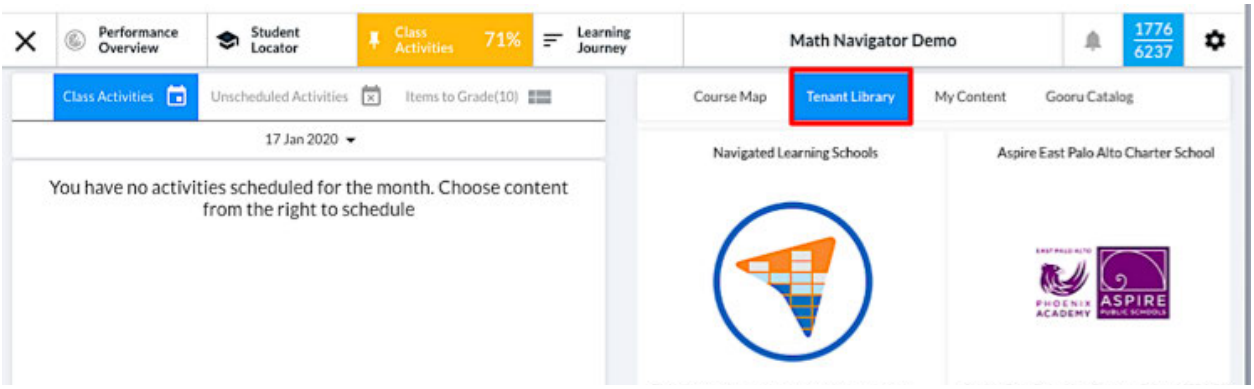


4. To preview the collection or assessment, click on its name. To assign, click the "+" icon. To schedule for later, click on the clock icon and select a date.

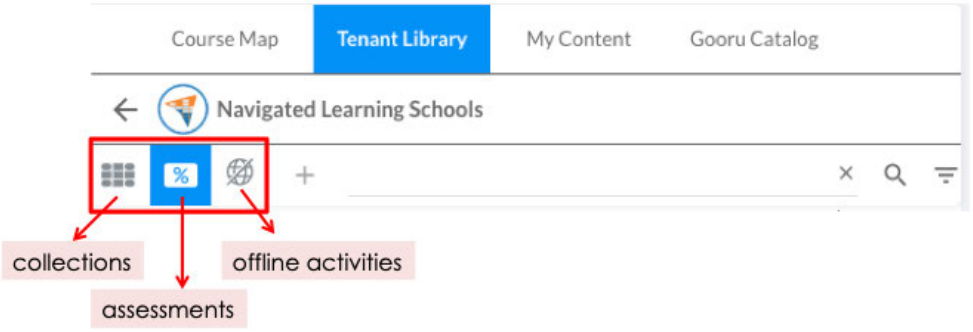
Option 2: Assigning Class Activities from the Tenant Library

1. Click on the "Class Activities" tab at the top of the classroom page.

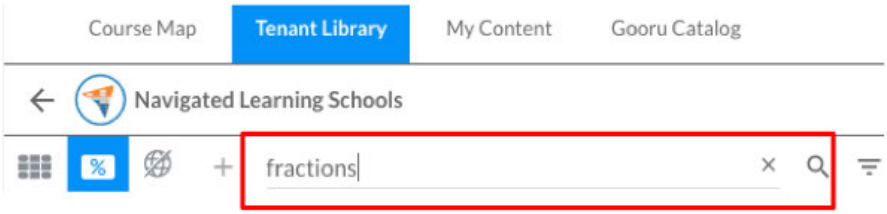
2. On the right side of the page, select "Tenant Library". Choose the desired library from the cards shown.



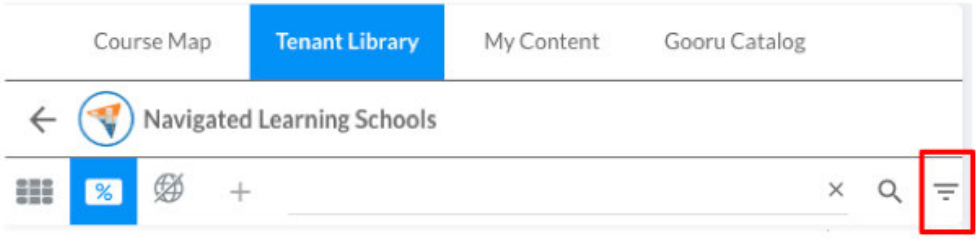
3. Use the icons (shown below) to narrow the search by activity type.

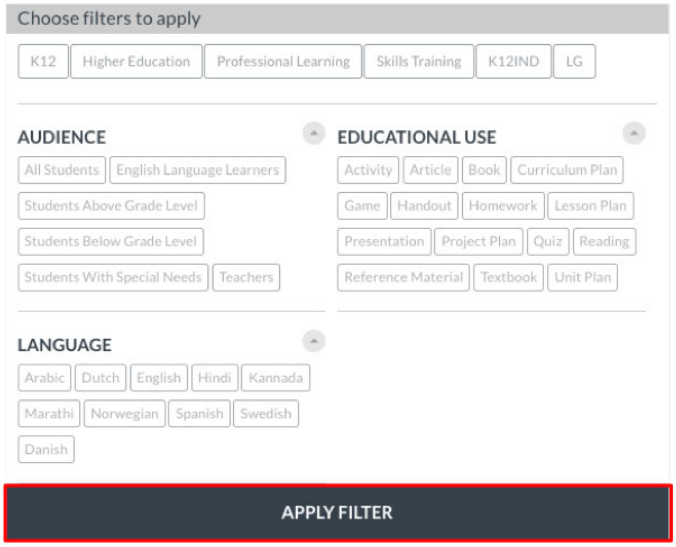


4. Use the search bar to narrow results.



5. Narrow the results further by clicking the filter icon. Select a framework and competency, and click "Apply Filter".



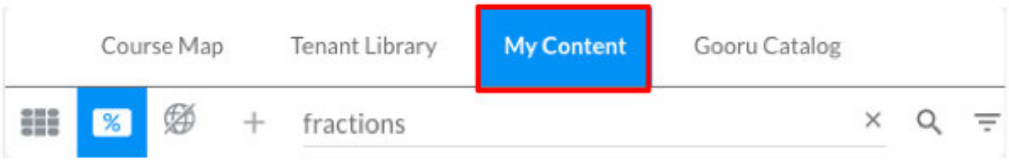


6. To preview the collection or assessment, click on its name. To assign, click the "+" icon. To schedule for later, click on the clock icon and select a date.

Option 3: Assigning Class Activities from My Content

1. Click on the “Class Activities” tab at the top of the classroom page.

2. On the right side of the page, select “My Content”.

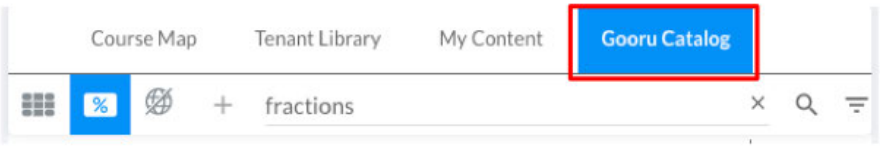


3. Use the search bar and filter icon to narrow the results, as in Option 2.

Option 4: Assigning Class Activities from the Gooru Catalog

1. Click on the “Class Activities” tab at the top of the classroom page.

2. On the right side of the page, select “Gooru Catalog.”



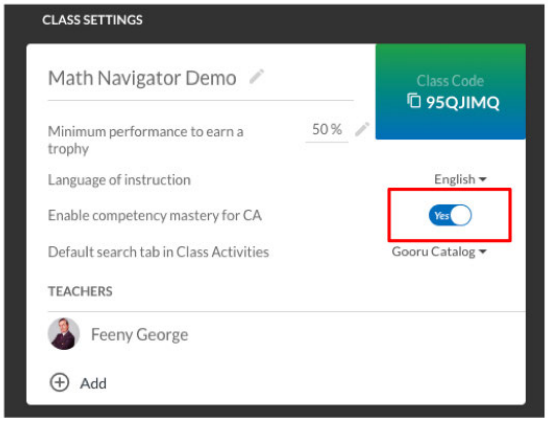
3. Use the search bar and filter icon to narrow the results, as in Option 2.

Enabling Competency Mastery for Class Activities

When enabled, all Class Activities will automatically count towards mastery.

1. Click on the wheel icon to access your Class Settings.

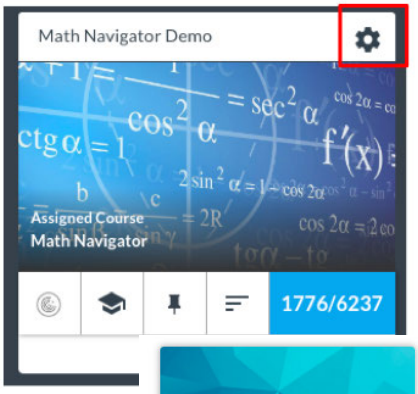
2. Under the Class Settings section, competency mastery can be enabled or disabled (below):



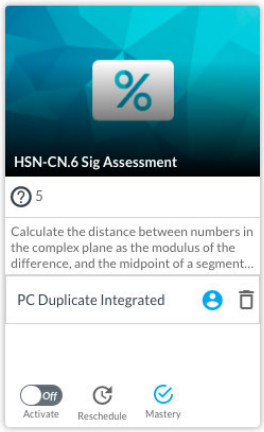
Go Live

Note: Go Live updates and individual and an assessment

1. On your classroom page, click on "Class Activities"

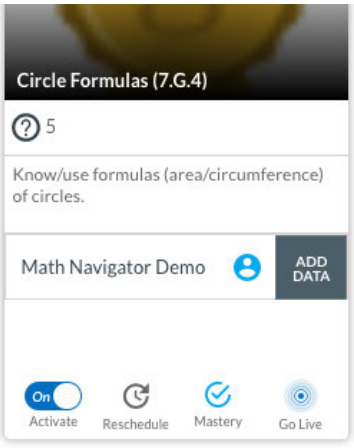


continuously displays class progress in real-time.

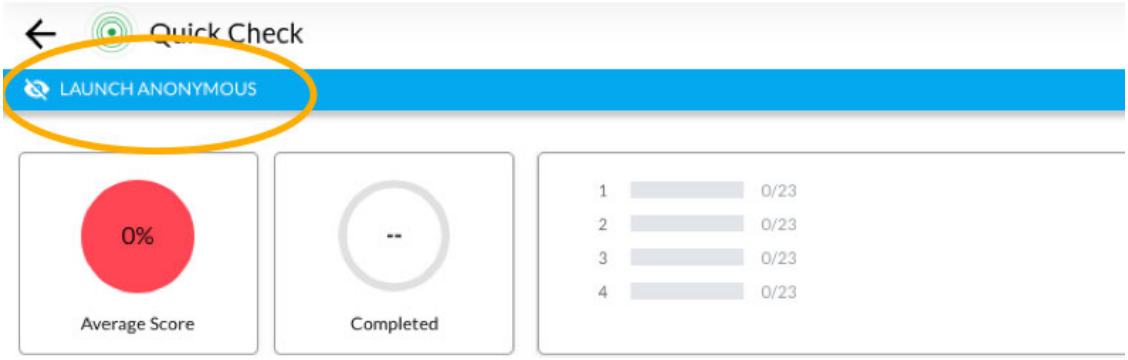


on

2. Add an assessment to your Class Activities page. Once the assessment is posted, click on the "Go Live" icon.

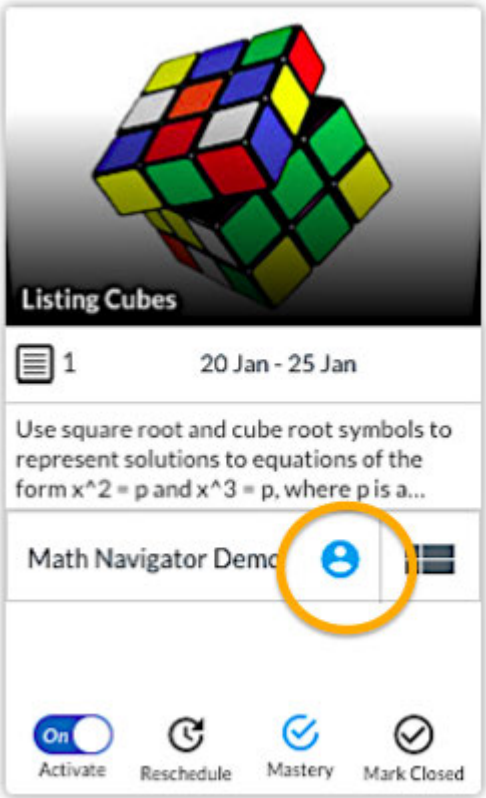


3. You can hide student names by selecting "Launch Anonymous."



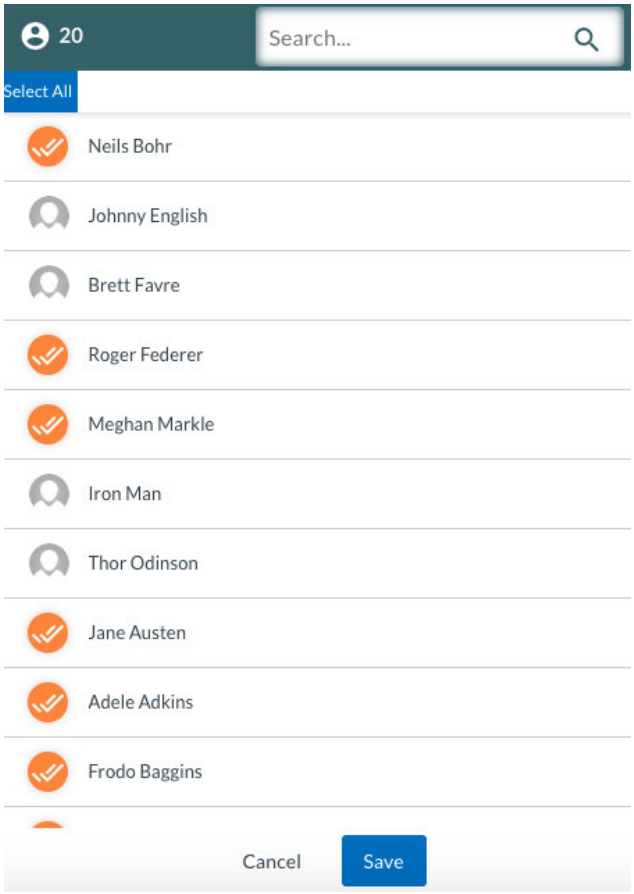
Selecting Students for a Class Activity

1. After you have posted a Class Activity to your page, click on the blue student icon next to the activity's name.



2. The pop-up window shows the selected audience for your activity. You can deselect students or select them by clicking on their avatars.

3. Click "Save" to apply the changes. (Note: an orange check mark indicates that a student *will* be assigned the activity).


20
Search...
Select All
Neils Bohr
Johnny English
Brett Favre
Roger Federer
Meghan Markle
Iron Man
Thor Odinson
Jane Austen
Adele Adkins
Frodo Baggins
Cancel
Save

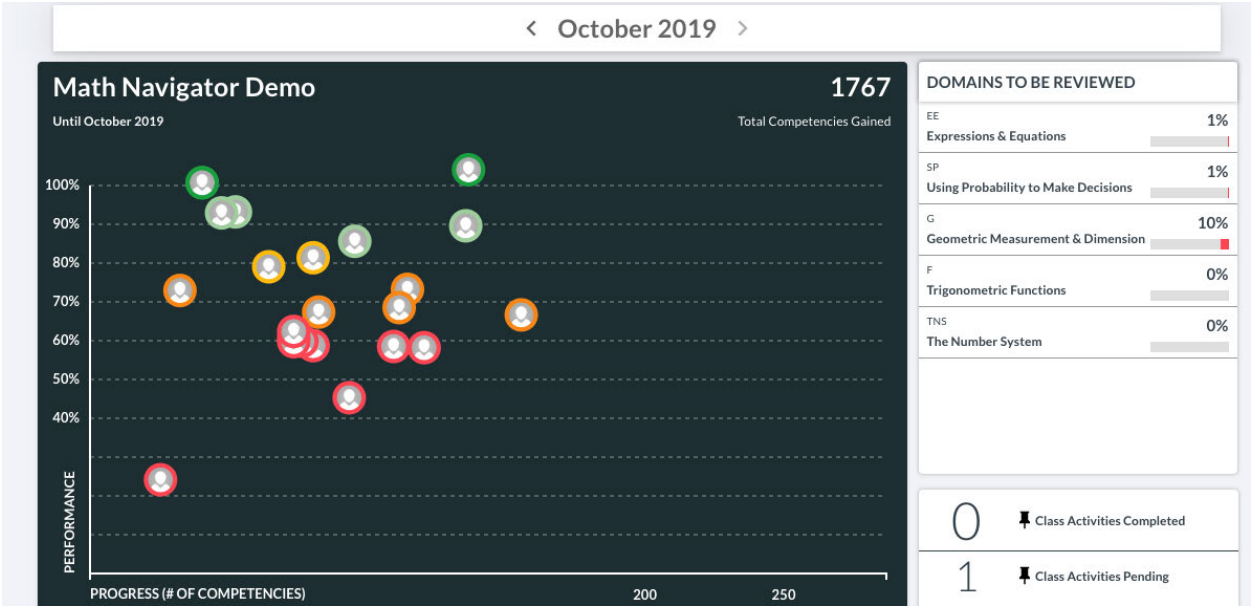
Accessing Data

Performance Overview

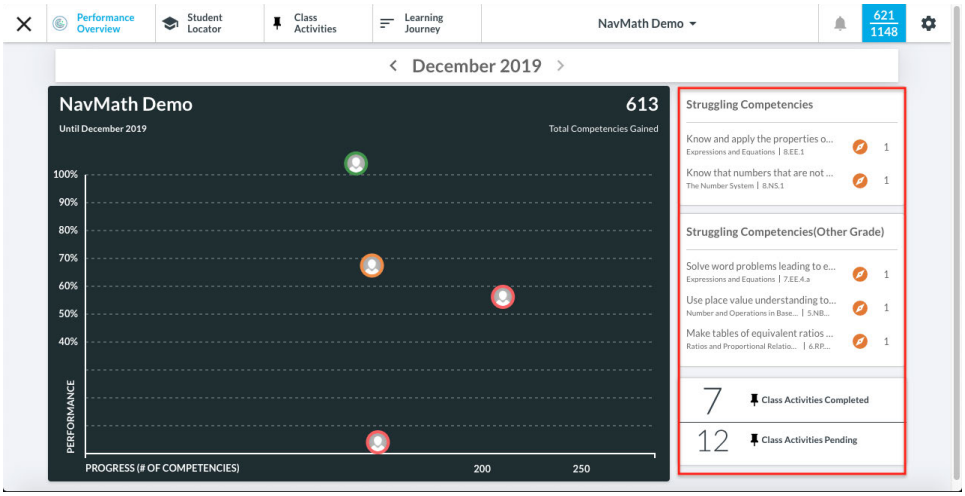
Note: Performance Overview provides a quick summary of your class's progress. Students are plotted on a graph of progress vs. individual performance.

1. On your classroom page, click "Performance Overview"

2. With Performance Overview, you can:
   - View the progress vs. performance of all students in your class
   - Toggle between months
   - Identify how many class activities are pending or completed
   - Hover over a student's image to view their individual data on progress and performance



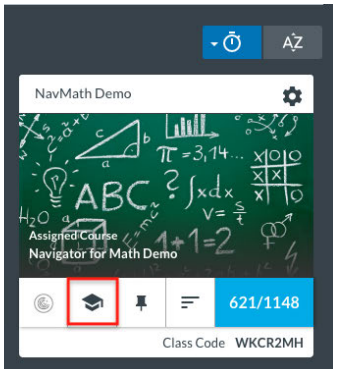
Competencies with which students are struggling (scoring <80%) are listed on the right side of the Performance Overview page. Suggestions can be assigned to struggling students directly from this box by clicking the orange suggestion icon. (See “Making Suggestions”)

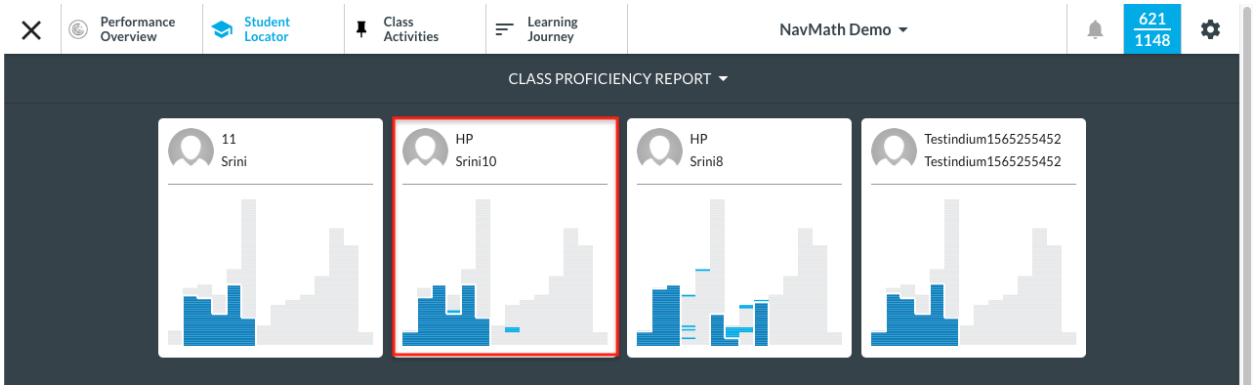


Making Suggestions

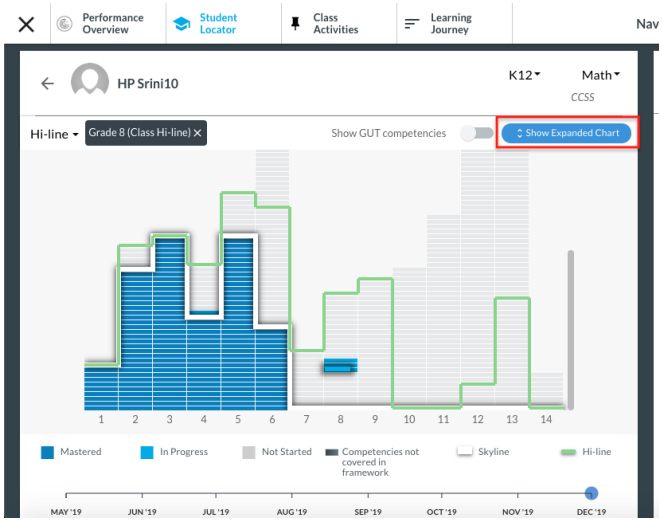
Instructors can choose recommended collections or assessments from a list of suggestions available in multiple locations:

Option 1: Single Student Suggestion via Skyline

1. Click on the “Student Locator” icon on the classroom  card.
2. Select the learner to whom you want to assign the resources or assessments.

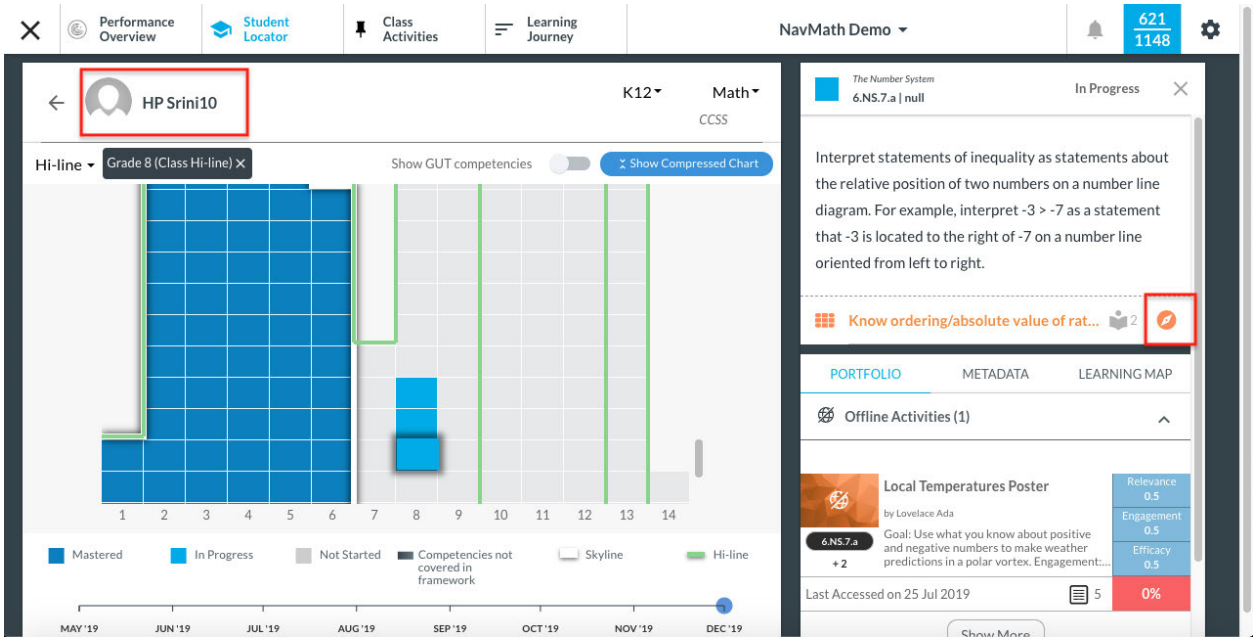


3. Click on the blue “Show Expanded Chart” button to get an enhanced view of the learner’s skyline.

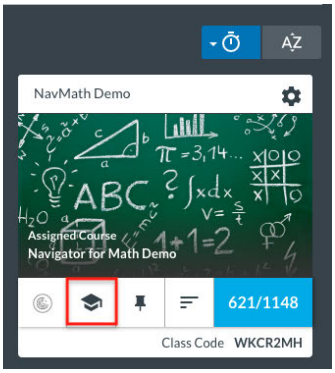


4. Then, select the competency from the skyline with which the learner needs more support by clicking on the corresponding block.

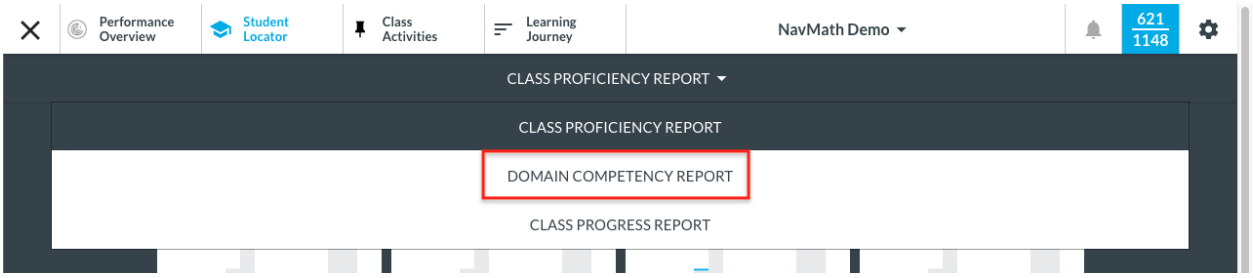
5. Click on the orange suggestion icon (on the right side of the screen) to send the suggested collection or assessment to the learner.



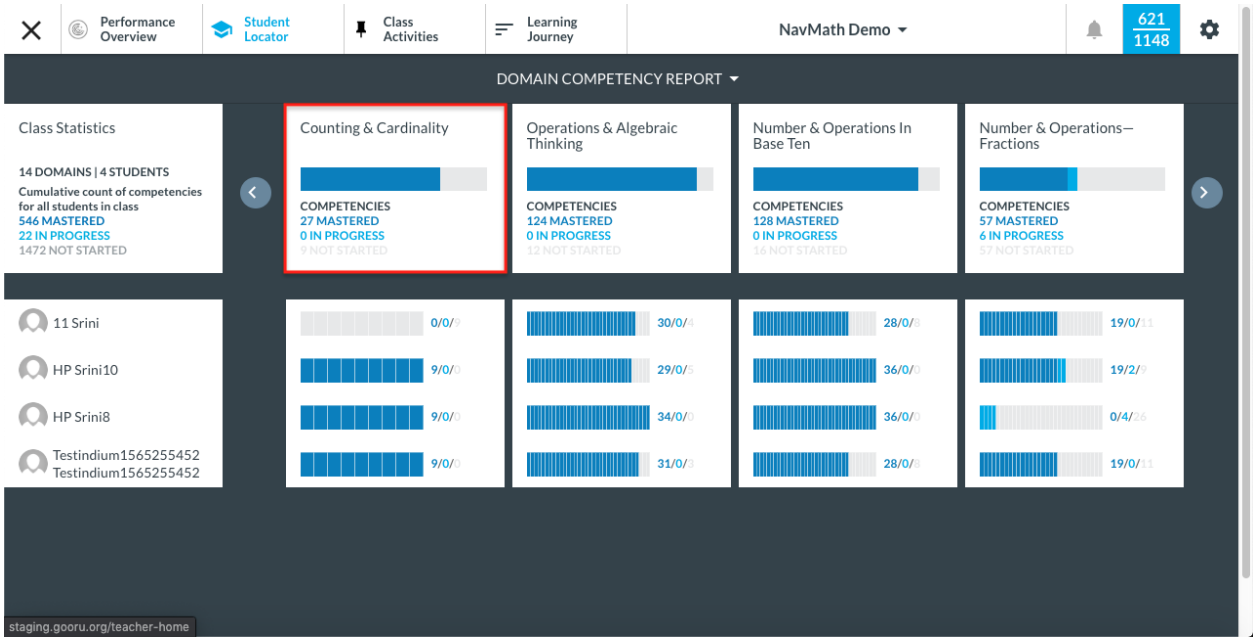
Option 2: Student Suggestion by Domain



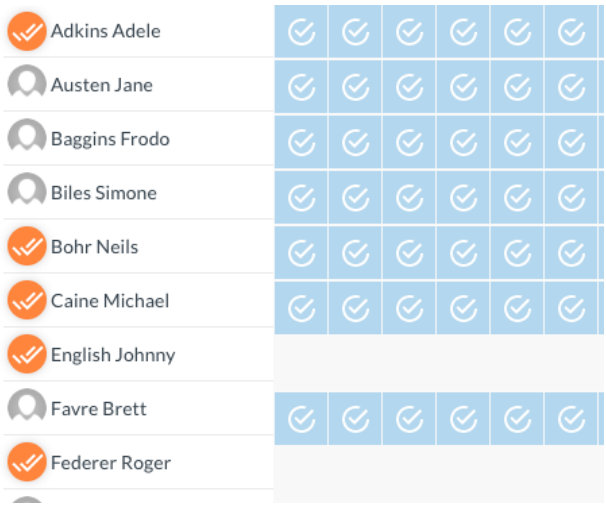
1. Click on the “Student Locator” icon on the classroom card.
2. Select “Domain Competency Report” from the dropdown menu.



3. Select a domain. You can scroll through the course’s domains using the left/right arrows on the screen.

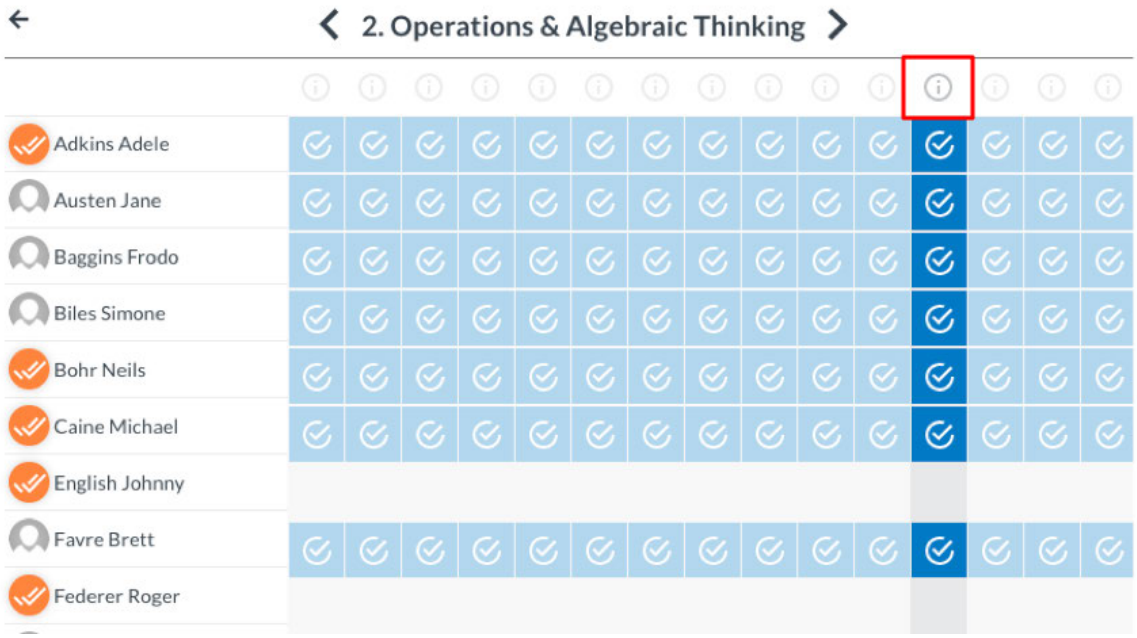


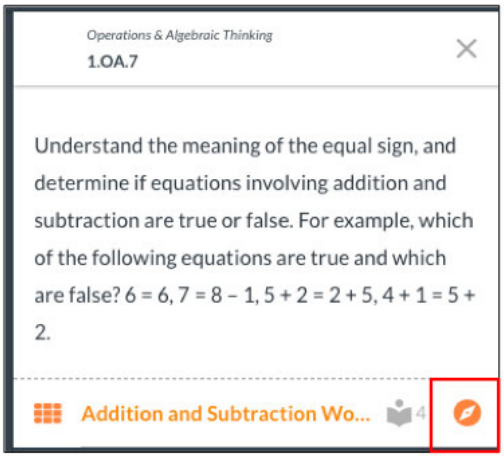
4. Identify which students you would like to receive suggested supplemental resources by clicking on their avatars. Orange check marks indicate that a student has been selected.



In this example, Adele, Neils, Michael, Johnny, and Roger have all been selected to receive the suggestion.

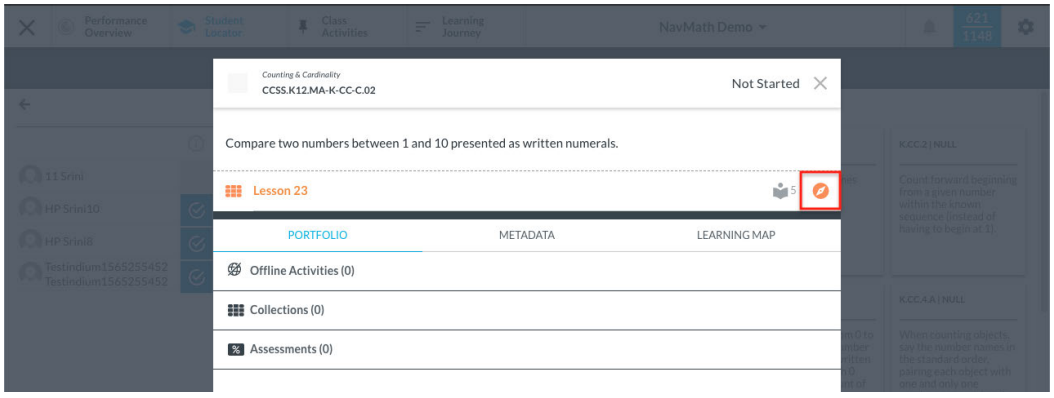
5. Hover over each “i” symbol to get a description of the competency. Click on the “i” of the targeted competency. This will highlight the column in dark blue (shown below).

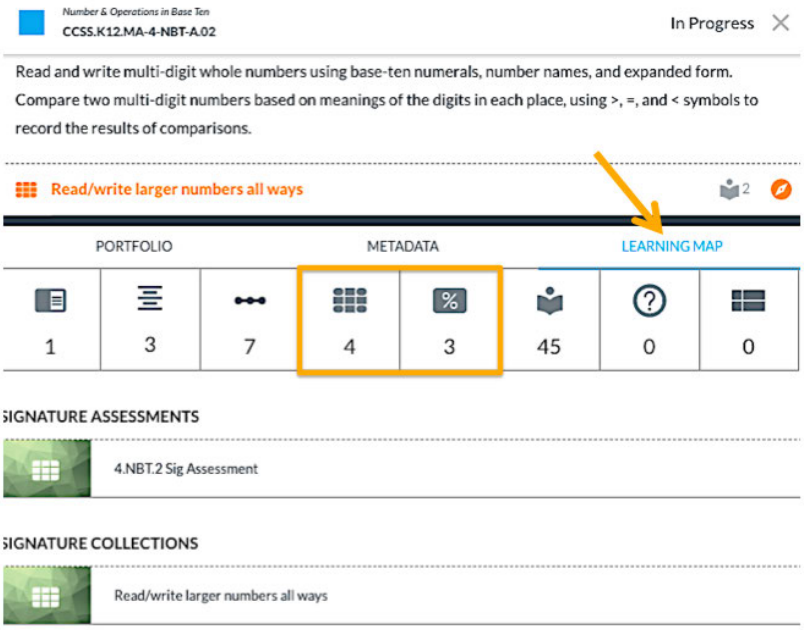




6. On the right side of the screen, click on the orange suggestion icon.

7. In the pop-up window, click on the orange suggestion icon to assign the additional resources and assessments to the cohort of students you selected.

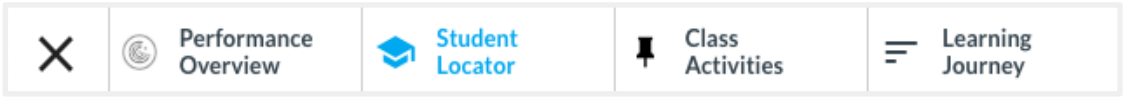


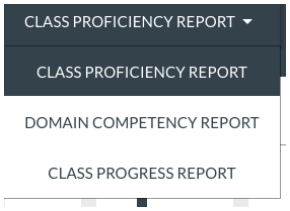


8. You can also select the “Learning Map” tab in the pop-up window, and then click on the collection or assessment tabs (highlighted in the image to the left) to assign any collection or assessment from the list to the student.

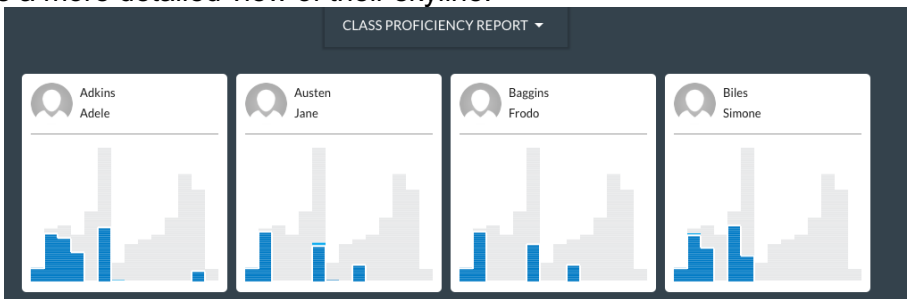
Class Proficiency Report

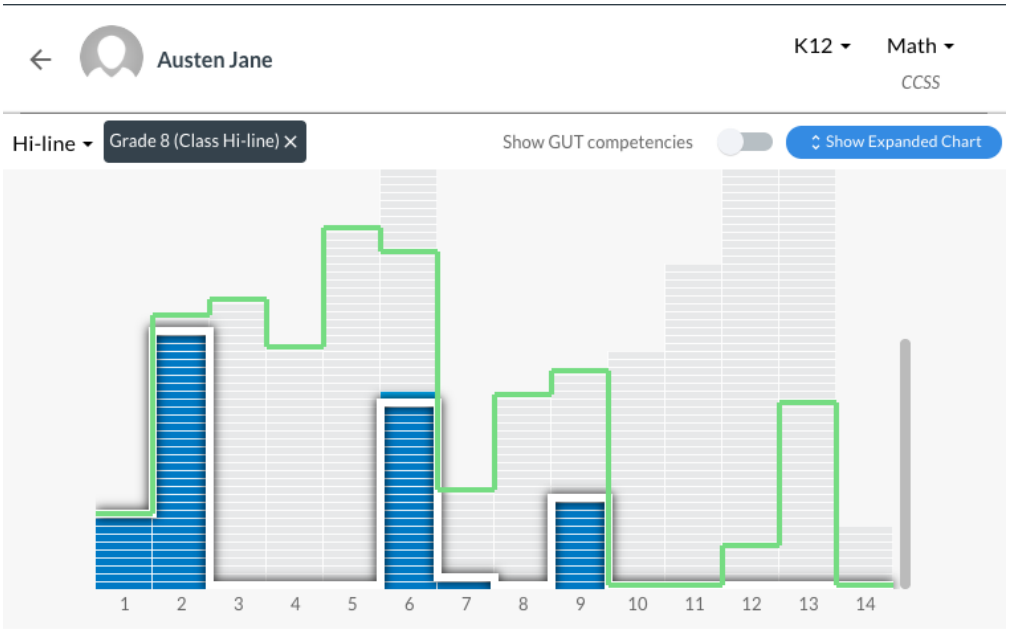
1. On your classroom page, click “Student Locator”



2. Select “Class Proficiency Report” from the drop-down  menu

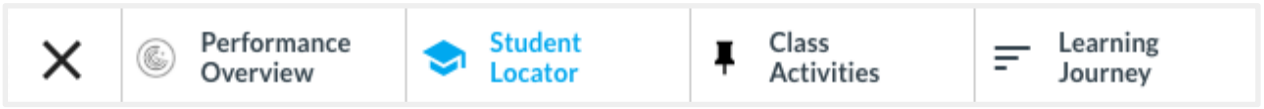
3. The Class Proficiency Report shows each student’s **skyline** - their individualized learning journey. A student builds their skyline as they show proficiency. Clicking on a student in the Class Proficiency Report shows a more detailed view of their skyline.



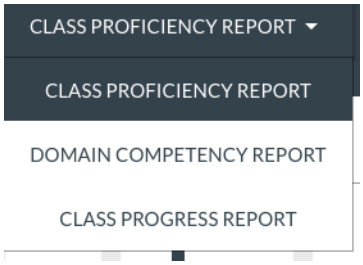


Class Progress Report

1. On your classroom page, click “Student Locator”

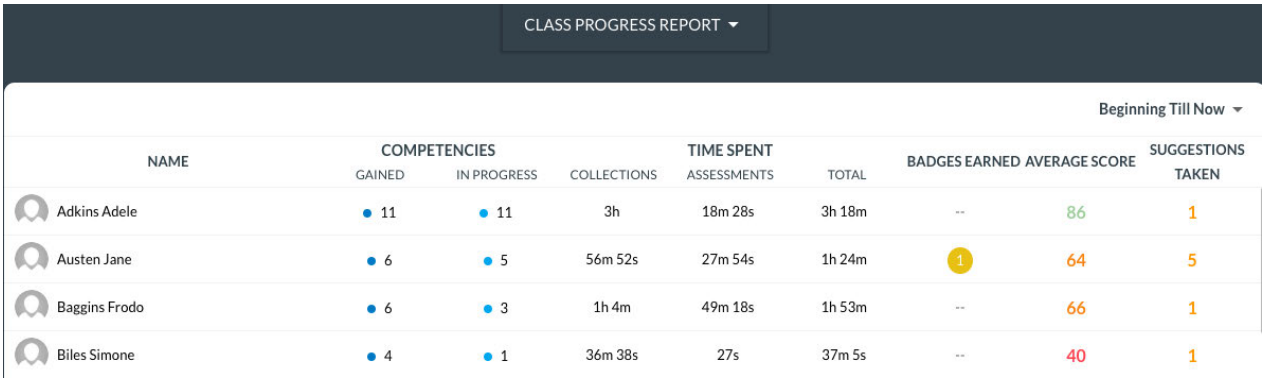


2. Select “Class Progress Report” from the drop-down menu



3. The Class Progress Report shows the comprehensive progress of each student over the last week, since the beginning of the course, or during a custom date range.

   Data include competencies gained, competencies in progress, time spent on collections and assessments, and average assessment score.

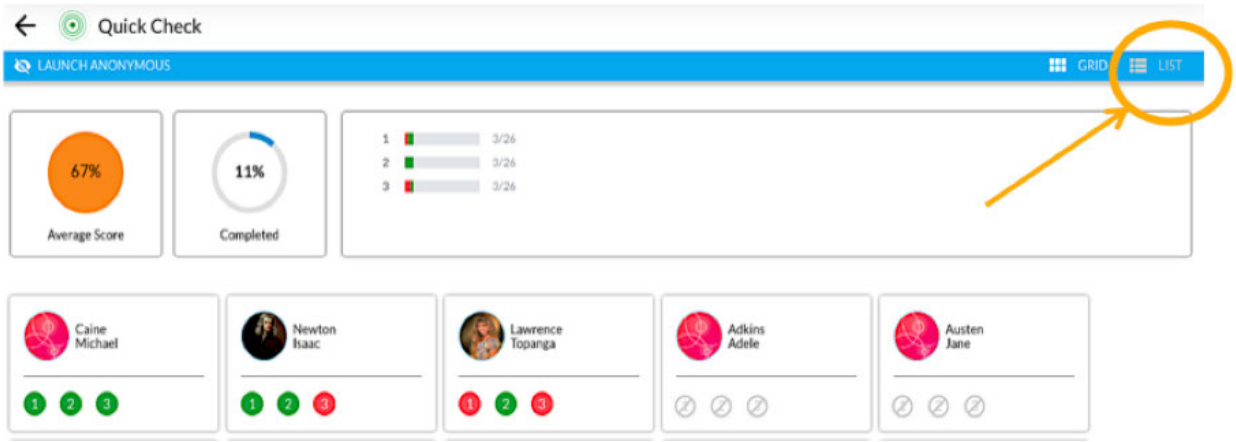


4. You can click on a student's name to see which competencies have been gained and which are in progress.

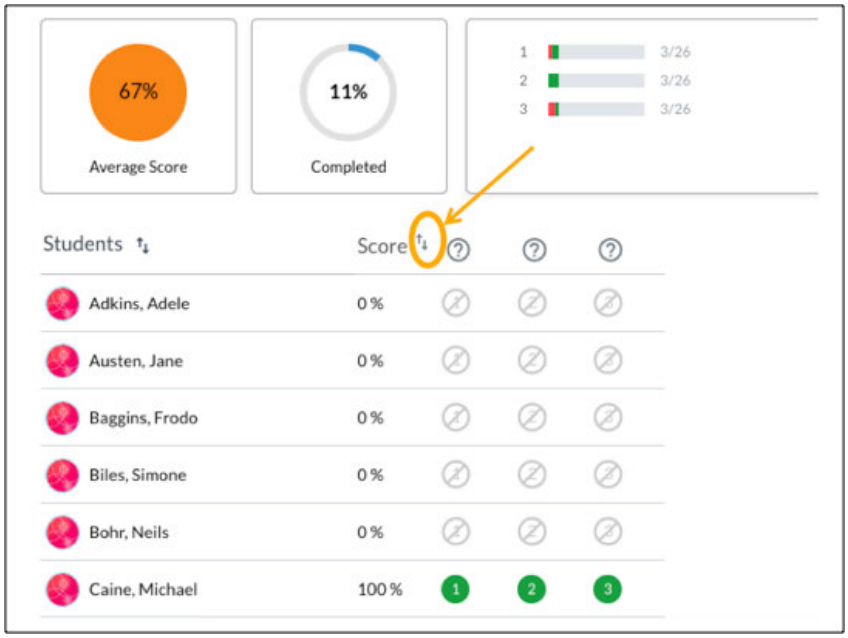
Sorting Go Live Data by Score

Note: Once you have the data sorted, we recommend taking a screenshot or printing the page. After you close Go Live, the data will no longer be sortable.

1. When you use Go Live for a Class Activity, data will populate in real-time as students work. When you're ready to **sort,** click "List" in the upper right corner.



2. Once your data are in list format, click the arrow next to **Score**. This will sort by score from low to high. Click the arrow again to sort by score from high to low.



To sort alphabetically, click the arrow next to "Students."

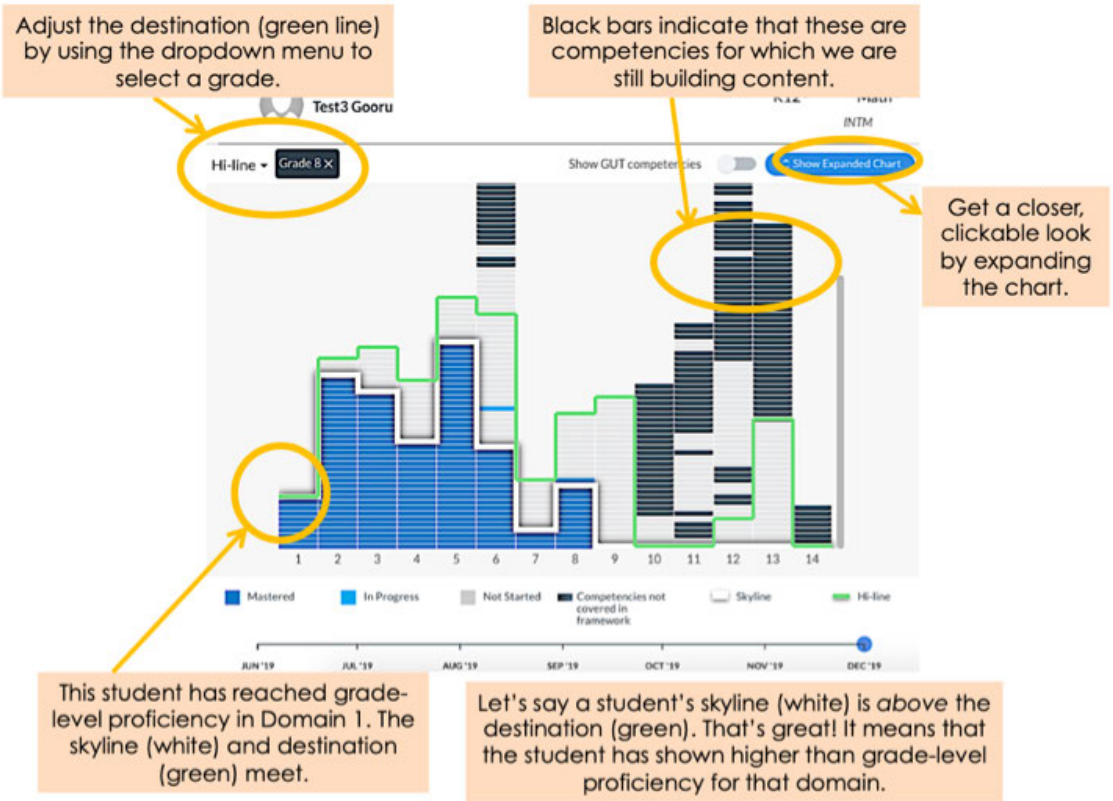
Interpreting a Skyline

In a student's proficiency chart, there are two important lines:

- **Skyline** (white) – "Where is the student now?"
  - Initially set using the student's NWEA MAP data to show the highest standard in each domain for which they showed proficiency
  - Moves upward as the student shows proficiency as they work
  - Travels with the student year-to-year, providing longitudinal data for learners

- **Destination** (green) – "What is the student aiming for?"
  - Also called hi-line
  - Student's goal based on their grade level (i.e., what each student should master to show grade-level proficiency)

The overall goal is for the student's skyline (white) to meet or surpass the destination (green).



Glossary

Terminology	Definition
Air Traffic Control View	Quick view of the entire class plotted on a graph, depicting progress vs performance of each student.
Assessment	A set of scored questions that teachers and students can use to monitor understanding and performance.
Baseline	A black line on the Proficiency Chart that indicates student's starting location, the base-level concepts and skills they have mastered in each domain.
Classroom	"Virtual classroom" within Gooru where teachers add students or students join the classroom and move through content that the teacher has "assigned" or made visible
Classroom Code	A code unique to each classroom that students can use to join the classroom
Classroom Roster	The students (Gooru users) who joined a classroom via classroom code
Collection	A playlist of multimedia resources (videos, website, images, etc) and questions
Competency	This term is synonymous with learning standard. K-12 educators generally understand competencies and content developed at the competency level. Much of Gooru's current content is developed around learning at the competency level.
Competency Report	Competency summary of all students in the class that includes the domain names, how many competencies are in each domain, and each student's mastery status.
Completion	Measured progression through a course by a student
Content Editor	Where a user can build and edit their content on Gooru. Also where they develop the course hierarchy (unit, lesson, and collection folders) and add and organize content
Content Information	A designated section in a collection, unit, or course to provide additional thoughts about the content
Content Settings	The status of content in terms of visibility - can be private, shared individually, assigned to a student or students, or publically visible.
Course	A "folder" that allows you to organize your learning content into units and lessons.
Course Map	View of all the content available in a classroom
Course Objectives	Goals for academic mastery within the course designated and written by the author
Current Location	Where a student is in a Gooru course in terms of mastery of a standard with regard to the learning targets. Demonstrates understanding and progression through the collections, lesson, unit, and course.

Class Activities	Collections and assessments that teachers want students to complete that day while monitoring their results in real-time. It allows teachers to provide real-time help to students, who may or may not complete the whole assessment or face difficulties. Teachers can view student performance data by launching the Go Live dashboard to see how students responded.
Data	Data of student assessments, attempts, scores, etc for a classroom
Drag and Drop Order	Question type in which student clicks and drags multiple answer choices so that they are in the order as requested in the question.
Fill-in-the-blank	Question type in which student enters text to complete a statement in the question that is missing words or phrases.
Gooru Catalog	The set of all content available in Gooru, searchable by collection, resource, and standard.
Highlight Text	Question type in which student highlights a text or a sentence to select the correct answer as requested in the question.
Lesson	A lesson is a sub-folder in the course folder hierarchy.
Library	The content on a user's profile--their courses, collections, assessments, questions, and uploaded resources
Metadata	Information associated/tagged on resources, assessments, collections, courses; includes areas such as standard, grade level, type of media, etc.
Multiple Answer	Question type in which student selects "Yes" or "No" out of multiple choices.
Multiple Choice	Question type in which student selects one answer out of multiple.
Multiple Select - Image	Question type in which student selects one or more images available as answer choices.
Narration	Text that pops up before a resource to describe it and/or give instructions to the student studying the collection
Go Live	When the teacher selects “Go Live” for an assessment, the results for each question for the whole class is visible on the teacher’s dashboard in real-time
Performance Data	View of progress within a unit, lesson, or collection
Proficiency	Student skill level.
Profiles	A collection of information on the user and the user's content (the user could be an individual or an organization) that the user can update. Profiles are publicly visible.
Reaction	Student response to a resource via a likert-type scale
Real-Time Performance	Visual representation of completion and mastery within a collection, lesson, unit and course
Remix	Copy and save content to your account
Resource	A resource is multimedia content in a variety of formats such as videos, interactives, websites, images, Google docs, and more.

Search (Collection)	User enters a keyword in the search box to find and filter Gooru's catalog to find relevant collections
Search (Resource)	User enters a keyword in the search box to find and filter Gooru's catalog to find relevant resources
Signature Assessment	Assessment offered to students when they receive >= 80% on their Course Assessment, allowing them the opportunity to gain mastery of a competency.
Signature Collection	Additional resources offered to students to enhance their understanding of the concept if they receive < 80% on the Course Assessment.
Skyline	A white line on the Proficiency Chart that represents the highest-level concepts and skills they have mastered in each domain.
Standards	District, state or national frameworks that define content areas and skills. Well-known standards include Common Core, Next Gen Science Standards
Teacher Suggestion	Additional collections or assessments offered to students to reinforce concepts where students may be struggling.
Time Spent	Amount of time spent on a question, resource, assessment, or collection
Unit	A unit is a sub-folder in the course folder hierarchy.

Updated NILE Architecture

Navigator Core

Overview

Navigator solution is expansive and supports an array of capabilities that can be leveraged by partners. This document covers the Navigator platform architecture detail.

Navigator Overview

Navigator supports teachers, and with the right tools, to enable them advance student outcomes and foster an innovative community that supports the success of every student. At the core of Navigator is the huge catalog of open education resources that are highly curated and metadata enhanced. This helps teachers pull together relevant content items for students. As students interact with the content items, the generated data is analyzed in various dimensions. The rich metadata at content helps Navigator track learner proficiency at various concepts.

Navigator Platform Components

Navigator platform comprises of various task specific microservices and orchestrators that can broadly be categorized into:

- **User Identity and Authorization**: Handles user identity management, authorization checks, auth token issue, and Single-Sign On (SSO) support.
- **Competency model**: Handles taxonomy model, framework setup and crosswalk support
- **Catalog services**: Handles content Create, Read, Update & Delete (CRUD) services, content index and content search
- **Suggest engine**: Handles suggestions based on learner identity (profile), learning maps (catalog) and principles of learning
- **Data write and read services**: Handles data logging, data reports, and data postback
- **Datascope** (Data stream processing pipeline)
- **Learner profile services**: Handles learner profile compute and read services
- **Navigator application**: Orchestrates all above to pull together various learner flows

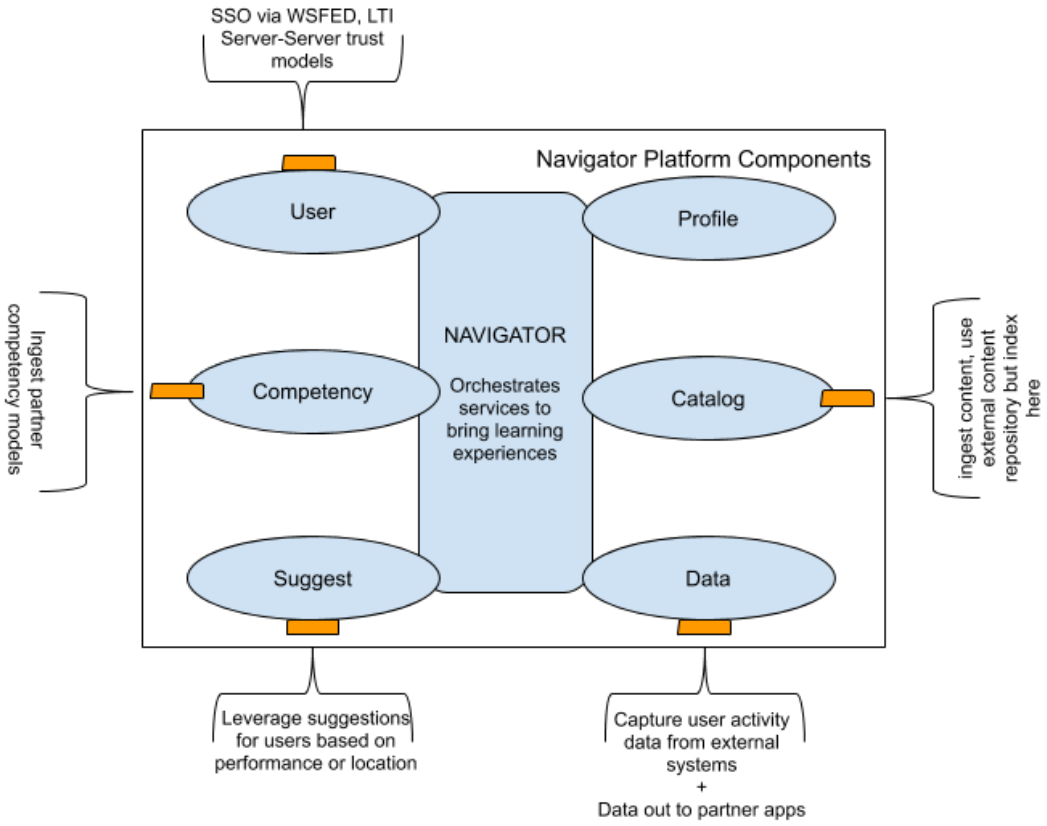
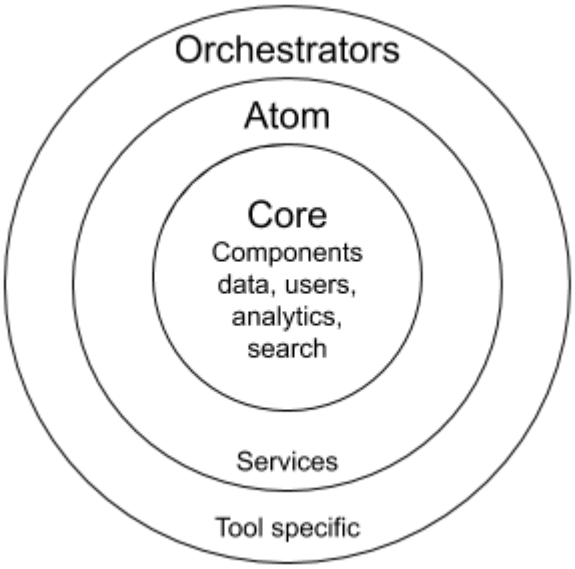


Figure: Navigator Platform Components & Integration Points

Navigator Core Design Principle

Core layer is responsible for data management, provides data level CRUD capability as API, manages user, content, class and usage data. Atoms and Orchestrators built around these Core capabilities extend to specific needs is the preferred model. This helps us manage extensions consistently and without a lot of code bloat.



Essentially, our design would follow the below structure:

- Different data sources core to our systems will be central and abstracted to external layers
- Core services expose the data sources to achieve one level of abstraction to avoid systems tied directly to data sources.
- Extension APIs work on top of Core APIs and will be a mechanism to feedback data / information into the data sources which will be assimilated and used
- Orchestration APIs enable different kinds of responses for custom usage of tools. This could be either single orchestration API, which could be reused, or sub system level orchestration layer, which talks to core APIs to assemble the required responses.
- Subsystems and Tools layer which will provide the experience are built on top of this
- Message bus will enable different types of communication between different nodes of platform

Further, each Core component and services comply with the following guidelines:

- RESTful interface exposes core features, JSON responses only
- Controller-service separation of concerns
- Cacheable and layered architecture
- API versioning followed when upgrading the same
- Event based update of other subsystems like search / analytics
- All API protected with authentication key to avoid denial of service (DOS) attacks
- Role based access control for authorization of API

The basic constraint behind the architectural choices is governed by scalability. The system is designed to be horizontally scalable. To achieve this most core components are written using async frameworks. Using async frameworks enables the blocking operations to happen on worker pool threads while main threads are available for servicing user requests. Note that because of JDBC layer, the architecture is not async end to end.

Technology Stack

Platform	Purpose
Ubuntu 16.04 or above	OS recommended; but you can use other variants as well
Java 8	All backend microservices are implemented in Java
Kafka 0.10.1.1	Messaging system for cross-service communications
Zookeeper 3.4.10	Distributed coordination of kafka nodes
Cassandra 3.0	Data store for raw/unprocessed usage events
Elasticsearch 5.6.0	Content index and search engine backbone
Postgres DB 10.3	Database to store all relational and transactional data
Redis 4.0	Cache storage for frequently accessed critical data
Memcached	Cache storage used at real-time subsystem
HAProxy 1.6 or above	To manage routing rules to various services / sub-systems as well as load balance traffic; other alternatives like Application Load Balancer can also be used for the same purpose
Nginx 1.14.0 or above	To manage multiple front-end sites; other alternatives can also be used for the same purpose
Gradle 2.7 & 4.4	Build tool for most of the back-end microservices; some services need higher version of gradle as noted at build.gradle files in respective service repository
NodeJs 4.4.*	For frontend build as well as to run specific identity provisioning service component
Tomcat 8.0 Maven 3.3.9	While most of the component redesign has moved away from Maven build and managed under Tomcat server, there is one server (search and suggest service) that still is not migrated away. Hence this is needed today but may move away from the requirement at a later time.
Ember-Cli, Bower, PhantomJs,	Build environment for front-end application

Grunt, Stubby	

Architecture Overview

Component Architecture

At the heart of the Navigator are the Core services. These are designed as microservices created using VertX. API access requires an auth token which is validated at every API entry point. The tokens are managed at Redis cache and expiry is extended every time an API is invoked. Postgres database is the key data store for all core entities - user, content, classes, competencies. Messaging across clustered sub-systems is managed via Kafka message bus. Search and Suggest subsystem uses ElasticSearch. All activity data is stored at Cassandra while processed user activity data required for explicit reporting purposes is captured at analytics Postgres database. All services are exposed over Restful API and responses are always in JSON format. Frontend is developed in EmberJS and orchestrates the various services per application flow needs.

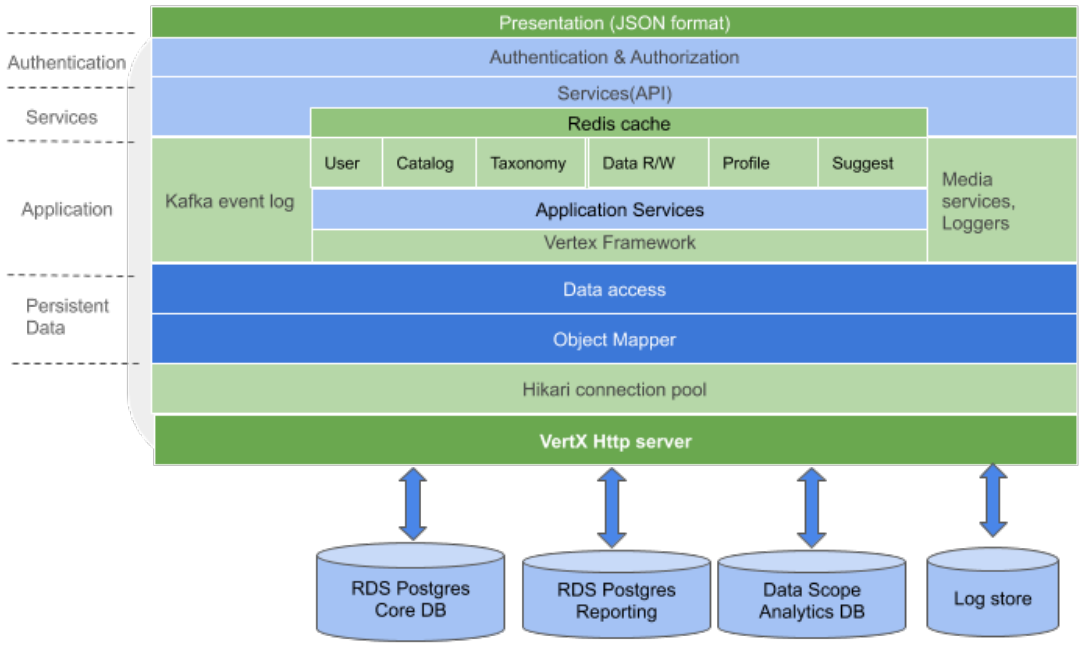


Figure: Core Architecture

Key Service Components are as below:

- Core Services: Content, Class, Metadata and Taxonomy
- Auth Services: Users, Authorization, Tenant, Partner, App
- Study Suggestions
- Class and User Reports
- Realtime
- User Interface: Navigator, Widgets

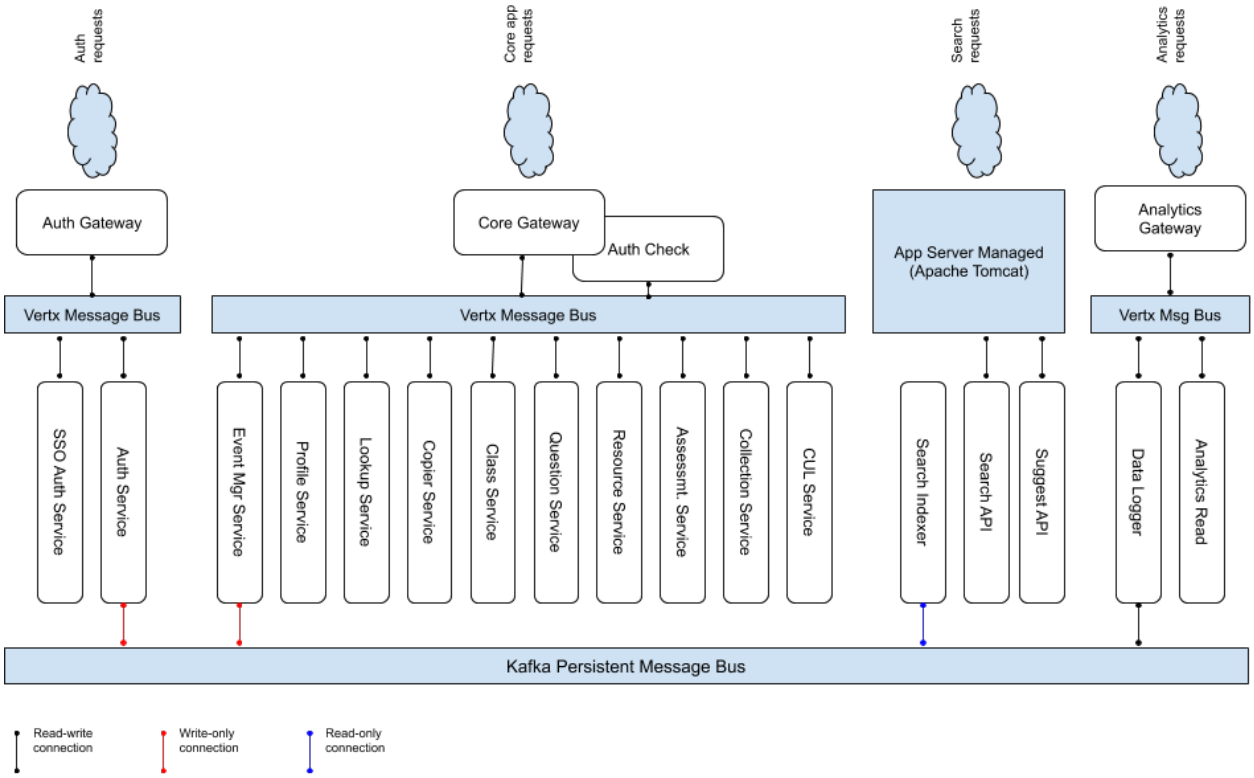
- Partner Integration Components:
    - SSO support for service providers LTI, GMail, SAML, WSFed, Shibboleth
    - Data In APIs
    - Data Out APIs / Post-back
    - Roster Sync

Clusters

Each component is a collection of one or more processes running. With such a view each component forms a cluster. In a cluster, there is at least one cluster member which is responsible for handling incoming requests. These requests are mostly HTTP based and thus the cluster member handling them is the API gateway. The gateway is aware of its members and what requests each member can process. Its primary task is to act as a web server listening for requests. Each cluster generally contains one cluster member which is responsible for auth checks. API gateway will bind this cluster member for all APIs to make sure that valid requests are reaching the downstream handlers.

Auth check handlers are responsible to verify the access token provided in request with respect to token store present at Redis. In case of auth failures, the response is returned to caller. If auth is successful, then request is dispatched to the next handler.

The communication between cluster members happen using TCP with a request response model which is asynchronous. The cluster management happens using Hazelcast where in a new cluster member is automatically detected and is made part of cluster.



When it comes to horizontal scaling, the architecture provides multiple avenues to scale, e.g.

- One gateway cluster member backed by multiple instances of processing handlers. The load balancing in this model happens in round robin fashion.
- The cluster itself is replicated and load balancing is managed at the level of entry point of deployment infra (e.g. haproxy, nginx etc)
- These two approaches can be mixed as well. E.g. if there is load on only single handler for a cluster, only one cluster can be maintained and handler can with multiple instances.

There is another level wherein scalability can be dealt with. Since the clusters are implemented to be asynchronous, there is a concept of event loop threads and worker threads. Their count is configuration governed and in case where scaling needs change, the configuration can be tweaked to provide better scalability without additional hardware.

While the intracluster communication happens with the TCP bus which is non persistent and local to the cluster, the inter cluster communication happens using different methodologies. Two of which are HTTP calls and Kafka. Kafka communication is used where the communication does not need to be request/response type. However, if there is a need for request/response type of communication across the cluster, then HTTP based communication is used.

Environments

There are different environments for different purposes. Stage, Beta and Production are the most important ones from release readiness perspective. Development and QA environments are important from an engineering perspective for feature creation.

Each environment is completely isolated from other environments to provide for better security. This is done using Virtual Private Clouds specific to that environment. This makes sure that no member from one environment is able to access any member from other environments. The communication, if it has to happen, needs to happen using authenticated APIs.

Data In & Data Out

At a broad level, event data coming in is streamed to Kafka message bus which keeps accumulating events while ensuring all consumers in the system get a copy of the message. There are task/metric specific consumers that pick up events from Kafka, process and store them as needed. All events are always stored at Cassandra DB for any future reference. Class and other learner specific events data is processed for reporting needs and processed report data is stored at PostgreSQL reports database for better query ability. Additional aggregates computed for purposes of Learner Profile are stored at Postgres database.

Data In & Data Out components expose REST api for data logging with XAPI compliance on request payloads related to learner study activities.

Navigator collects data related to Content, Learner Study activities, Search, Social and other User events, whether originating within Navigator system or due to ingestion by partner systems.

The activity data is stored in a manner to enable filter on multiple dimensions: time, user, action, and content. Data would be used to generate Aggregated Reports, Learner Profile, Learner Performance report, Admin Reports, depict User Journeys, and more.

Data In

Navigator collects data as users interact with the system. There are a variety of events that generate activity data. Some are events explicitly logged by application / components (client-side events); and some are generated as a side-effect of certain user actions (server-side events). Further, there are scenarios where external entities log data to Navigator via an exposed API.

Event Types

Navigator application is one of the primary event generators for the learner activities. Events are generated for learners':

- Independent Learning Activities where learner self-studies an available course
- In-Class Activities where learner is assisted by teacher but learning is self-paced
- Daily Class Activities where teacher focuses entire class on a specific day activity
- Content Activities where users create and update content, collaboration, reuse, view
- Other Activities related to user sign-up, login, class join, follow & other social interactions

Typically, for a learner, the progress, performance and proficiency are key performance indicators (KPIs) captured. So, Score, Time spent and reactions for the given context of learning are logged in the system, as learners work with the Navigator application. There are additional events captured related to teacher grading of free response questions, teacher override of score, etc that impact the above KPIs.

Additionally we also plan to associate Mindset Vectors, Community Vectors, Skills vectors to describe the Learner's profile and to help the system locate the learner more precisely and to enhance search and suggest for the learner. Some of the examples of these vectors are - Self Confidence, Grit, Perseverance, Engagement, Citizenship, Authority etc.

Note that using the Navigator application triggers the data collection automatically. However, the models where partner users are not using Navigator application is also supported. In this model, Partners and System Integrators send discrete events to Navigator. Partners may send events in the standard XAPI-formatted payload or a variant based upon mutually agreed API contracts.

Different CRUD operations on the Navigator Core entities like Class Create, Course Copy, Add Content are captured as internal server-generated events. Internal events are also generated for other activities like User Sign-in, Assigning Collaborators, Student joining Classes etc.

Event Logger

The events generated are stored as Raw Events at Cassandra and are also further processed by the Event Logger Component. The events are also stored as commit log in Kafka for a predefined time.

The Event Logger is setup as a scalable and a high availability (HA) Cluster. The requests coming to the Event Logger are HTTP which is handled by an API Gateway. Primarily the API Gateway is a Web Server capable of handling high volume requests.

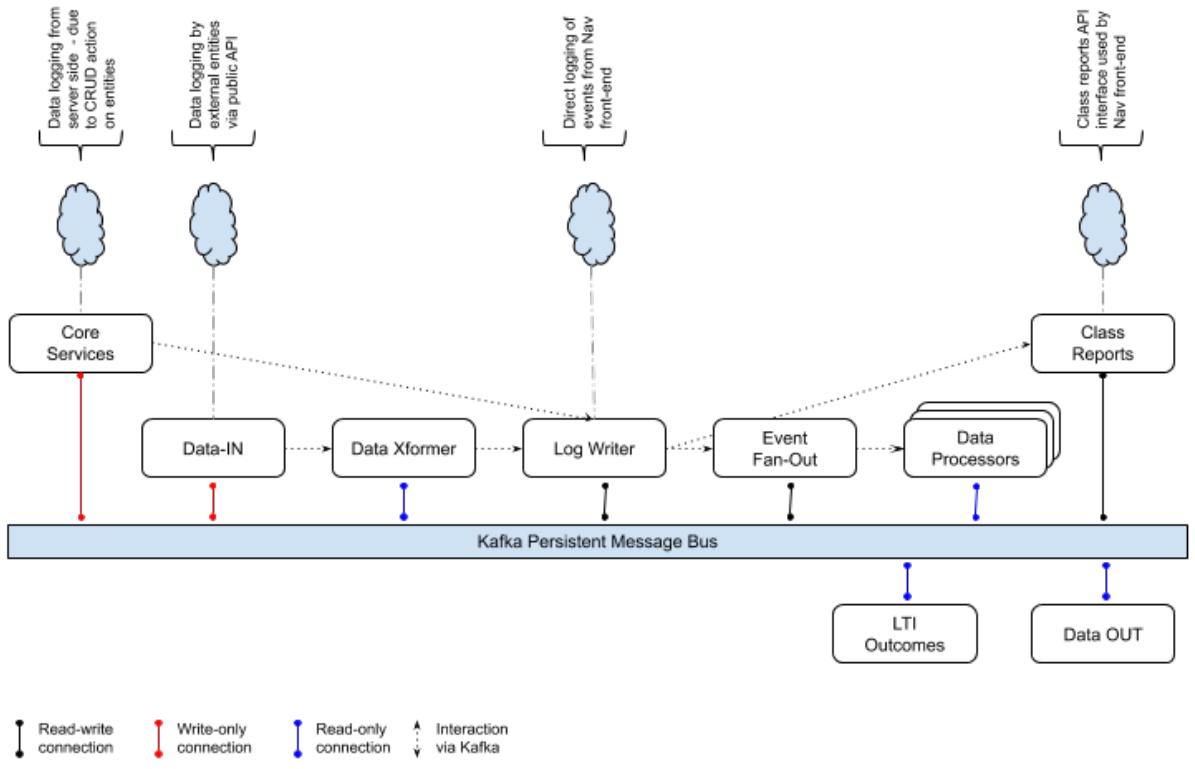
The gateway relays the requests (events) to the downstream processor(s) via Kafka. The downstream event processor is responsible to process every event and store the data into the underlying data store, PostgreSQL database.

Event Transformer

Partners & system Integrators can send data to Gooru using the shared Navigator APIs. Partners can send data to Navigator as a JSON Payload that is xAPI Standards compliant or any mutually agreed upon custom format. The Event Transformer component will transform the Partner messages into Navigator format and relay them down to other event processors and data stores.

Event Processing

All events are collected at data-in / event logger component which stores all raw events at Cassandra while propagating the events further for additional custom processing. The event fan-out component is responsible for generating necessary events and push through the message bus. Specific data processors then consume the messages and compute the specific metric that each one of the processors is responsible for. Based on the event type, the Data Out and outcome post-back components do the necessary additional transformation and relay of the same to the partner environment.



Event Flow Design

Global Event Processors

The events flowing in from the outside world are mostly stateless atomic events and they may not carry a lot of context information with them. Note that context implies the association or references to the metrics associated with the events and not necessarily imply the context of event attributes. For example, the content usage events are atomic and only project information related to the score of a particular question identified in that Events or time spent by the user on that particular resource. However, it is always desirable to know the cumulative time spent for the entire collection or the cumulative score in assessment.

This needs further processing of the events by correlating it to the current context in the system. Global Event Processors (GEP) enables this post-processing of events. There maybe one or more global event processors in the system.

Event Fan-Out

The GEP events are sourced to the Event Fan-Out / Demultiplexer component. The primary purpose of Event Demux is to segregate the factual information provided in the GEP events, create specific internal events per measure (metric) and emit these Discrete Events downstream to the Discrete Event Processors. For example, the key measures (metrics) associated with an Assessment are time spent, score and reaction. Once the assessment is completed an event containing information of the Overall time spent, score and reaction along with the dimension information (User, Course, Unit, Lesson, etc.) is sent to the Demux. Demux will create three discrete events for the three measures (including the event attributes/dimensions) and send it downstream for further processing.

Discrete Event Processors

The discrete event processor (DEP) is responsible to process every event and store data into the underlying data store. Based on the events being consumed by the DEP's, they are also chained, i.e. one DEP can act as an event producer for another DEP.

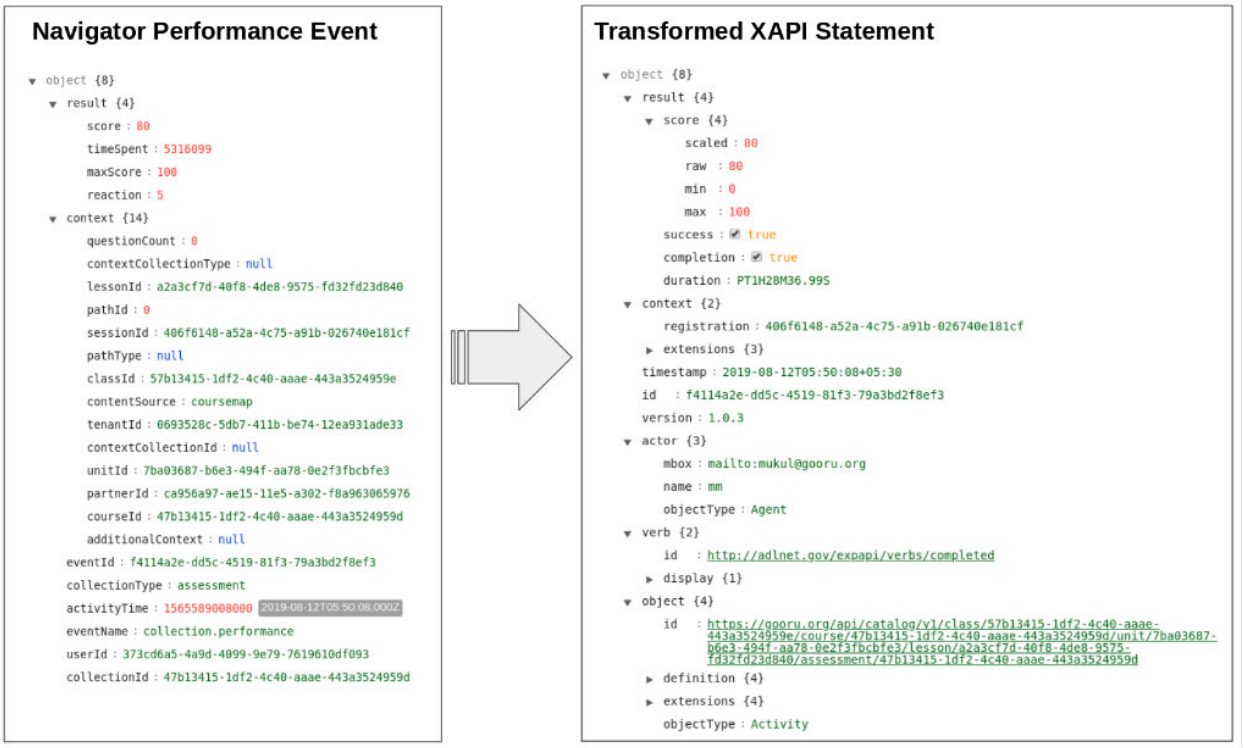
The processed Data from the DEP's will serve the Partner Data Out components and the Data Read API's.

Navigator XPI Transformer

xAPI is an interoperability specification that allows learning technologies to talk to one another. Integrating two xAPI conformant systems can be considerably faster and cheaper. This includes migrating data from old learning systems to new ones.It is possible to collect data about the wide range of experiences a Learner has (online and offline). xAPI captures data in a consistent format about a person or group's activities from many technologies. Heterogeneous systems are able to securely communicate by capturing and sharing this stream of activities using xAPI's simple vocabulary.

The Navigator Event Transformer will transform the relevant native Navigator Events into XAPI Statements. These XAPI Statements will be stored into Learning Record Store (LRS)

Sample Navigator Event Transformation to XAPI Statement



Data Out

Generically, data is served to the external world using the API Infrastructure. Navigator frontend application uses the available API for various in-product reporting purposes. Data can be pulled by the partners using the REST APIs. These data read / reporting APIs are tuned to report data at a specific context level - class level of learner activity and/or independent learner activity related reports.

Additional Data Out components optimized for Partners' specialized data requests are also created. Data Out component can be configured to handle a data pull request by the partners or data can be pushed to the partners at specified triggers. The trigger can be an incoming event or a specific time interval. Data Out also includes the IMS LTI outcomes service that allows Gooru to send information like score, time spent on content to its consumers (typically the Learning Management Systems).

For Interoperability with other Learning Technology Platforms/products Navigator Activity data is stored into Learning Record Store (LRS) as XAPI Statements. RESTful APIs are exposed to GET/POST XAPI Statements into this LRS. Apart from the Navigator Activity events, events from external learning systems can be stored and fetched from this LRS.

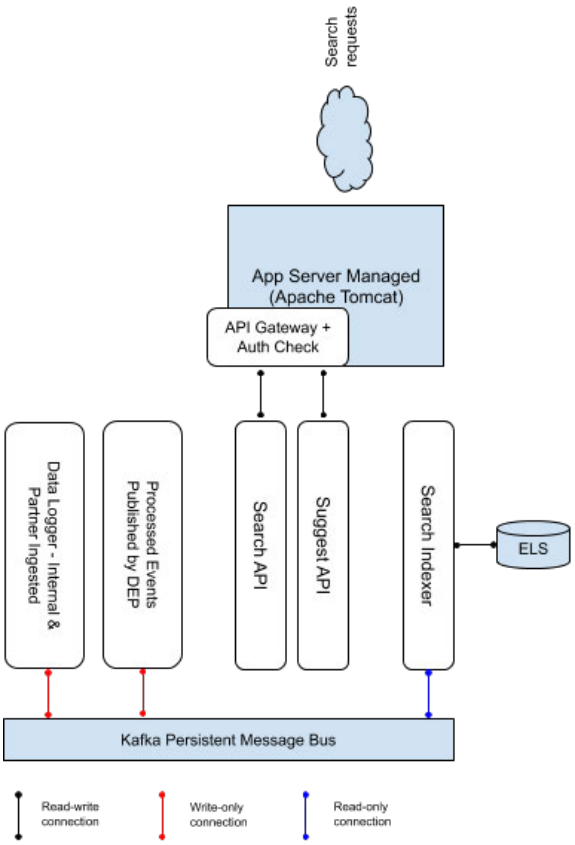
Search & Suggest

Navigator uses ElasticSearch (ES) at its core for Search and Suggest services. All content catalog items, along with its rich metadata, are stored at index as terms for faster search experience. There are multiple indexes in use that are used for specific purpose either standalone or in conjunction with the other: resource index, collection index, course index, unit index, lesson index, content publishers index and crosswalk index.

Architecture

Search infrastructure is distributed in nature and is designed to be horizontally scalable. Key components of the infrastructure include

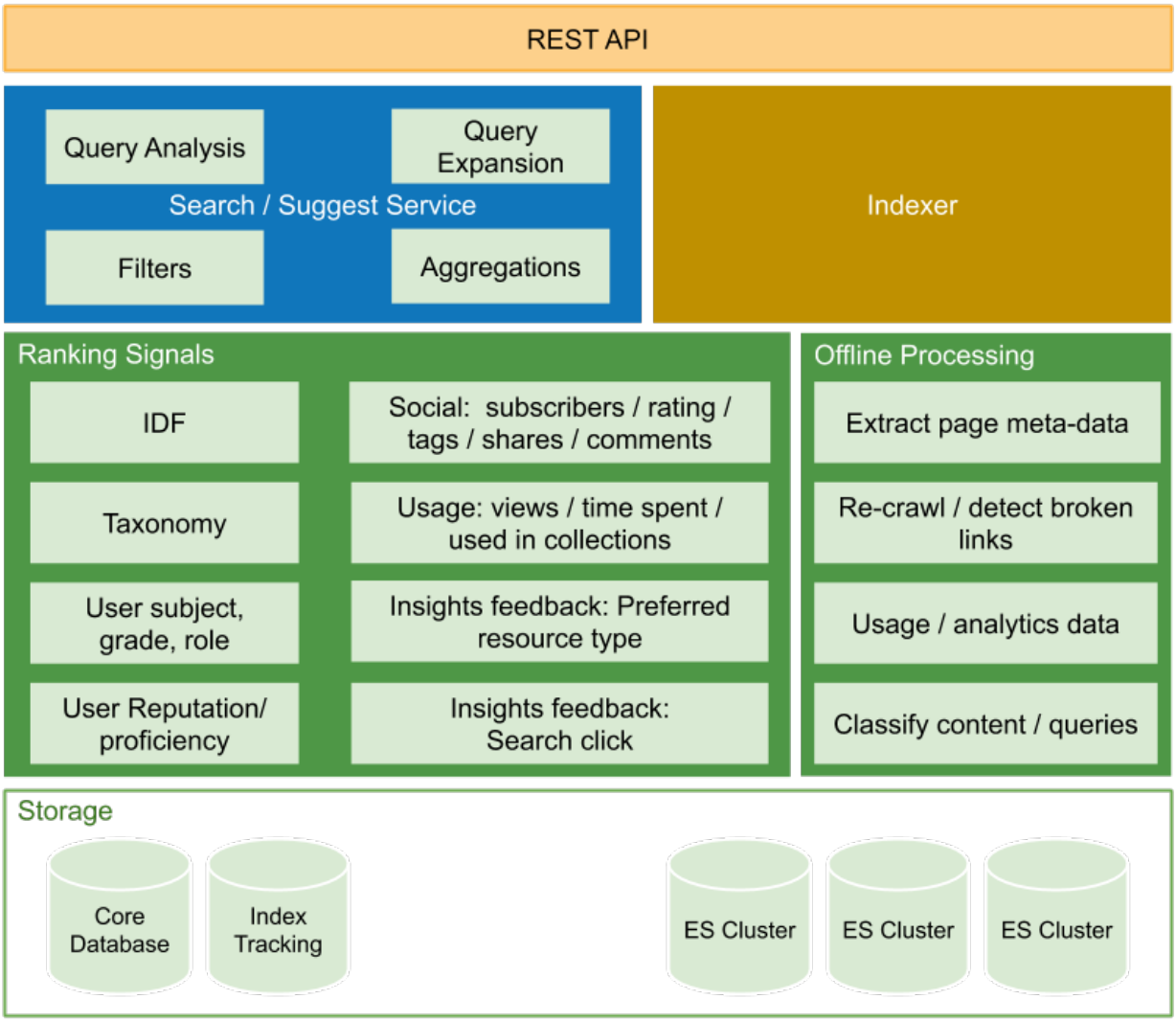
- Indexer adds / updates content and metadata into the search index.
- API layer handles search queries and is the user facing part of search & suggest system.



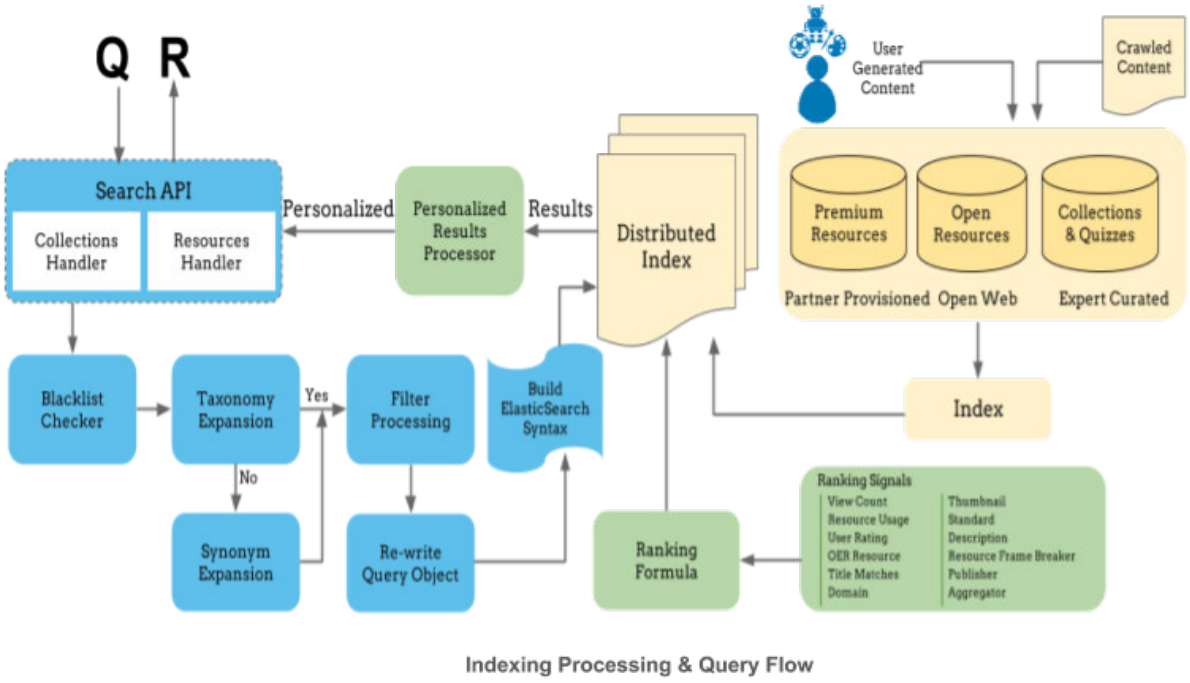
As content gets created or updated, Navigator core services publish this change data / events into the Kafka message bus. These messages are consumed by indexer component which then refreshes the documents that have changed and updates an index attributes across ES indexes. Index updates are in real-time for user updates & in batch mode for bulk content ingestion done via tools. 4-gram index is maintained currently based on usage trend analysis.

Search and Suggest services utilize various data points - user inputs, usage data gathered, content extracted metadata and other contextual information to serve response for that context.

Suggest, unlike Search, has no real user input query term but the same is assumed based on the context of invocation. The requests go through a processing pipeline to classify, expand, rewrite, apply necessary filters, search, rank using signals, and gets personalized.



Relevance score of content is computed based on Term Frequency (TF) and Inverse Document Frequency (IDF) and adjusted based on usage signals collected by Navigator, with TF/IDF weighted at 60% and usage signals weighted at 40%.



Indexing Processing & Query Flow

REST APIs are exposed for search across catalog with a variety of filters, related content suggestions, performance based suggestions, crosswalk, and index export.

Key learning path suggestions are offered to student are as below:

a. Profile based route suggestions
b. Profile based rescope suggestions
c. Performance based reroute suggestions
d. Teacher influenced/discretionary reroute suggestions

Profile based Route Suggestion

At every student journey, based on learner profile, more specifically learner current location as reflected at competency skyline, learners are offered a route that supports student in successful completion of a chosen target journey.

Route suggestions are always made in the context of a course and at the entry point into course as a way to better prepare students for what's coming ahead in the course. These suggestions are made only once at the start of the journey; and are not offered after users start the journey. But learners can go back anytime to review previous suggestions and accept the same route.

Route suggestion is a preparatory course, compiled as an array of units. Depending on the domains across which the competency gaps are spread, this preparatory course may contain multiple units, one Unit per domain, and multiple lessons per unit, one Lesson per competency. The route suggestion intends to help learner cover the competency gaps prior to starting the destination journey.

Profile based Rescope Suggestion

When students choose a destination Navigator course for study, Rescope of selected destination course happens to focus learner on competencies learner is yet to master. The rescope computation identifies the content at destination course that the learner can skip and focus on the content that will enhance the learner mastery.

Rescope suggestions are always made in the context of a course and at the entry point into course as a way to focus the student on competencies not yet mastered at the course.

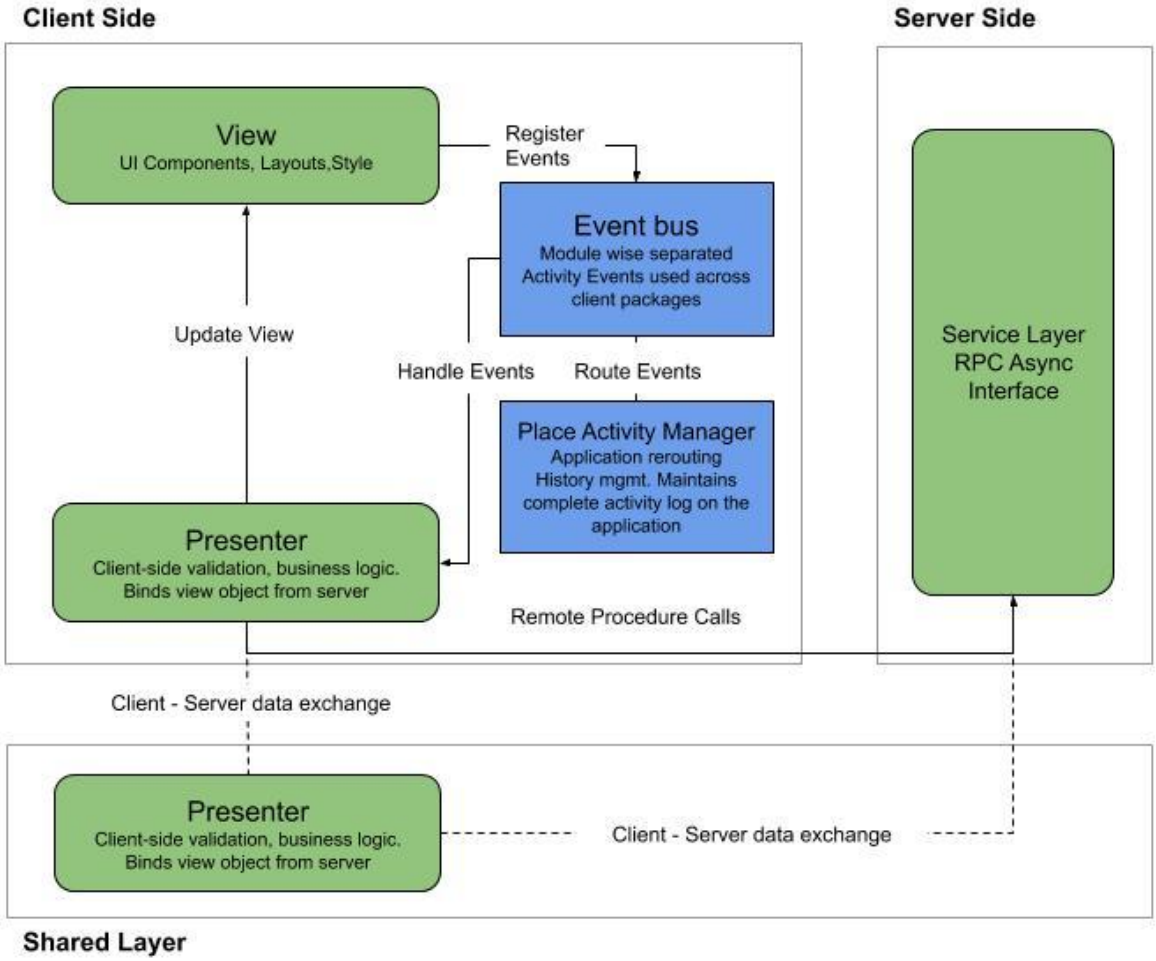
Performance based Reroute Suggestion

As students commit to a given journey and progress down a learning path, Navigator tracks a student activities and performance at every competency. Reroute always happens in the context of an assessment. Based on student performance, at the end of every assessment, a suggestion that can result in rerouting is made:

- If the student performs poorly (<80%) at the competencies covered by the assessment and if competency is not completed or mastered previously, an appropriate signature collection is suggested as a backfill.
- If the student performs very well (>= 80%) at the competencies covered by the assessment and if there is no mastery data (system may have completion data), student is offered a signature assessment as suggestion to earn mastery. If this assessment is part of a Route or Teacher suggestion, then reroute logic is not triggered, implies signature assessments are not suggested.

Navigator Application Design

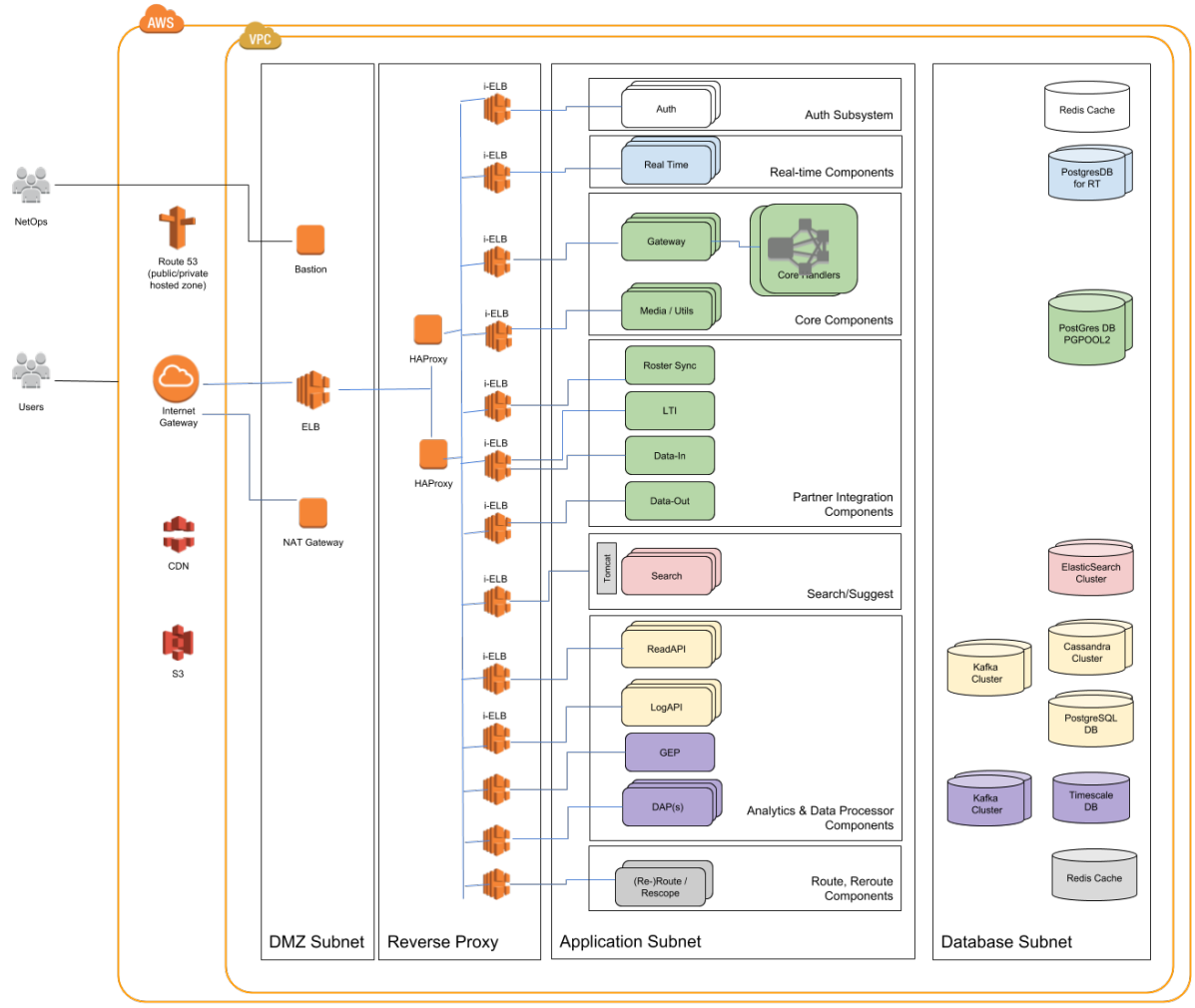
- EmberJS used for front-end application
- Reusable components developed as embeddable components
- Ember-Data layer for data encapsulation
- Bootstrap framework stitched together for responsive UI
- Localization managed via ember-i18n services
- (Limited) Feature configuration managed via external config files
- Headless testing of functionality using Ember Tests, PhantomJS & Stubby



Deploy Architecture

Navigator sources are spread across Github repositories with build and deploy process managed at Bamboo-based Continuous Integration / Continuous Deployment (CI/CD) implementation and via AWS Code Deploy agents. For detail on manual build process, refer to this document. For access to repositories and other infra, do contact us at partners@gooru.org.

Navigator is deployed at AWS in a high-availability mode. As such, at each layer there is redundancy setup so as not to impact application & services availability. Bamboo-based CI/CD implementation is in place to help with automated build and deploy of application components.



The first component which is outside world facing from infra perspective is Elastic Load Balancer (ELB). ELB serves purpose of:

- Termination of SSL
- End point of domains
- Load balancing between different haproxy(s)

Once the request is forwarded from ELB, it ends up on one of two parallel setups which are frontended by haproxy. These two setups are identical and they belong to same VPC. Haproxy rules are responsible to identify the cluster which is meant to serve the incoming request.

Most of the clusters only have specific ports opened to haproxy. Once the request is dispatched to cluster, the cluster determines its processing methodology. For haproxy the request is synchronous, even though cluster may process it asynchronously.

The infrastructure setup is continuously monitored using a combination of strategies:

- Server monitoring is done using Vistara agents and AWS health check services
- App level monitoring using scripts extracting log data and verifying HTTP response codes. Failures are flagged to team for review and action.

- API response monitoring passively by extracting haproxy logs and raise alerts if time taken for API exceeds thresholds
- 24 x 7 infra monitoring of the system by dedicated team

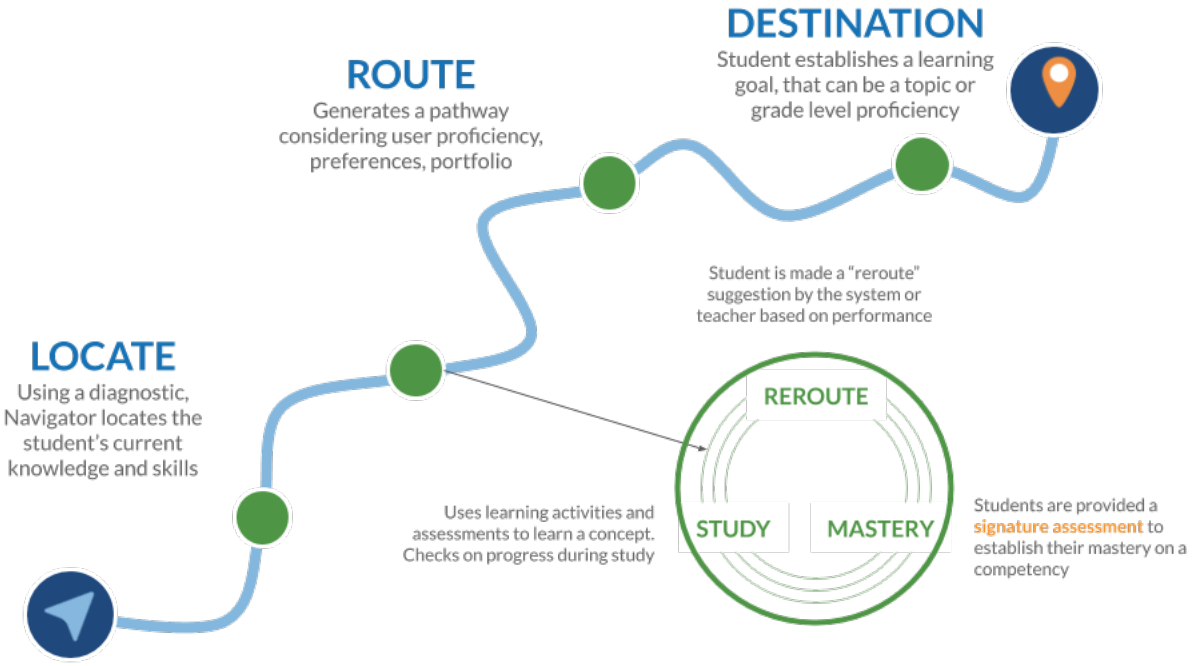
For more detail on deploy, refer to the Navigator Deployment document.

Updated NILE Pedagogy Design

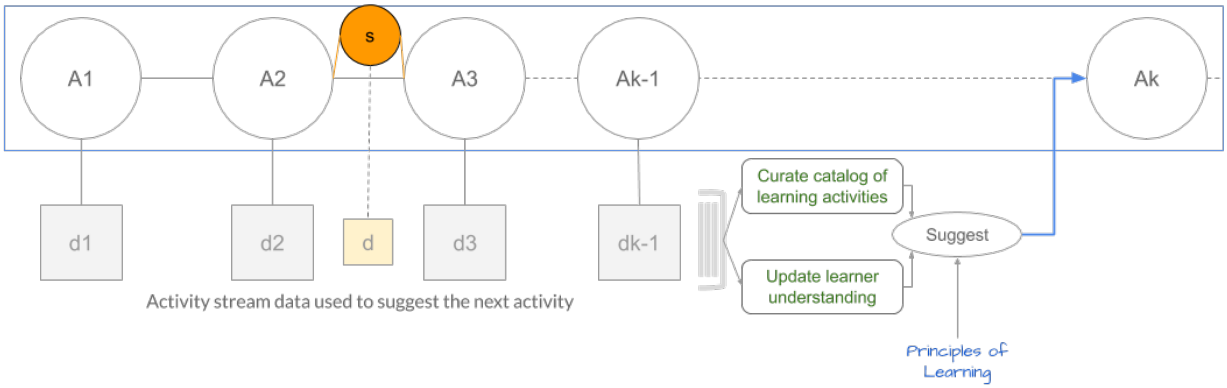
Science of Navigated Learning Operationalized with AI

Overview

Navigated Learning is about assured learning with simplicity of use. Navigator's GPS-like experience for learners begins by locating the learner's current knowledge, skills and mindsets.



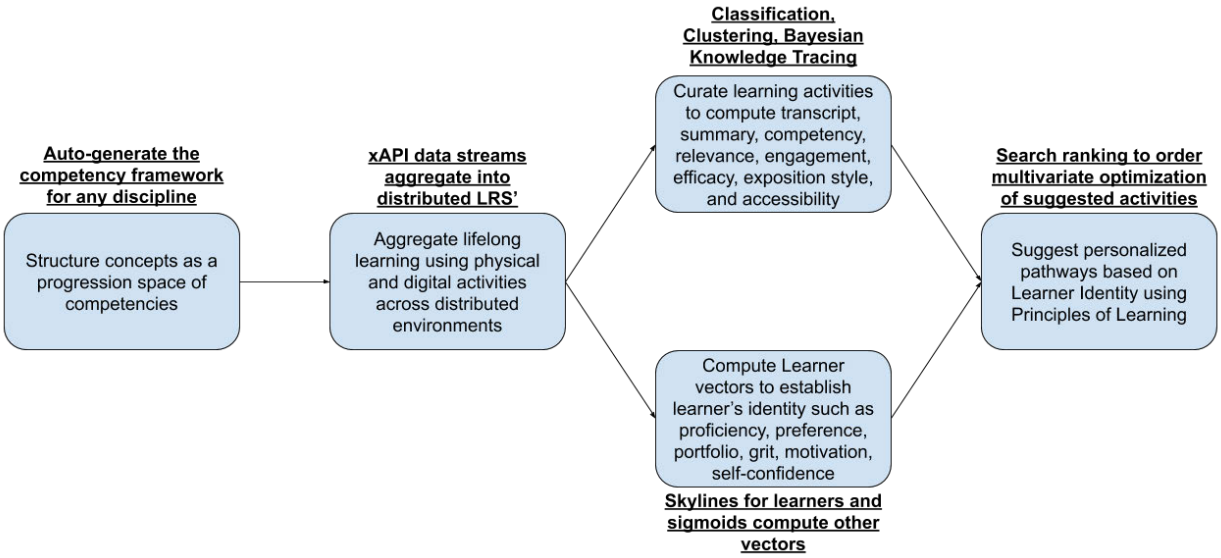
Navigator captures data after every learning activity. It uses the activity stream data to dynamically curate the learning activities and update its understanding of the learner to personalize the suggestion of learning pathways. We use the words "curate" to mean activity vectors and "understand" to mean learner vectors. To understand learners we need curated catalog and vice-versa. This is circular, so Navigator started with some seed values and then iteratively updates these.



Competency Framework enabling AI

To develop a "GPS for Learners", we

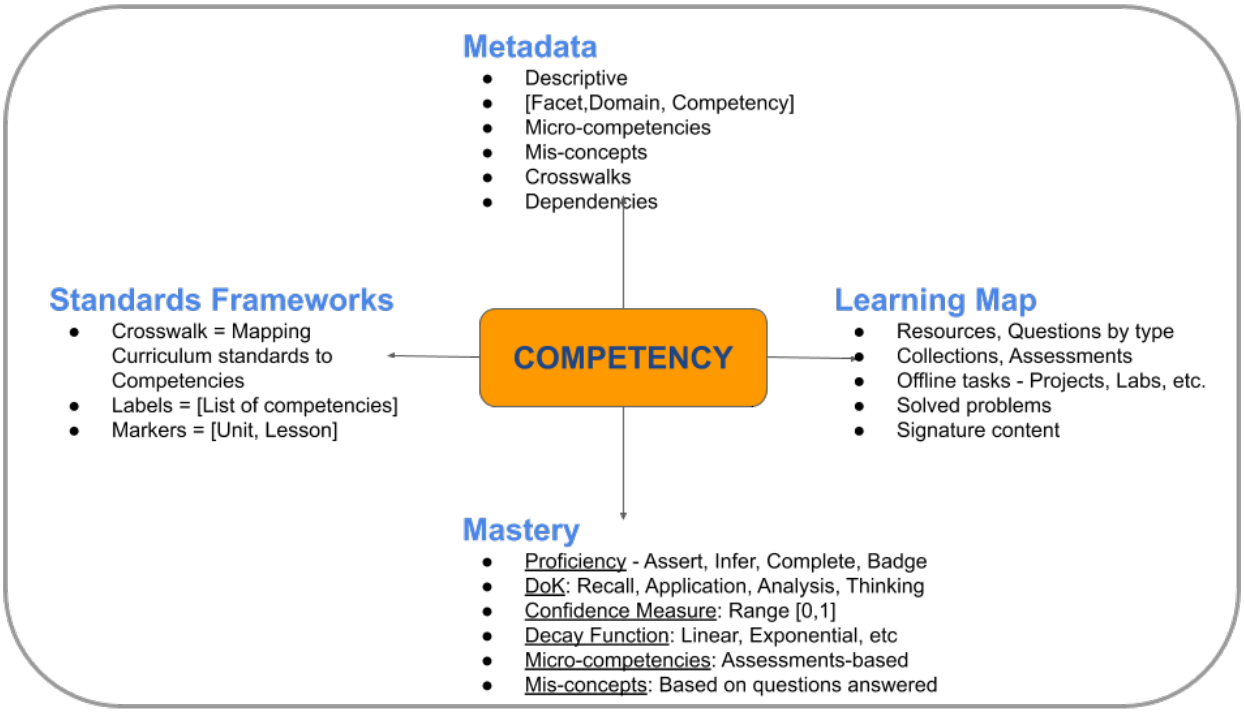
1. Structure learning as a progression space of competencies, so mathematical measures such as route, reroute and mean-time-to-learn can be computed,
2. Enable the development of competency frameworks by auto-generating from a corpus of content, the competencies and structuring them as a progression space.
3. Develop a competency editor, so experts can start with auto-generated competencies organized into a progression space, to review and finalize the competency framework.
4. Systematize the capture of data after every learning activity across distributed learning environments. xAPI statements across distributed systems are aggregated.
5. Curate resources with content analysis, machine classification and vector computations
6. Update learner identity with biographical data, knowledge & skills metadata, mindset metadata and community metadata. We capture bio metadata that exist from 3rd party systems - for e.g. grade-level of a student. Knowledge and skills metadata include progress, performance, proficiency, portfolio and preferences, Mindset metadata includes self-confidence, motivation, grit, and perseverance, Community metadata includes citizenship, authority and reputation. We refer to these collectively as to "locate the learner" in terms of knowledge, skills, and mindsets.
7. Use Principles of Learning to suggest personalized pathways given curated catalog of learning activities and learner identity.



Progression Space of Competencies

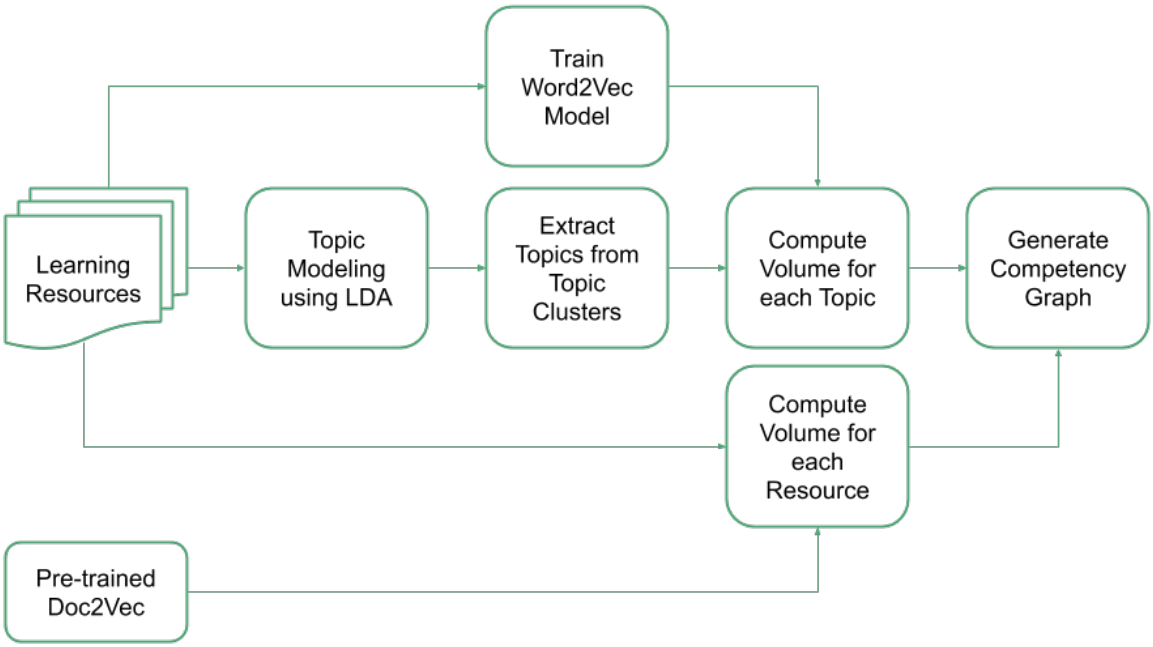
We structure the learning space as a progression space of competencies - i.e. a metric space with a partial ordering of competencies. The competencies are organized in a 3-dimensional space where the z-axis are the facets of learning, x-axis are the domains and y-axis are the competencies in increasing level of complexity. The competencies are nodes in a Navigator Competency Graph (NCG) with different types of edges - complementary edges to micro-competency, supplementary edges to concepts within a topic and dependency edges across topics and facets. NCGs for various facets are currently developed by subject matter experts. But, later in this document we will discuss, how we use AI techniques to auto-compute NCGs with expert validation.

Every competency is well defined with micro-competencies, dependencies, alternate conceptions, depth-of-knowledge, and a decay function. For user simplicity we represent the learning journey as a linear path, in actuality, learning pathways are spiral and competencies are revisited to reinforce a previously learned concept, to increase the depth of knowledge or to clarify a struggle or alternative conceptions.



Auto-generating competencies and structuring in a progression space

Establishing the progression space of competencies in any discipline is the foundation for enabling assured learning and simplicity of use with a “GPS for Learners”. Given the vastness of the learning space, it is critical that we auto-generate the progression space of competencies. Gooru has some very encouraging early results, where we take a corpus of content in a particular discipline, eg. textbook chapters on cyber security, or middle-school science textbooks used across schools in India, and the Navigator computes the transcript of the content, uses LDA topic analysis, doc2vec and word2vec to determine the topics, the competencies and the sequence. Experts review and finetune the progression space of competencies.

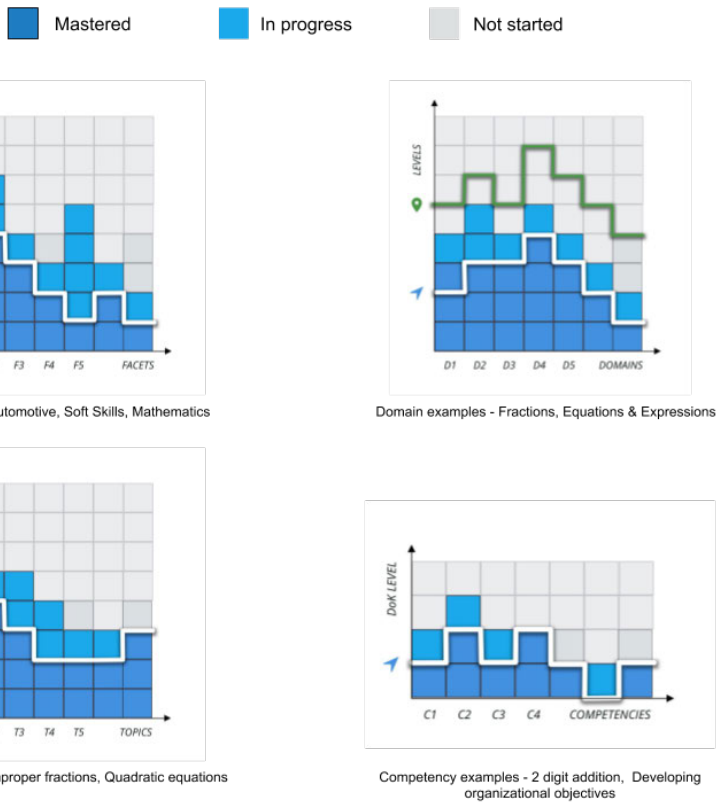


Learning Map for every competency

Every learning activity is mapped to a competency. We create and maintain a Learning Map - that is a set of activities of different types associated with every competency. For every competency, we have projects, solved examples, videos, written documents and assessments. The Learning Map is periodically updated to include learning activities that are effective for certain learners.

Learner Identity embedding in every competency

Navigator maintains for every user their complete identity against every competency. We represent the learner's progress, proficiency, performance, preference, portfolio, citizenship, authority, etc. for every competency.
This requires a uniform and thorough representational model for capturing our understanding of the learner across disciplines so, machines can personalize not just based on proficiency, but also based on the learner's disposition. We also need a visualization model that is simple for learners and their instructors to comprehend.
First, we maintain the learner's proficiency in the competency. While we represent whether users having mastery or not in a competency, the data about a learners understanding of a competency is very nuanced to include information such as DoK, or misconceptions associated with the competency.



Facet examples - Automotive, Soft Skills, Mathematics

Domain examples - Fractions, Equations & Expressions

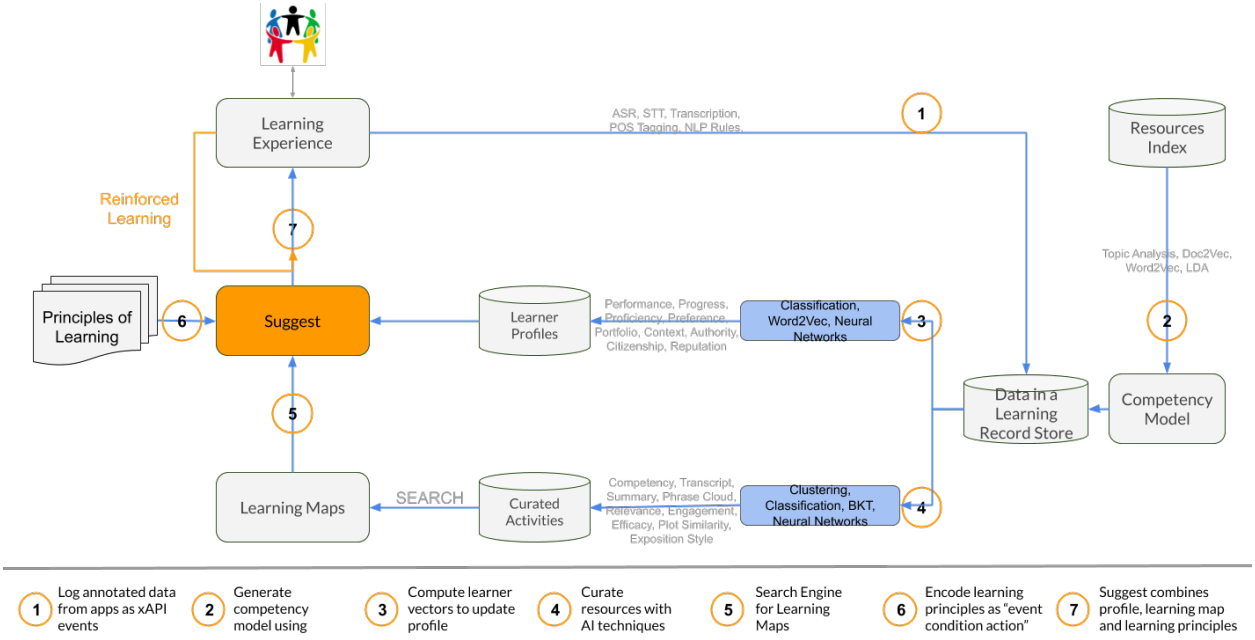
Topic examples - Improper fractions, Quadratic equations

Competency examples - 2 digit addition, Developing organizational objectives

Relate to local norms

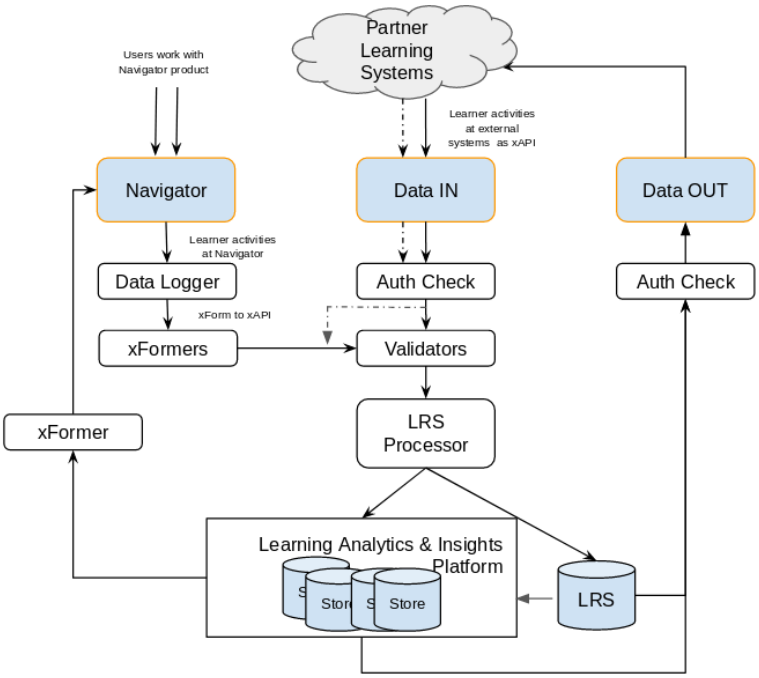
Competencies are also mapped to local norms. Learning in current education systems are organized by grades and courses. These are labels associated with a set of competencies. Institutions follow local standards frameworks. These are cross-walked to the lat-long for learning, so, the navigation is against a competency model, while the user experience displays their standards framework.

Personalized Learning Pathways



Activity Stream Data from Distributed Systems

Learning situations are highly varied during the lifelong journey of the learner. To address this extreme differences in learning and to navigate the learner to their learning goals, we need variety of learning content (offline tasks such as projects, essays, proofs and digital content such as videos, webpages, and multiple choice questions) and tools (assessments systems, simulations, games, interactives). All the offline and digital learning content and tools are collectively referred to as learning activities or learning resources. Lifelong learning demands that the learner profile has to be accessible over decades as technologies evolve and learning activities change. Learner data is continuously accumulated at Navigator via Data logging Environment. Navigator will store data from external systems based on a mutually agreed upon Event Specification or via xAPI. xAPI is an interoperability specification that allows learning technologies to talk to one another. Integrating two xAPI conformant systems can be considerably faster and cheaper. This includes migrating data from old learning systems to new ones.It is possible to collect data about the wide range of experiences a Learner has (online and offline) for heterogeneous/distributed systems using XAPI Specifications.



Curate Resources

We use the big-data from a logical LRS to curate learning activities using AI techniques. Beyond standard metadata such as Title, Description and Thumbnail, we use content analysis tools to compute transcript, summary, and phrase cloud for a learning resource. We use machine classification to determine the competency that the resource can be used to learn. We can use machine classification with supervised training and hierarchical classification to determine the lexile level or appropriateness of the resource for learners with special needs (hearing impairment, color blind, etc.). We use algebraic techniques to computer measures such as relevance and engagement, and use Bayesian Knowledge Tracing to estimate the efficacy of a learning activity.

Update Learner Identity

Navigator seeks to understand the learner across all facets - curriculum knowledge, non-cognitive skills, mindsets such as perseverance and motivation. Navigator computes a full-range of learner vectors such as progress, proficiency, performance, preference, portfolio, citizenship, and authority. We are working on ensuring that we capture data that can be analyzed to compute perseverance and motivation. With this complete understanding of the learner, we can personalize their pathways.

Principles of Learning

Based on the 20 principles of Learning that the American Psychological Association has published, we have focused on 5 principles.

- Students learn best when they are actively engaged in constructing new learning on a foundation of prior knowledge and experience.
- Students learn best when their learning opportunity is a stretch learning experience; e.g., it builds on what they know and provides guidance but also extends or applies what they know in a new

way, whether that extension is to a new context, a chance for them to make an inference, an analogy, or to a surprising next step.

- Students learn best when they have the opportunity to revisit an idea or concept multiple times, especially when revisiting is new flavors or variations on the original (e.g., Bruner; Reiser).
- Assessment is always an imperfect measure of what someone knows. Therefore, frequent embedded assessment, multiple levels of challenge and multiple kinds of evidence are the best means to generate a solid estimate of progress of critical thinking, knowledge and skills (e.g., Bransford; Schwartz; Barron).
- Choice, within reasonable limits and with supports, fosters engagement, confidence-building, and perseverance. Learning environments that foster trust and risk-taking with guidance foster deeper engagement, confidence-building, and perseverance (e.g., Resnick).

In the Event, Condition Action table below, the learning principles inform the suggest logic within the system.

Event	Condition	Action	Learning Principle
Demonstrates proficiency on 7.RP.2.c (Represents Proportional Relationships as Equations)			
	Above average performer	Offer a more challenging and engaging resource to expand learner's ZPD on competency	Actively engage in constructing new knowledge building on prior knowledge and experience Stretch learning experience
	Struggled and demonstrated mastery after 2nd attempt	Offer 7.RP.2.c additional CFU in a month from demonstrating proficiency	
Performs poorly in the check-for-understanding questions for 7.RP.2.c			
	Average or below performer	System suggests a lift-collection with curated resources in (7.RP.2)	Choice fosters engagement
	Has mastery in all the dependent competencies	Suggest a signature collection to reinforce the concept	Revisit concept in a new context
	Has gaps in prerequisite competencies	Offer collection in competency where there is a gap in learning (7.RP.1)	Revisit foundational concept
Performs poorly on second CFU for 7.RP.2.c			
	Has common alternative concepts/struggles in math concepts	Offered collection in foundational competency (6.RP.2 and 6.RP.3	Revisit foundational concept

	Struggling learner	Instructor is offered curated suggestions to give learner	
Demonstrates same gaps/struggles on multiple CFUs			
	Weakness in a dependent concept that has the same alternative conception	Suggest a lift-collection with resources related to the dependent concept + current concept	Revisit foundational concept
	Common struggle is present on multiple assessments	Show solved example with step by step answers	Revisit foundational concept
Demonstrates proficiency on 7.RP.2.c (Represents Proportional Relationships as Equations)			
	Above average performer	Offer a more challenging and engaging resource to expand learner's ZPD on competency	Actively engage in constructing new knowledge building on prior knowledge and experience Stretch learning experience
	Struggled and demonstrated mastery after 2nd attempt	Offer 7.RP.2.c additional CFU in a month from demonstrating proficiency	
Learner earns a badge	Learner is above average on this competency	Learner is offered collections on new competencies on learning pathway	Actively engage in constructing new knowledge building on prior knowledge and experience Stretch learning experience

Recommend Pathways

Depending on which principle of learning we use, we can obtain a large number of options for the next activity in the learning journey for the students. Navigator uses ranking signals similar to search ranking, to compute a total order of these options and suggests upto three activities, from which the learner can choose one to engage in. Using the search ranking as a multi-variate optimization technique allows us to continue to add to the principles of learning, consider not just the learning activities but also an understanding of the learner.

Big-data operationalizes science of navigated learning

Gooru along with its research partners are working on a number of research problems to increase the efficiency and effectiveness of Navigated Learning. The current list of AI techniques that we use include:

Technology	Description	AI/ML Techniques
Automatic Generation of Coherent Learning Pathways	In this work, we propose a model to automatically generate learning pathways from available open learning resources, such that the generated pathways are semantically coherent and pedagogically progressive. The proposed model has two components – a Greedy Generator and a Validator based on Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) models respectively. The Greedy Generator chooses the next resource in the learning pathway based on local considerations and the Validator validates the learning pathway as a whole. They work in tandem with each other connected by a feedback loop.	• Word embeddings -Word2Vec • Document embeddings - Paragraph2Vec • LDA (Latent Dirichlet Allocation) • SVM (Support Vector Machine) • LSTM (Long Short-Term Memory)
Creating Navigable Competency Maps	In this work, we build a two-dimensional *competency map* that is based on generating the semantic embeddings of learning resources. Here, learning is situated in a logical space called the *progression space*, where each point in this space represents a competency, or a unit of learning. Competency map is a two dimensional map with horizontal axis (X-axis) representing the topics and vertical axis (Y-axis) representing the complexity or pedagogic level of the learning resources, such that when we move from left to right on X-axis or top to bottom on Y-axis, there is a progression in terms of learning. Competency map also has different learning resources mapped on to it using the semantic embeddings.	• Word embeddings - Word2Vec • Document embeddings - Paragraph2Vec • FastText (Word Embeddings) • Glove (Word Embeddings) • LDA (Latent Dirichlet Allocation) • K-means clustering
Automatic Detection Of Topic Transitions In Lecture Videos	Videos are an increasingly popular method of delivering lectures online. A lot of learning videos on popular learning platforms have lengths up to 120 minutes, in which the lecturer discusses several sub-topics under the ambit of a broader topic. This creates the need for the video to be searchable so that the student can switch to the topics they are interested in learning. We seek to automate the process of generating timestamps where the transitions in topics take place in the videos since manual segmentation is expensive and slow. There is little visual information which can be made use of to detect scene transitions in learning videos. We hence make use of the audio information encoded in the video by generating the video transcript using Automatic Speech Recognition. We propose two models that automatically detect time stamps where topic transitions take place using Natural language Processing.	• Automatic Speech Recognition to generate video transcript • Pre-trained sentence embeddings (InferSent) to represent sentences • Pre-trained word embeddings (Elmo) to represent words • Graph convolutional networks for clustering similar sentences
Locating the Learner's Mindsets	**Perseverance:** A measure of the total amount of time a learner spends on competencies **Grit**: A measure of the total amount of time a learner spends on competencies in spite of failure **Self-confidence**: A measure of the learner's assessment of his own abilities.	• Probability distributions and Statistical techniques such as normalization and standardization to represent metrics within a desirable range
Curating Catalog of Learning Activities	**Efficacy**: Measures how effective a learning resource is in making an observable difference in the competency obtained by the learner **Engagement**: Measures how engaging a learning resource is to learners (in terms of visits, likes, shares) while they try to acquire a particular competency **Relevance**: Measures how relevant a learning resource is to acquiring a particular competency	• Bayesian knowledge tracing
Evidence Based Competency Modeling	In this work, we propose that data generated during the learning process, referred to as evidence, can be used to reason about the underlying competency. We focus on activities that learners do like consuming resources like videos, text books, articles, and documents.	• SVM (Support Vector Machine) • K Nearest Neighbor Clustering • Keras Sequential Model

	The work also shows that one single average model to predict the state of competency cannot be used and we need to build separate models for different earners, which validates “The Myth of Average”	
Automatic Trailer Generation of Narratives	Trailers usually provide the viewer with an eagle’s eye view of any Narrative. Since the time of creation of trailers, people have perfected the art of making effective trailers. With the enormous growth rate in user-generated academic resources such as videos, audios, texts, courses; efficient navigation of these resources has become very important. To get a quick sense of the narratives, it will be useful to have academic trailers of such resources. To generate these academic trailers manually is not only a task of skill but also of specialized content knowledge. Leveraging capacity of machine learning and automation we can have intelligent systems to generate academic trailers with minimum human interference. IBM Watson has automatically generated the first plausible trailer in 2016. We are proposing to generate automatic trailer of academic resources by using text processing, NLP and machine learning techniques. After testing it on academic resources, it could be extended to apply in the publishing industry to generate trailers automatically of 2-3 mins long for books, conference proceedings, research papers etc. Below diagram shows the full flow of the process. The input is the corpus that goes to Module 1 for extracting voice-over text, which is used by the remaining three modules for extracting title, video-text and TTS engine. Module 4 compiles all the inflow information with user defined some meta-information to produce a trailer.	• Sequence to Sequence Attention based Models. • Transformer Models • Word Embeddings - Meta Embedding [GloVe + Word2Vec] • Document Embeddings - [ELmo/BERT] • LSTM (Long Short-Term Memory) • Text-to-Speech (TTS) • Coreference Resolution

Updated NILE Requirements

NCG Requirements and High Level Design

Navigator Competency Model

See Navigator Competency Model document for more details

Competency visualization as Tree

Though the core idea of NCG is to model the Competency as Graph, however, in order for visualization for approval and to understand the process of creation of the whole FW, it may be beneficial to have a tree view.

Classification

The top level item under which the rest of the FW needs to be modeled. E.g. K12, Higher Ed

Classification exists as a global entity. It won't be FW specific. FW creators either need to create one of the Classification under which they will create new FW, or they first need to create a new classification, get it approved and then use it to create new FW.

Subject

The Classification will contain one or more subjects e.g. Math, ELA etc. which will be relevant to that classification.

Subjects exists as a global entity. It won't be FW specific. FW creators either need to create one of the Subjects under which they will create new FW, or they first need to create new subject, get it approved and then use it to create new FW.

Framework

Once the required Classification and Subject needed to create FW exists, FW could be created under them. Note that FW name or code won't be unique key and as such CCSS can be applied to multiple Classification/Subject combination as FW. Hence trying to access the FW using the FW code will be non-deterministic as multiple FW may exists. User should have context of Classification/Subject along with name/code OR they should be having ID (in which case context for the FW can be looked up).

Framework creation can be done using two different variations: - FW which is going to act as reference FW - FW which will be mapped to the reference FW

The default path would be to assume that FW is going to map to reference FW.

The difference is that when a reference FW is created, the domains and topics are allowed to be created. However, in case when FW is to be mapped, the domain/topic need to mapped to domain/topic from the reference FW. They can't be created in isolation. This is to avoid the domain/topic level crosswalk.

Framework Creation Prerequisites

FW creation is a multi step process. First steps would be to create the subject and classification for which FW needs to be created. - Fetch Classification list to verify that classification exists - If it does, then continue - Else, create classification - Apply for approval - If approved, then continue - Else modify as per comment provided - Fetch Subject under the classifications to verify that subject exists - If it does, then continue - Else, create subject - Apply for approval - If approved, then continue - Else modify as per comment provided

Framework Creation: Ref FW

- Create FW
- Create Domains (the sequence is inferred)
- Create Topics in domain (the sequence is inferred)
- Create competencies for domain/topic (the sequence which defines the complementary relationship is inferred)
- Define learning milestone group and then learning milestones in them
- Create relationship for pre-requisite
- Create relationship for related competency
- Create micro competencies
- Create alt concepts
- Create Labels (Optional)
- Trigger validation for whole FW (by creator)
- Apply for approval (approval can only happen if all mandatory steps are completed and validation is successful)
- Once approved, FW becomes read only

Framework Creation: Mapped FW

- Create FW
- Fetch domain from Ref FW and provided mapping with mapped FW domain. Note that not all domains will be covered by mapped FW. Also note that Ref FW is more granular and hence one domain from mapped FW should not be mapped to more than one Ref FW domain. The sequence of domains is inferred from Ref FW
- For each domain that is created in mapped FW, fetch the topics associated with the Ref FW domain. For each topic, provide a mapping with mapped FW topic. Also note that Ref FW is more granular and hence one topic from mapped FW should not be mapped to more than one Ref FW topic. The sequence of topics is inferred from Ref FW (as creator is just creating the mapping)
- For each domain/topic which is mapped, fetch the Ref FW competency and create a mapping for competencies in mapped FW. Also note that Ref FW is more granular and hence one competency from mapped FW should not be mapped to more than one Ref FW domain. The relationship of being complementary is inferred from Ref FW
- Define learning milestone group and then milestones in them

- For each mapped competency, fetch the Ref FW micro competencies and provide mapping
- Note that alt concepts exists in Ref FW so no creation provided
- Create Labels (Optional)
- Trigger validation for whole FW (by creator)
- Apply for approval (approval can only happen if all mandatory steps are completed and validation is successful)
- Once approved, FW becomes read only

Notable Points

- Only Ref FW can define domains/topic. Mapped FW can only map to them with their own name
- Only Ref FW can define relationships. Mapped FW will inherit it
- Crosswalk need to be defined before defining milestones
- The sequence of topics is just for display purpose. It has got no bearing on pre-req kind of topic dependency
- The competency will be flattened and there won't be any L0, L1 and L2 for comp/micro comp
- For cross walk to happen parent item should act as bound
- If crosswalk is being done for competency, then that should be within the crosswalked topic
- If crosswalk is being done for micro competency, then should be within the crosswalked competency

Also to note: If a new Ref FW is being created, the following need to be available before it can be considered as ready (for teachers, students etc.)

- Signature content
- Diagnostic

The FW without the above will be available for content builders who could create signature content and diagnostic.

This is not part of FW creation / approval of NCG. This should be part of post approval making this available in Platform. The NCG scope does not allow creation of content and hence it can't be verified here.

API List for NCG Core

Entities

Entities covered in Core Model backed with NCG Core APIs

- Classification
- Facet/Subject
- Framework
- Domains
- Topics
- Competencies
- Learning Milestones

- Competency Groups
    - Custom groups
- Micro competencies
- Alternate concepts

Helper Entities

- Review audit log

Entities not covered in core model

- Learner Profile (including prefs)
- Learning Activities (including Learning Maps, signature, diagnostic etc)

Relationships for entities covered in core model

- Dependencies (within same FW)
- Crosswalk (with reference FW)

Detailed APIs

Classification

Create Classification

Details:

HTTP Method	POST
End Point	http://{host}/ncg/api/classification
Sample	http://localhost:8080/ncg/api/classification

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value

raw	<pre>{
	"title": "Title 1",
	"displayCode": "CLS 1",
	"description": "Title 1 description"
	}</pre>

Response Payload:

Response Code: 200 OK

Update Classification

Details:

HTTP Method	PUT
End Point	http://{host}/ncg/api/classification/{ID}
Sample	http://localhost:8080/ncg/api/classification/1

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"title": "Title 1U","displayCode": "CLS 1U","description": "Title 1U description"}`

Response Payload:

Response Code: 200 OK

Request review for classification

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{ID}/review`
Sample	`http://localhost:8080/ncg/api/classification/1/review`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with the actual value of the access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Approve classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{ID}/review/approve`
Sample	`http://localhost:8080/ncg/api/classification/1/review/approve`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "This is simply awesome"}`

Response Payload:

Response Code: 200 OK

Reject classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{ID}/review/reject`
Sample	`http://localhost:8080/ncg/api/classification/1/review/reject`

Headers:

Header Name	Value

Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "This is plain stupid"}`

Response Payload:

Response Code: 200 OK

List classifications

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/classification/list?status=&offset=&max=&scope=`
Sample	`http://localhost:8080/ncg/api/classification/list?status=0&offset=&max=&scope=`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value

raw	`{`
	` "title": "Title1",`
	` "displayCode": "CLS1",`
	` "description": "Title 1 description",`
	` "i18nCode": "i18n.cls1"`
	`}`

Response Payload:

Response Code: 200 OK

Facet/Subject

Create subject in classification

Details:

HTTP Method	POST
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject`
Sample	`http://localhost:8080/ncg/api/classification/1/subject`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"title": "Title 1D","displayCode": "CLS 1D","description": "Title 1 description"}`

Response Payload:

Response Code: 200 OK

Update subject in classification

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject/{subjectID}`
Sample	`http://localhost:8080/ncg/api/classification/1/subject/1`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"title": "Title 3","displayCode": "CLS 4","description": "Title 3 description"}`

Response Payload:

Response Code: 200 OK

Request review for subject in classification

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject/{subjectID}/review`
Sample	`http://localhost:8080/ncg/api/classification/1/subject/1/review`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Approve subject in classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject/{subjectID}/review/approve`
Sample	`http://localhost:8080/ncg/api/classification/1/subject/1/review/approve`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "Sure, it is awesome"}`

Response Payload:

Response Code: 200 OK

Reject subject in classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject/{subjectID}/review/reject`
Sample	`http://localhost:8080/ncg/api/classification/1/subject/1/review/reject`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "This is plain stupid"}`

Response Payload:

Response Code: 200 OK

List subject in classification

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/classification/{classificationID}/subject/list?status=&offset=&max=&scope=`
Sample	`http://localhost:8080/ncg/api/classification/1/subject/list?status=&offset=&max=&scope=`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Framework (One flavour for Ref FW and one for mapped one)

Create framework for subject in classification

Details:

HTTP Method	POST
End Point	`http://{host}/ncg/api/framework`
Sample	`http://localhost:8080/ncg/api/framework`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value

raw	`{`
	`"subjectId": 1,`
	`"displayCode": "CLS 1",`
	`"description": "Title 1 description",`
	`"refFwId": null`
	`}`

Response Payload:

Response Code: 200 OK

Update framework for subject in classification

Details:

HTTP Method	PUT

End Point	`http://{host}/ncg/api/framework/{frameworkID}`
Sample	`http://localhost:8080/ncg/api/framework/1`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

Validate framework

- After all the creation is done, not just FW

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/validate`
Sample	`http://localhost:8080/ncg/api/framework/1/validate`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Request review for framework

- Framework for subject in classification (after all the creation is done, not just framework and validation is successful)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/review`
Sample	`http://localhost:8080/ncg/api/framework/1/review`

Headers:

Header Name	**Value**
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Approve framework for subject in classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/review/approve`
Sample	`http://localhost:8080/ncg/api/framework/1/review/approve`

Headers:

Header Name	**Value**

Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "This is simply awesome"}`

Response Payload:

Response Code: 200 OK

Reject framework for subject in classification (admin)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/review/reject`
Sample	`http://localhost:8080/ncg/api/framework/1/review/reject`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body:

Mode	Value
raw	`{"comment": "This is plain stupid"}`

Response Payload:

Response Code: 200 OK

List frameworks for subject in classification

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/list`
Sample	`http://localhost:8080/ncg/api/framework/list?status=0&offset=&max=&scope=&subjectId=`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Domain (One flavour for Ref FW and one for mapped one)

Framework Domains - Create/Update (Ref and Mapped)

- Create domain (if FW is ref, else create mapping of domains)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/domains`
Sample	`http://localhost:8080/ncg/api/framework/{frameworkID}/domains`

Headers:

Header Name	Value

Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body - Ref

Mode	Value

raw	{
	"domains": [
	{
	"displayCode": "DMN-1",
	"name": "DMN-1",
	"refFwDomainId": null
	},
	{
	"displayCode": "DMN-2",
	"name": "DMN-2",
	"refFwDomainId": null
	}
	]
	}

Request Body - Mapped

Mode	Value

raw	{
	"domains": [
	{
	"displayCode": "DMN-1",
	"name": "DMN-1",
	"refFwDomainId": 24
	},
	{
	"displayCode": "DMN-2",
	"name": "DMN-2",
	"refFwDomainId": 25
	}
	]
	}

Response Payload:

Response Code: 200 OK

List Domains in framework:

Details:

HTTP Method	GET

End Point	http://{host}/ncg/api/framework/{frameworkID}/domains
Sample	http://localhost:8080/ncg/api/framework/2/domains?offset=&max=

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Response Payload:

Response Code: 200 OK

Topic (One flavour for Ref FW and one for mapped one)

Domain Topics - Create/Update (Ref and Mapped)

- Create topics in domain (if FW is ref, else create mapping of topics)

Details:

HTTP Method	PUT
End Point	http://{host}/ncg/api/framework/{frameworkID}/domains/{domainID}/topics
Sample	http://localhost:8080/ncg/api/framework/1/domains/24/topics

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body - Ref

Mode	Value
raw	{"topics" : [{"displayCode": "TPC-1","name": "TPC-1","refFwTopicId": null},{"displayCode": "TPC-2","name": "TPC-2","refFwTopicId": null}]}

Request Body - Mapped

Mode	Value

raw	{
	"topics": [
	{
	"displayCode": "DMN-MAP-1",
	"name": "DMN-MAP-1",
	"refFwTopicId": 1
	},
	{
	"displayCode": "DMN-MAP-2",
	"name": "DMN-MAP_2",
	"refFwTopicId": 2
	}
	]
	}

Response Payload:

Response Code: 200 OK

List Framework topics

Details:

HTTP Method	GET
End Point	http://{host}/ncg/api/framework/{frameworkID}/domains/{domainID}/topics?offset=&max=
Sample	http://localhost:8080/ncg/api/framework/2/domains/26/topics?offset=&max=

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

Competencies (One flavour for Ref FW and one for mapped one)

Competencies in Domain - Create/Update (Ref and Mapped)

- Create competencies (if FW is ref, else create mapped)

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/domains/{domainID}/topics/{topicID}/competencies`
Sample	`http://localhost:8080/ncg/api/framework/1/domains/24/topics/1/competencies`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body - Ref

Mode	Value

raw	{
	"competencies": [
	{
	"displayCode": "C-1",
	"name": "C-1",
	"description": "description",
	"twoWordDescription": "twoWordDescription",
	"studentDescription": "studentDescription",
	"teacherDescription": "teacherDescription",
	"compType": "core",
	"refFwCompId": null
	},
	{
	"displayCode": "C-2",
	"name": "C-2",
	"description": "description",
	"twoWordDescription": "twoWordDescription",
	"studentDescription": "studentDescription",
	"teacherDescription": "teacherDescription",
	"compType": "additional",
	"refFwCompId": null
	}
	]
	}

Request Body - Mapped

Mode	Value

raw	{
	"competencies": [
	{
	"displayCode": "CM-1",
	"name": "CM-1",
	"description": "description",
	"twoWordDescription": "twoWordDescription",
	"studentDescription": "studentDescription",
	"teacherDescription": "teacherDescription",
	"compType": "core",
	"refFwCompId": 7
	},
	{
	"displayCode": "C-2",
	"name": "C-2",
	"description": "description",
	"twoWordDescription": "twoWordDescription",
	"studentDescription": "studentDescription",
	"teacherDescription": "teacherDescription",
	"compType": "additional",
	"refFwCompId": 8
	}
	]

	}

Response Payload:

Response Code: 200 OK

List Competencies in framework/domain/topic:

Details:

HTTP Method	GET
End Point	http://{host}/ncg/api/framework/{frameworkID}/domains/{domainID}/topics/{topicID}/competencies?offset=&max=
Sample	http://localhost:8080/ncg/api/framework/1/domains/1/topics/1/competencies?offset=&max=

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	{"displayCode": "CLS 1U","description": "Title 1 description U"}

Response Payload:

Response Code: 200 OK

Learning Milestones (Created only for Ref FW)

- Create learning milestones group in framework

Learning milestones group in framework - Create/Update

Details:

HTTP Method	PUT
End Point	http://{host}/ncg/api/framework/{frameworkID}/milestones
Sample	http://localhost:8080/ncg/api/framework/1/milestones

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value

raw	<pre>{
	  "milestones": [
	    {
	      "description": "MS-1",
	      "name": "MS-1",
	      "domains": [
	        {
	          "domainId": 1,
	          "topics": [
	            {
	              "topicId": 1,
	              "compIdHigh": 1,
	              "compIdLow": null
	            },
	            {
	              "topicId": 2,
	              "compIdHigh": 4,
	              "compIdLow": null
	            }
	          ]
	        },
	        {
	          "domainId": 2,
	          "topics": [
	            {
	              "topicId": 4,
	              "compIdHigh": 10,</pre>

```
              "compIdLow": null
            },
            {
              "topicId": 5,
              "compIdHigh": 14,
              "compIdLow": null
            }
          ]
        }
      ]
    },
    {
      "description": "MS-2",
      "name": "MS-2",
      "domains": [
        {
          "domainId": 1,
          "topics": [
            {
              "topicId": 1,
              "compIdHigh": 3,
              "compIdLow": 2
            },
            {
              "topicId": 2,
              "compIdHigh": 6,
              "compIdLow": 5
            }
          ]
        },
        {
          "domainId": 2,
          "topics": [
            {
              "topicId": 4,
              "compIdHigh": 12,
              "compIdLow": 11
            },
            {
              "topicId": 5,
              "compIdHigh": 15,
              "compIdLow": 15
            }
          ]
        }
      ]
    }
  ]
}
```


Response Payload:

Response Code: 200 OK

List milestones group in framework

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/{frameworkID}/milestones`
Sample	`http://localhost:8080/ncg/api/framework/1/milestones?offset=&max=`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

Competency Groups (One flavour for Ref FW and one for mapped one)

Competency group in framework- Create/Update

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/grouptypes/{groupTypeID}/groups`
Sample	`http://localhost:8080/ncg/api/framework/1/grouptypes/1/groups`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	{ "groups": [{ "description": "Grade-1", "name": "Grade-1", "competencyIds": [1, 11, 21, 31, 41] }, { "description": "Grade-2", "name": "Grade-2", "competencyIds": [2, 12, 22, 32, 42] }, { "description": "Grade-3", "name": "Grade-3", "competencyIds": [3, 13, 23, 33, 43] }, { "description": "Grade-4", "name": "Grade-4", "competencyIds": [

	<pre>4,
	14,
	24,
	34,
	44
	]
	}
	]
	}</pre>

Response Payload:

Response Code: 200 OK

List Competency groups in framework:

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/{frameworkID}/grouptypes/{groupTypeID}/groups`
Sample	`http://localhost:8080/ncg/api/framework/1/grouptypes/1/groups?offset=&max=`

Headers:

Header Name	**Value**
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	**Value**
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

Relationship for competencies (applicable for Ref FW only)

Create/Update relationship between two competencies of same framework:

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/competencies/{competencyID}/relationship/{relationtionShipID}`
Sample	`http://localhost:8080/ncg/api/framework/1/competencies/1/relationship/2`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"relatedTo" : [2,3,4]}`

Response Payload:

Response Code: 200 OK

- Query competencies under a framework based on specified relationship

Query competencies under a framework based on specified relationship

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/{frameworkID}/competencies/{competencyID}/relationship/{relationShipID}`

Sample	http://localhost:8080/ncg/api/framework/1/competencies/2/relationship/1

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	{"relatedTo" : [2,3,4]}

Response Payload:

Response Code: 200 OK

Alternate concepts (applicable for Ref FW only)

Create alternate concepts in a competency

Details:

HTTP Method	POST
End Point	http://{host}/ncg/api/framework/{frameworkID}/altconcepts
Sample	http://localhost:8080/ncg/api/framework/1/altconcepts

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "AC-1","name": "AC-1","description": "description","twoWordDescription": "twoWordDescription"}`

Response Payload:

Response Code: 200 OK

- Update alternate concepts in competency

Update alternate concepts in a competency

Details:

HTTP Method	PUT
End Point	`http://{host}/ncg/api/framework/{frameworkID}/altconcepts/{altconceptsID}`
Sample	`http://localhost:8080/ncg/api/framework/1/altconcepts/1`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "AC-1-modified","name": "AC-1-modified","description": "description-mod","twoWordDescription": "twoWordDescription-mod"}`

Response Payload:

Response Code: 200 OK

List alternate concepts in framework

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/{frameworkID}/altconcepts`
Sample	`http://localhost:8080/ncg/api/framework/1/altconcepts?offset=&max=`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

List alternate concepts in competency

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/framework/{frameworkID}/altconcepts/competency/{competencyID}`
Sample	`http://localhost:8080/ncg/api/framework/1/altconcepts/competency/1`

Headers:

Header Name	Value
Authorization	Token {TOKEN}

Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"displayCode": "CLS 1U","description": "Title 1 description U"}`

Response Payload:

Response Code: 200 OK

Review Trail

Fetch review trail associated with specified entity

Details:

HTTP Method	GET
End Point	`http://{host}/ncg/api/reviewtrail`
Sample	`http://localhost:8080/ncg/api/reviewtrail?id=1&type=CLASSIFICATION`

Headers:

Header Name	Value
Authorization	Token {TOKEN}
Content-Type	application/json

- Replace the {TOKEN} with actual value of access token obtained from authentication.

Request Body

Mode	Value
raw	`{"title": "Title1","displayCode": "CLS1","description": "Title 1 description","i18nCode": "i18n.cls1"}`

Response Payload:

Response Code: 200 OK

For every review done by approver, it may result in rejection or approval. For each of the state the approver need to provide reason for the same. They are stored in datastore. Similarly, when validation is triggered, system may produce some validation messages which are also stored here. They can be fetched on demand and are rendered in reverse chronological order along with the source/status (source = reviewer/system, status = approved/rejected/validation failed)

API Authentication and Authorization

The NCG API will require the authentication token as generated by the Navigator today. The NCG itself will not generate the authentication token. Once the authentication is done, the token is validated against redis for every API call. Only if the token is valid, user is allowed to do the specified operation. No APIs for NCG is provided for anonymous users.

Most of the APIs which are not approver specific, are not tied to any specific permission which users must have. However, for the approvers, it is mandatory that user should have permissions of "ncg.approver" otherwise user won't be authorized to do the approve/reject. The permissions of users are read from session stored in redis, which the platform stores while the user token is generated